# Sun Performance Library Reference Manual 

## Sun ${ }^{\text {TM }}$ Studio 8

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.
650-960-1300

Part No. 817-5071-10
May 2004 Revision A

Copyright © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.
U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements. Use is subject to license terms.
This distribution may include materials developed by third parties.
Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.
Sun, Sun Microsystems, the Sun logo, Java, and JavaHelp are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc.

This product is covered and controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.
DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.
L'utilisation est soumise aux termes de la Licence.
Cette distribution peut comprendre des composants développés par des tierces parties.
Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l’Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.
Sun, Sun Microsystems, le logo Sun, Java, et JavaHelp sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.
Ce produit est soumis à la législation américaine en matière de contrôle des exportations et peut être soumis à la règlementation en vigueur dans d'autres pays dans le domaine des exportations et importations. Les utilisations, ou utilisateurs finaux, pour des armes nucléaires,des missiles, des armes biologiques et chimiques ou du nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers les pays sous embargo américain, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exhaustive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.
LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.

## Sun Performance Library[tm] Reference Manual

## Sun [tm] Studio 8

This reference manual is the Sun Performance Library section 3P man pages, available in HTML and PDF formats. For additional information, see the Sun Performance Library User's Guide, available on docs. sun . com, or the LAPACK Users' Guide, available from the Society for Industrial and Applied Mathematics (SIAM).
available_threads - available_threads - returns information about current thread usage
blas_dpermute - blas_dpermute - permutes a real (double precision) array in terms of the permutation vector P , output by dsortv
blas_dsort - blas_dsort - sorts a real (double precision) vector X in increasing or decreasing order using quick sort algorithm
blas_dsortv - blas_dsortv - sorts a real (double precision) vector X in increasing or decreasing order using quick sort algorithm and overwrite P with the permutation vector
blas_ipermute - blas_ipermute - permutes an integer array in terms of the permutation vector P , output by dsortv
blas_isort - blas_isort - sorts an integer vector X in increasing or decreasing order using quick sort algorithm
blas_isortv - blas_isortv - sorts a real vector X in increasing or decreasing order using quick sort algorithm and overwrite P with the permutation vector
blas_spermute - blas_spermute - permutes a real array in terms of the permutation vector P , output by dsortv
blas_ssort - blas_ssort - sorts a real vector X in increasing or decreasing order using quick sort algorithm
blas_ssortv - blas_ssortv - sorts a real vector X in increasing or decreasing order using quick sort algorithm and overwrite P with the permutation vector
$\underline{\text { caxpy }- \text { caxpy }- \text { compute } y:=}$ alpha $* x+y$
$\underline{\text { caxpyi }- \text { caxpyi - Compute } y:=\text { alpha } * x+y ~}$
cbcomm - cbcomm - block coordinate matrix-matrix multiply
cbdimm - cbdimm - block diagonal format matrix-matrix multiply
cbdism - cbdism - block diagonal format triangular solve
cbdsqr - cbdsqr - compute the singular value decomposition (SVD) of a real N -by- N (upper or lower) bidiagonal matrix $B$.
cbelmm - cbelmm - block Ellpack format matrix-matrix multiply
cbelsm - cbelsm - block Ellpack format triangular solve
cbscmm - cbscmm - block sparse column matrix-matrix multiply
cbscsm - cbscsm - block sparse column format triangular solve
cbsrmm - cbsrmm - block sparse row format matrix-matrix multiply
cbsrsm - cbsrsm - block sparse row format triangular solve
cenvcor - cenvcor - compute the convolution or correlation of complex vectors
cenvcor2 - cenvcor2 - compute the convolution or correlation of complex matrices
ccoomm - ccoomm - coordinate matrix-matrix multiply
ccopy - ccopy - Copy $x$ to $y$
ccscmm - ccscmm - compressed sparse column format matrix-matrix multiply
$\underline{\text { ccscsm - ccscsm - compressed sparse column format triangular solve }}$
$\underline{\text { ccsrmm - ccsrmm - compressed sparse row format matrix-matrix multiply }}$
ccsrsm - ccsrsm - compressed sparse row format triangular solve
cdiamm - cdiamm - diagonal format matrix-matrix multiply
cdiasm - cdiasm - diagonal format triangular solve
$\underline{\text { cdotc }- \text { cdotc }- \text { compute the dot product of two vectors } \operatorname{conjg}(x) \text { and } y . ~}$
cdotci - cdotci - Compute the complex conjugated indexed dot product.
cdotu - cdotu - compute the dot product of two vectors x and y .
cdotui - cdotci - Compute the complex conjugated indexed dot product.
cellmm - cellmm - Ellpack format matrix-matrix multiply
cellsm - cellsm - Ellpack format triangular solve
$\underline{\text { cfft2b }}$ - cfft2b - compute a periodic sequence from its Fourier coefficients. The xFFT operations are unnormalized, so a call of $x$ FFT2F followed by a call of $x F F T 2 B$ will multiply the input sequence by $\mathrm{M}^{*} \mathrm{~N}$.
cfft2f - cfft2f - compute the Fourier coefficients of a periodic sequence. The xFFT operations are unnormalized, so a call of xFFT2F followed by a call of xFFT2B will multiply the input sequence by $\mathrm{M}^{*} \mathrm{~N}$.
cfft 2 i - cfft 2 i - initialize the array WSAVE, which is used in both the forward and backward transforms.
$\underline{\text { cfft } 3 \mathrm{~b}}$ - cfft3b-compute a periodic sequence from its Fourier coefficients. The FFT operations are unnormalized, so a call of CFFT3F followed by a call of CFFT3B will multiply the input sequence by $\mathrm{M}^{*} \mathrm{~N} * \mathrm{~K}$.
cfft3f - cfft3f - compute the Fourier coefficients of a periodic sequence. The FFT operations are unnormalized, so a call of CFFT3F followed by a call of CFFT3B will multiply the input sequence by $\mathrm{M}^{*} \mathrm{~N}^{*} \mathrm{~K}$.
cfft3i - cfft3i - initialize the array WSAVE, which is used in both CFFT3F and CFFT3B.
cfftb - cfftb - compute a periodic sequence from its Fourier coefficients. The FFT operations are unnormalized, so a call of CFFTF followed by a call of CFFTB will multiply the input sequence by N .
cfftc - cfftc - initialize the trigonometric weight and factor tables or compute the Fast Fourier transform (forward or inverse) of a complex sequence.
cfftc2 - cfftc2 - initialize the trigonometric weight and factor tables or compute the two-dimensional Fast Fourier Transform (forward or inverse) of a two-dimensional complex array.
cfftc3 - cfftc3 - initialize the trigonometric weight and factor tables or compute the three-dimensional Fast Fourier Transform (forward or inverse) of a three-dimensional complex array.
cfftcm - cfftcm - initialize the trigonometric weight and factor tables or compute the one-dimensional Fast Fourier Transform (forward or inverse) of a set of data sequences stored in a two-dimensional complex array.
cfftf - cfftf - compute the Fourier coefficients of a periodic sequence. The FFT operations are unnormalized, so a call of CFFTF followed by a call of CFFTB will multiply the input sequence by N .
cffti - cffti - initialize the array WSAVE, which is used in both CFFTF and CFFTB.
cfftopt - cfftopt - compute the length of the closest fast FFT
cffts - cffts - initialize the trigonometric weight and factor tables or compute the inverse Fast Fourier Transform of a complex sequence as follows.
cffts2 - cffts2 - initialize the trigonometric weight and factor tables or compute the two-dimensional inverse Fast Fourier Transform of a two-dimensional complex array.
cffts3 - cffts3 - initialize the trigonometric weight and factor tables or compute the three-dimensional inverse Fast Fourier Transform of a three-dimensional complex array.
cfftsm - cfftsm - initialize the trigonometric weight and factor tables or compute the one-dimensional inverse Fast Fourier Transform of a set of complex data sequences stored in a two-dimensional array.
cgbbrd - cgbbrd - reduce a complex general m-by-n band matrix A to real upper bidiagonal form B by a unitary transformation
cgbcon - cgbcon - estimate the reciprocal of the condition number of a complex general band matrix A , in either the 1-norm or the infinity-norm,
cgbequ - cgbequ - compute row and column scalings intended to equilibrate an M -by-N band matrix A and reduce its condition number
cgbmv - cgbmv - perform one of the matrix-vector operations $y:=a l p h a^{*} A^{*} x+$ beta*y, or $y:=a l p h a * A^{\prime} * x+$ beta*y, or y $:=\operatorname{alpha}{ }^{*} \operatorname{conjg}\left(\mathrm{~A}^{\prime}\right)^{*} \mathrm{x}+$ beta* $^{\mathrm{y}}$
cgbrfs - cgbrfs - improve the computed solution to a system of linear equations when the coefficient matrix is banded, and provides error bounds and backward error estimates for the solution
cgbsv - cgbsv - compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$, where A is a band matrix of order N with KL subdiagonals and KU superdiagonals, and X and B are N -by-NRHS matrices
cgbsvx - cgbsvx - use the LU factorization to compute the solution to a complex system of linear equations $A * X$ $=\mathrm{B}, \mathrm{A}^{*} * \mathrm{~T} * \mathrm{X}=\mathrm{B}$, or $\mathrm{A}^{*} \mathrm{H}^{*} \mathrm{X}=\mathrm{B}$,
cgbtf2 - cgbtf2 - compute an LU factorization of a complex m-by-n band matrix A using partial pivoting with row interchanges
cgbtrf - cgbtrf - compute an LU factorization of a complex m-by-n band matrix A using partial pivoting with row interchanges
cgbtrs - cgbtrs - solve a system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}, \mathrm{A} * * \mathrm{~T} * \mathrm{X}=\mathrm{B}$, or $\mathrm{A}^{*} * \mathrm{H} * \mathrm{X}=\mathrm{B}$ with a general band matrix A using the LU factorization computed by CGBTRF
cgebak - cgebak - form the right or left eigenvectors of a complex general matrix by backward transformation on the computed eigenvectors of the balanced matrix output by CGEBAL
cgebal - cgebal - balance a general complex matrix A
cgebrd - cgebrd - reduce a general complex M-by-N matrix A to upper or lower bidiagonal form B by a unitary transformation
cgecon - cgecon - estimate the reciprocal of the condition number of a general complex matrix A , in either the 1norm or the infinity-norm, using the LU factorization computed by CGETRF
cgeequ - cgeequ - compute row and column scalings intended to equilibrate an M-by-N matrix A and reduce its condition number
cgees - cgees - compute for an N -by-N complex nonsymmetric matrix A, the eigenvalues, the Schur form T, and, optionally, the matrix of Schur vectors Z
cgeesx - cgeesx - compute for an N-by-N complex nonsymmetric matrix A, the eigenvalues, the Schur form T, and, optionally, the matrix of Schur vectors $Z$
cgeev - cgeev - compute for an N -by- N complex nonsymmetric matrix A , the eigenvalues and, optionally, the left and/or right eigenvectors
cgeevx - cgeevx - compute for an N -by-N complex nonsymmetric matrix A , the eigenvalues and, optionally, the left and/or right eigenvectors
cgegs - cgegs - routine is deprecated and has been replaced by routine CGGES
cgegv - cgegv - routine is deprecated and has been replaced by routine CGGEV
cgehrd - cgehrd - reduce a complex general matrix A to upper Hessenberg form H by a unitary similarity transformation
cgelqf - cgelqf - compute an LQ factorization of a complex M-by-N matrix A
cgels - cgels - solve overdetermined or underdetermined complex linear systems involving an M-by-N matrix A, or its conjugate-transpose, using a QR or LQ factorization of A
cgelsd - cgelsd - compute the minimum-norm solution to a real linear least squares problem
cgelss - cgelss - compute the minimum norm solution to a complex linear least squares problem
cgelsx - cgelsx - routine is deprecated and has been replaced by routine CGELSY
cgelsy - cgelsy - compute the minimum-norm solution to a complex linear least squares problem
cgemm - cgemm - perform one of the matrix-matrix operations $C:=$ alpha*op( A )*op( B ) + beta*C
cgemv - cgemv - perform one of the matrix-vector operations y := alpha*A*x + beta*y, or y := alpha*A'*x+ beta* y , or $\mathrm{y}:=\operatorname{alpha}{ }^{*} \operatorname{conjg}\left(\mathrm{~A}^{\prime}\right)^{*} \mathrm{x}+$ beta* $^{\mathrm{y}}$
cgeqlf - cgeqlf - compute a QL factorization of a complex M-by-N matrix A
cgeqp3 - cgeqp3 - compute a QR factorization with column pivoting of a matrix A
cgeqpf - cgeqpf - routine is deprecated and has been replaced by routine CGEQP3
cgeqrf - cgeqrf - compute a QR factorization of a complex M -by-N matrix A
cgerc - cgerc - perform the rank 1 operation $\mathrm{A}:=\operatorname{alpha}{ }^{*} x^{*} \operatorname{conjg}\left(y^{\prime}\right)+A$
cgerfs - cgerfs - improve the computed solution to a system of linear equations and provides error bounds and backward error estimates for the solution
cgerqf - cgerqf - compute an RQ factorization of a complex M-by-N matrix A
cgeru - cgeru - perform the rank 1 operation A := alpha*x*y' + A
cgesdd - cgesdd - compute the singular value decomposition (SVD) of a complex M-by-N matrix A, optionally computing the left and/or right singular vectors, by using divide-and-conquer method
cgesv - cgesv - compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
cgesvd - cgesvd - compute the singular value decomposition (SVD) of a complex M-by-N matrix A, optionally computing the left and/or right singular vectors
cgesvx - cgesvx - use the LU factorization to compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}$ $=\mathrm{B}$,
cgetf2 - cgetf2 - compute an LU factorization of a general m-by-n matrix A using partial pivoting with row interchanges
cgetrf - cgetrf - compute an LU factorization of a general M-by-N matrix A using partial pivoting with row interchanges
cgetri - cgetri - compute the inverse of a matrix using the LU factorization computed by CGETRF
cgetrs - cgetrs - solve a system of linear equations $A * X=B, A^{*} * T * X=B$, or $A^{* *} H * X=B$ with a general $N-$ by-N matrix A using the LU factorization computed by CGETRF
cggbak - cggbak - form the right or left eigenvectors of a complex generalized eigenvalue problem $A^{*} \mathrm{x}=$ lambda*B*x, by backward transformation on the computed eigenvectors of the balanced pair of matrices output by CGGBAL
cggbal - cggbal - balance a pair of general complex matrices (A,B)
cgges - cgges - compute for a pair of $\mathrm{N}-\mathrm{by}-\mathrm{N}$ complex nonsymmetric matrices ( $\mathrm{A}, \mathrm{B}$ ), the generalized eigenvalues, the generalized complex Schur form (S, T), and optionally left and/or right Schur vectors (VSL and VSR)
cggesx - cggesx - compute for a pair of $N-b y-N$ complex nonsymmetric matrices (A,B), the generalized eigenvalues, the complex Schur form (S,T),
cggev - cggev - compute for a pair of N -by- N complex nonsymmetric matrices ( $\mathrm{A}, \mathrm{B}$ ), the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors
cggevx - cggevx - compute for a pair of $N$-by-N complex nonsymmetric matrices ( $\mathrm{A}, \mathrm{B}$ ) the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors
cggglm - cggglm - solve a general Gauss-Markov linear model (GLM) problem
cgghrd - cgghrd - reduce a pair of complex matrices (A,B) to generalized upper Hessenberg form using unitary transformations, where A is a general matrix and B is upper triangular
cgglse - cgglse - solve the linear equality-constrained least squares (LSE) problem
cggqrf - cggqrf - compute a generalized QR factorization of an N -by-M matrix A and an N -by-P matrix B .
cggrqf - cggrqf - compute a generalized RQ factorization of an M-by-N matrix A and a P-by-N matrix B
cggsvd - cggsvd - compute the generalized singular value decomposition (GSVD) of an M-by-N complex matrix A and P-by-N complex matrix B
cggsvp - cggsvp - compute unitary matrices $\mathrm{U}, \mathrm{V}$ and Q such that $\mathrm{N}-\mathrm{K}-\mathrm{L} K \mathrm{~L} \mathrm{U}^{*} * \mathrm{~A} * \mathrm{Q}=\mathrm{K}(0 \mathrm{~A} 12 \mathrm{~A} 13)$ if $\mathrm{M}-\mathrm{K}-$ $\mathrm{L}>=0$
cgssco - cgssco - General sparse solver condition number estimate.
cgssda - cgssda - Deallocate working storage for the general sparse solver.
cgssfa - cgssfa - General sparse solver numeric factorization.
cgssfs - cgssfs - General sparse solver one call interface.
cgssin - cgssin - Initialize the general sparse solver.
cgssor - cgssor - General sparse solver ordering and symbolic factorization.
cgssps - cgssps - Print general sparse solver statics.
cgssrp - cgssrp - Return permutation used by the general sparse solver.
cgsssl - cgsssl - Solve routine for the general sparse solver.
cgssuo - cgssuo - User supplied permutation for ordering used in the general sparse solver.
cgtcon - cgtcon - estimate the reciprocal of the condition number of a complex tridiagonal matrix A using the LU factorization as computed by CGTTRF
cgthr - cgthr - Gathers specified elements from y into $x$.
cgthrz - cgthrz - Gather and zero.
cgtrfs - cgtrfs - improve the computed solution to a system of linear equations when the coefficient matrix is tridiagonal, and provides error bounds and backward error estimates for the solution
cgtsv - cgtsv - solve the equation $\mathrm{A}^{*} \mathrm{X}=\mathrm{B}$,
cgtsvx - cgtsvx - use the LU factorization to compute the solution to a complex system of linear equations $A * X$ $=\mathrm{B}, \mathrm{A} * * \mathrm{~T} * \mathrm{X}=\mathrm{B}$, or $\mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B}$,
cgttrf - cgttrf - compute an LU factorization of a complex tridiagonal matrix A using elimination with partial pivoting and row interchanges
cgttrs - cgttrs - solve one of the systems of equations $\mathrm{A} * \mathrm{X}=\mathrm{B}, \mathrm{A}^{*} * \mathrm{~T} * \mathrm{X}=\mathrm{B}$, or $\mathrm{A}^{*}{ }^{*} \mathrm{H}^{*} \mathrm{X}=\mathrm{B}$,
$\underline{\text { chbev - chbev - compute all the eigenvalues and, optionally, eigenvectors of a complex Hermitian band matrix A }}$
chbevd - chbevd - compute all the eigenvalues and, optionally, eigenvectors of a complex Hermitian band matrix A
chbevx - chbevx - compute selected eigenvalues and, optionally, eigenvectors of a complex Hermitian band matrix A
chbgst - chbgst - reduce a complex Hermitian-definite banded generalized eigenproblem $A * x=l a m b d a * B * x$ to standard form $\mathrm{C}^{*} \mathrm{y}=$ lambda* y ,
chbgv - chbgv - compute all the eigenvalues, and optionally, the eigenvectors of a complex generalized Hermitiandefinite banded eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\operatorname{lambda}) * \mathrm{~B} * \mathrm{x}$
chbgvd - chbgvd - compute all the eigenvalues, and optionally, the eigenvectors of a complex generalized Hermitian-definite banded eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{~B}^{*} \mathrm{x}$
chbgvx - chbgvx - compute all the eigenvalues, and optionally, the eigenvectors of a complex generalized Hermitian-definite banded eigenproblem, of the form $A * x=($ lambda $) * B * x$

chbtrd - chbtrd - reduce a complex Hermitian band matrix A to real symmetric tridiagonal form T by a unitary similarity transformation
checon - checon - estimate the reciprocal of the condition number of a complex Hermitian matrix A using the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D}^{*} \mathrm{U}^{* *} \mathrm{H}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D}^{*} \mathrm{~L}^{* *} \mathrm{H}$ computed by CHETRF
$\underline{\text { cheev - cheev - compute all eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix A }}$
cheevd - cheevd - compute all eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix A
cheevr - cheevr - compute selected eigenvalues and, optionally, eigenvectors of a complex Hermitian tridiagonal matrix T
cheevx - cheevx - compute selected eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix A
chegs2 - chegs2 - reduce a complex Hermitian-definite generalized eigenproblem to standard form
chegst - chegst - reduce a complex Hermitian-definite generalized eigenproblem to standard form
chegv - chegv - compute all the eigenvalues, and optionally, the eigenvectors of a complex generalized Hermitiandefinite eigenproblem, of the form $\mathrm{A}^{*} \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{~B}^{*} \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda})^{*} \mathrm{x}$, or $\mathrm{B}^{*} \mathrm{~A}^{*} \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{x}$
chegvd - chegvd - compute all the eigenvalues, and optionally, the eigenvectors of a complex generalized

Hermitian-definite eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{~B} * \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda})^{*} \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lambda}) * \mathrm{x}$
chegvx - chegvx - compute selected eigenvalues, and optionally, eigenvectors of a complex generalized Hermitian-definite eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{~B}^{*} \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda})^{*} \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lambda}) * \mathrm{x}$
chemm - chemm - perform one of the matrix-matrix operations $\mathrm{C}:=$ alpha*A* $\mathrm{B}+$ beta* $^{*} \mathrm{C}$ or $\mathrm{C}:=$ alpha* $\mathrm{B}^{*} \mathrm{~A}+$ beta*C
$\underline{\text { chemv }}$ - chemv - perform the matrix-vector operation $\mathrm{y}:=\operatorname{alpha*} \mathrm{A}^{*} \mathrm{x}+$ beta* $^{\mathrm{y}}$
cher - cher - perform the hermitian rank 1 operation $\mathrm{A}:=\operatorname{alpha} \mathrm{x}^{*} \operatorname{conjg}\left(\mathrm{x}^{\prime}\right)+\mathrm{A}$
cher2 - cher2 - perform the hermitian rank 2 operation $\mathrm{A}:=\operatorname{alpha} *^{*}{ }^{*} \operatorname{conjg}\left(\mathrm{y}^{\prime}\right)+\operatorname{conjg}(\operatorname{alpha}) * \mathrm{y}^{*} \operatorname{conjg}\left(\mathrm{x}^{\prime}\right)+\mathrm{A}$
cher $2 \mathrm{k}-$ cher 2 k - perform one of the Hermitian rank 2 k operations $\mathrm{C}:=\operatorname{alpha} \mathrm{A}^{*}$ *conjg( $\left.\mathrm{B}^{\prime}\right)+\operatorname{conjg}($ alpha $)$ *B*conjg ( $\mathrm{A}^{\prime}$ ) + beta* C or $\mathrm{C}:=\operatorname{alpha*} \operatorname{conjg}\left(\mathrm{A}^{\prime}\right) * \mathrm{~B}+\operatorname{conjg}(\operatorname{alpha}) * \operatorname{conjg}\left(\mathrm{~B}^{\prime}\right) * \mathrm{~A}+$ beta$^{*} \mathrm{C}$
cherfs - cherfs - improve the computed solution to a system of linear equations when the coefficient matrix is Hermitian indefinite, and provides error bounds and backward error estimates for the solution
cherk - cherk - perform one of the Hermitian rank k operations $C:=\operatorname{alpha*} A^{*} \operatorname{conjg}\left(\mathrm{~A}^{\prime}\right)+$ beta* C or $\mathrm{C}:=$ alpha*conjg( $\left.\mathrm{A}^{\prime}\right)^{*} \mathrm{~A}+$ beta* $^{*} \mathrm{C}$
chesv - chesv - compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
chesvx - chesvx - use the diagonal pivoting factorization to compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
chetf2 - chetf2 - compute the factorization of a complex Hermitian matrix A using the Bunch-Kaufman diagonal pivoting method
chetrd - chetrd - reduce a complex Hermitian matrix A to real symmetric tridiagonal form T by a unitary similarity transformation
chetrf - chetrf - compute the factorization of a complex Hermitian matrix A using the Bunch-Kaufman diagonal pivoting method
chetri - chetri - compute the inverse of a complex Hermitian indefinite matrix A using the factorization $\mathrm{A}=$ $\mathrm{U}^{*} \mathrm{D}^{*} \mathrm{U}^{* *} \mathrm{H}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{D}^{*} \mathrm{~L}^{* *} \mathrm{H}$ computed by CHETRF
chetrs - chetrs - solve a system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$ with a complex Hermitian matrix A using the factorization $\mathrm{A}=\mathrm{U}^{*} \mathrm{D}^{*} \mathrm{U}^{*} * \mathrm{H}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{D}^{*} \mathrm{~L}^{* *} \mathrm{H}$ computed by CHETRF
chgeqz - chgeqz - implement a single-shift version of the QZ method for finding the generalized eigenvalues $\mathrm{w}(\mathrm{i})$ $=\operatorname{ALPHA}(\mathrm{i}) / \mathrm{BETA}(\mathrm{i})$ of the equation $\operatorname{det}(\mathrm{A}-\mathrm{w}(\mathrm{i}) \mathrm{B})=0$ If $\mathrm{JOB}=$ 'S', then the pair $(\mathrm{A}, \mathrm{B})$ is simultaneously reduced to Schur form (i.e., A and B are both upper triangular) by applying one unitary tranformation (usually called Q) on the left and another (usually called Z) on the right
chpcon - chpcon - estimate the reciprocal of the condition number of a complex Hermitian packed matrix A using the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D}^{*} \mathrm{U}^{* *} \mathrm{H}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D}^{*} \mathrm{~L}^{* *} \mathrm{H}$ computed by CHPTRF
$\underline{\text { chpev - chpev - compute all the eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix in }}$ packed storage
chpevd - chpevd - compute all the eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix A in packed storage
chpevx - chpevx - compute selected eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix A in packed storage
chpgst - chpgst - reduce a complex Hermitian-definite generalized eigenproblem to standard form, using packed storage
chpgv - chpgv - compute all the eigenvalues and, optionally, the eigenvectors of a complex generalized Hermitiandefinite eigenproblem, of the form $\mathrm{A}^{*} \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{~B}^{*} \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda}) * \mathrm{x}$, or $\mathrm{B}^{*} \mathrm{~A} * \mathrm{x}=(\operatorname{lambda}) * \mathrm{x}$
chpgvd - chpgvd - compute all the eigenvalues and, optionally, the eigenvectors of a complex generalized Hermitian-definite eigenproblem, of the form $\mathrm{A}^{*} \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{~B}^{*} \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda})^{*} \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{x}$
chpgvx - chpgvx - compute selected eigenvalues and, optionally, eigenvectors of a complex generalized Hermitian-definite eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\operatorname{lambda}){ }^{*} \mathrm{~B}^{*} \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda})^{*} \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{x}$
$\underline{\text { chpmv }}$ - chpmv - perform the matrix-vector operation $\mathrm{y}:=$ alpha* $\mathrm{A} * \mathrm{x}+$ beta*y
$\underline{\text { chpr }}$ - chpr - perform the hermitian rank 1 operation $\mathrm{A}:=\operatorname{alpha}{ }^{*} \mathrm{x}^{*} \operatorname{conjg}\left(\mathrm{x}^{\prime}\right)+\mathrm{A}$
chpr2 - chpr2 - perform the Hermitian rank 2 operation $A:=\operatorname{alpha}{ }^{*} x^{*} \operatorname{conjg}\left(y^{\prime}\right)+\operatorname{conjg}($ alpha $) * y * \operatorname{conjg}\left(x^{\prime}\right)+$ A
chprfs - chprfs - improve the computed solution to a system of linear equations when the coefficient matrix is Hermitian indefinite and packed, and provides error bounds and backward error estimates for the solution
chpsv - chpsv - compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
chpsvx - chpsvx - use the diagonal pivoting factorization $A=U^{*} D^{*} U^{* *} H$ or $A=L^{*} D^{*} L^{* *} H$ to compute the solution to a complex system of linear equations $A * X=B$, where $A$ is an $N$-by-N Hermitian matrix stored in packed format and X and B are N -by-NRHS matrices
chptrd - chptrd - reduce a complex Hermitian matrix A stored in packed form to real symmetric tridiagonal form T by a unitary similarity transformation
chptrf - chptrf - compute the factorization of a complex Hermitian packed matrix A using the Bunch-Kaufman diagonal pivoting method
chptri - chptri - compute the inverse of a complex Hermitian indefinite matrix A in packed storage using the factorization $\mathrm{A}=\mathrm{U}^{*} \mathrm{D} * \mathrm{U}^{* *} \mathrm{H}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D}^{*} \mathrm{~L}^{* *} \mathrm{H}$ computed by CHPTRF
chptrs - chptrs - solve a system of linear equations $A * X=B$ with a complex Hermitian matrix A stored in packed format using the factorization $A=U^{*} D^{*} U^{* *} H$ or $A=L^{*} D^{*} L^{* *} H$ computed by CHPTRF
chsein - chsein - use inverse iteration to find specified right and/or left eigenvectors of a complex upper Hessenberg matrix H
chseqr - chseqr - compute the eigenvalues of a complex upper Hessenberg matrix H , and, optionally, the matrices T and Z from the Schur decomposition $\mathrm{H}=\mathrm{Z} \mathrm{T} \mathrm{Z}{ }^{* *} \mathrm{H}$, where T is an upper triangular matrix (the Schur form), and Z is the unitary matrix of Schur vectors
cjadmm - cjadmm - Jagged diagonal matrix-matrix multiply (modified Ellpack)
cjadrp - cjadrp - right permutation of a jagged diagonal matrix
cjadsm - cjadsm - Jagged-diagonal format triangular solve
$\underline{\text { clarz - clarz - applie a complex elementary reflector } \mathrm{H} \text { to a complex } \mathrm{M} \text {-by-N matrix } \mathrm{C} \text {, from either the left or the }}$ right
 from the left or the right
clarzt - clarzt - form the triangular factor T of a complex block reflector H of order $>\mathrm{n}$, which is defined as a product of $k$ elementary reflectors
clatzm - clatzm - routine is deprecated and has been replaced by routine CUNMRZ
$\underline{\text { cosqb - cosqb - synthesize a Fourier sequence from its representation in terms of a cosine series with odd wave }}$ numbers. The COSQ operations are unnormalized inverses of themselves, so a call to COSQF followed by a call to COSQB will multiply the input sequence by $4 * \mathrm{~N}$.
cosqf - cosqf - compute the Fourier coefficients in a cosine series representation with only odd wave numbers. The COSQ operations are unnormalized inverses of themselves, so a call to COSQF followed by a call to COSQB will multiply the input sequence by $4 * \mathrm{~N}$.
cosqi - cosqi - initialize the array WSAVE, which is used in both COSQF and COSQB.
cost - cost - compute the discrete Fourier cosine transform of an even sequence. The COST transforms are unnormalized inverses of themselves, so a call of COST followed by another call of COST will multiply the input sequence by $2 *(\mathrm{~N}-1)$.
$\underline{\text { costi - costi - initialize the array WSAVE, which is used in COST. }}$
cpbcon - cpbcon - estimate the reciprocal of the condition number (in the 1-norm) of a complex Hermitian positive definite band matrix using the Cholesky factorization $A=U^{*} * H^{*} U$ or $A=L^{*} L^{* *} H$ computed by CPBTRF
cpbequ - cpbequ - compute row and column scalings intended to equilibrate a Hermitian positive definite band matrix A and reduce its condition number (with respect to the two-norm)
cpbrfs - cpbrfs - improve the computed solution to a system of linear equations when the coefficient matrix is Hermitian positive definite and banded, and provides error bounds and backward error estimates for the solution cpbstf - cpbstf - compute a split Cholesky factorization of a complex Hermitian positive definite band matrix A cpbsv - cpbsv - compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
cpbsvx - cpbsvx - use the Cholesky factorization $A=U^{* *} H^{*} U$ or $A=L^{*} L^{* *} H$ to compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
cpbtf2 - cpbtf2 - compute the Cholesky factorization of a complex Hermitian positive definite band matrix A
cpbtrf - cpbtrf - compute the Cholesky factorization of a complex Hermitian positive definite band matrix A
cpbtrs - cpbtrs - solve a system of linear equations $A * X=B$ with a Hermitian positive definite band matrix $A$ using the Cholesky factorization $A=U^{* *} H^{*} U$ or $A=L^{*} L^{* *} H$ computed by CPBTRF
cpocon - cpocon - estimate the reciprocal of the condition number (in the 1-norm) of a complex Hermitian positive definite matrix using the Cholesky factorization $A=U^{* *} H^{*} U$ or $A=L^{*} L^{* *} \mathrm{H}$ computed by CPOTRF
cpoequ - cpoequ - compute row and column scalings intended to equilibrate a Hermitian positive definite matrix A and reduce its condition number (with respect to the two-norm)
cporfs - cporfs - improve the computed solution to a system of linear equations when the coefficient matrix is Hermitian positive definite,
cposv - cposv - compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
cposvx - cposvx - use the Cholesky factorization $A=U^{* *} H^{*} U$ or $A=L^{*} L^{* *} H$ to compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
cpotf2 - cpotf2 - compute the Cholesky factorization of a complex Hermitian positive definite matrix A
cpotrf - cpotrf - compute the Cholesky factorization of a complex Hermitian positive definite matrix A
cpotri - cpotri - compute the inverse of a complex Hermitian positive definite matrix A using the Cholesky factorization $\mathrm{A}=\mathrm{U}^{* *} \mathrm{H}^{*} \mathrm{U}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{~L}^{* *} \mathrm{H}$ computed by CPOTRF
cpotrs - cpotrs - solve a system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$ with a Hermitian positive definite matrix A using the Cholesky factorization $A=U^{* *} H^{*} U$ or $A=L^{*} L^{* *} H$ computed by CPOTRF
cppcon - cppcon - estimate the reciprocal of the condition number (in the 1-norm) of a complex Hermitian positive definite packed matrix using the Cholesky factorization $A=U^{* *} H^{*} U$ or $A=L^{*} L^{* *} H$ computed by CPPTRF
cppequ - cppequ - compute row and column scalings intended to equilibrate a Hermitian positive definite matrix A in packed storage and reduce its condition number (with respect to the two-norm)
cpprfs - cpprfs - improve the computed solution to a system of linear equations when the coefficient matrix is Hermitian positive definite and packed, and provides error bounds and backward error estimates for the solution
cppsv - cppsv - compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
cppsvx - cppsvx - use the Cholesky factorization $A=U^{*} * H^{*} U$ or $A=L^{*} L^{* *} \mathrm{H}$ to compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
cpptrf - cpptrf - compute the Cholesky factorization of a complex Hermitian positive definite matrix A stored in packed format
cpptri - cpptri - compute the inverse of a complex Hermitian positive definite matrix A using the Cholesky factorization $\mathrm{A}=\mathrm{U}^{* *} \mathrm{H}^{*} \mathrm{U}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{~L}^{* *} \mathrm{H}$ computed by CPPTRF
cpptrs - cpptrs - solve a system of linear equations A*X $=\mathrm{B}$ with a Hermitian positive definite matrix A in packed storage using the Cholesky factorization $A=U^{* *} H^{*} U$ or $A=L^{*} L^{* *} H$ computed by CPPTRF
cptcon - cptcon - compute the reciprocal of the condition number (in the 1-norm) of a complex Hermitian positive definite tridiagonal matrix using the factorization $A=L^{*} D^{*} L^{* *} H$ or $A=U^{* *} H^{*} D^{*} U$ computed by CPTTRF
cpteqr - cpteqr - compute all eigenvalues and, optionally, eigenvectors of a symmetric positive definite tridiagonal matrix by first factoring the matrix using SPTTRF and then calling CBDSQR to compute the singular values of the bidiagonal factor
cptrfs - cptrfs - improve the computed solution to a system of linear equations when the coefficient matrix is Hermitian positive definite and tridiagonal, and provides error bounds and backward error estimates for the solution
cptsv - cptsv - compute the solution to a complex system of linear equations $A * X=B$, where $A$ is an $N-b y-N$ Hermitian positive definite tridiagonal matrix, and X and B are N -by-NRHS matrices.
cptsvx - cptsvx - use the factorization $A=L * D^{*} L^{* *} H$ to compute the solution to a complex system of linear equations $A^{*} X=B$, where $A$ is an $N$-by-N Hermitian positive definite tridiagonal matrix and $X$ and $B$ are N -byNRHS matrices
cpttrf - cpttrf - compute the $L^{*} D^{*} L^{\prime}$ factorization of a complex Hermitian positive definite tridiagonal matrix $A$
cpttrs - cpttrs - solve a tridiagonal system of the form $A * X=B$ using the factorization $A=U^{*} * D^{*} U$ or $A=$ L*D*L' computed by CPTTRF
cptts2 - cptts2 - solve a tridiagonal system of the form $A * X=B$ using the factorization $A=U^{\prime} * D^{*} U$ or $A=$ L*D*L' computed by CPTTRF
crot - crot - apply a plane rotation, where the $\cos (\mathrm{C})$ is real and the $\sin (\mathrm{S})$ is complex, and the vectors X and Y are complex
crotg - crotg - Construct a Given's plane rotation
cscal - cscal - Compute y $:=$ alpha * y
$\underline{\text { csctr }}$ - csctr - Scatters elements from $x$ into $y$.
cskymm - cskymm - Skyline format matrix-matrix multiply
cskysm - cskysm - Skyline format triangular solve
cspcon - cspcon - estimate the reciprocal of the condition number (in the 1-norm) of a complex symmetric packed matrix A using the factorization $\mathrm{A}=\mathrm{U}^{*} \mathrm{D}^{*} \mathrm{U}^{* *} \mathrm{~T}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{D}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ computed by CSPTRF
csprfs - csprfs - improve the computed solution to a system of linear equations when the coefficient matrix is symmetric indefinite and packed, and provides error bounds and backward error estimates for the solution

cspsvx - cspsvx - use the diagonal pivoting factorization $A=U^{*} D^{*} U^{*} * T$ or $A=L^{*} D^{*} L^{* *} T$ to compute the solution to a complex system of linear equations $A * X=B$, where $A$ is an $N$-by-N symmetric matrix stored in
packed format and X and B are N -by-NRHS matrices
csptrf - csptrf - compute the factorization of a complex symmetric matrix A stored in packed format using the Bunch-Kaufman diagonal pivoting method
csptri - csptri - compute the inverse of a complex symmetric indefinite matrix A in packed storage using the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D}^{*} \mathrm{U}^{*} * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{D}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ computed by CSPTRF
csptrs - csptrs - solve a system of linear equations A*X $=\mathrm{B}$ with a complex symmetric matrix A stored in packed format using the factorization $\mathrm{A}=\mathrm{U}^{*} \mathrm{D}^{*} \mathrm{U}^{*} * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D}^{*} \mathrm{~L} * * \mathrm{~T}$ computed by CSPTRF
csrot - csrot - Apply a plane rotation.
csscal - csscal - Compute y $:=$ alpha * y
cstedc - cstedc - compute all eigenvalues and, optionally, eigenvectors of a symmetric tridiagonal matrix using the divide and conquer method
cstegr - cstegr - Compute T-sigma_i = L_i D_i L_i^T, such that L_i D_i L_i^T is a relatively robust representation
cstein - cstein - compute the eigenvectors of a real symmetric tridiagonal matrix T corresponding to specified eigenvalues, using inverse iteration
csteqr - csteqr - compute all eigenvalues and, optionally, eigenvectors of a symmetric tridiagonal matrix using the implicit QL or QR method
cstsv - cstsv - compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$ where A is a Hermitian tridiagonal matrix
csttrf - csttrf - compute the factorization of a complex Hermitian tridiagonal matrix A
$\underline{\text { csttrs }}$ - csttrs - computes the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$
cswap - cswap - Exchange vectors x and y .
csycon - csycon - estimate the reciprocal of the condition number (in the 1-norm) of a complex symmetric matrix A using the factorization $A=U^{*} D^{*} U^{* *} T$ or $A=L^{*} D^{*} L^{* *} T$ computed by CSYTRF
csymm - csymm - perform one of the matrix-matrix operations $\mathrm{C}:=$ alpha*A*B + beta* C or $\mathrm{C}:=$ alpha*B*A + beta*C
$\underline{\text { csyr } 2 \mathrm{k}}-\operatorname{csyr} 2 \mathrm{k}$ - perform one of the symmetric rank 2 k operations $\mathrm{C}:=\operatorname{alpha} \mathrm{A}^{*} \mathrm{~A}^{\prime}+\operatorname{alpha} \mathrm{B}^{*} \mathrm{~A}^{\prime}+$ beta* C or C :
$=$ alpha* ${ }^{\prime} *$ B + alpha* ${ }^{\prime} * A+$ beta $^{*} \mathrm{C}$
csyrfs - csyrfs - improve the computed solution to a system of linear equations when the coefficient matrix is symmetric indefinite, and provides error bounds and backward error estimates for the solution
 beta*C

csysvx - csysvx - use the diagonal pivoting factorization to compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
csytf2 - csytf2 - compute the factorization of a complex symmetric matrix A using the Bunch-Kaufman diagonal pivoting method
csytrf - csytrf - compute the factorization of a complex symmetric matrix A using the Bunch-Kaufman diagonal pivoting method
csytri - csytri - compute the inverse of a complex symmetric indefinite matrix A using the factorization $\mathrm{A}=$ $\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ computed by CSYTRF
csytrs - csytrs - solve a system of linear equations $A * X=B$ with a complex symmetric matrix $A$ using the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D}^{*} \mathrm{U} * * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ computed by CSYTRF
ctbcon - ctbcon - estimate the reciprocal of the condition number of a triangular band matrix A , in either the 1norm or the infinity-norm

$\underline{\text { ctbrfs - ctbrfs - provide error bounds and backward error estimates for the solution to a system of linear equations }}$ with a triangular band coefficient matrix
$\underline{\text { ctbsv }}$ - ctbsv - solve one of the systems of equations $A^{*} x=b$, or $A^{\prime} * x=b$, or $\operatorname{conjg}\left(A^{\prime}\right) * x=b$
$\underline{\text { ctbtrs }}$ - ctbtrs - solve a triangular system of the form $\mathrm{A} * \mathrm{X}=\mathrm{B}, \mathrm{A}^{*} * \mathrm{~T} * \mathrm{X}=\mathrm{B}$, or $\mathrm{A}^{* *} \mathrm{H} * \mathrm{X}=\mathrm{B}$,
ctgevc - ctgevc - compute some or all of the right and/or left generalized eigenvectors of a pair of complex upper triangular matrices (A,B)
ctgexc - ctgexc - reorder the generalized Schur decomposition of a complex matrix pair ( $\mathrm{A}, \mathrm{B}$ ), using an unitary equivalence transformation $(A, B):=Q *(A, B) * Z$, so that the diagonal block of $(A, B)$ with row index IFST is moved to row ILST
ctgsen - ctgsen - reorder the generalized Schur decomposition of a complex matrix pair (A, B) (in terms of an unitary equivalence trans- formation $\left.\mathrm{Q}^{\prime} *(\mathrm{~A}, \mathrm{~B}) * \mathrm{Z}\right)$, so that a selected cluster of eigenvalues appears in the leading diagonal blocks of the pair $(\mathrm{A}, \mathrm{B})$
ctgsja - ctgsja - compute the generalized singular value decomposition (GSVD) of two complex upper triangular (or trapezoidal) matrices A and B
ctgsna - ctgsna - estimate reciprocal condition numbers for specified eigenvalues and/or eigenvectors of a matrix pair (A, B)
ctgsyl - ctgsyl - solve the generalized Sylvester equation
ctpcon - ctpcon - estimate the reciprocal of the condition number of a packed triangular matrix A , in either the 1norm or the infinity-norm
$\underline{\text { ctpmv }}$ - ctpmv - perform one of the matrix-vector operations $x:=A^{*} x$, or $x:=A^{\prime *} x$, or $x:=\operatorname{conjg}\left(A^{\prime}\right)^{*} x$
ctprfs - ctprfs - provide error bounds and backward error estimates for the solution to a system of linear equations with a triangular packed coefficient matrix
$\underline{\text { ctpsv }}$ - ctpsv - solve one of the systems of equations $A^{*} x=b$, or $A^{\prime} * x=b$, or $\operatorname{conjg}\left(A^{\prime}\right)^{*} x=b$
ctptri - ctptri - compute the inverse of a complex upper or lower triangular matrix A stored in packed format
ctptrs - ctptrs - solve a triangular system of the form $\mathrm{A}^{*} \mathrm{X}=\mathrm{B}, \mathrm{A}^{*} * \mathrm{~T} * \mathrm{X}=\mathrm{B}$, or $\mathrm{A}^{*} * \mathrm{H} * \mathrm{X}=\mathrm{B}$,
ctrans - ctrans - transpose and scale source matrix
ctrcon - ctrcon - estimate the reciprocal of the condition number of a triangular matrix A , in either the 1-norm or the infinity-norm

ctrexc - ctrexc - reorder the Schur factorization of a complex matrix $\mathrm{A}=\mathrm{Q}^{*} \mathrm{~T}^{*} \mathrm{Q} * * \mathrm{H}$, so that the diagonal element of T with row index IFST is moved to row ILST
ctrmm - ctrmm - perform one of the matrix-matrix operations B :=alpha*op(A)*B, or B :=alpha*B*op(A ) where alpha is a scalar, $B$ is an $m$ by $n$ matrix, $A$ is a unit, or non-unit, upper or lower triangular matrix and op $(A)$ is one of op $(A)=A$ or op( $A)=A^{\prime}$ or op $(A)=\operatorname{conjg}\left(A^{\prime}\right)$
$\underline{\text { ctrmv }}-\operatorname{ctrmv}$ - perform one of the matrix-vector operations $x:=A^{*} x$, or $x:=A^{\prime}{ }^{*} x$, or $x:=\operatorname{conjg}\left(A^{\prime}\right)^{*} x$
ctrrfs - ctrrfs - provide error bounds and backward error estimates for the solution to a system of linear equations with a triangular coefficient matrix
 eigenvalues appears in the leading positions on the diagonal of the upper triangular matrix T , and the leading columns of Q form an orthonormal basis of the corresponding right invariant subspace
ctrsm - ctrsm - solve one of the matrix equations op (A)*X $=$ alpha*B, or $X^{*} o p(A)=$ alpha*B
ctrsna - ctrsna - estimate reciprocal condition numbers for specified eigenvalues and/or right eigenvectors of a complex upper triangular matrix T (or of any matrix $\mathrm{Q}^{*} \mathrm{~T}^{*} \mathrm{Q}^{* *} \mathrm{H}$ with Q unitary)
$\underline{\text { ctrsv }}-\operatorname{ctrsv}$ - solve one of the systems of equations $A^{*} x=b$, or $A^{\prime *} x=b$, or $\operatorname{conjg}\left(A^{\prime}\right)^{*} x=b$
ctrsyl - ctrsyl - solve the complex Sylvester matrix equation
$\underline{\text { ctrti2 }}$ - ctrti2 - compute the inverse of a complex upper or lower triangular matrix
ctrtri - ctrtri - compute the inverse of a complex upper or lower triangular matrix A
$\underline{\text { ctrtrs }-\operatorname{ctrtrs}-\text { solve a triangular system of the form } \mathrm{A} * \mathrm{X}=\mathrm{B}, \mathrm{A}^{*} * \mathrm{~T} * \mathrm{X}=\mathrm{B}, \text { or } \mathrm{A}^{*} * \mathrm{H} * \mathrm{X}=\mathrm{B}, ~}$
ctzrqf - ctzrqf - routine is deprecated and has been replaced by routine CTZRZF
ctzrzf - ctzrzf - reduce the M-by-N ( $\mathrm{M}<=\mathrm{N}$ ) complex upper trapezoidal matrix A to upper triangular form by means of unitary transformations
cung21 - cung21-generate an m by $n$ complex matrix Q with orthonormal columns,
cung 2 r - cung 2 r - generate an m by n complex matrix Q with orthonormal columns,
cungbr - cungbr - generate one of the complex unitary matrices Q or $\mathrm{P}^{* *} \mathrm{H}$ determined by CGEBRD when reducing a complex matrix A to bidiagonal form
cunghr - cunghr - generate a complex unitary matrix Q which is defined as the product of IHI-ILO elementary reflectors of order N , as returned by CGEHRD
cung12 - cung12 - generate an m-by-n complex matrix Q with orthonormal rows,
cunglq - cunglq - generate an M-by-N complex matrix Q with orthonormal rows,
cungql - cungql - generate an M-by-N complex matrix Q with orthonormal columns,
cungqr - cungqr - generate an M-by-N complex matrix Q with orthonormal columns,
cungr2 - cungr2 - generate an $m$ by $n$ complex matrix $Q$ with orthonormal rows,
cungrq - cungrq - generate an M-by-N complex matrix Q with orthonormal rows,
cungtr - cungtr - generate a complex unitary matrix Q which is defined as the product of $\mathrm{n}-1$ elementary reflectors of order N , as returned by CHETRD
cunmbr - cunmbr - VECT = ' Q ', CUNMBR overwrites the general complex M -by-N matrix C with SIDE = ' L ' SIDE $=$ 'R' TRANS $=$ 'N'

cunml2 - cunml2 - overwrite the general complex m-by-n matrix C with $\mathrm{Q} * \mathrm{C}$ if $\operatorname{SIDE}=$ ' L ' and TRANS = ' N ', or $\mathrm{Q}^{\prime *} \mathrm{C}$ if SIDE $=$ 'L' and TRANS $=$ ' C ', or $\mathrm{C} * \mathrm{Q}$ if SIDE $=$ 'R' and TRANS $=$ ' N ', or $\mathrm{C} * \mathrm{Q}$ ' if SIDE $=$ ' R ' and TRANS = 'C',
cunmlq - cunmlq - overwrite the general complex M -by- N matrix C with $\operatorname{SIDE}=$ 'L' $\operatorname{SIDE}=$ ' R ' TRANS $=$ ' $\mathrm{N}^{\prime}$
cunmql - cunmql - overwrite the general complex M -by- N matrix C with $\operatorname{SIDE}=$ 'L' SIDE = 'R' TRANS = ' N '
$\underline{\text { cunmqr }}$ - cunmqr - overwrite the general complex $\mathrm{M}-$ by- N matrix C with $\operatorname{SIDE}=\mathrm{L}^{\prime} \mathrm{L}$ SIDE $=$ ' $\mathrm{R}^{\prime}$ TRANS $=$ ' $\mathrm{N}^{\prime}$
cunmr2 - cunmr2 - overwrite the general complex m-by-n matrix C with $\mathrm{Q} * \mathrm{C}$ if SIDE $=$ ' L ' and TRANS $=$ ' N ', or $\mathrm{Q}{ }^{*} \mathrm{C}$ if SIDE $=$ 'L' and TRANS $=$ ' C ', or $\mathrm{C} * \mathrm{Q}$ if SIDE $=$ ' R ' and TRANS $=$ ' N ', or $\mathrm{C} * \mathrm{Q}$ ' if SIDE $=$ ' R ' and TRANS = 'C',


cunmtr - cunmtr - overwrite the general complex M-by-N matrix C with $\operatorname{SIDE}={ }^{\prime} \mathrm{L}^{\prime} \operatorname{SIDE}={ }^{\prime} \mathrm{R}$ ' TRANS $=$ ' $\mathrm{N}^{\prime}$
cupgtr - cupgtr - generate a complex unitary matrix Q which is defined as the product of $\mathrm{n}-1$ elementary reflectors $\mathrm{H}(\mathrm{i})$ of order n , as returned by CHPTRD using packed storage
cupmtr - cupmtr - overwrite the general complex M-by-N matrix C with SIDE $=$ 'L' SIDE $=$ ' R ' TRANS $=$ ' $\mathrm{N}^{\prime}$
cvbrmm - cvbrmm - variable block sparse row format matrix-matrix multiply
cvbrsm - cvbrsm - variable block sparse row format triangular solve
cvmul - cvmul - compute the scaled product of complex vectors
dasum - dasum - Return the sum of the absolute values of a vector x .
daxpy - daxpy - compute $\mathrm{y}:=$ alpha $* \mathrm{x}+\mathrm{y}$
daxpyi - daxpyi - Compute $\mathrm{y}:=$ alpha $* \mathrm{x}+\mathrm{y}$
dbcomm - dbcomm - block coordinate matrix-matrix multiply
dbdimm - dbdimm - block diagonal format matrix-matrix multiply
dbdism - dbdism - block diagonal format triangular solve
 matrix B
dbdsqr - dbdsqr - compute the singular value decomposition (SVD) of a real N -by- N (upper or lower) bidiagonal matrix $B$.
dbelmm - dbelmm - block Ellpack format matrix-matrix multiply
dbelsm - dbelsm - block Ellpack format triangular solve
dbscmm - dbscmm - block sparse column matrix-matrix multiply
$\underline{\text { dbscsm - dbscsm - block sparse column format triangular solve }}$
dbsrmm - dbsrmm - block sparse row format matrix-matrix multiply
$\underline{\text { dbsrsm - dbsrsm - block sparse row format triangular solve }}$
denvcor - denveor - compute the convolution or correlation of real vectors
denvcor2 - denvcor2 - compute the convolution or correlation of real matrices
dcoomm - dcoomm - coordinate matrix-matrix multiply
dcopy - dcopy - Copy x to y
dcosqb - dcosqb - synthesize a Fourier sequence from its representation in terms of a cosine series with odd wave numbers. The COSQ operations are unnormalized inverses of themselves, so a call to COSQF followed by a call to COSQB will multiply the input sequence by $4 * \mathrm{~N}$.
dcosqf - dcosqf - compute the Fourier coefficients in a cosine series representation with only odd wave numbers. The COSQ operations are unnormalized inverses of themselves, so a call to COSQF followed by a call to COSQB will multiply the input sequence by $4 * \mathrm{~N}$.
dcosqi - dcosqi - initialize the array WSAVE, which is used in both COSQF and COSQB.
dcost - dcost - compute the discrete Fourier cosine transform of an even sequence. The COST transforms are unnormalized inverses of themselves, so a call of COST followed by another call of COST will multiply the input sequence by $2 *(\mathrm{~N}-1)$.
dcosti - dcosti - initialize the array WSAVE, which is used in COST.
dcscmm - dcscmm - compressed sparse column format matrix-matrix multiply
$\underline{\text { dcscsm - dcscsm - compressed sparse column format triangular solve }}$
dcsrmm - dcsrmm - compressed sparse row format matrix-matrix multiply
$\underline{\text { dcsrsm - dcsrsm - compressed sparse row format triangular solve }}$
ddiamm - ddiamm - diagonal format matrix-matrix multiply
ddiasm - ddiasm - diagonal format triangular solve
ddisna - ddisna - compute the reciprocal condition numbers for the eigenvectors of a real symmetric or complex Hermitian matrix or for the left or right singular vectors of a general m-by-n matrix
ddot - ddot - compute the dot product of two vectors x and y .
ddoti - ddoti - Compute the indexed dot product.
dellmm - dellmm - Ellpack format matrix-matrix multiply
dellsm - dellsm - Ellpack format triangular solve
$\underline{\text { dezftb }}$ - dezftb - computes a periodic sequence from its Fourier coefficients. DEZFTB is a simplified but slower version of DFFTB.
dezftf - dezftf - computes the Fourier coefficients of a periodic sequence. DEZFTF is a simplified but slower version of DFFTF.
dezfti - dezfti - initializes the array WSAVE, which is used in both DEZFTF and DEZFTB.
$\underline{\mathrm{dfft} 2 \mathrm{~b}}$ - dfft2b - compute a periodic sequence from its Fourier coefficients. The DFFT operations are unnormalized, so a call of DFFT2F followed by a call of DFFT2B will multiply the input sequence by $\mathrm{M}^{*} \mathrm{~N}$.
dfft2f - dfft2f - compute the Fourier coefficients of a periodic sequence. The DFFT operations are unnormalized, so a call of DFFT2F followed by a call of DFFT2B will multiply the input sequence by $\mathrm{M}^{*} \mathrm{~N}$.
dfft2i - dfft2i - initialize the array WSAVE, which is used in both the forward and backward transforms.
$\underline{\mathrm{dfft} 3 \mathrm{~b}}$ - dfft3b - compute a periodic sequence from its Fourier coefficients. The DFFT operations are unnormalized, so a call of DFFT3F followed by a call of DFFT3B will multiply the input sequence by $\mathrm{M}^{*} \mathrm{~N}^{*} \mathrm{~K}$.
dfft3f - dfft3f - compute the Fourier coefficients of a real periodic sequence. The DFFT operations are unnormalized, so a call of DFFT3F followed by a call of DFFT3B will multiply the input sequence by $\mathrm{M}^{*} \mathrm{~N}^{*} \mathrm{~K}$.
dfft3i - dfft3i - initialize the array WSAVE, which is used in both DFFT3F and DFFT3B.
dfftb - dfftb - compute a periodic sequence from its Fourier coefficients. The DFFT operations are unnormalized, so a call of DFFTF followed by a call of DFFTB will multiply the input sequence by N .
dfftf - dfftf - compute the Fourier coefficients of a periodic sequence. The FFT operations are unnormalized, so a call of DFFTF followed by a call of DFFTB will multiply the input sequence by N .
dffti - dffti - initialize the array WSAVE, which is used in both DFFTF and DFFTB.
dfftopt - dfftopt - compute the length of the closest fast FFT
dfftz - dfftz - initialize the trigonometric weight and factor tables or compute the forward Fast Fourier Transform of a double precision sequence.
dfftz2 - dfftz2 - initialize the trigonometric weight and factor tables or compute the two-dimensional forward Fast Fourier Transform of a two-dimensional double precision array.
dfftz3 - dfftz3 - initialize the trigonometric weight and factor tables or compute the three-dimensional forward Fast Fourier Transform of a three-dimensional double complex array.
dfftzm - dfftzm - initialize the trigonometric weight and factor tables or compute the one-dimensional forward Fast Fourier Transform of a set of double precision data sequences stored in a two-dimensional array.
dgbbrd - dgbbrd - reduce a real general m-by-n band matrix A to upper bidiagonal form B by an orthogonal transformation
dgbcon - dgbcon - estimate the reciprocal of the condition number of a real general band matrix A , in either the 1 norm or the infinity-norm,
dgbequ - dgbequ - compute row and column scalings intended to equilibrate an M-by-N band matrix A and reduce its condition number
dgbmv - dgbmv - perform one of the matrix-vector operations $y:=a l p h a * A * x+b^{*} \operatorname{beta}^{*} y$ or $y:=\operatorname{alpha}{ }^{*} A^{\prime} * x+$ beta*y
dgbrfs - dgbrfs - improve the computed solution to a system of linear equations when the coefficient matrix is banded, and provides error bounds and backward error estimates for the solution
dgbsv - dgbsv - compute the solution to a real system of linear equations $A * X=B$, where $A$ is a band matrix of order N with KL subdiagonals and KU superdiagonals, and X and B are N -by-NRHS matrices
dgbsvx - dgbsvx - use the LU factorization to compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$, $\mathrm{A}^{*} \mathrm{~T}^{*} \mathrm{X}=\mathrm{B}$, or $\mathrm{A}^{*}{ }^{*} \mathrm{H}^{*} \mathrm{X}=\mathrm{B}$,
dgbtf2 - dgbtf2 - compute an LU factorization of a real m-by-n band matrix A using partial pivoting with row interchanges
dgbtrf - dgbtrf - compute an LU factorization of a real m-by-n band matrix A using partial pivoting with row interchanges
dgbtrs - dgbtrs - solve a system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$ or $\mathrm{A}^{\prime} * \mathrm{X}=\mathrm{B}$ with a general band matrix A using the LU factorization computed by SGBTRF
dgebak - dgebak - form the right or left eigenvectors of a real general matrix by backward transformation on the computed eigenvectors of the balanced matrix output by SGEBAL
dgebal - dgebal - balance a general real matrix A
dgebrd - dgebrd - reduce a general real M-by-N matrix A to upper or lower bidiagonal form B by an orthogonal transformation
dgecon - dgecon - estimate the reciprocal of the condition number of a general real matrix A , in either the 1-norm or the infinity-norm, using the LU factorization computed by SGETRF
dgeequ - dgeequ - compute row and column scalings intended to equilibrate an M -by- N matrix A and reduce its condition number
dgees - dgees - compute for an N -by- N real nonsymmetric matrix A, the eigenvalues, the real Schur form T , and, optionally, the matrix of Schur vectors Z
dgeesx - dgeesx - compute for an N -by-N real nonsymmetric matrix A, the eigenvalues, the real Schur form T, and, optionally, the matrix of Schur vectors $Z$
dgeev - dgeev - compute for an N -by-N real nonsymmetric matrix A, the eigenvalues and, optionally, the left and/ or right eigenvectors
dgeevx - dgeevx - compute for an N-by-N real nonsymmetric matrix A, the eigenvalues and, optionally, the left and/or right eigenvectors
dgegs - dgegs - routine is deprecated and has been replaced by routine SGGES
dgegv - dgegv - routine is deprecated and has been replaced by routine SGGEV
dgehrd - dgehrd - reduce a real general matrix A to upper Hessenberg form H by an orthogonal similarity transformation
dgelqf - dgelqf - compute an LQ factorization of a real M-by-N matrix A
dgels - dgels - solve overdetermined or underdetermined real linear systems involving an M-by-N matrix A, or its transpose, using a QR or LQ factorization of A
dgelsd - dgelsd - compute the minimum-norm solution to a real linear least squares problem
dgelss - dgelss - compute the minimum norm solution to a real linear least squares problem
dgelsx - dgelsx - routine is deprecated and has been replaced by routine SGELSY
dgelsy - dgelsy - compute the minimum-norm solution to a real linear least squares problem
dgemm - dgemm - perform one of the matrix-matrix operations C :=alpha*op(A)*op(B)+beta*C
dgemv - dgemv - perform one of the matrix-vector operations $y:=a l p h a * A * x+$ beta*y or $y:=a l p h a * A^{\prime} * x+$ beta*y
dgeqlf - dgeqlf - compute a QL factorization of a real M-by-N matrix A
dgeqp3 - dgeqp3 - compute a QR factorization with column pivoting of a matrix A
dgeqpf - dgeqpf - routine is deprecated and has been replaced by routine SGEQP3
dgeqrf - dgeqrf - compute a QR factorization of a real M-by-N matrix A
dger - dger - perform the rank 1 operation $\mathrm{A}:=$ alpha* $\mathrm{x}^{*} \mathrm{y}^{\prime}+\mathrm{A}$
dgerfs - dgerfs - improve the computed solution to a system of linear equations and provides error bounds and backward error estimates for the solution
dgerqf - dgerqf - compute an RQ factorization of a real M-by-N matrix A
dgesdd - dgesdd - compute the singular value decomposition (SVD) of a real M-by-N matrix A, optionally computing the left and right singular vectors
dgesv - dgesv - compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
dgesvd - dgesvd - compute the singular value decomposition (SVD) of a real M-by-N matrix A, optionally computing the left and/or right singular vectors
dgesvx - dgesvx - use the LU factorization to compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
dgetf2 - dgetf2 - compute an LU factorization of a general m-by-n matrix A using partial pivoting with row interchanges
dgetrf - dgetrf - compute an LU factorization of a general M-by-N matrix A using partial pivoting with row interchanges
dgetri - dgetri - compute the inverse of a matrix using the LU factorization computed by SGETRF
dgetrs - dgetrs - solve a system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$ or $\mathrm{A}^{\prime} * \mathrm{X}=\mathrm{B}$ with a general N -by-N matrix A using the LU factorization computed by SGETRF
dggbak - dggbak - form the right or left eigenvectors of a real generalized eigenvalue problem $A * x=$ lambda*B*x, by backward transformation on the computed eigenvectors of the balanced pair of matrices output by SGGBAL
dggbal - dggbal - balance a pair of general real matrices (A,B)
dgges - dgges - compute for a pair of N -by- N real nonsymmetric matrices ( $\mathrm{A}, \mathrm{B}$ ),
dggesx - dggesx - compute for a pair of $N-b y-N$ real nonsymmetric matrices $(A, B)$, the generalized eigenvalues, the real Schur form ( $\mathrm{S}, \mathrm{T}$ ), and,
dggev - dggev - compute for a pair of N-by-N real nonsymmetric matrices (A,B)
dggevx - dggevx - compute for a pair of N-by-N real nonsymmetric matrices (A,B)
dggglm - dggglm - solve a general Gauss-Markov linear model (GLM) problem
dgghrd - dgghrd - reduce a pair of real matrices (A,B) to generalized upper Hessenberg form using orthogonal transformations, where A is a general matrix and B is upper triangular
dgglse - dgglse - solve the linear equality-constrained least squares (LSE) problem
dggqrf - dggqrf - compute a generalized QR factorization of an N -by-M matrix A and an N -by-P matrix B .
dggrqf - dggrqf - compute a generalized RQ factorization of an M-by-N matrix A and a P-by-N matrix B
dggsvd - dggsvd - compute the generalized singular value decomposition (GSVD) of an M-by-N real matrix A and P-by-N real matrix B
dggsvp - dggsvp - compute orthogonal matrices $U, V$ and $Q$ such that $N-K-L K L U^{\prime} * A * Q=K(0$ A12 A13 ) if M-K-L >=0
dgssco - dgssco - General sparse solver condition number estimate.
dgssda - dgssda - Deallocate working storage for the general sparse solver.
dgssfa - dgssfa - General sparse solver numeric factorization.
dgssfs - dgssfs - General sparse solver one call interface.
dgssin - dgssin - Initialize the general sparse solver.
dgssor - dgssor - General sparse solver ordering and symbolic factorization.
dgssps - dgssps - Print general sparse solver statics.
dgssrp - dgssrp - Return permutation used by the general sparse solver.
dgsssl - dgsssl - Solve routine for the general sparse solver.
dgssuo - dgssuo - User supplied permutation for ordering used in the general sparse solver.
dgtcon - dgtcon - estimate the reciprocal of the condition number of a real tridiagonal matrix A using the LU factorization as computed by SGTTRF
dgthr - dgthr - Gathers specified elements from y into x .
dgthrz - dgthrz - Gather and zero.
dgtrfs - dgtrfs - improve the computed solution to a system of linear equations when the coefficient matrix is tridiagonal, and provides error bounds and backward error estimates for the solution
dgtsv - dgtsv - solve the equation $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
dgtsvx - dgtsvx - use the LU factorization to compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$ or $\mathrm{A} * * \mathrm{~T} * \mathrm{X}=\mathrm{B}$,
dgttrf - dgttrf - compute an LU factorization of a real tridiagonal matrix A using elimination with partial pivoting and row interchanges
dgttrs - dgttrs - solve one of the systems of equations $\mathrm{A}^{*} \mathrm{X}=\mathrm{B}$ or $\mathrm{A}^{\prime} * \mathrm{X}=\mathrm{B}$,
dhgeqz - dhgeqz - implement a single-/double-shift version of the QZ method for finding the generalized eigenvalues $w(j)=\left(\operatorname{ALPHAR}(\mathrm{j})+\mathrm{i}^{*} \operatorname{ALPHAI}(\mathrm{j})\right) / \operatorname{BETAR}(\mathrm{j})$ of the equation $\operatorname{det}(\mathrm{A}-\mathrm{w}(\mathrm{i}) \mathrm{B})=0 \operatorname{In}$ addition, the pair A,B may be reduced to generalized Schur form
dhsein - dhsein - use inverse iteration to find specified right and/or left eigenvectors of a real upper Hessenberg matrix H
dhseqr - dhseqr - compute the eigenvalues of a real upper Hessenberg matrix H and, optionally, the matrices T and Z from the Schur decomposition $\mathrm{H}=\mathrm{Z} \mathrm{T} \mathrm{Z**T}$,where T is an upper quasi-triangular matrix (the Schur form), and Z is the orthogonal matrix of Schur vectors
djadmm - djadmm - Jagged diagonal matrix-matrix multiply (modified Ellpack)
djadrp - djadrp - right permutation of a jagged diagonal matrix
djadsm - djadsm - Jagged-diagonal format triangular solve
dlagtf - dlagtf - factorize the matrix (T-lambda*I), where T is an n by n tridiagonal matrix and lambda is a scalar, as T-lambda*I = PLU
dlamrg - dlamrg - will create a permutation list which will merge the elements of A (which is composed of two independently sorted sets) into a single set which is sorted in ascending order

dlarzb - dlarzb - applies a real block reflector H or its transpose $\mathrm{H}^{*}$ T to a real distributed M -by- N C from the left or the right
dlarzt - dlarzt - form the triangular factor T of a real block reflector H of order $>\mathrm{n}$, which is defined as a product of k elementary reflectors
dlasrt - dlasrt - the numbers in D in increasing order (if ID = 'I') or in decreasing order (if ID = 'D' )
dlatzm - dlatzm - routine is deprecated and has been replaced by routine SORMRZ
dnrm2 - dnrm2 - Return the Euclidian norm of a vector.
dopgtr - dopgtr - generate a real orthogonal matrix Q which is defined as the product of $\mathrm{n}-1$ elementary reflectors $\mathrm{H}(\mathrm{i})$ of order n , as returned by SSPTRD using packed storage
$\underline{\text { dopmtr }}$ - dopmtr - overwrite the general real M-by-N matrix C with $\operatorname{SIDE}=$ 'L' SIDE $=$ 'R' TRANS $=$ ' N '
dorg21-dorg21-generate an m by n real matrix Q with orthonormal columns,
$\underline{\text { dorg } 2 r}$ - dorg $2 r$ - generate an $m$ by $n$ real matrix $Q$ with orthonormal columns,
dorgbr - dorgbr - generate one of the real orthogonal matrices Q or $\mathrm{P}^{* *}$ T determined by SGEBRD when reducing a real matrix A to bidiagonal form
dorghr - dorghr - generate a real orthogonal matrix Q which is defined as the product of IHI-ILO elementary reflectors of order N , as returned by SGEHRD
dorg12 - dorg12 - generate an $m$ by $n$ real matrix Q with orthonormal rows,
dorglq - dorglq - generate an M-by-N real matrix Q with orthonormal rows,
dorgql - dorgql - generate an M-by-N real matrix Q with orthonormal columns,
dorgqr - dorgqr - generate an M-by-N real matrix Q with orthonormal columns,
dorgr2 - dorgr2 - generate an $m$ by $n$ real matrix Q with orthonormal rows,
dorgrq - dorgrq - generate an M-by-N real matrix Q with orthonormal rows,
dorgtr - dorgtr - generate a real orthogonal matrix Q which is defined as the product of $\mathrm{n}-1$ elementary reflectors of order N , as returned by SSYTRD
dormbr - dormbr - VECT = 'Q', SORMBR overwrites the general real M-by-N matrix C with SIDE = 'L' SIDE = 'R' TRANS = 'N'
$\underline{\text { dormhr }}$ - dormhr - overwrite the general real M-by-N matrix C with $\operatorname{SIDE}=$ 'L' SIDE $=$ 'R' TRANS $=$ ' $\mathrm{N}^{\prime}$
 dormql - dormql - overwrite the general real M-by-N matrix C with $\mathrm{SIDE}=$ 'L' SIDE $=$ 'R' TRANS $=$ ' N '
 dormrq - dormrq - overwrite the general real M-by-N matrix C with $\operatorname{SIDE}=$ 'L' SIDE $=$ 'R'TRANS $=$ ' N '
 $\underline{\text { dormtr }}$ - dormtr - overwrite the general real M-by-N matrix C with SIDE = 'L' SIDE = 'R' TRANS $=$ ' N '
dpbcon - dpbcon - estimate the reciprocal of the condition number (in the 1-norm) of a real symmetric positive definite band matrix using the Cholesky factorization $A=U * * T * U$ or $A=L^{*} L^{* *} T$ computed by SPBTRF
dpbequ - dpbequ - compute row and column scalings intended to equilibrate a symmetric positive definite band matrix A and reduce its condition number (with respect to the two-norm)
dpbrfs - dpbrfs - improve the computed solution to a system of linear equations when the coefficient matrix is symmetric positive definite and banded, and provides error bounds and backward error estimates for the solution
dpbstf - dpbstf - compute a split Cholesky factorization of a real symmetric positive definite band matrix A
dpbsv - dpbsv - compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
dpbsvx - dpbsvx - use the Cholesky factorization $\mathrm{A}=\mathrm{U}^{*} * \mathrm{~T}^{*} \mathrm{U}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ to compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
dpbtf2 - dpbtf2 - compute the Cholesky factorization of a real symmetric positive definite band matrix A dpbtrf - dpbtrf - compute the Cholesky factorization of a real symmetric positive definite band matrix A
dpbtrs - dpbtrs - solve a system of linear equations $\mathrm{A}^{*} \mathrm{X}=\mathrm{B}$ with a symmetric positive definite band matrix A using the Cholesky factorization $\mathrm{A}=\mathrm{U}^{*} * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ computed by SPBTRF
dpocon - dpocon - estimate the reciprocal of the condition number (in the 1-norm) of a real symmetric positive definite matrix using the Cholesky factorization $A=U^{*} * T^{*} U$ or $A=L^{*} L^{* *} T$ computed by SPOTRF
dpoequ - dpoequ - compute row and column scalings intended to equilibrate a symmetric positive definite matrix A and reduce its condition number (with respect to the two-norm)
dporfs - dporfs - improve the computed solution to a system of linear equations when the coefficient matrix is symmetric positive definite,
dposv - dposv - compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
dposvx - dposvx - use the Cholesky factorization $\mathrm{A}=\mathrm{U}^{*} * \mathrm{~T}^{*} \mathrm{U}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ to compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
dpotf2 - dpotf2 - compute the Cholesky factorization of a real symmetric positive definite matrix A
dpotrf - dpotrf - compute the Cholesky factorization of a real symmetric positive definite matrix A
dpotri - dpotri - compute the inverse of a real symmetric positive definite matrix A using the Cholesky factorization $\mathrm{A}=\mathrm{U}^{*} * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ computed by SPOTRF
dpotrs - dpotrs - solve a system of linear equations $A * X=B$ with a symmetric positive definite matrix $A$ using the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ computed by SPOTRF
dppcon - dppcon - estimate the reciprocal of the condition number (in the 1-norm) of a real symmetric positive definite packed matrix using the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ computed by SPPTRF
dppequ - dppequ - compute row and column scalings intended to equilibrate a symmetric positive definite matrix A in packed storage and reduce its condition number (with respect to the two-norm)
dpprfs - dpprfs - improve the computed solution to a system of linear equations when the coefficient matrix is symmetric positive definite and packed, and provides error bounds and backward error estimates for the solution
dppsv - dppsv - compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
dppsvx - dppsvx - use the Cholesky factorization $A=U^{* *} T^{*} U$ or $A=L^{*} L^{* *} T$ to compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
dpptrf - dpptrf - compute the Cholesky factorization of a real symmetric positive definite matrix A stored in packed format
dpptri - dpptri - compute the inverse of a real symmetric positive definite matrix A using the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ computed by SPPTRF
dpptrs - dpptrs - solve a system of linear equations $A * X=B$ with a symmetric positive definite matrix $A$ in packed storage using the Cholesky factorization $\mathrm{A}=\mathrm{U}^{* *} \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=\mathrm{L} * \mathrm{~L}^{*} * \mathrm{~T}$ computed by SPPTRF
dptcon - dptcon - compute the reciprocal of the condition number (in the 1-norm) of a real symmetric positive definite tridiagonal matrix using the factorization $A=L * D * L * * T$ or $A=U * * T * D * U$ computed by SPTTRF
dpteqr - dpteqr - compute all eigenvalues and, optionally, eigenvectors of a symmetric positive definite tridiagonal matrix by first factoring the matrix using SPTTRF, and then calling SBDSQR to compute the singular values of the bidiagonal factor
dptrfs - dptrfs - improve the computed solution to a system of linear equations when the coefficient matrix is symmetric positive definite and tridiagonal, and provides error bounds and backward error estimates for the solution
dptsv - dptsv - compute the solution to a real system of linear equations $A * X=B$, where $A$ is an $N-b y-N$ symmetric positive definite tridiagonal matrix, and X and B are N -by-NRHS matrices.
dptsvx - dptsvx - use the factorization $A=L^{*} D^{*} L^{* *} T$ to compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$, where A is an N -by- N symmetric positive definite tridiagonal matrix and X and B are N -by-NRHS matrices
dpttrf - dpttrf - compute the $L^{*} D^{*} L^{\prime}$ factorization of a real symmetric positive definite tridiagonal matrix A
dpttrs - dpttrs - solve a tridiagonal system of the form $A * X=B$ using the $L * D^{*} L^{\prime}$ factorization of A computed by SPTTRF
dptts2 - dptts2 - solve a tridiagonal system of the form $A * X=B$ using the $L^{*} D^{*} L^{\prime}$ factorization of A computed by SPTTRF
dqdota - dqdota - compute a double precision constant plus an extended precision constant plus the extended precision dot product of two double precision vectors x and y .
dqdoti - dqdoti - compute a constant plus the extended precision dot product of two double precision vectors x and $y$.
drot - drot - Apply a Given's rotation constructed by SROTG.
drotg - drotg - Construct a Given's plane rotation
droti - droti - Apply an indexed Givens rotation.
drotm - drotm - Apply a Gentleman's modified Given's rotation constructed by SROTMG.
drotmg - drotmg - Construct a Gentleman's modified Given's plane rotation
$\underline{\text { dsbev - dsbev - compute all the eigenvalues and, optionally, eigenvectors of a real symmetric band matrix A }}$
dsbevd - dsbevd - compute all the eigenvalues and, optionally, eigenvectors of a real symmetric band matrix A
dsbevx - dsbevx - compute selected eigenvalues and, optionally, eigenvectors of a real symmetric band matrix A
dsbgst - dsbgst - reduce a real symmetric-definite banded generalized eigenproblem $\mathrm{A} * \mathrm{x}=\operatorname{lambda} * \mathrm{~B} * \mathrm{x}$ to standard form $\mathrm{C} * \mathrm{y}=$ lambda*y,
dsbgv - dsbgv - compute all the eigenvalues, and optionally, the eigenvectors of a real generalized symmetricdefinite banded eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\operatorname{lambda}) * \mathrm{~B} * \mathrm{x}$
dsbgvd - dsbgvd - compute all the eigenvalues, and optionally, the eigenvectors of a real generalized symmetricdefinite banded eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\operatorname{lambda}) * \mathrm{~B} * \mathrm{x}$
$\underline{\text { dsbgvx - dsbgvx - compute selected eigenvalues, and optionally, eigenvectors of a real generalized symmetric- }}$ definite banded eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\operatorname{lambda}) * \mathrm{~B} * \mathrm{x}$

dsbtrd - dsbtrd - reduce a real symmetric band matrix A to symmetric tridiagonal form T by an orthogonal similarity transformation
dscal - dscal - Compute y $:=$ alpha * y
$\underline{\text { dsctr }}$ - dsctr - Scatters elements from x into y .
dsdot - dsdot - compute the double precision dot product of two single precision vectors x and y .
dsecnd - dsecnd - return the user time for a process in seconds
dsinqb - dsinqb - synthesize a Fourier sequence from its representation in terms of a sine series with odd wave numbers. The SINQ operations are unnormalized inverses of themselves, so a call to SINQF followed by a call to SINQB will multiply the input sequence by $4 * \mathrm{~N}$.
dsinqf - dsinqf - compute the Fourier coefficients in a sine series representation with only odd wave numbers. The SINQ operations are unnormalized inverses of themselves, so a call to SINQF followed by a call to SINQB will multiply the input sequence by $4 * \mathrm{~N}$.
dsinqi - dsinqi - initialize the array xWSAVE, which is used in both SINQF and SINQB.
dsint - dsint - compute the discrete Fourier sine transform of an odd sequence. The SINT transforms are unnormalized inverses of themselves, so a call of SINT followed by another call of SINT will multiply the input
sequence by $2 *(\mathrm{~N}+1)$.
dsinti - dsinti - initialize the array WSAVE, which is used in subroutine SINT.
dskymm - dskymm - Skyline format matrix-matrix multiply
dskysm - dskysm - Skyline format triangular solve
dspcon - dspcon - estimate the reciprocal of the condition number (in the 1-norm) of a real symmetric packed matrix A using the factorization $\mathrm{A}=\mathrm{U}^{*} \mathrm{D}^{*} \mathrm{U}^{* *} \mathrm{~T}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{D}^{*} \mathrm{~L}^{* *} \mathrm{~T}$ computed by SSPTRF
dspev - dspev - compute all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix A in packed storage
dspevd - dspevd - compute all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix A in packed storage
dspevx - dspevx - compute selected eigenvalues and, optionally, eigenvectors of a real symmetric matrix A in packed storage
dspgst - dspgst - reduce a real symmetric-definite generalized eigenproblem to standard form, using packed storage
dspgv - dspgv - compute all the eigenvalues and, optionally, the eigenvectors of a real generalized symmetricdefinite eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{~B}{ }^{*} \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda})^{*} \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{x}$
dspgvd - dspgvd - compute all the eigenvalues, and optionally, the eigenvectors of a real generalized symmetricdefinite eigenproblem, of the form $\mathrm{A}^{*} \mathrm{x}=(\operatorname{lambda}) * \mathrm{~B} * \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda}) * \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lambda}) * \mathrm{x}$
dspgvx - dspgvx - compute selected eigenvalues, and optionally, eigenvectors of a real generalized symmetricdefinite eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\mathrm{lambda}) * \mathrm{~B} * \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\mathrm{lambda}) * \mathrm{x}$, or $\mathrm{B}^{*} \mathrm{~A}^{*} \mathrm{x}=(\operatorname{lambda}) * \mathrm{x}$

dspr - dspr - perform the symmetric rank 1 operation A := alpha*x*x' +A
$\underline{\text { dspr2 }}$ - dspr2 - perform the symmetric rank 2 operation A := alpha*x*y' + alpha* $y^{*} x^{\prime}+$ A
dsprfs - dsprfs - improve the computed solution to a system of linear equations when the coefficient matrix is symmetric indefinite and packed, and provides error bounds and backward error estimates for the solution

dspsvx - dspsvx - use the diagonal pivoting factorization $A=U * D * U * * T$ or $A=L * D * L * * T$ to compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$, where A is an $\mathrm{N}-$ by-N symmetric matrix stored in packed format and X and B are N-by-NRHS matrices
dsptrd - dsptrd - reduce a real symmetric matrix A stored in packed form to symmetric tridiagonal form T by an orthogonal similarity transformation
dsptrf - dsptrf - compute the factorization of a real symmetric matrix A stored in packed format using the BunchKaufman diagonal pivoting method
dsptri - dsptri - compute the inverse of a real symmetric indefinite matrix A in packed storage using the factorization $\mathrm{A}=\mathrm{U}^{*} \mathrm{D}^{*} \mathrm{U}^{* *} \mathrm{~T}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{D}^{*} \mathrm{~L}^{* *} \mathrm{~T}$ computed by SSPTRF
dsptrs - dsptrs - solve a system of linear equations A*X $=\mathrm{B}$ with a real symmetric matrix A stored in packed format using the factorization $\mathrm{A}=\mathrm{U}^{*} \mathrm{D}^{*} \mathrm{U}^{*} * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D}^{*} \mathrm{~L}^{* *} \mathrm{~T}$ computed by SSPTRF
dstebz - dstebz - compute the eigenvalues of a symmetric tridiagonal matrix T
dstedc - dstedc - compute all eigenvalues and, optionally, eigenvectors of a symmetric tridiagonal matrix using the divide and conquer method
dstegr - dstegr - (a) Compute T-sigma_i = L_i $\mathrm{D}_{-} \mathrm{i} \mathrm{L}_{-} \mathrm{i}^{\wedge} \mathrm{T}$, such that $\mathrm{L}_{-} \mathrm{i} \mathrm{D}_{-} \mathrm{i} \mathrm{L}_{-} \mathrm{i}^{\wedge} \mathrm{T}$ is a relatively robust representation
dstein - dstein - compute the eigenvectors of a real symmetric tridiagonal matrix T corresponding to specified eigenvalues, using inverse iteration
dsteqr - dsteqr - compute all eigenvalues and, optionally, eigenvectors of a symmetric tridiagonal matrix using the implicit QL or QR method
dsterf - dsterf - compute all eigenvalues of a symmetric tridiagonal matrix using the Pal-Walker-Kahan variant of the QL or QR algorithm
dstev - dstev - compute all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix A
dstevd - dstevd - compute all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix
dstevr - dstevr - compute selected eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix T
dstevx - dstevx - compute selected eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix A
$\underline{\text { dstsv }}$ - dstsv - compute the solution to a system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$ where A is a symmetric tridiagonal matrix
dsttrf - dsttrf - compute the factorization of a symmetric tridiagonal matrix A
dsttrs - dsttrs - computes the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$
dswap - dswap - Exchange vectors $x$ and $y$.
dsycon - dsycon - estimate the reciprocal of the condition number (in the 1-norm) of a real symmetric matrix A using the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D}^{*} \mathrm{U}^{*} * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D}^{*} \mathrm{~L}^{* *} \mathrm{~T}$ computed by SSYTRF
dsyev - dsyev - compute all eigenvalues and, optionally, eigenvectors of a real symmetric matrix A
dsyevd - dsyevd - compute all eigenvalues and, optionally, eigenvectors of a real symmetric matrix A
$\underline{\text { dsyevr - dsyevr - compute selected eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal }}$ matrix T
$\underline{\text { dsyevx - dsyevx - compute selected eigenvalues and, optionally, eigenvectors of a real symmetric matrix A }}$
dsygs2 - dsygs2 - reduce a real symmetric-definite generalized eigenproblem to standard form
dsygst - dsygst - reduce a real symmetric-definite generalized eigenproblem to standard form
dsygv - dsygv - compute all the eigenvalues, and optionally, the eigenvectors of a real generalized symmetricdefinite eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{~B}^{*} \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda})^{*} \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{x}$
dsygvd - dsygvd - compute all the eigenvalues, and optionally, the eigenvectors of a real generalized symmetricdefinite eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\operatorname{lambda}) * \mathrm{~B} * \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda}) * \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lambda}) * \mathrm{x}$
dsygvx - dsygvx - compute selected eigenvalues, and optionally, eigenvectors of a real generalized symmetricdefinite eigenproblem, of the form $\mathrm{A}^{*} \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{~B}^{*} \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda})^{*} \mathrm{x}$, or $\mathrm{B}^{*} \mathrm{~A}^{*} \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{x}$
$\underline{\text { dsymm }}-$ dsymm - perform one of the matrix-matrix operations $\mathrm{C}:=$ alpha*A*B + beta* C or $\mathrm{C}:=$ alpha*B*A + beta*C

dsyr - dsyr - perform the symmetric rank 1 operation A:= alpha*x*x' + A

dsyr2k - dsyr 2 k - perform one of the symmetric rank 2 k operations $\mathrm{C}:=$ alpha* $\mathrm{A}^{*} \mathrm{~B}^{\prime}+$ alpha* $\mathrm{B}^{*} \mathrm{~A}^{\prime}+$ beta* $\mathrm{C}^{\text {or }} \mathrm{C}$ : $=$ alpha* ${ }^{\prime} *$ B + alpha* ${ }^{\prime} * \mathrm{~A}+$ beta $^{*} \mathrm{C}$
dsyrfs - dsyrfs - improve the computed solution to a system of linear equations when the coefficient matrix is symmetric indefinite, and provides error bounds and backward error estimates for the solution
dsyrk - dsyrk - perform one of the symmetric rank k operations $\mathrm{C}:=\operatorname{alpha} \mathrm{a}^{*} \mathrm{~A}^{*} \mathrm{~A}^{\prime}+\operatorname{beta}^{*} \mathrm{C}$ or $\mathrm{C}:=\operatorname{alpha} \mathrm{A}^{\prime} * \mathrm{~A}+$ beta*C

dsysvx - dsysvx - use the diagonal pivoting factorization to compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
dsytd2 - dsytd2 - reduce a real symmetric matrix A to symmetric tridiagonal form T by an orthogonal similarity transformation
dsytf2 - dsytf2 - compute the factorization of a real symmetric matrix A using the Bunch-Kaufman diagonal pivoting method
dsytrd - dsytrd - reduce a real symmetric matrix A to real symmetric tridiagonal form T by an orthogonal similarity transformation
dsytrf - dsytrf - compute the factorization of a real symmetric matrix A using the Bunch-Kaufman diagonal pivoting method
dsytri - dsytri - compute the inverse of a real symmetric indefinite matrix A using the factorization $\mathrm{A}=$ $\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ computed by SSYTRF
dsytrs - dsytrs - solve a system of linear equations $\mathrm{A}^{*} \mathrm{X}=\mathrm{B}$ with a real symmetric matrix A using the factorization $\mathrm{A}=\mathrm{U}^{*} \mathrm{D}^{*} \mathrm{U}^{*} * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ computed by SSYTRF
dtbcon - dtbcon - estimate the reciprocal of the condition number of a triangular band matrix A , in either the 1norm or the infinity-norm

dtbrfs - dtbrfs - provide error bounds and backward error estimates for the solution to a system of linear equations with a triangular band coefficient matrix
$\underline{\text { dtbsv }}$ - dtbsv - solve one of the systems of equations $A^{*} \mathrm{x}=\mathrm{b}$, or $\mathrm{A}^{\prime}{ }^{*} \mathrm{X}=\mathrm{b}$
$\underline{\text { dtbtrs - dtbtrs - solve a triangular system of the form } \mathrm{A} * \mathrm{X}=\mathrm{B} \text { or } \mathrm{A} * * \mathrm{~T} * \mathrm{X}=\mathrm{B}, ~}$
$\underline{\text { dtgevc - dtgevc - compute some or all of the right and/or left generalized eigenvectors of a pair of real upper }}$ triangular matrices $(A, B)$
dtgexc - dtgexc - reorder the generalized real Schur decomposition of a real matrix pair (A,B) using an orthogonal equivalence transformation $(A, B)=Q^{*}(A, B) * Z^{\prime}$,
dtgsen - dtgsen - reorder the generalized real Schur decomposition of a real matrix pair (A, B) (in terms of an orthonormal equivalence trans- formation $\mathrm{Q}^{\prime *}(\mathrm{~A}, \mathrm{~B}) * \mathrm{Z}$ ), so that a selected cluster of eigenvalues appears in the leading diagonal blocks of the upper quasi-triangular matrix A and the upper triangular B
dtgsja - dtgsja - compute the generalized singular value decomposition (GSVD) of two real upper triangular (or trapezoidal) matrices A and B
dtgsna - dtgsna - estimate reciprocal condition numbers for specified eigenvalues and/or eigenvectors of a matrix pair (A, B) in generalized real Schur canonical form (or of any matrix pair ( $Q^{*} A^{*} Z^{\prime}, Q^{*} B^{*} Z^{\prime}$ ) with orthogonal matrices Q and Z , where $\mathrm{Z}^{\prime}$ denotes the transpose of Z
dtgsyl - dtgsyl - solve the generalized Sylvester equation
dtpcon - dtpcon - estimate the reciprocal of the condition number of a packed triangular matrix A , in either the 1norm or the infinity-norm
$\underline{\text { dtpmv }}$ - dtpmv - perform one of the matrix-vector operations $\mathrm{x}:=\mathrm{A}^{*} \mathrm{x}$, or $\mathrm{x}:=\mathrm{A}^{\prime *} \mathrm{x}$
dtprfs - dtprfs - provide error bounds and backward error estimates for the solution to a system of linear equations with a triangular packed coefficient matrix
$\underline{d t p s v}-\mathrm{dtpsv}$ - solve one of the systems of equations $\mathrm{A}^{*} \mathrm{x}=\mathrm{b}$, or $\mathrm{A}^{\prime}{ }^{*} \mathrm{x}=\mathrm{b}$
dtptri - dtptri - compute the inverse of a real upper or lower triangular matrix A stored in packed format
$\underline{\text { dtptrs }-d t p t r s ~-~ s o l v e ~ a ~ t r i a n g u l a r ~ s y s t e m ~ o f ~ t h e ~ f o r m ~} \mathrm{~A} * \mathrm{X}=\mathrm{B}$ or $\mathrm{A} * * \mathrm{~T} * \mathrm{X}=\mathrm{B}$,
dtrans - dtrans - transpose and scale source matrix
dtrcon - dtrcon - estimate the reciprocal of the condition number of a triangular matrix A , in either the 1-norm or the infinity-norm
dtrevc - dtrevc - compute some or all of the right and/or left eigenvectors of a real upper quasi-triangular matrix T
dtrexc - dtrexc - reorder the real Schur factorization of a real matrix $\mathrm{A}=\mathrm{Q}^{*} \mathrm{~T}^{*} \mathrm{Q} * * \mathrm{~T}$, so that the diagonal block of T with row index IFST is moved to row ILST
$\underline{\text { dtrmm }}$ - dtrmm - perform one of the matrix-matrix operations B $:=\operatorname{alpha*op(A)*B,~or~B:=alpha*B*op(A)~}$ $\underline{\text { dtrmv }}$ - dtrmv - perform one of the matrix-vector operations $\mathrm{x}:=\mathrm{A}^{*} \mathrm{x}$, or $\mathrm{x}:=\mathrm{A}^{\prime *} \mathrm{x}$
dtrrfs - dtrrfs - provide error bounds and backward error estimates for the solution to a system of linear equations with a triangular coefficient matrix
dtrsen - dtrsen - reorder the real Schur factorization of a real matrix $A=Q^{*} T^{*} Q^{* *} T$, so that a selected cluster of eigenvalues appears in the leading diagonal blocks of the upper quasi-triangular matrix T ,
$\underline{\text { dtrsm }}$ - dtrsm - solve one of the matrix equations $o p(A) * X=$ alpha*B, or $X * o p(A)=$ alpha*B
dtrsna - dtrsna - estimate reciprocal condition numbers for specified eigenvalues and/or right eigenvectors of a real upper quasi-triangular matrix T (or of any matrix $\mathrm{Q} * \mathrm{~T} * \mathrm{Q} * * \mathrm{~T}$ with Q orthogonal)
$\underline{\text { dtrsv }}$ - dtrsv - solve one of the systems of equations $A^{*} x=b$, or $A^{*} x=b$
dtrsyl - dtrsyl - solve the real Sylvester matrix equation
$\underline{\text { dtrti2 }}$ - dtrti2 - compute the inverse of a real upper or lower triangular matrix
dtrtri - dtrtri - compute the inverse of a real upper or lower triangular matrix A
$\underline{\text { dtrtrs }}$ - dtrtrs - solve a triangular system of the form $\mathrm{A} * \mathrm{X}=\mathrm{B}$ or $\mathrm{A} * * \mathrm{~T} * \mathrm{X}=\mathrm{B}$,
dtzrqf - dtzrqf - routine is deprecated and has been replaced by routine STZRZF
dtzrzf - dtzrzf - reduce the $\mathrm{M}-\mathrm{by}-\mathrm{N}(\mathrm{M}<=\mathrm{N})$ real upper trapezoidal matrix A to upper triangular form by means of orthogonal transformations
dvbrmm - dvbrmm - variable block sparse row format matrix-matrix multiply
dvbrsm - dvbrsm - variable block sparse row format triangular solve
dwiener - dwiener - perform Wiener deconvolution of two signals
dzasum - dzasum - Return the sum of the absolute values of a vector x .
dznrm2 - dznrm2-Return the Euclidian norm of a vector.
$\underline{\text { ezfftb - ezfftb - computes a periodic sequence from its Fourier coefficients. EZFFTB is a simplified but slower }}$
version of RFFTB.
ezfftf - ezfftf - computes the Fourier coefficients of a periodic sequence. EZFFTF is a simplified but slower version of RFFTF.
ezffti - ezffti - initializes the array WSAVE, which is used in both EZFFTF and EZFFTB.
fft - Overview of Fast Fourier Transform subroutines
icamax - icamax - return the index of the element with largest absolute value.
idamax - idamax - return the index of the element with largest absolute value.
ilaenv - The name of the calling subroutine, in either upper case or lower case.
isamax - isamax - return the index of the element with largest absolute value.
izamax - izamax - return the index of the element with largest absolute value.

1same - lsame - returns .TRUE. if CA is the same letter as CB regardless of case
$\underline{\mathrm{rfft} 2 \mathrm{~b}}-\mathrm{rfft} 2 \mathrm{~b}$ - compute a periodic sequence from its Fourier coefficients. The RFFT operations are unnormalized, so a call of RFFT2F followed by a call of RFFT2B will multiply the input sequence by $\mathrm{M} * \mathrm{~N}$.
$\underline{\mathrm{rfft} 2 \mathrm{f}}$ - rfft 2 f - compute the Fourier coefficients of a periodic sequence. The RFFT operations are unnormalized, so a call of RFFT2F followed by a call of RFFT2B will multiply the input sequence by $M^{*} N$.
$\underline{\mathrm{rfft} 2 \mathrm{i}}-\mathrm{rfft} 2 \mathrm{i}$ - initialize the array WSAVE, which is used in both the forward and backward transforms.
$\underline{\mathrm{rfft} 3 \mathrm{~b}}$ - rfft3b - compute a periodic sequence from its Fourier coefficients. The RFFT operations are unnormalized, so a call of RFFT3F followed by a call of RFFT3B will multiply the input sequence by $\mathrm{M}^{*} \mathrm{~N}^{*} \mathrm{~K}$.
$\underline{\mathrm{rfft} 3 \mathrm{f}}-\mathrm{rfft} 3 \mathrm{f}$ - compute the Fourier coefficients of a real periodic sequence. The RFFT operations are unnormalized, so a call of RFFT3F followed by a call of RFFT3B will multiply the input sequence by $\mathrm{M}^{*} \mathrm{~N}^{*} \mathrm{~K}$.
$\underline{\mathrm{rfft} 3 \mathrm{i}}-\mathrm{rfft} 3 \mathrm{i}$ - initialize the array WSAVE, which is used in both RFFT3F and RFFT3B.
$\underline{\text { rfftb }}$ - rfftb - compute a periodic sequence from its Fourier coefficients. The RFFT operations are unnormalized, so a call of RFFTF followed by a call of RFFTB will multiply the input sequence by N .
$\underline{\mathrm{rfftf}}-\mathrm{rfftf}$ - compute the Fourier coefficients of a periodic sequence. The FFT operations are unnormalized, so a call of RFFTF followed by a call of RFFTB will multiply the input sequence by N .
rffti - rffti - initialize the array WSAVE, which is used in both RFFTF and RFFTB.
rfftopt - rfftopt - compute the length of the closest fast FFT
sasum - sasum - Return the sum of the absolute values of a vector x .
saxpy - saxpy - compute $\mathrm{y}:=$ alpha $* \mathrm{x}+\mathrm{y}$
saxpyi - saxpyi - Compute y := alpha * $\mathrm{x}+\mathrm{y}$
sbcomm - sbcomm - block coordinate matrix-matrix multiply
sbdimm - sbdimm - block diagonal format matrix-matrix multiply
sbdism - sbdism - block diagonal format triangular solve
 matrix B
sbdsqr - sbdsqr - compute the singular value decomposition (SVD) of a real N-by-N (upper or lower) bidiagonal matrix $B$.
sbelmm - sbelmm - block Ellpack format matrix-matrix multiply
sbelsm - sbelsm - block Ellpack format triangular solve
sbscmm - sbscmm - block sparse column matrix-matrix multiply
$\underline{\text { sbscsm - sbscsm - block sparse column format triangular solve }}$
sbsrmm - sbsrmm - block sparse row format matrix-matrix multiply
sbsrsm - sbsrsm - block sparse row format triangular solve
scasum - scasum - Return the sum of the absolute values of a vector x .
scnrm2 - scnrm2-Return the Euclidian norm of a vector.
scnvcor - scnvcor - compute the convolution or correlation of real vectors
scnvcor2 - scnvcor2 - compute the convolution or correlation of real matrices
scoomm - scoomm - coordinate matrix-matrix multiply
scopy - scopy - Copy x to y
scscmm - scscmm - compressed sparse column format matrix-matrix multiply
scscsm - scscsm - compressed sparse column format triangular solve
scsrmm - scsrmm - compressed sparse row format matrix-matrix multiply
scsrsm - scsrsm - compressed sparse row format triangular solve
sdiamm - sdiamm - diagonal format matrix-matrix multiply
sdiasm - sdiasm - diagonal format triangular solve
sdisna - sdisna - compute the reciprocal condition numbers for the eigenvectors of a real symmetric or complex Hermitian matrix or for the left or right singular vectors of a general m-by-n matrix
sdot - sdot - compute the dot product of two vectors x and y .
sdoti - sdoti - Compute the indexed dot product.
sdsdot - sdsdot - compute a constant plus the double precision dot product of two single precision vectors x and y
second - second - return the user time for a process in seconds
sellmm - sellmm - Ellpack format matrix-matrix multiply
sellsm - sellsm - Ellpack format triangular solve
$\underline{\text { sfftc }}$ - sfftc - initialize the trigonometric weight and factor tables or compute the forward Fast Fourier Transform of a real sequence.
sfftc2 - sfftc2 - initialize the trigonometric weight and factor tables or compute the two-dimensional forward Fast Fourier Transform of a two-dimensional real array.
sfftc3 - sfftc3 - initialize the trigonometric weight and factor tables or compute the three-dimensional forward Fast Fourier Transform of a three-dimensional complex array.
sfftcm - sfftcm - initialize the trigonometric weight and factor tables or compute the one-dimensional forward

Fast Fourier Transform of a set of real data sequences stored in a two-dimensional array.
sgbbrd - sgbbrd - reduce a real general m-by-n band matrix A to upper bidiagonal form B by an orthogonal transformation
sgbcon - sgbcon - estimate the reciprocal of the condition number of a real general band matrix A, in either the 1norm or the infinity-norm,
sgbequ - sgbequ - compute row and column scalings intended to equilibrate an M-by-N band matrix A and reduce its condition number
sgbmv - sgbmv - perform one of the matrix-vector operations y $:=$ alpha*A*x + beta* $^{2} y$ or $y:=$ alpha* $^{\prime} A^{\prime} x^{x}+$ beta*y
sgbrfs - sgbrfs - improve the computed solution to a system of linear equations when the coefficient matrix is banded, and provides error bounds and backward error estimates for the solution
sgbsv - sgbsv - compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$, where A is a band matrix of order N with KL subdiagonals and KU superdiagonals, and X and B are N -by-NRHS matrices
sgbsvx - sgbsvx - use the LU factorization to compute the solution to a real system of linear equations $A * X=B$, $\mathrm{A}^{* *} \mathrm{~T} * \mathrm{X}=\mathrm{B}$, or $\mathrm{A}^{* *} \mathrm{H}^{*} \mathrm{X}=\mathrm{B}$,
sgbtf2 - sgbtf2 - compute an LU factorization of a real m-by-n band matrix A using partial pivoting with row interchanges
sgbtrf - sgbtrf - compute an LU factorization of a real m-by-n band matrix A using partial pivoting with row interchanges
sgbtrs - sgbtrs - solve a system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$ or $\mathrm{A}^{\prime} * \mathrm{X}=\mathrm{B}$ with a general band matrix A using the LU factorization computed by SGBTRF
sgebak - sgebak - form the right or left eigenvectors of a real general matrix by backward transformation on the computed eigenvectors of the balanced matrix output by SGEBAL
sgebal - sgebal - balance a general real matrix A
sgebrd - sgebrd - reduce a general real M-by-N matrix A to upper or lower bidiagonal form B by an orthogonal transformation
sgecon - sgecon - estimate the reciprocal of the condition number of a general real matrix A , in either the 1-norm or the infinity-norm, using the LU factorization computed by SGETRF
sgeequ - sgeequ - compute row and column scalings intended to equilibrate an M -by- N matrix A and reduce its condition number
sgees - sgees - compute for an N -by- N real nonsymmetric matrix A , the eigenvalues, the real Schur form T , and, optionally, the matrix of Schur vectors Z
sgeesx - sgeesx - compute for an N-by-N real nonsymmetric matrix A, the eigenvalues, the real Schur form T, and, optionally, the matrix of Schur vectors Z
sgeev - sgeev - compute for an N-by-N real nonsymmetric matrix A, the eigenvalues and, optionally, the left and/ or right eigenvectors
sgeevx - sgeevx - compute for an N-by-N real nonsymmetric matrix A, the eigenvalues and, optionally, the left and/or right eigenvectors
sgegs - sgegs - routine is deprecated and has been replaced by routine SGGES
sgegv - sgegv - routine is deprecated and has been replaced by routine SGGEV
sgehrd - sgehrd - reduce a real general matrix A to upper Hessenberg form H by an orthogonal similarity transformation
sgelqf - sgelqf - compute an LQ factorization of a real M-by-N matrix A
sgels - sgels - solve overdetermined or underdetermined real linear systems involving an M-by-N matrix A, or its transpose, using a QR or LQ factorization of A
sgelsd - sgelsd - compute the minimum-norm solution to a real linear least squares problem
sgelss - sgelss - compute the minimum norm solution to a real linear least squares problem
sgelsx - sgelsx - routine is deprecated and has been replaced by routine SGELSY
sgelsy - sgelsy - compute the minimum-norm solution to a real linear least squares problem
sgemm - sgemm - perform one of the matrix-matrix operations C :=alpha*op(A )*op(B)+beta*C
sgemv - sgemv - perform one of the matrix-vector operations $y:=a l p h a * A * x+$ beta* $^{*} y$ or $y:=a l p h a * A^{\prime} * x+$ beta*y
sgeqlf - sgeqlf - compute a QL factorization of a real M-by-N matrix A
sgeqp3 - sgeqp3 - compute a QR factorization with column pivoting of a matrix A
sgeqpf - sgeqpf - routine is deprecated and has been replaced by routine SGEQP3
sgeqrf - sgeqrf - compute a QR factorization of a real M-by-N matrix $A$
sger - sger - perform the rank 1 operation $\mathrm{A}:=\operatorname{alpha}{ }^{*} \mathrm{x}^{*} \mathrm{y}^{\prime}+\mathrm{A}$
sgerfs - sgerfs - improve the computed solution to a system of linear equations and provides error bounds and backward error estimates for the solution
sgerqf - sgerqf - compute an RQ factorization of a real M-by-N matrix A
sgesdd - sgesdd - compute the singular value decomposition (SVD) of a real M-by-N matrix A, optionally computing the left and right singular vectors
sgesv - sgesv - compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
sgesvd - sgesvd - compute the singular value decomposition (SVD) of a real M-by-N matrix A, optionally computing the left and/or right singular vectors
sgesvx - sgesvx - use the LU factorization to compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
sgetf2 - sgetf2 - compute an LU factorization of a general m-by-n matrix A using partial pivoting with row interchanges
sgetrf - sgetrf - compute an LU factorization of a general M-by-N matrix A using partial pivoting with row interchanges
sgetri - sgetri - compute the inverse of a matrix using the LU factorization computed by SGETRF
sgetrs - sgetrs - solve a system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$ or $\mathrm{A}^{\prime} * \mathrm{X}=\mathrm{B}$ with a general N -by-N matrix A using the LU factorization computed by SGETRF
sggbak - sggbak - form the right or left eigenvectors of a real generalized eigenvalue problem $A * x=$ lambda*B*x, by backward transformation on the computed eigenvectors of the balanced pair of matrices output by SGGBAL
sggbal - sggbal - balance a pair of general real matrices (A,B)
sgges - sgges - compute for a pair of N-by-N real nonsymmetric matrices (A,B),
sggesx - sggesx - compute for a pair of $N$-by-N real nonsymmetric matrices $(A, B)$, the generalized eigenvalues, the real Schur form (S,T), and,
sggev - sggev - compute for a pair of N -by-N real nonsymmetric matrices (A,B)
sggevx - sggevx - compute for a pair of N-by-N real nonsymmetric matrices (A,B)
sggglm - sggglm - solve a general Gauss-Markov linear model (GLM) problem
sgghrd - sgghrd - reduce a pair of real matrices (A,B) to generalized upper Hessenberg form using orthogonal transformations, where $A$ is a general matrix and $B$ is upper triangular
sgglse - sgglse - solve the linear equality-constrained least squares (LSE) problem
sggqrf - sggqrf - compute a generalized QR factorization of an N -by-M matrix A and an N -by-P matrix B .
sggrqf - sggrqf - compute a generalized RQ factorization of an M-by-N matrix A and a P-by-N matrix B
sggsvd - sggsvd - compute the generalized singular value decomposition (GSVD) of an M-by-N real matrix A and P-by-N real matrix B
sggsvp - sggsvp - compute orthogonal matrices $U, V$ and $Q$ such that $N-K-L K L U^{*} * A^{*} Q=K(0 A 12 A 13)$ if $M-$ $K-L>=0$
sgssco - sgssco - General sparse solver condition number estimate.
sgssda - sgssda - Deallocate working storage for the general sparse solver.
sgssfa - sgssfa - General sparse solver numeric factorization.
sgssfs - sgssfs - General sparse solver one call interface.
sgssin - sgssin - Initialize the general sparse solver.
sgssor - sgssor - General sparse solver ordering and symbolic factorization.
sgssps - sgssps - Print general sparse solver statics.
sgssrp - sgssrp - Return permutation used by the general sparse solver.
sgsssl - sgsssl-Solve routine for the general sparse solver.
sgssuo - sgssuo - User supplied permutation for ordering used in the general sparse solver.
sgtcon - sgtcon - estimate the reciprocal of the condition number of a real tridiagonal matrix A using the LU factorization as computed by SGTTRF
sgthr - sgthr - Gathers specified elements from y into $x$.
sgthrz - sgthrz - Gather and zero.
sgtrfs - sgtrfs - improve the computed solution to a system of linear equations when the coefficient matrix is tridiagonal, and provides error bounds and backward error estimates for the solution
sgtsv - sgtsv - solve the equation $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
sgtsvx - sgtsvx - use the LU factorization to compute the solution to a real system of linear equations A * $\mathrm{X}=\mathrm{B}$ or $\mathrm{A}^{*} * \mathrm{~T} * \mathrm{X}=\mathrm{B}$,
sgttrf - sgttrf - compute an LU factorization of a real tridiagonal matrix A using elimination with partial pivoting and row interchanges
sgttrs - sgttrs - solve one of the systems of equations $A^{*} X=B$ or $A^{*} * X=B$,
shgeqz - shgeqz - implement a single-/double-shift version of the QZ method for finding the generalized eigenvalues $w(j)=\left(\operatorname{ALPHAR}(\mathrm{j})+\mathrm{i}^{*} \operatorname{ALPHAI}(\mathrm{j})\right) / \operatorname{BETAR}(\mathrm{j})$ of the equation $\operatorname{det}(\mathrm{A}-\mathrm{w}(\mathrm{i}) \mathrm{B})=0$ In addition, the pair A,B may be reduced to generalized Schur form
shsein - shsein - use inverse iteration to find specified right and/or left eigenvectors of a real upper Hessenberg matrix H
shseqr - shseqr - compute the eigenvalues of a real upper Hessenberg matrix H and, optionally, the matrices T and Z from the Schur decomposition $\mathrm{H}=\mathrm{Z} \mathrm{T} \mathrm{Z} * *$, where T is an upper quasi-triangular matrix (the Schur form), and Z is the orthogonal matrix of Schur vectors
$\underline{\text { sinqb }}$ - sinqb - synthesize a Fourier sequence from its representation in terms of a sine series with odd wave numbers. The SINQ operations are unnormalized inverses of themselves, so a call to SINQF followed by a call to SINQB will multiply the input sequence by $4 * \mathrm{~N}$.
sinqf - sinqf - compute the Fourier coefficients in a sine series representation with only odd wave numbers. The SINQ operations are unnormalized inverses of themselves, so a call to SINQF followed by a call to SINQB will multiply the input sequence by $4 * \mathrm{~N}$.
sinqi - sinqi - initialize the array xWSAVE, which is used in both SINQF and SINQB.
sint - sint - compute the discrete Fourier sine transform of an odd sequence. The SINT transforms are unnormalized inverses of themselves, so a call of SINT followed by another call of SINT will multiply the input sequence by $2 *(\mathrm{~N}+1)$.
sinti - sinti - initialize the array WSAVE, which is used in subroutine SINT.
sjadmm - sjadmm - Jagged diagonal matrix-matrix multiply (modified Ellpack)
sjadrp - sjadrp - right permutation of a jagged diagonal matrix
sjadsm - sjadsm - Jagged-diagonal format triangular solve
slagtf - slagtf - factorize the matrix (T-lambda*I), where T is an n by n tridiagonal matrix and lambda is a scalar, as T-lambda* $\mathrm{I}=\mathrm{PLU}$
slamrg - slamrg - will create a permutation list which will merge the elements of A (which is composed of two independently sorted sets) into a single set which is sorted in ascending order
slarz - slarz - applies a real elementary reflector H to a real M-by-N matrix C, from either the left or the right
slarzb - slarzb - applies a real block reflector H or its transpose $\mathrm{H}^{* *} \mathrm{~T}$ to a real distributed $\mathrm{M}-\mathrm{by}-\mathrm{N} \mathrm{C}$ from the left or the right
slarzt - slarzt - form the triangular factor T of a real block reflector H of order $>\mathrm{n}$, which is defined as a product of k elementary reflectors
slasrt - slasrt - the numbers in D in increasing order (if ID = 'I') or in decreasing order (if ID = 'D' )
slatzm - slatzm - routine is deprecated and has been replaced by routine SORMRZ
snrm2 - snrm2 - Return the Euclidian norm of a vector.
sopgtr - sopgtr - generate a real orthogonal matrix Q which is defined as the product of $\mathrm{n}-1$ elementary reflectors H (i) of order n , as returned by SSPTRD using packed storage
sopmtr - sopmtr - overwrite the general real M-by-N matrix C with SIDE = 'L' SIDE $=$ ' R ' TRANS $=$ ' N '
sorg21-sorg21 - generate an m by n real matrix Q with orthonormal columns,
sorg2r - sorg2r - generate an $m$ by $n$ real matrix Q with orthonormal columns,
sorgbr - sorgbr - generate one of the real orthogonal matrices Q or $\mathrm{P}^{* *} \mathrm{~T}$ determined by SGEBRD when reducing
a real matrix A to bidiagonal form
sorghr - sorghr - generate a real orthogonal matrix Q which is defined as the product of IHI-ILO elementary reflectors of order N , as returned by SGEHRD
sorgl2 - sorgl2 - generate an $m$ by $n$ real matrix Q with orthonormal rows,
sorglq - sorglq - generate an M-by-N real matrix Q with orthonormal rows,
sorgql - sorgql - generate an M-by-N real matrix Q with orthonormal columns,
sorgqr - sorgqr - generate an M-by-N real matrix Q with orthonormal columns,
sorgr2 - sorgr2 - generate an m by $n$ real matrix Q with orthonormal rows,
sorgrq - sorgrq - generate an M-by-N real matrix Q with orthonormal rows,
sorgtr - sorgtr - generate a real orthogonal matrix Q which is defined as the product of $\mathrm{n}-1$ elementary reflectors of order N , as returned by SSYTRD
sormbr - sormbr - VECT = 'Q', SORMBR overwrites the general real $\mathrm{M}-\mathrm{by}-\mathrm{N}$ matrix C with $\operatorname{SIDE}=$ 'L' $\mathrm{SIDE}=$ 'R' TRANS = 'N'
sormhr - sormhr - overwrite the general real M-by-N matrix C with SIDE = 'L' SIDE = 'R' TRANS = 'N'
sormlq - sormlq - overwrite the general real M-by-N matrix C with SIDE $=$ 'L' SIDE $=$ 'R' TRANS $=$ ' N '
sormql - sormql - overwrite the general real M-by-N matrix C with SIDE $=$ 'L' SIDE $=$ ' $\mathrm{R}^{\prime}$ TRANS $=$ ' N '
sormqr - sormqr - overwrite the general real M-by-N matrix C with SIDE $=$ 'L' SIDE $=$ ' $\mathrm{R}^{\prime}$ TRANS $=$ ' N '
sormrq - sormrq - overwrite the general real M-by-N matrix C with SIDE = 'L' SIDE $=$ ' R ' TRANS $=$ ' N '
sormrz - sormrz - overwrite the general real M-by-N matrix C with SIDE = 'L' SIDE = 'R' TRANS = 'N'
sormtr - sormtr - overwrite the general real M-by-N matrix C with $\operatorname{SIDE}=$ 'L' SIDE $=$ ' R ' TRANS $=$ ' N '
spbcon - spbcon - estimate the reciprocal of the condition number (in the 1-norm) of a real symmetric positive definite band matrix using the Cholesky factorization $\mathrm{A}=\mathrm{U}^{*} * \mathrm{~T}^{*} \mathrm{U}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ computed by SPBTRF
spbequ - spbequ - compute row and column scalings intended to equilibrate a symmetric positive definite band matrix A and reduce its condition number (with respect to the two-norm)
spbrfs - spbrfs - improve the computed solution to a system of linear equations when the coefficient matrix is symmetric positive definite and banded, and provides error bounds and backward error estimates for the solution
spbstf - spbstf - compute a split Cholesky factorization of a real symmetric positive definite band matrix A
$\underline{\text { spbsv }}$ - spbsv - compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
spbsvx - spbsvx - use the Cholesky factorization $A=U^{*} * T^{*} U$ or $A=L^{*} L^{* *} T$ to compute the solution to a real system of linear equations $A * X=B$,
spbtf2 - spbtf2 - compute the Cholesky factorization of a real symmetric positive definite band matrix A
spbtrf - spbtrf - compute the Cholesky factorization of a real symmetric positive definite band matrix A
spbtrs - spbtrs - solve a system of linear equations $A * X=B$ with a symmetric positive definite band matrix $A$ using the Cholesky factorization $\mathrm{A}=\mathrm{U}^{* *} \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=\mathrm{L} * \mathrm{~L}^{*} * \mathrm{~T}$ computed by SPBTRF
spocon - spocon - estimate the reciprocal of the condition number (in the 1-norm) of a real symmetric positive definite matrix using the Cholesky factorization $\mathrm{A}=\mathrm{U}^{*} * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ computed by SPOTRF
spoequ - spoequ - compute row and column scalings intended to equilibrate a symmetric positive definite matrix A and reduce its condition number (with respect to the two-norm)
sporfs - sporfs - improve the computed solution to a system of linear equations when the coefficient matrix is symmetric positive definite,
sposv - sposv - compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
sposvx - sposvx - use the Cholesky factorization $A=U^{*} * T^{*} U$ or $A=L^{*} L^{* *} T$ to compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
spotf2 - spotf2 - compute the Cholesky factorization of a real symmetric positive definite matrix A
$\underline{\text { spotrf }}$ - spotrf - compute the Cholesky factorization of a real symmetric positive definite matrix A
spotri - spotri - compute the inverse of a real symmetric positive definite matrix A using the Cholesky factorization $\mathrm{A}=\mathrm{U}^{*} * \mathrm{~T}^{*} \mathrm{U}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{~L}^{* *} \mathrm{~T}$ computed by SPOTRF
spotrs - spotrs - solve a system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$ with a symmetric positive definite matrix A using the Cholesky factorization $\mathrm{A}=\mathrm{U}^{*} * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=\mathrm{L} * \mathrm{~L}^{*} * \mathrm{~T}$ computed by SPOTRF
sppcon - sppcon - estimate the reciprocal of the condition number (in the 1-norm) of a real symmetric positive
definite packed matrix using the Cholesky factorization $A=U * * T * U$ or $A=L * L * * T$ computed by SPPTRF
sppequ - sppequ - compute row and column scalings intended to equilibrate a symmetric positive definite matrix A in packed storage and reduce its condition number (with respect to the two-norm)
spprfs - spprfs - improve the computed solution to a system of linear equations when the coefficient matrix is symmetric positive definite and packed, and provides error bounds and backward error estimates for the solution
$\underline{\text { sppsv }}$ - sppsv - compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
sppsvx - sppsvx - use the Cholesky factorization $A=U^{*} * T^{*} U$ or $A=L^{*} L^{*} * T$ to compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
spptrf - spptrf - compute the Cholesky factorization of a real symmetric positive definite matrix A stored in packed format
spptri - spptri - compute the inverse of a real symmetric positive definite matrix A using the Cholesky factorization $\mathrm{A}=\mathrm{U}^{*} * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ computed by SPPTRF
spptrs - spptrs - solve a system of linear equations $A * X=B$ with a symmetric positive definite matrix $A$ in packed storage using the Cholesky factorization $A=U * * T * U$ or $A=L * L * * T$ computed by SPPTRF
sptcon - sptcon - compute the reciprocal of the condition number (in the 1-norm) of a real symmetric positive definite tridiagonal matrix using the factorization $A=L * D * L * * T$ or $A=U * * T * D * U$ computed by SPTTRF
spteqr - spteqr - compute all eigenvalues and, optionally, eigenvectors of a symmetric positive definite tridiagonal matrix by first factoring the matrix using SPTTRF, and then calling SBDSQR to compute the singular values of the bidiagonal factor
sptrfs - sptrfs - improve the computed solution to a system of linear equations when the coefficient matrix is symmetric positive definite and tridiagonal, and provides error bounds and backward error estimates for the solution
sptsv - sptsv - compute the solution to a real system of linear equations $A * X=B$, where $A$ is an $N-b y-N$ symmetric positive definite tridiagonal matrix, and X and B are N -by-NRHS matrices.
sptsvx - sptsvx - use the factorization $A=L^{*} D^{*} L^{*} * T$ to compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$, where A is an N -by- N symmetric positive definite tridiagonal matrix and X and B are N -by-NRHS matrices
spttrf - spttrf - compute the $L^{*} D^{*} L^{\prime}$ factorization of a real symmetric positive definite tridiagonal matrix A
spttrs - spttrs - solve a tridiagonal system of the form $\mathrm{A} * \mathrm{X}=\mathrm{B}$ using the $\mathrm{L} * \mathrm{D} * \mathrm{~L}$ ' factorization of A computed by SPTTRF
sptts2 - sptts2 - solve a tridiagonal system of the form $A * X=B$ using the $L * D * L$ factorization of A computed by SPTTRF
srot - srot - Apply a Given's rotation constructed by SROTG.
srotg - srotg - Construct a Given's plane rotation
sroti - sroti - Apply an indexed Givens rotation.
srotm - srotm - Apply a Gentleman's modified Given's rotation constructed by SROTMG.
srotmg - srotmg - Construct a Gentleman's modified Given's plane rotation
ssbev - ssbev - compute all the eigenvalues and, optionally, eigenvectors of a real symmetric band matrix A
ssbevd - ssbevd - compute all the eigenvalues and, optionally, eigenvectors of a real symmetric band matrix A
ssbevx - ssbevx - compute selected eigenvalues and, optionally, eigenvectors of a real symmetric band matrix A
ssbgst - ssbgst - reduce a real symmetric-definite banded generalized eigenproblem $A * x=1 a m b d a * B * x$ to standard form $\mathrm{C}^{*} \mathrm{y}=$ lambda* y ,
ssbgv - ssbgv - compute all the eigenvalues, and optionally, the eigenvectors of a real generalized symmetricdefinite banded eigenproblem, of the form $A * x=(l a m b d a) * B * x$
ssbgvd - ssbgvd - compute all the eigenvalues, and optionally, the eigenvectors of a real generalized symmetricdefinite banded eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\mathrm{lambda}) * \mathrm{~B}$ * x
ssbgvx - ssbgvx - compute selected eigenvalues, and optionally, eigenvectors of a real generalized symmetricdefinite banded eigenproblem, of the form $A * x=(l a m b d a) * B * x$
ssbmv - ssbmv - perform the matrix-vector operation $y:=a l p h a * A * x+$ beta* $y$
ssbtrd - ssbtrd - reduce a real symmetric band matrix A to symmetric tridiagonal form T by an orthogonal similarity transformation
sscal - sscal - Compute y $:=$ alpha $* y$
ssctr - ssctr - Scatters elements from x into y .
sskymm - sskymm - Skyline format matrix-matrix multiply
sskysm - sskysm - Skyline format triangular solve
sspcon - sspcon - estimate the reciprocal of the condition number (in the 1-norm) of a real symmetric packed matrix A using the factorization $\mathrm{A}=\mathrm{U}^{*} \mathrm{D}^{*} \mathrm{U}^{*} * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{D}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ computed by SSPTRF
sspev - sspev - compute all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix A in packed storage
sspevd - sspevd - compute all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix A in packed storage
sspevx - sspevx - compute selected eigenvalues and, optionally, eigenvectors of a real symmetric matrix A in packed storage
sspgst - sspgst - reduce a real symmetric-definite generalized eigenproblem to standard form, using packed storage
sspgv - sspgv - compute all the eigenvalues and, optionally, the eigenvectors of a real generalized symmetricdefinite eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\operatorname{lambda}) * \mathrm{~B} * \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda}) * \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lambda}) * \mathrm{x}$
sspgvd - sspgvd - compute all the eigenvalues, and optionally, the eigenvectors of a real generalized symmetricdefinite eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\operatorname{lambda}) * \mathrm{~B} * \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda}) * \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lambda}) * \mathrm{x}$
sspgvx - sspgvx - compute selected eigenvalues, and optionally, eigenvectors of a real generalized symmetricdefinite eigenproblem, of the form $\mathrm{A}^{*} \mathrm{x}=(\operatorname{lambda}) * \mathrm{~B} * \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda}) * \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lambda}) * \mathrm{x}$
$\underline{\text { sspmv }}-$ sspmv - perform the matrix-vector operation $y:=a l p h a * A * x+$ beta* $y$
sspr - sspr - perform the symmetric rank 1 operation $\mathrm{A}:=$ alpha* $\mathrm{x}^{*} \mathrm{x}^{\prime}+\mathrm{A}$
sspr2 - sspr2 - perform the symmetric rank 2 operation A :=alpha* $x^{*} y^{\prime}+$ alpha* $y^{*} x^{\prime}+A$
ssprfs - ssprfs - improve the computed solution to a system of linear equations when the coefficient matrix is symmetric indefinite and packed, and provides error bounds and backward error estimates for the solution
sspsv - sspsv - compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
sspsvx - sspsvx - use the diagonal pivoting factorization $A=U * D^{*} U^{* *} T$ or $A=L^{*} D^{*} L^{*} * T$ to compute the solution to a real system of linear equations $A * X=B$, where $A$ is an $N-b y-N$ symmetric matrix stored in packed format and X and B are N -by-NRHS matrices
ssptrd - ssptrd - reduce a real symmetric matrix A stored in packed form to symmetric tridiagonal form T by an orthogonal similarity transformation
ssptrf - ssptrf - compute the factorization of a real symmetric matrix A stored in packed format using the BunchKaufman diagonal pivoting method
ssptri - ssptri - compute the inverse of a real symmetric indefinite matrix A in packed storage using the factorization $\mathrm{A}=\mathrm{U}^{*} \mathrm{D}^{*} \mathrm{U}^{* *} \mathrm{~T}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{D}^{*} \mathrm{~L}^{* *} \mathrm{~T}$ computed by SSPTRF
ssptrs - ssptrs - solve a system of linear equations $A * X=B$ with a real symmetric matrix $A$ stored in packed format using the factorization $A=U^{*} D^{*} U^{*} * T$ or $A=L * D * L * T$ computed by SSPTRF
sstebz - sstebz - compute the eigenvalues of a symmetric tridiagonal matrix T
sstedc - sstedc - compute all eigenvalues and, optionally, eigenvectors of a symmetric tridiagonal matrix using the divide and conquer method
sstegr - sstegr - (a) Compute T-sigma_i = L_i D_i L_i^T, such that L_i D_i L_i^ ${ }^{\wedge}$ is a relatively robust representation
sstein - sstein - compute the eigenvectors of a real symmetric tridiagonal matrix T corresponding to specified eigenvalues, using inverse iteration
ssteqr - ssteqr - compute all eigenvalues and, optionally, eigenvectors of a symmetric tridiagonal matrix using the implicit QL or QR method
ssterf - ssterf - compute all eigenvalues of a symmetric tridiagonal matrix using the Pal-Walker-Kahan variant of the QL or QR algorithm
sstev - sstev - compute all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix A
sstevd - sstevd - compute all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix
sstevr - sstevr - compute selected eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix T
sstevx - sstevx - compute selected eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix A
sstsv - sstsv - compute the solution to a system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$ where A is a symmetric tridiagonal matrix
ssttrf - ssttrf - compute the factorization of a symmetric tridiagonal matrix A
ssttrs - ssttrs - computes the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$
sswap - sswap - Exchange vectors $x$ and $y$.
ssycon - ssycon - estimate the reciprocal of the condition number (in the 1-norm) of a real symmetric matrix A using the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}^{*} * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ computed by SSYTRF
ssyev - ssyev - compute all eigenvalues and, optionally, eigenvectors of a real symmetric matrix A
ssyevd - ssyevd - compute all eigenvalues and, optionally, eigenvectors of a real symmetric matrix A
ssyevr - ssyevr - compute selected eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix T
ssyevx - ssyevx - compute selected eigenvalues and, optionally, eigenvectors of a real symmetric matrix A
ssygs2 - ssygs2 - reduce a real symmetric-definite generalized eigenproblem to standard form
ssygst - ssygst - reduce a real symmetric-definite generalized eigenproblem to standard form
ssygv - ssygv - compute all the eigenvalues, and optionally, the eigenvectors of a real generalized symmetricdefinite eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{~B} * \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda})^{*} \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{x}$
ssygvd - ssygvd - compute all the eigenvalues, and optionally, the eigenvectors of a real generalized symmetricdefinite eigenproblem, of the form $\mathrm{A}^{*} \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{~B}^{*} \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda})^{*} \mathrm{x}$, or $\mathrm{B}^{*} \mathrm{~A}^{*} \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{x}$
ssygvx - ssygvx - compute selected eigenvalues, and optionally, eigenvectors of a real generalized symmetricdefinite eigenproblem, of the form $\mathrm{A} * \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{~B} * \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda}) * \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{x}$
 beta*C

$\underline{\text { ssyr }- \text { ssyr }- \text { perform the symmetric rank } 1 \text { operation } \mathrm{A}:=\operatorname{alpha}{ }^{*} \mathrm{x}^{*} \mathrm{x}^{\prime}+\mathrm{A}, ~}$

ssyr2k - ssyr2k - perform one of the symmetric rank $2 k$ operations $C:=$ alpha* $A^{*} B^{\prime}+$ alpha* $B^{*} A^{\prime}+$ beta* $C$ or $C$ : $=$ alpha* ${ }^{\prime} * \mathrm{~B}+$ alpha* ${ }^{\prime} * \mathrm{~A}+$ beta* C
ssyrfs - ssyrfs - improve the computed solution to a system of linear equations when the coefficient matrix is symmetric indefinite, and provides error bounds and backward error estimates for the solution
 beta*C
$\underline{\text { ssysv - ssysv - compute the solution to a real system of linear equations } \mathrm{A} * \mathrm{X}=\mathrm{B}, ~}$
ssysvx - ssysvx - use the diagonal pivoting factorization to compute the solution to a real system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
ssytd2 - ssytd2 - reduce a real symmetric matrix A to symmetric tridiagonal form T by an orthogonal similarity transformation
ssytf2 - ssytf2 - compute the factorization of a real symmetric matrix A using the Bunch-Kaufman diagonal pivoting method
ssytrd - ssytrd - reduce a real symmetric matrix A to real symmetric tridiagonal form T by an orthogonal similarity transformation
ssytrf - ssytrf - compute the factorization of a real symmetric matrix A using the Bunch-Kaufman diagonal pivoting method
ssytri - ssytri - compute the inverse of a real symmetric indefinite matrix A using the factorization $\mathrm{A}=$ $\mathrm{U}^{*} \mathrm{D}^{*} \mathrm{U}^{* *} \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L}^{*} * \mathrm{~T}$ computed by SSYTRF
ssytrs - ssytrs - solve a system of linear equations $A * X=B$ with a real symmetric matrix $A$ using the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D}^{*} \mathrm{U}^{*} * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ computed by SSYTRF
 norm or the infinity-norm

stbrfs - stbrfs - provide error bounds and backward error estimates for the solution to a system of linear equations with a triangular band coefficient matrix


stgevc - stgevc - compute some or all of the right and/or left generalized eigenvectors of a pair of real upper triangular matrices ( $\mathrm{A}, \mathrm{B}$ )
stgexc - stgexc - reorder the generalized real Schur decomposition of a real matrix pair $(A, B)$ using an orthogonal equivalence transformation $(A, B)=Q^{*}(A, B) * Z^{\prime}$,
stgsen - stgsen - reorder the generalized real Schur decomposition of a real matrix pair (A, B) (in terms of an orthonormal equivalence trans- formation $\mathrm{Q}^{\prime *}(\mathrm{~A}, \mathrm{~B}) * \mathrm{Z}$ ), so that a selected cluster of eigenvalues appears in the leading diagonal blocks of the upper quasi-triangular matrix A and the upper triangular B
stgsja - stgsja - compute the generalized singular value decomposition (GSVD) of two real upper triangular (or trapezoidal) matrices A and B
stgsna - stgsna - estimate reciprocal condition numbers for specified eigenvalues and/or eigenvectors of a matrix pair (A, B) in generalized real Schur canonical form (or of any matrix pair ( $Q^{*} A^{*} Z^{\prime}, Q^{*} B^{*} Z^{\prime}$ ) with orthogonal matrices Q and Z , where $\mathrm{Z}^{\prime}$ denotes the transpose of Z
stgsyl - stgsyl - solve the generalized Sylvester equation
stpcon - stpcon - estimate the reciprocal of the condition number of a packed triangular matrix A , in either the 1norm or the infinity-norm

stprfs - stprfs - provide error bounds and backward error estimates for the solution to a system of linear equations with a triangular packed coefficient matrix
$\underline{\text { stpsv }}$ - stpsv - solve one of the systems of equations $A^{*} x=b$, or $A^{*} x=b$
stptri - stptri - compute the inverse of a real upper or lower triangular matrix A stored in packed format
stptrs - stptrs - solve a triangular system of the form $\mathrm{A} * \mathrm{X}=\mathrm{B}$ or $\mathrm{A}^{*} * \mathrm{~T} * \mathrm{X}=\mathrm{B}$,
strans - strans - transpose and scale source matrix
strcon - strcon - estimate the reciprocal of the condition number of a triangular matrix A, in either the 1-norm or the infinity-norm
$\underline{\text { strevc - strevc - compute some or all of the right and/or left eigenvectors of a real upper quasi-triangular matrix } \mathrm{T}}$
strexc - strexc - reorder the real Schur factorization of a real matrix $A=Q^{*} T^{*} Q^{*} * T$, so that the diagonal block of T with row index IFST is moved to row ILST
strmm - strmm - perform one of the matrix-matrix operations B :=alpha*op(A)*B, or B :=alpha*B*op(A)
$\underline{\text { strmv }}$ - strmv - perform one of the matrix-vector operations $x:=A^{*} x$, or $x:=A^{\prime} * x$
strrfs - strrfs - provide error bounds and backward error estimates for the solution to a system of linear equations
with a triangular coefficient matrix
strsen - strsen - reorder the real Schur factorization of a real matrix $A=Q^{*} T^{*} Q^{*} * T$, so that a selected cluster of eigenvalues appears in the leading diagonal blocks of the upper quasi-triangular matrix T ,
$\underline{\operatorname{strsm}}-\operatorname{strsm}-$ solve one of the matrix equations op(A)$)^{*} X=\operatorname{alpha} * B$, or $X * o p(A)=$ alpha*B
strsna - strsna - estimate reciprocal condition numbers for specified eigenvalues and/or right eigenvectors of a real upper quasi-triangular matrix T (or of any matrix $\mathrm{Q}^{*} \mathrm{~T}^{*} \mathrm{Q}^{* *} \mathrm{~T}$ with Q orthogonal)
$\underline{\text { strsv }}$ - strsv - solve one of the systems of equations $A^{*} x=b$, or $A^{\prime} * x=b$
strsyl - strsyl - solve the real Sylvester matrix equation
strti2 - strti2 - compute the inverse of a real upper or lower triangular matrix
strtri - strtri - compute the inverse of a real upper or lower triangular matrix A
strtrs - strtrs - solve a triangular system of the form $\mathrm{A} * \mathrm{X}=\mathrm{B}$ or $\mathrm{A}^{*} \mathrm{~T}^{*} \mathrm{X}=\mathrm{B}$,
stzrqf - stzrqf - routine is deprecated and has been replaced by routine STZRZF
stzrzf - stzrzf - reduce the M-by-N ( $\mathrm{M}<=\mathrm{N}$ ) real upper trapezoidal matrix A to upper triangular form by means of orthogonal transformations
sunperf_version - sunperf_version - gets library information .HP 1i SUBROUTINE SUNPERF_VERSION (VERSION, PATCH, UPDATE) .HP 1i INTEGER VERSION, PATCH, UPDATE .HP 1i
svbrmm - svbrmm - variable block sparse row format matrix-matrix multiply
svbrsm - svbrsm - variable block sparse row format triangular solve
swiener - swiener - perform Wiener deconvolution of two signals
use_threads - use_threads - set the upper bound on the number of threads that the calling thread wants used
using_threads - using_threads - returns the current Use number set by the USE_THREADS subroutine
vcfftb - vcfftb - compute a periodic sequence from its Fourier coefficients. The VCFFT operations are normalized, so a call of VCFFTF followed by a call of VCFFTB will return the original sequence.
vcfftf - vcfftf - compute the Fourier coefficients of a periodic sequence. The VCFFT operations are normalized,
so a call of VCFFTF followed by a call of VCFFTB will return the original sequence.
vcffti - vcffti - initialize the array WSAVE, which is used in both VCFFTF and VCFFTB.
vcosqb - vcosqb - synthesize a Fourier sequence from its representation in terms of a cosine series with odd wave numbers. The VCOSQ operations are normalized, so a call of VCOSQF followed by a call of VCOSQB will return the original sequence.
vcosqf - vcosqf - compute the Fourier coefficients in a cosine series representation with only odd wave numbers. The VCOSQ operations are normalized, so a call of VCOSQF followed by a call of VCOSQB will return the original sequence.
vcosqi - vcosqi - initialize the array WSAVE, which is used in both VCOSQF and VCOSQB.
vcost - vcost - compute the discrete Fourier cosine transform of an even sequence. The VCOST transform is normalized, so a call of VCOST followed by a call of VCOST will return the original sequence.
vcosti - vcosti - initialize the array WSAVE, which is used in VCOST.
vdcosqb - vdcosqb - synthesize a Fourier sequence from its representation in terms of a cosine series with odd wave numbers. The VCOSQ operations are normalized, so a call of VCOSQF followed by a call of VCOSQB will return the original sequence.
vdcosqf - vdcosqf - compute the Fourier coefficients in a cosine series representation with only odd wave numbers. The VCOSQ operations are normalized, so a call of VCOSQF followed by a call of VCOSQB will return the original sequence.
vdcosqi - vdcosqi - initialize the array WSAVE, which is used in both VCOSQF and VCOSQB.
vdcost - vdcost - compute the discrete Fourier cosine transform of an even sequence. The VCOST transform is normalized, so a call of VCOST followed by a call of VCOST will return the original sequence.
vdcosti - vdcosti - initialize the array WSAVE, which is used in VCOST.
vdfftb - vdfftb - compute a periodic sequence from its Fourier coefficients. The VRFFT operations are normalized, so a call of VRFFTF followed by a call of VRFFTB will return the original sequence.
vdfftf - vdfftf - compute the Fourier coefficients of a periodic sequence. The VRFFT operations are normalized, so a call of VRFFTF followed by a call of VRFFTB will return the original sequence.
vdffti - vdffti - initialize the array WSAVE, which is used in both VRFFTF and VRFFTB.
$\underline{\text { vdsinqb }}$ - vdsinqb - synthesize a Fourier sequence from its representation in terms of a sine series with odd wave
numbers. The VSINQ operations are normalized, so a call of VSINQF followed by a call of VSINQB will return the original sequence.
vdsinqf - vdsinqf - compute the Fourier coefficients in a sine series representation with only odd wave numbers. The VSINQ operations are normalized, so a call of VSINQF followed by a call of VSINQB will return the original sequence.
vdsinqi - vdsinqi - initialize the array WSAVE, which is used in both VSINQF and VSINQB.
vdsint - vdsint - compute the discrete Fourier sine transform of an odd sequence. The VSINT transforms are unnormalized inverses of themselves, so a call of VSINT followed by another call of VSINT will multiply the input sequence by $2 *(\mathrm{~N}+1)$. The VSINT transforms are normalized, so a call of VSINT followed by a call of VSINT will return the original sequence.
vdsinti - vdsinti - initialize the array WSAVE, which is used in subroutine VSINT.
vrfftb - vrfftb - compute a periodic sequence from its Fourier coefficients. The VRFFT operations are normalized, so a call of VRFFTF followed by a call of VRFFTB will return the original sequence.
vrfftf - vrfftf - compute the Fourier coefficients of a periodic sequence. The VRFFT operations are normalized, so a call of VRFFTF followed by a call of VRFFTB will return the original sequence.
vrffti - vrffti - initialize the array WSAVE, which is used in both VRFFTF and VRFFTB.
vsinqb - vsinqb - synthesize a Fourier sequence from its representation in terms of a sine series with odd wave numbers. The VSINQ operations are normalized, so a call of VSINQF followed by a call of VSINQB will return the original sequence.
vsinqf - vsinqf - compute the Fourier coefficients in a sine series representation with only odd wave numbers. The VSINQ operations are normalized, so a call of VSINQF followed by a call of VSINQB will return the original sequence.
vsinqi - vsinqi - initialize the array WSAVE, which is used in both VSINQF and VSINQB.
vsint - vsint - compute the discrete Fourier sine transform of an odd sequence. The VSINT transforms are unnormalized inverses of themselves, so a call of VSINT followed by another call of VSINT will multiply the input sequence by $2 *(\mathrm{~N}+1)$. The VSINT transforms are normalized, so a call of VSINT followed by a call of VSINT will return the original sequence.
vsinti - vsinti - initialize the array WSAVE, which is used in subroutine VSINT.
vzfftb - vzfftb - compute a periodic sequence from its Fourier coefficients. The VZFFT operations are normalized, so a call of VZFFTF followed by a call of VZFFTB will return the original sequence.
vzfftf - vzfftf - compute the Fourier coefficients of a periodic sequence. The VZFFT operations are normalized, so a call of VZFFTF followed by a call of VZFFTB will return the original sequence.
vzffti - vzffti - initialize the array WSAVE, which is used in both VZFFTF and VZFFTB.
zaxpy - zaxpy - compute $\mathrm{y}:=$ alpha $* \mathrm{x}+\mathrm{y}$
zaxpyi - zaxpyi - Compute $\mathrm{y}:=$ alpha $* \mathrm{x}+\mathrm{y}$
zbcomm - zbcomm - block coordinate matrix-matrix multiply
zbdimm - zbdimm - block diagonal format matrix-matrix multiply
zbdism - zbdism - block diagonal format triangular solve
zbdsqr - zbdsqr - compute the singular value decomposition (SVD) of a real N-by-N (upper or lower) bidiagonal matrix $B$.
zbelmm - zbelmm - block Ellpack format matrix-matrix multiply
zbelsm - zbelsm - block Ellpack format triangular solve
zbscmm - zbscmm - block sparse column matrix-matrix multiply
zbscsm - zbscsm - block sparse column format triangular solve
zbsrmm - zbsrmm - block sparse row format matrix-matrix multiply
zbsrsm - zbsrsm - block sparse row format triangular solve
$\underline{\text { zenvcor - zenvcor - compute the convolution or correlation of complex vectors }}$
zcnvcor2 - zcnvcor2 - compute the convolution or correlation of complex matrices
zcoomm - zcoomm - coordinate matrix-matrix multiply

Zcopy - zcopy - Copy x to y
zcscmm - zcscmm - compressed sparse column format matrix-matrix multiply
$\underline{\text { zcscsm }}$ - zcscsm - compressed sparse column format triangular solve
zcsrmm - zcsrmm - compressed sparse row format matrix-matrix multiply
zcsrsm - zcsrsm - compressed sparse row format triangular solve
zdiamm - zdiamm - diagonal format matrix-matrix multiply
zdiasm - zdiasm - diagonal format triangular solve
$\underline{\text { zdotc }-z d o t c}$ - compute the dot product of two vectors conjg(x) and y .
zdotci - zdotci - Compute the complex conjugated indexed dot product.

zdotui - zdotui - Compute the complex unconjugated indexed dot product.
zdrot - zdrot - Apply a plane rotation.
zdscal - zdscal - Compute y $:=$ alpha * y
zellmm - zellmm - Ellpack format matrix-matrix multiply
zellsm - zellsm - Ellpack format triangular solve
$\underline{\text { zfft2b }}$ - zfft2b - compute a periodic sequence from its Fourier coefficients. The FFT operations are unnormalized, so a call of ZFFT2F followed by a call of ZFFT2B will multiply the input sequence by $\mathrm{M} * \mathrm{~N}$.
zfft2f - zfft2f - compute the Fourier coefficients of a periodic sequence. The FFT operations are unnormalized, so a call of ZFFT2F followed by a call of ZFFT2B will multiply the input sequence by $\mathrm{M}^{*} \mathrm{~N}$.
zfft2i - zfft2i - initialize the array WSAVE, which is used in both the forward and backward transforms.
$\underline{\text { zfft3b }}$ - zfft3b - compute a periodic sequence from its Fourier coefficients. The FFT operations are unnormalized, so a call of ZFFT3F followed by a call of ZFFT3B will multiply the input sequence by $\mathrm{M}^{*} \mathrm{~N} * \mathrm{~K}$.
zfft3f - zfft3f - compute the Fourier coefficients of a periodic sequence. The FFT operations are unnormalized, so a call of ZFFT3F followed by a call of ZFFT3B will multiply the input sequence by $\mathrm{M}^{*} \mathrm{~N}^{*} \mathrm{~K}$.
zfft3i - zfft3i - initialize the array WSAVE, which is used in both ZFFT3F and ZFFT3B.
zfftb - zfftb - compute a periodic sequence from its Fourier coefficients. The FFT operations are unnormalized, so a call of ZFFTF followed by a call of ZFFTB will multiply the input sequence by N .
zfftd - zfftd - initialize the trigonometric weight and factor tables or compute the inverse Fast Fourier Transform of a double complex sequence.
zfftd2 - zfftd2 - initialize the trigonometric weight and factor tables or compute the two-dimensional inverse Fast Fourier Transform of a two-dimensional double complex array.
zfftd3 - zfftd3 - initialize the trigonometric weight and factor tables or compute the three-dimensional inverse Fast Fourier Transform of a three-dimensional double complex array.
zfftdm - zfftdm - initialize the trigonometric weight and factor tables or compute the one-dimensional inverse Fast Fourier Transform of a set of double complex data sequences stored in a two-dimensional array.
zfftf - zfftf - compute the Fourier coefficients of a periodic sequence. The FFT operations are unnormalized, so a call of ZFFTF followed by a call of ZFFTB will multiply the input sequence by N .
zffti - zffti - initialize the array WSAVE, which is used in both ZFFTF and ZFFTB.
zfftopt - zfftopt - compute the length of the closest fast FFT
zfftz - zfftz - initialize the trigonometric weight and factor tables or compute the Fast Fourier transform (forward or inverse) of a double complex sequence.
zfftz2 - zfftz2 - initialize the trigonometric weight and factor tables or compute the two-dimensional Fast Fourier Transform (forward or inverse) of a two-dimensional double complex array.
zfftz3 - zfftz3 - initialize the trigonometric weight and factor tables or compute the three-dimensional Fast Fourier Transform (forward or inverse) of a three-dimensional double complex array.
zfftzm - zfftzm - initialize the trigonometric weight and factor tables or compute the one-dimensional Fast Fourier Transform (forward or inverse) of a set of data sequences stored in a two-dimensional double complex array.

> zgbbrd - zgbbrd - reduce a complex general m-by-n band matrix A to real upper bidiagonal form B by a unitary transformation
zgbcon - zgbcon - estimate the reciprocal of the condition number of a complex general band matrix A, in either the 1-norm or the infinity-norm,
zgbequ - zgbequ - compute row and column scalings intended to equilibrate an M-by-N band matrix A and reduce its condition number
 beta* y , or y $:=\operatorname{alpha}{ }^{*} \operatorname{conjg}\left(\mathrm{~A}^{\prime}\right)^{*} \mathrm{x}+$ beta* $^{\mathrm{y}}$
zgbrfs - zgbrfs - improve the computed solution to a system of linear equations when the coefficient matrix is banded, and provides error bounds and backward error estimates for the solution
zgbsv - zgbsv - compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$, where A is a band matrix of order N with KL subdiagonals and KU superdiagonals, and X and B are N -by-NRHS matrices
zgbsvx - zgbsvx - use the LU factorization to compute the solution to a complex system of linear equations A * X $=\mathrm{B}, \mathrm{A} * * \mathrm{~T} * \mathrm{X}=\mathrm{B}$, or $\mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B}$,
zgbtf2 - zgbtf2 - compute an LU factorization of a complex m-by-n band matrix A using partial pivoting with row interchanges
zgbtrf - zgbtrf - compute an LU factorization of a complex m-by-n band matrix A using partial pivoting with row interchanges
zgbtrs - zgbtrs - solve a system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}, \mathrm{A}^{*} * \mathrm{~T} * \mathrm{X}=\mathrm{B}$, or $\mathrm{A}^{*}{ }^{*} \mathrm{H}^{*} \mathrm{X}=\mathrm{B}$ with a general band matrix A using the LU factorization computed by CGBTRF
zgebak - zgebak - form the right or left eigenvectors of a complex general matrix by backward transformation on the computed eigenvectors of the balanced matrix output by CGEBAL
zgebal - zgebal - balance a general complex matrix A
zgebrd - zgebrd - reduce a general complex M-by-N matrix A to upper or lower bidiagonal form B by a unitary transformation
zgecon - zgecon - estimate the reciprocal of the condition number of a general complex matrix A , in either the 1norm or the infinity-norm, using the LU factorization computed by CGETRF
zgeequ - zgeequ - compute row and column scalings intended to equilibrate an M-by-N matrix A and reduce its condition number
zgees - zgees - compute for an N -by-N complex nonsymmetric matrix A , the eigenvalues, the Schur form T, and, optionally, the matrix of Schur vectors Z
zgeesx - zgeesx - compute for an N-by-N complex nonsymmetric matrix A, the eigenvalues, the Schur form T, and, optionally, the matrix of Schur vectors Z
zgeev - zgeev - compute for an N -by-N complex nonsymmetric matrix A, the eigenvalues and, optionally, the left and/or right eigenvectors
zgeevx - zgeevx - compute for an N-by-N complex nonsymmetric matrix A, the eigenvalues and, optionally, the left and/or right eigenvectors
zgegs - zgegs - routine is deprecated and has been replaced by routine CGGES
zgegv - zgegv - routine is deprecated and has been replaced by routine CGGEV
zgehrd - zgehrd - reduce a complex general matrix A to upper Hessenberg form H by a unitary similarity transformation
zgelqf - zgelqf - compute an $L Q$ factorization of a complex M-by-N matrix A
zgels - zgels - solve overdetermined or underdetermined complex linear systems involving an M-by-N matrix A, or its conjugate-transpose, using a QR or LQ factorization of A
zgelsd - zgelsd - compute the minimum-norm solution to a real linear least squares problem
zgelss - zgelss - compute the minimum norm solution to a complex linear least squares problem
zgelsx - zgelsx - routine is deprecated and has been replaced by routine CGELSY
zgelsy - zgelsy - compute the minimum-norm solution to a complex linear least squares problem
zgemm - zgemm - perform one of the matrix-matrix operations C :=alpha*op(A)*op(B)+beta*C
zgemv - zgemv - perform one of the matrix-vector operations y $:=$ alpha* $A^{*} x+$ beta* $y$, or $y:=a l p h a * A^{\prime} * x+$ beta*y, or y $:=$ alpha*conjg( $\left.A^{\prime}\right)^{*} \mathrm{x}+$ beta*y $^{\text {y }}$
zgeqlf - zgeqlf - compute a QL factorization of a complex M-by-N matrix A
zgeqp3 - zgeqp3 - compute a QR factorization with column pivoting of a matrix A
zgeqpf - zgeqpf - routine is deprecated and has been replaced by routine CGEQP3
zgeqrf - zgeqrf - compute a QR factorization of a complex M-by-N matrix A
zgerc - zgerc - perform the rank 1 operation A := alpha*x*conjg( $\left.\mathrm{y}^{\prime}\right)+\mathrm{A}$
zgerfs - zgerfs - improve the computed solution to a system of linear equations and provides error bounds and backward error estimates for the solution
zgerqf - zgerqf - compute an RQ factorization of a complex M-by-N matrix A
zgeru - zgeru - perform the rank 1 operation $\mathrm{A}:=$ alpha* $\mathrm{x}^{*} \mathrm{y}^{\prime}+\mathrm{A}$
zgesdd - zgesdd - compute the singular value decomposition (SVD) of a complex M-by-N matrix A, optionally computing the left and/or right singular vectors, by using divide-and-conquer method
zgesv - zgesv - compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
zgesvd - zgesvd - compute the singular value decomposition (SVD) of a complex M-by-N matrix A, optionally computing the left and/or right singular vectors
zgesvx - zgesvx - use the LU factorization to compute the solution to a complex system of linear equations A*X $=\mathrm{B}$,
zgetf2 - zgetf2 - compute an LU factorization of a general m-by-n matrix A using partial pivoting with row interchanges
zgetrf - zgetrf - compute an LU factorization of a general M-by-N matrix A using partial pivoting with row interchanges
zgetri - zgetri - compute the inverse of a matrix using the LU factorization computed by CGETRF
zgetrs - zgetrs - solve a system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}, \mathrm{A} * * \mathrm{~T} * \mathrm{X}=\mathrm{B}$, or $\mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B}$ with a general $\mathrm{N}-$ by-N matrix A using the LU factorization computed by CGETRF
zggbak - zggbak - form the right or left eigenvectors of a complex generalized eigenvalue problem $\mathrm{A}^{*} \mathrm{x}=$ lambda*B*x, by backward transformation on the computed eigenvectors of the balanced pair of matrices output by CGGBAL
zggbal - zggbal - balance a pair of general complex matrices (A,B)
zgges - zgges - compute for a pair of $N$-by- $N$ complex nonsymmetric matrices ( $\mathrm{A}, \mathrm{B}$ ), the generalized eigenvalues, the generalized complex Schur form (S, T), and optionally left and/or right Schur vectors (VSL and VSR)

Zggesx - zggesx - compute for a pair of $N$-by-N complex nonsymmetric matrices (A,B), the generalized eigenvalues, the complex Schur form (S,T),
zggev - zggev - compute for a pair of $N$-by-N complex nonsymmetric matrices ( $\mathrm{A}, \mathrm{B}$ ), the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors
zggevx - zggevx - compute for a pair of $N$-by-N complex nonsymmetric matrices ( $\mathrm{A}, \mathrm{B}$ ) the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors
zggglm - zggglm - solve a general Gauss-Markov linear model (GLM) problem
zgghrd - zgghrd - reduce a pair of complex matrices (A,B) to generalized upper Hessenberg form using unitary transformations, where A is a general matrix and B is upper triangular
zgglse - zgglse - solve the linear equality-constrained least squares (LSE) problem
zggqrf - zggqrf - compute a generalized QR factorization of an N -by-M matrix A and an N -by- P matrix B .
zggrqf - zggrqf - compute a generalized RQ factorization of an M-by-N matrix A and a P-by-N matrix B
zggsvd - zggsvd - compute the generalized singular value decomposition (GSVD) of an M-by-N complex matrix A and P-by-N complex matrix B
zggsvp - zggsvp - compute unitary matrices $\mathrm{U}, \mathrm{V}$ and Q such that $\mathrm{N}-\mathrm{K}-\mathrm{L} K \mathrm{~L} \mathrm{U}^{\prime} * \mathrm{~A} * \mathrm{Q}=\mathrm{K}(0 \mathrm{~A} 12 \mathrm{~A} 13)$ if $\mathrm{M}-\mathrm{K}-$ $\mathrm{L}>=0$
zgssco - zgssco - General sparse solver condition number estimate.
zgssda - zgssda - Deallocate working storage for the general sparse solver.
zgssfa - zgssfa - General sparse solver numeric factorization.
zgssfs - zgssfs - General sparse solver one call interface.

Zgssin - zgssin - Initialize the general sparse solver.
zgssor - zgssor - General sparse solver ordering and symbolic factorization.
zgssps - zgssps - Print general sparse solver statics.
zgssrp - zgssrp - Return permutation used by the general sparse solver.
zgsssl-zgsssl-Solve routine for the general sparse solver.
zgssuo - zgssuo - User supplied permutation for ordering used in the general sparse solver.
zgtcon - zgtcon - estimate the reciprocal of the condition number of a complex tridiagonal matrix A using the LU factorization as computed by CGTTRF
zgthr - zgthr - Gathers specified elements from y into x .
zgthrz - zgthrz - Gather and zero.
zgtrfs - zgtrfs - improve the computed solution to a system of linear equations when the coefficient matrix is tridiagonal, and provides error bounds and backward error estimates for the solution
zgtsv - zgtsv - solve the equation $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
zgtsvx - zgtsvx - use the LU factorization to compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}$ $=\mathrm{B}, \mathrm{A}^{*} * \mathrm{~T} * \mathrm{X}=\mathrm{B}$, or $\mathrm{A}^{*}{ }^{*} \mathrm{H} * \mathrm{X}=\mathrm{B}$,
zgttrf - zgttrf - compute an LU factorization of a complex tridiagonal matrix A using elimination with partial pivoting and row interchanges
$\underline{\text { zgttrs }}-\operatorname{zgttrs}-$ solve one of the systems of equations $A * X=B, A * * T * X=B$, or $A * * H * X=B$,
zhbev - zhbev - compute all the eigenvalues and, optionally, eigenvectors of a complex Hermitian band matrix A
zhbevd - zhbevd - compute all the eigenvalues and, optionally, eigenvectors of a complex Hermitian band matrix A
zhbevx - zhbevx - compute selected eigenvalues and, optionally, eigenvectors of a complex Hermitian band matrix A
zhbgst - zhbgst - reduce a complex Hermitian-definite banded generalized eigenproblem $A * x=l a m b d a * B * x$ to standard form $\mathrm{C} * \mathrm{y}=$ lambda*y,
zhbgv - zhbgv - compute all the eigenvalues, and optionally, the eigenvectors of a complex generalized Hermitiandefinite banded eigenproblem, of the form $A * x=(l a m b d a) * B * x$
zhbgvd - zhbgvd - compute all the eigenvalues, and optionally, the eigenvectors of a complex generalized Hermitian-definite banded eigenproblem, of the form $\mathrm{A}^{*} \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{~B}^{*} \mathrm{x}$
zhbgvx - zhbgvx - compute all the eigenvalues, and optionally, the eigenvectors of a complex generalized Hermitian-definite banded eigenproblem, of the form $\mathrm{A}^{*} \mathrm{X}=($ lambda $){ }^{*} \mathrm{~B}^{*} \mathrm{x}$
zhbmv - zhbmv - perform the matrix-vector operation $\mathrm{y}:=\operatorname{alpha*} \mathrm{A}^{*} \mathrm{x}+$ beta*y
zhbtrd - zhbtrd - reduce a complex Hermitian band matrix A to real symmetric tridiagonal form T by a unitary similarity transformation
zhecon - zhecon - estimate the reciprocal of the condition number of a complex Hermitian matrix A using the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}^{* *} \mathrm{H}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{D}^{*} \mathrm{~L}^{* *} \mathrm{H}$ computed by CHETRF
zheev - zheev - compute all eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix A
zheevd - zheevd - compute all eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix A
zheevr - zheevr - compute selected eigenvalues and, optionally, eigenvectors of a complex Hermitian tridiagonal matrix T
zheevx - zheevx - compute selected eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix A
zhegs2 - zhegs2 - reduce a complex Hermitian-definite generalized eigenproblem to standard form
zhegst - zhegst - reduce a complex Hermitian-definite generalized eigenproblem to standard form
zhegv - zhegv - compute all the eigenvalues, and optionally, the eigenvectors of a complex generalized Hermitiandefinite eigenproblem, of the form $\mathrm{A}^{*} \mathrm{x}=(\operatorname{lambda}) * \mathrm{~B} * \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda})^{*} \mathrm{x}$, or $\mathrm{B} * \mathrm{~A}^{*} \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{x}$
zhegvd - zhegvd - compute all the eigenvalues, and optionally, the eigenvectors of a complex generalized Hermitian-definite eigenproblem, of the form $\mathrm{A}^{*} \mathrm{x}=(\operatorname{lambda}) * \mathrm{~B} * \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda})^{*} \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{x}$
zhegvx - zhegvx - compute selected eigenvalues, and optionally, eigenvectors of a complex generalized Hermitian-definite eigenproblem, of the form $\mathrm{A}^{*} \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{~B}^{*} \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda})^{*} \mathrm{x}$, or $\mathrm{B}^{*} \mathrm{~A}^{*} \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{x}$
zhemm - zhemm - perform one of the matrix-matrix operations $\mathrm{C}:=$ alpha* $\mathrm{A} * \mathrm{~B}+$ beta* C or $\mathrm{C}:=$ alpha* $\mathrm{B}^{*} \mathrm{~A}+$ beta*C
zhemv - zhemv - perform the matrix-vector operation $y:=a l p h a * A * x+$ beta*y

zher2 - zher2 - perform the hermitian rank 2 operation $A:=\operatorname{alpha}{ }^{*} x^{*} \operatorname{conjg}\left(y^{\prime}\right)+\operatorname{conjg}(\operatorname{alpha})^{*} y^{*} \operatorname{conjg}\left(x^{\prime}\right)+A$
zher2k - zher2k - perform one of the Hermitian rank $2 k$ operations $C:=a l p h *^{*} A^{*} \operatorname{conjg}\left(B^{\prime}\right)+\operatorname{conjg}($ alpha $)$ ${ }^{*} \mathrm{~B}^{*} \operatorname{conjg}\left(\mathrm{~A}^{\prime}\right)+$ beta* $^{\mathrm{C}}$ or $\mathrm{C}:=\operatorname{alpha*} \operatorname{conjg}\left(\mathrm{A}^{\prime}\right) * \mathrm{~B}+\operatorname{conjg}(\operatorname{alpha})^{*} \operatorname{conjg}\left(\mathrm{~B}^{\prime}\right)^{*} \mathrm{~A}+\operatorname{beta}^{*} \mathrm{C}$
zherfs - zherfs - improve the computed solution to a system of linear equations when the coefficient matrix is Hermitian indefinite, and provides error bounds and backward error estimates for the solution
zherk - zherk - perform one of the Hermitian rank k operations $C:=\operatorname{alpha} A^{*} \operatorname{conjg}\left(\mathrm{~A}^{\prime}\right)+$ beta* C or $\mathrm{C}:=$ alpha*conjg ( $\mathrm{A}^{\prime}$ )*A + beta*C

zhesvx - zhesvx - use the diagonal pivoting factorization to compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
zhetf2 - zhetf2 - compute the factorization of a complex Hermitian matrix A using the Bunch-Kaufman diagonal pivoting method
zhetrd - zhetrd - reduce a complex Hermitian matrix A to real symmetric tridiagonal form T by a unitary similarity transformation
zhetrf - zhetrf - compute the factorization of a complex Hermitian matrix A using the Bunch-Kaufman diagonal pivoting method
zhetri - zhetri - compute the inverse of a complex Hermitian indefinite matrix A using the factorization $\mathrm{A}=$ $\mathrm{U} * \mathrm{D} * \mathrm{U}^{*} * \mathrm{H}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L}^{*} * \mathrm{H}$ computed by CHETRF
zhetrs - zhetrs - solve a system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$ with a complex Hermitian matrix A using the factorization $\mathrm{A}=\mathrm{U}^{*} \mathrm{D}^{*} \mathrm{U}^{* *} \mathrm{H}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D}^{*} \mathrm{~L}^{*} * \mathrm{H}$ computed by CHETRF
zhgeqz - zhgeqz - implement a single-shift version of the QZ method for finding the generalized eigenvalues w(i) $=A L P H A(i) / B E T A(i)$ of the equation $\operatorname{det}(A-w(i) B)=0$ If $\mathrm{JOB}=$ 'S', then the pair (A,B) is simultaneously reduced to Schur form (i.e., A and B are both upper triangular) by applying one unitary tranformation (usually called Q) on the left and another (usually called Z) on the right
zhpcon - zhpcon - estimate the reciprocal of the condition number of a complex Hermitian packed matrix A using the factorization $\mathrm{A}=\mathrm{U}^{*} \mathrm{D}^{*} \mathrm{U}^{* *} \mathrm{H}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D}^{*} \mathrm{~L}^{* *} \mathrm{H}$ computed by CHPTRF
zhpev - zhpev - compute all the eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix in packed storage
zhpevd - zhpevd - compute all the eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix A in packed storage
zhpevx - zhpevx - compute selected eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix A in packed storage
zhpgst - zhpgst - reduce a complex Hermitian-definite generalized eigenproblem to standard form, using packed storage
zhpgv - zhpgv - compute all the eigenvalues and, optionally, the eigenvectors of a complex generalized Hermitiandefinite eigenproblem, of the form $\mathrm{A}^{*} \mathrm{x}=(\operatorname{lambda}) * \mathrm{~B} * \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda}) * \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lambda})^{*} \mathrm{x}$
zhpgvd - zhpgvd - compute all the eigenvalues and, optionally, the eigenvectors of a complex generalized Hermitian-definite eigenproblem, of the form $\mathrm{A}^{*} \mathrm{x}=(\operatorname{lambda}) * \mathrm{~B} * \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda})^{*} \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lambda}) * \mathrm{x}$
zhpgvx - zhpgvx - compute selected eigenvalues and, optionally, eigenvectors of a complex generalized Hermitian-definite eigenproblem, of the form $\mathrm{A}^{*} \mathrm{x}=(\operatorname{lambda}) * \mathrm{~B} * \mathrm{x}, \mathrm{A} * \mathrm{Bx}=(\operatorname{lambda})^{*} \mathrm{x}$, or $\mathrm{B} * \mathrm{~A}^{*} \mathrm{x}=(\operatorname{lambda}) * \mathrm{x}$
zhpmv - zhpmv - perform the matrix-vector operation $\mathrm{y}:=\operatorname{alpha*} \mathrm{A}^{*} \mathrm{x}+$ beta*y $^{\mathrm{y}}$
zhpr - zhpr - perform the hermitian rank 1 operation $\mathrm{A}:=\operatorname{alpha}{ }^{*} \mathrm{x}^{*} \operatorname{conjg}\left(\mathrm{x}^{\prime}\right)+\mathrm{A}$
zhpr2 - zhpr2 - perform the Hermitian rank 2 operation A :=alpha* $x^{*} \operatorname{conjg}\left(y^{\prime}\right)+\operatorname{conjg}(\text { alpha })^{*} y^{*} \operatorname{conjg}\left(x^{\prime}\right)+$ A
zhprfs - zhprfs - improve the computed solution to a system of linear equations when the coefficient matrix is Hermitian indefinite and packed, and provides error bounds and backward error estimates for the solution

zhpsvx - zhpsvx - use the diagonal pivoting factorization $A=U * D * U * * H$ or $A=L * D * L^{*} H$ to compute the solution to a complex system of linear equations $A * X=B$, where $A$ is an $N$-by-N Hermitian matrix stored in packed format and X and B are N-by-NRHS matrices
zhptrd - zhptrd - reduce a complex Hermitian matrix A stored in packed form to real symmetric tridiagonal form T by a unitary similarity transformation
zhptrf - zhptrf - compute the factorization of a complex Hermitian packed matrix A using the Bunch-Kaufman diagonal pivoting method
zhptri - zhptri - compute the inverse of a complex Hermitian indefinite matrix A in packed storage using the factorization $\mathrm{A}=\mathrm{U}^{*} \mathrm{D}^{*} \mathrm{U}^{* *} \mathrm{H}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D}^{*} \mathrm{~L}^{* *} \mathrm{H}$ computed by CHPTRF
zhptrs - zhptrs - solve a system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$ with a complex Hermitian matrix A stored in packed format using the factorization $A=U^{*} D^{*} U^{* *} H$ or $A=L^{*} D^{*} L^{* *} H$ computed by CHPTRF
zhsein - zhsein - use inverse iteration to find specified right and/or left eigenvectors of a complex upper Hessenberg matrix H
zhseqr - zhseqr - compute the eigenvalues of a complex upper Hessenberg matrix H , and, optionally, the matrices T and Z from the Schur decomposition $\mathrm{H}=\mathrm{Z} \mathrm{T} \mathrm{Z}{ }^{* *} \mathrm{H}$, where T is an upper triangular matrix (the Schur form), and Z is the unitary matrix of Schur vectors
zjadmm - zjadmm - Jagged diagonal matrix-matrix multiply (modified Ellpack)
zjadrp - zjadrp - right permutation of a jagged diagonal matrix
zjadsm - zjadsm - Jagged-diagonal format triangular solve
zlarz - zlarz - applie a complex elementary reflector H to a complex M-by-N matrix C, from either the left or the right
zlarzb - zlarzb - applie a complex block reflector H or its transpose $\mathrm{H}^{* *} \mathrm{H}$ to a complex distributed M-by-N C from the left or the right
zlarzt - zlarzt - form the triangular factor T of a complex block reflector H of order $>\mathrm{n}$, which is defined as a product of $k$ elementary reflectors
zlatzm - zlatzm - routine is deprecated and has been replaced by routine CUNMRZ
zpbcon - zpbcon - estimate the reciprocal of the condition number (in the 1-norm) of a complex Hermitian positive definite band matrix using the Cholesky factorization $A=U^{*} * H^{*} U$ or $A=L^{*} L^{* *} H$ computed by CPBTRF
zpbequ - zpbequ - compute row and column scalings intended to equilibrate a Hermitian positive definite band matrix A and reduce its condition number (with respect to the two-norm)
zpbrfs - zpbrfs - improve the computed solution to a system of linear equations when the coefficient matrix is Hermitian positive definite and banded, and provides error bounds and backward error estimates for the solution
zpbstf - zpbstf - compute a split Cholesky factorization of a complex Hermitian positive definite band matrix A
zpbsv - zpbsv - compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
zpbsvx - zpbsvx - use the Cholesky factorization $\mathrm{A}=\mathrm{U}^{* *} \mathrm{H}^{*} \mathrm{U}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{~L}^{* *} \mathrm{H}$ to compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
zpbtf2 - zpbtf2 - compute the Cholesky factorization of a complex Hermitian positive definite band matrix A
zpbtrf - zpbtrf - compute the Cholesky factorization of a complex Hermitian positive definite band matrix A
zpbtrs - zpbtrs - solve a system of linear equations $A * X=B$ with a Hermitian positive definite band matrix $A$ using the Cholesky factorization $\mathrm{A}=\mathrm{U}^{* *} \mathrm{H}^{*} \mathrm{U}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{~L}^{* *} \mathrm{H}$ computed by CPBTRF
zpocon - zpocon - estimate the reciprocal of the condition number (in the 1-norm) of a complex Hermitian positive definite matrix using the Cholesky factorization $A=U^{* *} H^{*} U$ or $A=L^{*} L^{* *} \mathrm{H}$ computed by CPOTRF
zpoequ - zpoequ - compute row and column scalings intended to equilibrate a Hermitian positive definite matrix A and reduce its condition number (with respect to the two-norm)
zporfs - zporfs - improve the computed solution to a system of linear equations when the coefficient matrix is Hermitian positive definite,
zposv - zposv - compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
zposvx - zposvx - use the Cholesky factorization $A=U^{* *} H^{*} U$ or $A=L^{*} L^{* *} H$ to compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
zpotf2 - zpotf2 - compute the Cholesky factorization of a complex Hermitian positive definite matrix A
zpotrf - zpotrf - compute the Cholesky factorization of a complex Hermitian positive definite matrix A
zpotri - zpotri - compute the inverse of a complex Hermitian positive definite matrix A using the Cholesky factorization $\mathrm{A}=\mathrm{U}^{* *} \mathrm{H}^{*} \mathrm{U}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{~L}^{* *} \mathrm{H}$ computed by CPOTRF
zpotrs - zpotrs - solve a system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$ with a Hermitian positive definite matrix A using the Cholesky factorization $A=U^{* *} \mathrm{H}^{*} \mathrm{U}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{~L}^{* *} \mathrm{H}$ computed by CPOTRF
zppcon - zppcon - estimate the reciprocal of the condition number (in the 1-norm) of a complex Hermitian positive definite packed matrix using the Cholesky factorization $A=U^{* *} H^{*} U$ or $A=L^{*} L^{* *} H$ computed by CPPTRF
zppequ - zppequ - compute row and column scalings intended to equilibrate a Hermitian positive definite matrix A in packed storage and reduce its condition number (with respect to the two-norm)
zpprfs - zpprfs - improve the computed solution to a system of linear equations when the coefficient matrix is Hermitian positive definite and packed, and provides error bounds and backward error estimates for the solution
zppsv - zppsv - compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
zppsvx - zppsvx - use the Cholesky factorization $A=U^{* *} H^{*} U$ or $A=L^{*} L^{* *} \mathrm{H}$ to compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
zpptrf - zpptrf - compute the Cholesky factorization of a complex Hermitian positive definite matrix A stored in packed format
zpptri - zpptri - compute the inverse of a complex Hermitian positive definite matrix A using the Cholesky factorization $\mathrm{A}=\mathrm{U}^{* *} \mathrm{H}^{*} \mathrm{U}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{~L}^{* *} \mathrm{H}$ computed by CPPTRF
zpptrs - zpptrs - solve a system of linear equations $A * X=B$ with a Hermitian positive definite matrix A in packed storage using the Cholesky factorization $A=U^{*} * H^{*} U$ or $A=L^{*} L^{* *} H$ computed by CPPTRF
zptcon - zptcon - compute the reciprocal of the condition number (in the 1-norm) of a complex Hermitian positive definite tridiagonal matrix using the factorization $A=L * D * L * * H$ or $A=U^{* *} H^{*} D^{*} U$ computed by CPTTRF
zpteqr - zpteqr - compute all eigenvalues and, optionally, eigenvectors of a symmetric positive definite tridiagonal matrix by first factoring the matrix using SPTTRF and then calling CBDSQR to compute the singular values of the bidiagonal factor
zptrfs - zptrfs - improve the computed solution to a system of linear equations when the coefficient matrix is Hermitian positive definite and tridiagonal, and provides error bounds and backward error estimates for the solution
zptsv - zptsv - compute the solution to a complex system of linear equations $A * X=B$, where $A$ is an $N-b y-N$ Hermitian positive definite tridiagonal matrix, and X and B are N -by-NRHS matrices.
zptsvx - zptsvx - use the factorization $A=L^{*} D^{*} L^{* *} H$ to compute the solution to a complex system of linear equations $A * X=B$, where $A$ is an $N$-by-N Hermitian positive definite tridiagonal matrix and $X$ and $B$ are $N$-byNRHS matrices
zpttrf - zpttrf - compute the $L^{*} D^{*} L^{\prime}$ factorization of a complex Hermitian positive definite tridiagonal matrix A
zpttrs - zpttrs - solve a tridiagonal system of the form $A * X=B$ using the factorization $A=U^{*} * D * U$ or $A=$ L*D*L' computed by CPTTRF
zptts2 - zptts2 - solve a tridiagonal system of the form $A * X=B$ using the factorization $A=U^{\prime} * D^{*} U$ or $A=$ L*D*L' computed by CPTTRF
zrot - zrot - apply a plane rotation, where the $\cos (\mathrm{C})$ is real and the $\sin (\mathrm{S})$ is complex, and the vectors X and Y are complex
zrotg - zrotg - Construct a Given's plane rotation
zscal - zscal - Compute y $:=$ alpha * y
zsctr - zsctr - Scatters elements from $x$ into $y$.
zskymm - zskymm - Skyline format matrix-matrix multiply
zskysm - zskysm - Skyline format triangular solve
zspcon - zspcon - estimate the reciprocal of the condition number (in the 1-norm) of a complex symmetric packed matrix A using the factorization $\mathrm{A}=\mathrm{U}^{*} \mathrm{D}^{*} \mathrm{U}^{* *} \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D}^{*} \mathrm{~L}^{* *} \mathrm{~T}$ computed by CSPTRF
zsprfs - zsprfs - improve the computed solution to a system of linear equations when the coefficient matrix is symmetric indefinite and packed, and provides error bounds and backward error estimates for the solution
zspsv - zspsv - compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
zspsvx - zspsvx - use the diagonal pivoting factorization $A=U^{*} D^{*} U^{*} * T$ or $A=L^{*} D^{*} L^{* *} T$ to compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$, where A is an N -by-N symmetric matrix stored in packed format and X and B are N-by-NRHS matrices
zsptrf - zsptrf - compute the factorization of a complex symmetric matrix A stored in packed format using the Bunch-Kaufman diagonal pivoting method
zsptri - zsptri - compute the inverse of a complex symmetric indefinite matrix A in packed storage using the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D}^{*} \mathrm{U}^{*} * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L}^{*} \mathrm{D}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ computed by CSPTRF
zsptrs - zsptrs - solve a system of linear equations $A * X=B$ with a complex symmetric matrix A stored in packed format using the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D}^{*} \mathrm{U}^{*} * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D}^{*} \mathrm{~L} * * \mathrm{~T}$ computed by CSPTRF
zstedc - zstedc - compute all eigenvalues and, optionally, eigenvectors of a symmetric tridiagonal matrix using the divide and conquer method
zstegr - zstegr - Compute T-sigma_i $=L_{-} \mathrm{i} \mathrm{D}_{-} \mathrm{i} \mathrm{L}_{-} \mathrm{i}^{\wedge} T$, such that $\mathrm{L}_{-} \mathrm{i} \mathrm{D}_{-} \mathrm{i} \mathrm{L}_{-} \mathrm{i}^{\wedge} \mathrm{T}$ is a relatively robust representation
zstein - zstein - compute the eigenvectors of a real symmetric tridiagonal matrix T corresponding to specified eigenvalues, using inverse iteration
zsteqr - zsteqr - compute all eigenvalues and, optionally, eigenvectors of a symmetric tridiagonal matrix using the implicit QL or QR method
$\underline{\text { zstsv }}$ - zstsv - compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$ where A is a Hermitian tridiagonal matrix
zsttrf - zsttrf - compute the factorization of a complex Hermitian tridiagonal matrix A
$\underline{\text { zsttrs }}$ - zsttrs - computes the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$
zswap - zswap - Exchange vectors x and y .
zsycon - zsycon - estimate the reciprocal of the condition number (in the 1-norm) of a complex symmetric matrix A using the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D}^{*} \mathrm{U}^{*} * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ computed by CSYTRF
$\underline{\text { zsymm }}-$ zsymm - perform one of the matrix-matrix operations $\mathrm{C}:=\operatorname{alpha*} \mathrm{A} * \mathrm{~B}+$ beta* $^{\mathrm{C}}$ or $\mathrm{C}:=$ alpha* $\mathrm{B} * \mathrm{~A}+$ beta*C
zsyr2k-zsyr2k - perform one of the symmetric rank $2 k$ operations $C:=$ alpha* $A^{*} B^{\prime}+$ alpha* $B^{*} A^{\prime}+$ beta* $C$ or $C$ : $=$ alpha* ${ }^{\prime} * B+$ alpha $* B^{\prime} * \mathrm{~A}+$ beta $^{*} \mathrm{C}$
zsyrfs - zsyrfs - improve the computed solution to a system of linear equations when the coefficient matrix is symmetric indefinite, and provides error bounds and backward error estimates for the solution
zsyrk - zsyrk - perform one of the symmetric rank k operations C :=alpha*A*A' + beta*C or C :=alpha*A'*A + beta*C

zsysvx - zsysvx - use the diagonal pivoting factorization to compute the solution to a complex system of linear equations $\mathrm{A} * \mathrm{X}=\mathrm{B}$,
zsytf2 - zsytf2 - compute the factorization of a complex symmetric matrix A using the Bunch-Kaufman diagonal pivoting method
zsytrf - zsytrf - compute the factorization of a complex symmetric matrix A using the Bunch-Kaufman diagonal pivoting method
zsytri - zsytri - compute the inverse of a complex symmetric indefinite matrix A using the factorization $\mathrm{A}=$ $\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D}^{*} \mathrm{~L}^{*} * \mathrm{~T}$ computed by CSYTRF
zsytrs - zsytrs - solve a system of linear equations $A * X=B$ with a complex symmetric matrix $A$ using the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D}^{*} \mathrm{U}^{*} * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L}^{*} * \mathrm{~T}$ computed by CSYTRF
ztbcon - ztbcon - estimate the reciprocal of the condition number of a triangular band matrix A , in either the 1norm or the infinity-norm

ztbrfs - ztbrfs - provide error bounds and backward error estimates for the solution to a system of linear equations with a triangular band coefficient matrix
$\underline{\text { ztbsv }}$ - ztbsv - solve one of the systems of equations $A^{*} x=b$, or $A^{\prime} * x=b$, or $\operatorname{conjg}\left(A^{\prime}\right)^{*} x=b$
ztbtrs - ztbtrs - solve a triangular system of the form $\mathrm{A}^{*} \mathrm{X}=\mathrm{B}, \mathrm{A}^{*} * \mathrm{~T} * \mathrm{X}=\mathrm{B}$, or $\mathrm{A}^{*} * \mathrm{H} * \mathrm{X}=\mathrm{B}$,
ztgevc - ztgevc - compute some or all of the right and/or left generalized eigenvectors of a pair of complex upper triangular matrices (A,B)

Ztgexc - ztgexc - reorder the generalized Schur decomposition of a complex matrix pair (A,B), using an unitary equivalence transformation $(A, B):=Q *(A, B) * Z$, so that the diagonal block of $(A, B)$ with row index IFST is moved to row ILST
ztgsen - ztgsen - reorder the generalized Schur decomposition of a complex matrix pair (A, B) (in terms of an unitary equivalence trans- formation $\left.\mathrm{Q}^{\prime *}(\mathrm{~A}, \mathrm{~B}) * \mathrm{Z}\right)$, so that a selected cluster of eigenvalues appears in the leading diagonal blocks of the pair $(\mathrm{A}, \mathrm{B})$
ztgsja - ztgsja - compute the generalized singular value decomposition (GSVD) of two complex upper triangular (or trapezoidal) matrices A and B
ztgsna - ztgsna - estimate reciprocal condition numbers for specified eigenvalues and/or eigenvectors of a matrix pair (A, B)

Ztgsyl - ztgsyl - solve the generalized Sylvester equation
ztpcon - ztpcon - estimate the reciprocal of the condition number of a packed triangular matrix A , in either the 1norm or the infinity-norm
ztpmv - ztpmv - perform one of the matrix-vector operations $\mathrm{x}:=\mathrm{A}^{*} \mathrm{x}$, or $\mathrm{x}:=\mathrm{A}^{\prime}{ }^{*} \mathrm{x}$, or $\mathrm{x}:=\operatorname{conjg}\left(\mathrm{A}^{\prime}\right)^{*} \mathrm{x}$
ztprfs - ztprfs - provide error bounds and backward error estimates for the solution to a system of linear equations with a triangular packed coefficient matrix
$\underline{\text { ztpsv }}-\mathrm{ztpsv}$ - solve one of the systems of equations $\mathrm{A}^{*} \mathrm{x}=\mathrm{b}$, or $\mathrm{A}^{\prime} * \mathrm{x}=\mathrm{b}$, or $\operatorname{conjg}\left(\mathrm{A}^{\prime}\right)^{*} \mathrm{x}=\mathrm{b}$
ztptri - ztptri - compute the inverse of a complex upper or lower triangular matrix A stored in packed format
ztptrs - ztptrs - solve a triangular system of the form $\mathrm{A} * \mathrm{X}=\mathrm{B}, \mathrm{A}^{*} * \mathrm{~T} * \mathrm{X}=\mathrm{B}$, or $\mathrm{A}^{*} * \mathrm{H} * \mathrm{X}=\mathrm{B}$,
ztrans - ztrans - transpose and scale source matrix
$\underline{z t r c o n}$ - ztrcon - estimate the reciprocal of the condition number of a triangular matrix A , in either the 1-norm or the infinity-norm

ztrexc - ztrexc - reorder the Schur factorization of a complex matrix $\mathrm{A}=\mathrm{Q}^{*} \mathrm{~T}^{*} \mathrm{Q}^{* *} \mathrm{H}$, so that the diagonal element of T with row index IFST is moved to row ILST
ztrmm - ztrmm - perform one of the matrix-matrix operations B := alpha*op(A)*B, or B := alpha*B*op(A) where alpha is a scalar, $B$ is an $m$ by $n$ matrix, $A$ is a unit, or non-unit, upper or lower triangular matrix and op $(A)$ is one of op $(A)=A$ or op $(A)=A^{\prime}$ or op $(A)=\operatorname{conjg}\left(A^{\prime}\right)$
$\underline{\text { ztrmv }}-\mathrm{ztrmv}$ - perform one of the matrix-vector operations $\mathrm{x}:=\mathrm{A}^{*} \mathrm{x}$, or $\mathrm{x}:=\mathrm{A}^{\prime}{ }^{*} \mathrm{x}$, or $\mathrm{x}:=\operatorname{conjg}\left(\mathrm{A}^{\prime}\right)^{*} \mathrm{x}$
ztrrfs - ztrrfs - provide error bounds and backward error estimates for the solution to a system of linear equations with a triangular coefficient matrix
$\underline{\text { ztrsen }}$ - ztrsen - reorder the Schur factorization of a complex matrix $A=Q^{*} T^{*} Q^{* *} H$, so that a selected cluster of eigenvalues appears in the leading positions on the diagonal of the upper triangular matrix T , and the leading
columns of Q form an orthonormal basis of the corresponding right invariant subspace

ztrsna - ztrsna - estimate reciprocal condition numbers for specified eigenvalues and/or right eigenvectors of a complex upper triangular matrix T (or of any matrix $\mathrm{Q}^{*} \mathrm{~T}^{*} \mathrm{Q}^{* *} \mathrm{H}$ with Q unitary)
$\underline{\text { ztrsv }}$ - ztrsv - solve one of the systems of equations $A^{*} x=b$, or $A^{\prime} * x=b$, or conjg $\left(A^{\prime}\right)^{*} x=b$
ztrsyl - ztrsyl - solve the complex Sylvester matrix equation
ztrti2 - ztrti2 - compute the inverse of a complex upper or lower triangular matrix
ztrtri - ztrtri - compute the inverse of a complex upper or lower triangular matrix A
$\underline{\text { ztrtrs }-z t r t r s ~-~ s o l v e ~ a ~ t r i a n g u l a r ~ s y s t e m ~ o f ~ t h e ~ f o r m ~} \mathrm{~A} * \mathrm{X}=\mathrm{B}, \mathrm{A}^{*} * \mathrm{~T} * \mathrm{X}=\mathrm{B}$, or $\mathrm{A}^{*} * \mathrm{H} * \mathrm{X}=\mathrm{B}$,
ztzrqf - ztzrqf - routine is deprecated and has been replaced by routine CTZRZF
ztzrzf - ztzrzf - reduce the M-by-N ( $\mathrm{M}<=\mathrm{N}$ ) complex upper trapezoidal matrix A to upper triangular form by means of unitary transformations
zung21-zung21-generate an m by $n$ complex matrix Q with orthonormal columns,
zung2r - zung2r - generate an $m$ by $n$ complex matrix $Q$ with orthonormal columns,
zungbr - zungbr - generate one of the complex unitary matrices Q or $\mathrm{P}^{* *} \mathrm{H}$ determined by CGEBRD when reducing a complex matrix A to bidiagonal form
zunghr - zunghr - generate a complex unitary matrix Q which is defined as the product of IHI-ILO elementary reflectors of order N , as returned by CGEHRD
zung12 - zung12 - generate an m-by-n complex matrix Q with orthonormal rows,
zunglq - zunglq - generate an M-by-N complex matrix Q with orthonormal rows,
zungql - zungql - generate an M-by-N complex matrix Q with orthonormal columns,
zungqr - zungqr - generate an M-by-N complex matrix Q with orthonormal columns,
zungr2 - zungr2 - generate an $m$ by $n$ complex matrix Q with orthonormal rows,
zungrq - zungrq - generate an M-by-N complex matrix Q with orthonormal rows,
zungtr - zungtr - generate a complex unitary matrix $Q$ which is defined as the product of $n-1$ elementary reflectors of order N , as returned by CHETRD
zunmbr - zunmbr - VECT = 'Q', CUNMBR overwrites the general complex M-by-N matrix C with SIDE = 'L' SIDE $=$ 'R' TRANS $=$ 'N'

zunml2 - zunml2 - overwrite the general complex m-by-n matrix C with Q * C if $\operatorname{SIDE}=$ ' L ' and TRANS = ' N ', or $\mathrm{Q}^{\prime *} \mathrm{C}$ if SIDE $=$ 'L' and TRANS $=$ ' C ', or $\mathrm{C} * \mathrm{Q}$ if SIDE $=$ ' R ' and TRANS $=$ ' N ', or $\mathrm{C} * \mathrm{Q}$ ' if SIDE $=$ ' R ' and TRANS = 'C',

zunmql - zunmql - overwrite the general complex M-by-N matrix C with SIDE $=$ 'L' SIDE $=$ ' R ' TRANS $=$ ' $\mathrm{N}^{\prime}$

zunmr2 - zunmr2- overwrite the general complex m-by-n matrix C with $\mathrm{Q} * \mathrm{C}$ if $\operatorname{SIDE}=$ ' L ' and TRANS $=$ ' N ', or $\mathrm{Q}^{*} \mathrm{C}$ if SIDE = 'L' and TRANS $=$ ' C ', or $\mathrm{C} * \mathrm{Q}$ if SIDE $=$ 'R' and TRANS $=$ ' N ', or $\mathrm{C} * \mathrm{Q}$ ' if SIDE $=$ 'R' and TRANS = 'C',



zupgtr - zupgtr - generate a complex unitary matrix Q which is defined as the product of $\mathrm{n}-1$ elementary reflectors $\mathrm{H}(\mathrm{i})$ of order n , as returned by CHPTRD using packed storage
$\underline{\text { zupmtr }}$ - zupmtr - overwrite the general complex M-by-N matrix C with SIDE = 'L' SIDE = 'R'TRANS = ' N '
zvbrmm - zvbrmm - variable block sparse row format matrix-matrix multiply
zvbrsm - zvbrsm - variable block sparse row format triangular solve

Zvmul - zvmul - compute the scaled product of complex vectors

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

available_threads - retums inform ation about cument thread usage

## SYNOPSIS

SUBROUTINEAVA ILABLE_THREADS $\mathbb{N} T O T A L, N U S \mathbb{N} G)$
$\mathbb{N}$ TEGERNTOTAL,NUSNG

SU BROUTINEAVA ILABLE_THREADS_64 $\mathbb{N} T O T A L, N U S \mathbb{N} G)$
$\mathbb{N} T E G E R * 8 N T O T A L, N U S \mathbb{N} G$

F95 INTERFACE
SU BROUTINEAVAIABLE_THREADS $\mathbb{N} T O T A L, N U S \mathbb{N} G)$
$\mathbb{N} T E G E R:: N T O T A L, N U S \mathbb{N} G$

SU BROUTINEAVA $\left.\mathbb{L} A B L E \_T H R E A D S \_64 \mathbb{N} T O T A L, N U S \mathbb{N} G\right)$
$\mathbb{N}$ TEGER (8) ::NTOTAL,NUSNG

## C INTERFACE

\#include < sunperfh>
void available_threads(int *ntotal, int *nusing);
void available_threads_64 (long *ntotal, long *nusing);

## PURPOSE

available_threads threads retums N TO TA L , w hich is the total num ber of C PU s available to the job (generally the num ber of CPU s presently on-line in the partition), and N U S $\mathbb{N}$ G , which is the sum of the currentU se num bers for all threads specified in USE_THREADS. IfNTOTAL <NUSING then the system is potentially overcom m itted.

## ARGUMENTS

NTOTAL (output)
Totalnum ber of CPU s available.
NUSING (output)
Sum of currentU se num bers forall threads specified in USE_THREADS.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- ARGUMENTS
- SEE ALSO


## NAME

blas_dperm ute -perm utes a real (double precision) anay in term sof the perm utation vectorP, outputby dsortv

## SYNOPSIS

```
SUBROUT\mathbb{NE BLAS_DPERMUTE N,P, INCP,X,\mathbb{NCX)}}\mathbf{N}=()
INTEGER N
\mathbb{NTEGER P (*)}
\mathbb{NTEGER INCP}
REAL*8X (*)
\mathbb{NTEGER \mathbb{NCX}}\mathbf{N}=\mp@code{N}
SUBROUT\mathbb{NE BLAS_DPERMUTE_64 N,P,\mathbb{NCP,X,INCX)}}\mathbf{N}=1
INTEGER*8 N
\mathbb{NTEGER*8 P (*)}
INTEGER*8 \mathbb{NCP}
REAL*8 X (*)
\mathbb{NTEGER*8 }\mathbb{N}CX
```

F95 INTERFACE

```
    SUBROUT\mathbb{NE PERMUTE (X,P)}
    USE SUNPERF
    SUBROUT\mathbb{NE PERMUTE_64 (X,P)}
    USE SUNPERF
```


## ARGUMENTS

N (input) $\mathbb{N}$ TEGER, the num ber ofelem ents to be perm uted in X If $N<=1$, the subroutine retums w ithout trying to perm ute X .

P (input) $\mathbb{N}$ TEGER $(\mathbb{N}-1) \star \mathbb{N} C P \mid+1$ ), the perm utation (index) vectordefined follows the same conventions as that for DTYPE SORTV. It records the details of the interchanges of the elem ents of $X$ during sorting. That is $X=P * X$. In current im plem entation, $P$ contains the index of sorted $X$.
$\mathbb{N} C P$ (input) $\mathbb{N} T E G E R$, increm ent for $P$
$\mathbb{N} C P$ m ustnotbe zero. $\mathbb{N}$ CP could be negative. If
$\mathbb{N} C P<0$, the perm utation is applied in the oppo-
site direction. That is
If $\mathbb{N} C P>0$, if $\mathbb{N} C X>0$, sorted $X((i-1) * \mathbb{N} C X+1)=X(P(i-1) * \mathbb{N} C P+1))$, if $\mathbb{N} C X<0$, sorted $X(\mathbb{N}-i) \star \mathbb{N} C X+1)=X(P((i-1) \star \mathbb{N} C P+1))$;
If $\mathbb{N} C P<0$, if $\mathbb{N} C X>0$, sorted $X((i-1) * \mathbb{N} C X+1)=X(P(\mathbb{N}-i) * \mathbb{N} C P \mid+1))$. if $\mathbb{N} C X<0$, sorted $X(\mathbb{N}-i) \star \mid \mathbb{N} C X+1)$
$=X(\mathbb{P}(\mathbb{N}-i) \star|\mathbb{N} C P|+1))$.
X (input/output) REA $\left.L * 8(\mathbb{K} \mathbb{N D})(\mathbb{N}-1)^{\star}|\mathbb{N} C X|+1\right)$, the array to be perm uted. M inim um size $\mathbb{N}-1)^{\star} \mathbb{N} C X+1$ is required
$\mathbb{N} C X$ (input) $\mathbb{N} T E G E R$, increm ent for $X$
$\mathbb{N} C X$ m ustnotbe zero. $\mathbb{N} C X$ could be negative. If $\mathbb{N} C X<0, X$ w illbe perm uted in a reverse way (see the description for $\mathbb{N}$ CP above).

## SEE ALSO

blas_dsortv (3P ),blas_dsort(3P )

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- ARGUMENTS
- SEE ALSO


## NAME

blas_dsort-sorts a real (double precision) vector X in increasing ordecreasing orderusing quick sort algonithm

## SYNOPSIS

```
SUBROUT\mathbb{NE BLAS_DSORT (SORT,N,X,\mathbb{NCX)}}\mathbf{N}=(
INTEGER SORT
INTEGER N
REAL*8X (*)
\mathbb{NTEGER INCX}
```



```
\mathbb{NTEGER*8 SORT}
INTEGER*8 N
REAL*8 X (*)
\mathbb{NTEGER*8 INCX}
F95 INTERFACE
SUBROUT\mathbb{NE SORT (X [r,SORT])}
USE SUNPERF
SUBROUTINESORT_64 (X [,SORT])
USE SUNPERF
```

The functionality of SORT is covered by SORTV

## ARGUMENTS

SORT (input) $\mathbb{N}$ TEGER, indicating sortdirections
SORT $=0$, descending
SORT = 1, ascending
SORT = othervalue, emror
SORT is default to 1 forF $95 \mathbb{I N}$ TERFACE
$N$ (input) $\mathbb{N} T E G E R$, the num ber of elem ents to be sorted in $X$
If $\mathrm{N}<=1$, the subroutine retums w thout trying to sortX.

X (input/output) REAL*8 $(\mathbb{N}-1) * \mathbb{N} C X \mid+1)$, the anay to be sorted
M inim um size $\mathbb{N}-1)^{*} \mid \mathbb{N} C X+1$ is required
$\mathbb{N} C X$ (input) $\mathbb{N} T E G E R$, increm ent for $X$
$\mathbb{N} C X$ m ustnotbe zero. $\mathbb{N} C X$ could be negative. If
$\mathbb{N} C X<0$, change the sorting direction defined by
SORT.That is
If $S O R T=0$, let $S O R T=1, \mathbb{N} C X=\mathbb{N} C X ;$
If $S O R T=1$ letSORT $=0, \mathbb{N} C X=|\mathbb{N} C X|$.

## SEE ALSO

blas_dsortv (3P ) , blas_dperm ute (3P )

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- ARGUMENTS
- SEE ALSO


## NAME

blas_dsortv - sorts a real (double precision) vector X in increasing or decreasing orderusing quick sort algorithm and overw rite P w th the perm utation vector

## SYNOPSIS



```
INTEGER SORT
INTEGERN
REAL*8X (*)
INTEGER \mathbb{NCX}
INTEGER P(*)
```



```
SU BROUT\mathbb{NE BLAS_D SORTV_64(SORT,N,X,NNCX,P, INCP)}
INTEGER*8 SORT
INTEGER*8 N
REAL*8 ( (*)
\mathbb{NTEGER*8 INCX}
INTEGER*8 P (*)
\mathbb{NTEGER*8 INCP}
```


## F95 INTERFACE

```
SU BROUTINE SORTV ( X [, SORT] [, P])
USE SUNPERF
SU BROUTINE SORTV_64 (X [,SORT] [,P])
U SE SUNPERF
SORTV covers the functionality of SO RT
```


## ARGUMENTS

SORT (input) $\mathbb{N}$ TEGER, indicating sortdirections
SORT $=0$, descending
SORT = 1, ascending
SORT = othervalue, emror
SORT is default to 1 forF $95 \mathbb{I N}$ TERFACE
N (input) $\mathbb{N}$ TEGER, the num ber of elem ents to be sorted in X If $\mathrm{N}<=1$, the subroutine retums w ithout trying to sortX .

X (input/output) REA L*8 $(\mathbb{N}-1) \star \mathbb{N} C X+1)$, the array to be sorted
M inim um size $(\mathbb{N}-1)^{*}|\mathbb{N} C X|+1$ is required
$\mathbb{N} C X$ (input) $\mathbb{N} T E G E R$, increm ent for $X$
$\mathbb{N} C X$ m ustnotbe zero. $\mathbb{N} C X$ could be negative. If
$\mathbb{N} C X<0$, change the sorting direction defined by
SORT.That is
If SORT $=0$, letSORT $=1, \mathbb{N} C X=\mathbb{N} C X ;$
If $S O R T=1$, let $S O R T=0, \mathbb{N} C X=|\mathbb{N} C X|$.
P (output) $\mathbb{N}$ TEGER ( $\mathbb{N}-1$ )* $\mathbb{N} C P \mid+1$ ), the perm utation (index)
vector recording the details of the interchanges
of the elem ents of $X$ during sorting. That is $X=$ $P * X$. In this im plem entation, $P$ contains the index of sorted X.
$\mathbb{N} C P$ (input) $\mathbb{N} T E G E R$, increm ent fipr $P$
$\mathbb{N} C P$ m ustnotbe zero. $\mathbb{N}$ CP could be negative. If
$\mathbb{N} C P<0$, store $P$ (i) in reverse order. That is
If $\mathbb{N} C P>0$,
if $\mathbb{N} C X>0$, sorted $X((i-1) \star \mathbb{N} C X+1)=X(P(i-1) * \mathbb{N} C P+1))$,
if $\mathbb{N} C X<0$, sorted $X(\mathbb{N}-i) \star \mathbb{N} C X+1)=X(\mathbb{P}((i-1) * \mathbb{N} C P+1))$;
If $\mathbb{N}$ CP $<0$,
if $\mathbb{N} C X>0$, sorted $X((i-1) \star \mathbb{N} C X+1)=X(P(\mathbb{N}-i) * \mathbb{N} C P+1))$,
if $\mathbb{N} C X<0$, sorted $X(\mathbb{N}-i) \star|\mathbb{N} C X|+1)$
$\left.=X\left(\mathbb{P}(\mathbb{N}-i)^{\star} \mathbb{N} C P+1\right)\right)$.

## SEE ALSO

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- ARGUMENTS
- SEE ALSO


## NAME

blas_iperm ute -perm utes an integeranray in term s of the perm utation vectorP, outputby dsortv

## SYNOPSIS



```
INTEGER N
\mathbb{NTEGER P (*)}
\mathbb{NTEGER INCP}
\mathbb{NTEGERX (*)}
INTEGER \mathbb{NCX}
SUBROUT\mathbb{NE BLAS_\mathbb{PERMUTE_64 N,P,}\mathbb{N}CP,X,\mathbb{NCX)}}\mathbf{N}=(
INTEGER*8 N
\mathbb{NTEGER*8 P (*)}
INTEGER*8 \mathbb{NCP}
\mathbb{N TEGER *8 X (*)}
```



## F95 INTERFACE

```
SU BROUTINE PERMUTE ( \(X, \mathrm{P}\) )
USE SUNPERF
SU BROUTINE PERMUTE_64 (X,P)
USE SUNPERF
```


## ARGUMENTS

N (input) $\mathbb{N}$ TEGER, the num ber ofelem ents to be perm uted in X If $N<=1$, the subroutine retums w thout trying to perm ute X .
$\operatorname{P}$ (input) $\mathbb{N} \operatorname{TEGER}(\mathbb{N}-1) \star \mathbb{N} C P \mid+1)$, the perm utation (index) vectordefined follow s the same conventions as that for DTYPE SORTV. It records the details of the interchanges of the elem ents of $X$ during sorting. That is $\mathrm{X}=\mathrm{P} * \mathrm{X}$. In current im plem entation, P contains the index of sorted $X$.
$\mathbb{N} C P$ (input) $\mathbb{N} T E G E R$, increm ent for $P$
$\mathbb{N}$ CP m ustnotbe zero. $\mathbb{N}$ CP could be negative. If
$\mathbb{N} C P<0$, the perm utation is applied in the oppo-
site direction. That is
If $\mathbb{N}$ CP $>0$, if $\mathbb{N} C X>0$, sorted $X((i-1) \star \mathbb{N} C X+1)=X(\mathbb{P}((i-1) * \mathbb{N} C P+1))$, if $\mathbb{N} C X<0$, sorted $X(\mathbb{N}-i) \star \mathbb{N} C X+1)=X(P((i-1) * \mathbb{N} C P+1))$;
If $\mathbb{N} C P<0$, if $\mathbb{N} C X>0$, sorted $X((i-1) * \mathbb{N} C X+1)=X(P(\mathbb{N}-i) * \mathbb{N} C P+1))$. if $\mathbb{N} C X<0$, sorted $X(\mathbb{N}-i) \star \mathbb{N} C X+1)$
$\left.=X\left(\mathbb{P}(\mathbb{N}-i)^{\star} \mid \mathbb{N} C P+1\right)\right)$.
$X$ (input/output) $\mathbb{N} \operatorname{TEGER}(\mathbb{K} \mathbb{N} D)(\mathbb{N}-1) \star \mathbb{N} C X+1)$, the array to be perm uted. M inim um size $\mathbb{N}-1)^{\star}|\mathbb{N} C X|+1$ is required
$\mathbb{N} C X$ (input) $\mathbb{N} T E G E R$, increm ent for $X$
$\mathbb{N} C X$ m ustnotbe zero. $\mathbb{N} C X$ could be negative. If $\mathbb{N} C X<0, X$ w illbe perm uted in a reverse w ay (see the description for $\mathbb{N} C P$ above).

## SEE ALSO

blas_isortv (3P ),blas_isort(3P )

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- ARGUMENTS
- SEE ALSO


## NAME

blas_isort-sorts an integer vector X in increasing or decreasing orderusing quick sort algorithm

## SYNOPSIS

```
SUBROUT\mathbb{NE BLAS_ISORT (SORT,N,X,INCX)}
INTEGER SORT
INTEGER N
\mathbb{NTEGERX (*)}
INTEGER \mathbb{NCX}
SUBROUT\mathbb{NE BLAS_ISORT_64(SORT,N,X,\mathbb{NCX)}}\mathbf{N}\mathrm{ (SO}
INTEGER*8 SORT
\mathbb{NTEGER*8 N}
INTEGER*8 X (*)
\mathbb{NTEGER*8 INCX}
F95 INTERFACE
    SUBROUT\mathbb{NE SORT (X [r,SORT])}
    USE SUNPERF
    SUBROUTINESORT_64 (X [,SORT])
    USE SUNPERF
```

    The functionality of SORT is covered by SORTV
    
## ARGUMENTS

SORT (input) $\mathbb{N}$ TEGER, indicating sortdirections
SORT $=0$, descending
SORT = 1, ascending
SORT = othervalue, emror
SORT is default to 1 forF $95 \mathbb{I N}$ TERFACE
$N$ (input) $\mathbb{N} T E G E R$, the num ber of elem ents to be sorted in $X$ If $\mathrm{N}<=1$, the subroutine retums w thout trying to sortX.

X (input/output) $\mathbb{N} \operatorname{TEGER}(\mathbb{N}-1) * \mathbb{N} C X \mid+1)$, the amay to be sorted
$M$ inim um size $(\mathbb{N}-1)^{\star}|\mathbb{N} C X|+1$ is required
$\mathbb{N} C X$ (input) $\mathbb{N} T E G E R$, increm ent for $X$
$\mathbb{N} C X$ m ustnotbe zero. $\mathbb{N} C X$ could be negative. If
$\mathbb{N} C X<0$, change the sorting direction defined by
SORT.That is
If SORT $=0$, letSORT $=1, \mathbb{N} C X=\mathbb{N} C X ;$
If $S O R T=1$ letSORT $=0, \mathbb{N} C X=|\mathbb{N} C X|$.

## SEE ALSO

blas_isortv (3P ) , blas_iperm ute (3P)

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- ARGUMENTS
- SEE ALSO


## NAME

blas_isortv - sorts a real vector X in increasing or decreasing order using quick sort algorithm and overw rite $P$ w ith the perm utation vector

## SYNOPSIS



```
INTEGER SORT
INTEGER N
\mathbb{NTEGERX (*)}
INTEGER \mathbb{NCX}
\mathbb{NTEGERP(*)}
INTEGER \mathbb{NCP}
SUBROUTINE BLAS_ISORTV_64(SORT,N,X,\mathbb{NCX,P, NNCP)}
INTEGER*8 SORT
\mathbb{N}TEGER*8 N
INTEGER*8 X (*)
INTEGER*8 \mathbb{NCX}
\mathbb{NTEGER*8 P (*)}
\mathbb{N}TEGER*8 \mathbb{N CP}
```


## F95 INTERFACE

```
SU BROUTINE SORTV ( \(\times\) [, SORT] [,P])
USE SUNPERF
SU BROUTINE SORTV_64 (X [,SORT] [,P])
USE SUNPERF
```

SORTV covers the functionality of SO RT

## ARGUMENTS

SORT (input) $\mathbb{N}$ TEGER, indicating sortdirections
SORT $=0$, descending
SORT = 1, ascending
SORT = othervalue, emror
SORT is default to 1 forF $95 \mathbb{I N}$ TERFACE
N (input) $\mathbb{N}$ TEGER, the num ber of elem ents to be sorted in X If $\mathrm{N}<=1$, the subroutine retums w ithout trying to sortX .

X (input/output) $\mathbb{N} \operatorname{TEGER}(\mathbb{N}-1) \star|\mathbb{N} C X|+1)$, the amay to be sorted
M inim um size $(\mathbb{N}-1)^{\star}|\mathbb{N} C X|+1$ is required
$\mathbb{N} C X$ (input) $\mathbb{N} T E G E R$, increm ent for $X$
$\mathbb{N} C X$ m ustnotbe zero. $\mathbb{N} C X$ could be negative. If
$\mathbb{N} C X<0$, change the sorting direction defined by
SORT.That is
If SORT $=0$, letSORT $=1, \mathbb{N} C X=\mathbb{N} C X ;$
If $S O R T=1$, let $S O R T=0, \mathbb{N} C X=|\mathbb{N} C X|$.
P (output) $\mathbb{N}$ TEGER ( $\mathbb{N}-1$ )* $\mathbb{N} C P \mid+1$ ), the perm utation (index)
vector recording the details of the interchanges
of the elem ents of $X$ during sorting. That is $X=$ $P * X$. In this im plem entation, $P$ contains the index of sorted X.
$\mathbb{N} C P$ (input) $\mathbb{N} T E G E R$, increm ent fipr $P$
$\mathbb{N} C P$ m ustnotbe zero. $\mathbb{N}$ CP could be negative. If
$\mathbb{N} C P<0$, store $P$ (i) in reverse order. That is
If $\mathbb{N} C P>0$,
if $\mathbb{N} C X>0$, sorted $X((i-1) \star \mathbb{N} C X+1)=X(P(i-1) * \mathbb{N} C P+1))$,
if $\mathbb{N} C X<0$, sorted $X(\mathbb{N}-i) \star \mathbb{N} C X+1)=X(\mathbb{P}((i-1) * \mathbb{N} C P+1))$;
If $\mathbb{N}$ CP $<0$,
if $\mathbb{N} C X>0$, sorted $X((i-1) \star \mathbb{N} C X+1)=X(P(\mathbb{N}-i) * \mathbb{N} C P+1))$,
if $\mathbb{N} C X<0$, sorted $X(\mathbb{N}-i) \star|\mathbb{N} C X|+1)$
$\left.=X\left(\mathbb{P}(\mathbb{N}-i)^{\star} \mathbb{N} C P+1\right)\right)$.

## SEE ALSO

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- ARGUMENTS
- SEE ALSO


## NAME

blas_sperm ute - perm utes a realaray in term s of the perm utation vector $P$, outputby dsortv

## SYNOPSIS

```
SUBROUTINE BLAS_SPERMUTE N,P, INCP,X,INCX)
INTEGER N
\mathbb{NTEGER P (*)}
\mathbb{NTEGER INCP}
REALX (*)
\mathbb{NTEGER \mathbb{NCX}}\mathbf{N}=\mp@code{N}
```



```
INTEGER*8 N
\mathbb{NTEGER*8 P (*)}
INTEGER*8 \mathbb{NCP}
REAL X (*)
\mathbb{NTEGER*8 }\mathbb{N}CX
```

F95 INTERFACE

```
    SU BROUTINE PERMUTE (X,P)
    USE SUNPERF
    SUBROUT\mathbb{NE PERMUTE_64 (X,P)}
    USE SUNPERF
```


## ARGUMENTS

N (input) $\mathbb{N}$ TEGER, the num ber ofelem ents to be perm uted in X If $N<=1$, the subroutine retums w ithout trying to perm ute X .

P (input) $\mathbb{N}$ TEGER $(\mathbb{N}-1) \star \mathbb{N} C P \mid+1$ ), the perm utation (index) vectordefined follows the same conventions as that for DTYPE SORTV. It records the details of the interchanges of the elem ents of $X$ during sorting. That is $X=P * X$. In current im plem entation, $P$ contains the index of sorted $X$.
$\mathbb{N} C P$ (input) $\mathbb{N} T E G E R$, increm ent for $P$
$\mathbb{N} C P$ m ustnotbe zero. $\mathbb{N}$ CP could be negative. If
$\mathbb{N} C P<0$, the perm utation is applied in the oppo-
site direction. That is
If $\mathbb{N} C P>0$, if $\mathbb{N} C X>0$, sorted $X((i-1) \star \mathbb{N} C X+1)=X(\mathbb{P}((i-1) * \mathbb{N} C P+1))$, if $\mathbb{N} C X<0$, sorted $X(\mathbb{N}-i) \star \mathbb{N} C X+1)=X(P((i-1) \star \mathbb{N} C P+1))$;
If $\mathbb{N} C P<0$, if $\mathbb{N} C X>0$, sorted $X((i-1) * \mathbb{N} C X+1)=X(P(\mathbb{N}-i) * \mathbb{N} C P \mid+1))$. if $\mathbb{N} C X<0$, sorted $X(\mathbb{N}-i) \star \mid \mathbb{N} C X+1)$

$$
=X(\mathbb{P}(\mathbb{N}-i) \star \mathbb{N} C P \mid+1)) .
$$

X (input/output) REAL $(\mathbb{K} \mathbb{N} D)(\mathbb{N}-1) * \mathbb{N} C X+1)$, the anray to be perm uted. $M$ inim um size $(\mathbb{N}-1) * \mathbb{N} C X+1$ is required
$\mathbb{N} C X$ (input) $\mathbb{N} T E G E R$, increm ent for $X$
$\mathbb{N} C X$ mustnotbe zero. $\mathbb{N} C X$ could be negative. If $\mathbb{N} C X<0, X$ w illbe perm uted in a reverse way (see the description for $\mathbb{N} C P$ above).

## SEE ALSO

blas_ssortv (3P ),blas_ssort(3P)

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- ARGUMENTS
- SEE ALSO


## NAME

blas_ssort-sorts a realvectorX in increasing ordecreasing orderusing quick sort algorithm

## SYNOPSIS

```
SUBROUT\mathbb{NE BLAS_SSORT (SORT,N,X,INCX)}
INTEGER SORT
INTEGER N
REALX (*)
INTEGER \mathbb{NCX}
SU BROUT\mathbb{NE BLAS_SSORT_64 (SORT,N,X,\mathbb{NCX)}}\mathbf{N}=(
\mathbb{NTEGER*8 SORT}
INTEGER*8 N
REALX (*)
\mathbb{NTEGER*8 INCX}
F95 INTERFACE
SUBROUT\mathbb{NE SORT (X [r,SORT])}
USE SUNPERF
SUBROUTINESORT_64 (X [,SORT])
USE SUNPERF
```

The fiunctionality of SORT is covered by SORTV

## ARGUMENTS

SORT (input) $\mathbb{N}$ TEGER, indicating sortdirections
SORT $=0$, descending
SORT = 1, ascending
SORT = othervalue, emror
SORT is default to 1 forF $95 \mathbb{I N}$ TERFACE
N (input) $\mathbb{N}$ TEGER, the num ber of elem ents to be sorted in X
If $\mathrm{N}<=1$, the subroutine retums w ithout trying to sortX.

X (input/output) REAL $(\mathbb{N}-1) * \mathbb{N} C X+1)$, the anay to be sorted
M inim um size $\mathbb{N}-1)^{*} \mid \mathbb{N} C X+1$ is required
$\mathbb{N} C X$ (input) $\mathbb{N} T E G E R$, increm ent for $X$
$\mathbb{N} C X$ m ustnotbe zero. $\mathbb{N} C X$ could be negative. If
$\mathbb{N} C X<0$, change the sorting direction defined by
SORT.That is
If SORT $=0$, letSORT $=1, \mathbb{N} C X=\mathbb{N} C X ;$
If $S O R T=1$ letSORT $=0, \mathbb{N} C X=|\mathbb{N} C X|$.

## SEE ALSO

blas_ssortv (3P ) , blas_sperm ute (3P)

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- ARGUMENTS
- SEE ALSO


## NAME

blas_ssortv - sorts a real vector X in increasing or decreasing order using quick sortalgorithm and overw rite $P$ w ith the perm utation vector

## SYNOPSIS



```
INTEGER SORT
INTEGERN
REALX(*)
INTEGER INCX
\mathbb{NTEGER P (*)}
\mathbb{NTEGER INCP}
SUBROUT\mathbb{NE BLAS_SSORTV_64 (SORT,N,X,\mathbb{NCX,P,INCP)}}\mathbf{N}=1
INTEGER*8 SORT
\mathbb{NTEGER*8 N}
REAL X (*)
\mathbb{NTEGER*8 INCX}
INTEGER*8 P (*)
\mathbb{NTEGER*8 \mathbb{NCP}}\mathbf{}=1
```


## F95 INTERFACE

```
SU BROUTINE SORTV ( \(\times\) [, SORT] [,P])
USE SUNPERF
SU BROUTINE SORTV_64 (X [,SORT] [,P])
USE SUNPERF
SORTV covers the functionality of SO RT
```


## ARGUMENTS

SORT (input) $\mathbb{N}$ TEGER, indicating sortdirections
SORT $=0$, descending
SORT = 1, ascending
SORT = othervalue, emror
SORT is default to 1 forF $95 \mathbb{I N}$ TERFACE
N (input) $\mathbb{N}$ TEGER, the num ber of elem ents to be sorted in X
If $\mathrm{N}<=1$, the subroutine retums w ithout trying to sortX .

X (input/output) REAL $(\mathbb{N}-1) \star \mathbb{N} C X+1)$, the array to be sorted
$M$ inim um size $(\mathbb{N}-1) *|\mathbb{N} C X|+1$ is required
$\mathbb{N} C X$ (input) $\mathbb{N} T E G E R$, increm ent for $X$
$\mathbb{N} C X$ m ustnotbe zero. $\mathbb{N} C X$ could be negative. If
$\mathbb{N} C X<0$, change the sorting direction defined by
SORT.That is
If SORT $=0$, letSORT $=1, \mathbb{N} C X=\mathbb{N} C X ;$
If $S O R T=1$, let $S O R T=0, \mathbb{N} C X=|\mathbb{N} C X|$.
P (output) $\mathbb{N}$ TEGER ( $\mathbb{N}-1$ )* $\mathbb{N} C P \mid+1$ ), the perm utation (index)
vector recording the details of the interchanges
of the elem ents of $X$ during sorting. That is $X=$ $P * X$. In this im plem entation, $P$ contains the index of sorted X.
$\mathbb{N} C P$ (input) $\mathbb{N} T E G E R$, increm ent fipr $P$
$\mathbb{N} C P$ m ustnotbe zero. $\mathbb{N}$ CP could be negative. If
$\mathbb{N} C P<0$, store $P$ (i) in reverse order. That is
If $\mathbb{N} C P>0$,
if $\mathbb{N} C X>0$, sorted $X((i-1) \star \mathbb{N} C X+1)=X(P(i-1) * \mathbb{N} C P+1))$,
if $\mathbb{N} C X<0$, sorted $X(\mathbb{N}-i) \star \mathbb{N} C X+1)=X(\mathbb{P}((i-1) * \mathbb{N} C P+1))$;
If $\mathbb{N}$ CP $<0$,
if $\mathbb{N} C X>0$, sorted $X((i-1) \star \mathbb{N} C X+1)=X(P(\mathbb{N}-i) * \mathbb{N} C P+1))$,
if $\mathbb{N} C X<0$, sorted $X(\mathbb{N}-i) \star|\mathbb{N} C X|+1)$
$\left.=X\left(\mathbb{P}(\mathbb{N}-i)^{\star} \mathbb{N} C P+1\right)\right)$.

## SEE ALSO

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
caxpy - com pute y := alpha * x + y
```


## SYNOPSIS

```
SUBROUT\mathbb{NE CAXPY N,ALPHA,X, NNCX,Y,\mathbb{NCY)}}\mathbf{N},\mp@code{N}
COM PLEX A LPHA
COM PLEX X (*),Y (*)
```



```
SUBROUT\mathbb{NE CAXPY_64 N,ALPHA,X,INCX,Y,}\mathbb{NCY)}
COM PLEX A LPHA
COM PLEX X (*),Y (*)
INTEGER*8N,\mathbb{NCX,INCY}
F95 INTERFACE
```



```
    COMPLEX ::ALPHA
    COM PLEX,D IM ENSION (:) ::X,Y
    \mathbb{NTEGER ::N,\mathbb{NCX,INCY}}\mathbf{N}={
```



```
    COMPLEX ::ALPHA
    COM PLEX,D IM ENSION (:) ::X,Y
    \mathbb{NTEGER (8)::N, INCX,INCY}
C INTERFACE
    #include <sunperfh>
```

void caxpy (intn, com plex *alpha, com plex *x, int incx, com plex *y, intincy);
void caxpy_64 (long n, com plex *alpha, com plex *x, long incx, com plex *y, long incy);

## PURPOSE

caxpy com pute $\mathrm{y}:=$ alpha * $\mathrm{x}+\mathrm{y}$ w here alpha is a scalar and $x$ and $y$ are $n$-vectors.

## ARGUMENTS

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
array ofD $\mathbb{I M}$ ENSION at least ( $1+(\mathrm{n}-1)$ *abs( $\mathbb{N C X}$ ) ). Before entry, the increm ented amay $X$ $m$ ust contain the vectorx. U nchanged on exit.
$\mathbb{N C X}$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of $\mathrm{X} . \mathbb{N} C X$ m ustnotbe zero. U nchanged on exit.

Y (input/output)
aray ofD $\mathbb{I M}$ ENSION at least ( $1+(\mathrm{n}-1$ )*abs( $\mathbb{N} C Y$ )). On entry, the increm ented amay $Y \mathrm{~m}$ ust contain the vectory. On exit, $Y$ is overw rilten by the updated vectory.
$\mathbb{N} C Y$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $Y . \mathbb{N} C Y$ m ustnotbe zero. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS


## NAME

```
caxpyi-C om pute y := alpha * x + y
```


## SYNOPSIS

```
SUBROUT\mathbb{NE CAXPYINZ,A,X,INDX,Y)}
COM PLEX A
COM PLEX X (*),Y (*)
INTEGER NZ
INTEGER INDX(*)
SUBROUTINE CAXPYI_64NZ,A,X,NNDX,Y)
COM PLEX A
COM PLEX X (*),Y (*)
INTEGER*8NZ
INTEGER*8 \mathbb{NDX (*)}
F95 IN TERFACE
SUBROUT\mathbb{NE AXPYI(NZ],[A],X,NNDX,Y)}
COM PLEX ::A
COMPLEX,D IM ENSION (:) ::X,Y
INTEGER ::NZ
\mathbb{NTEGER,D IM ENSION (:) :: \mathbb{NDX}}\mathbf{N}=\mp@code{N}
SUBROUTINE AXPYI_64(NZ ],[A],X,\mathbb{NDX,Y)}
COM PLEX ::A
COMPLEX,D IM ENSION (:) ::X,Y
\mathbb{NTEGER (8) ::N Z}
\mathbb{NTEGER (8),D IM ENSION (:) :: \mathbb{NDX}}\mathbf{N}=\mp@code{l}
```

CAXPY IC om pute $\mathrm{y}:=$ alpha * $\mathrm{x}+\mathrm{y}$ where alpha is a scalar, x is a sparse vector, and $y$ is a vector in full storage form

```
do \(i=1, n\)
    \(y\) (indx (i)) = alpha * x (i) \(+y\) (indx (i))
enddo
```


## ARGUMENTS

NZ (input) - $\mathbb{N}$ TEGER
$N$ um ber of elem ents in the com pressed form .
U nchanged on exit.

A (input)
On entry, A (LPH A ) specifies the scaling value.
U nchanged on exit. A is defaulted to ( $1.0 \mathrm{E} 0,0.0 \mathrm{E} 0$ )
forF95 $\mathbb{I N}$ TERFACE.
X (input)
V ector containing the values of the com pressed form .
U nchanged on exit.
$\mathbb{N} D \mathrm{X}$ (input) - $\mathbb{N}$ TEGER
$V$ ector containing the indices of the com pressed form. It is assum ed that the elem ents in $\mathbb{N} D \mathrm{X}$ are distinct and greater than zero. U nchanged on exit.

Y (output)
V ectoron inputw hich contains the vector $Y$ in full storage form. On exit, only the elem ents
comesponding to the indices in $\mathbb{N} D \mathrm{X}$ have been
m odified.

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

```
cbcomm -block coordinatem atrix m atrix m ultiply
```


## SYNOPSIS

```
SUBROUTINE CBCOMM(TRANSA,MB,N,KB,ALPHA,DESCRA,
* VAL,BINDX,BJNDX,BNNZ,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,MB,N,KB,DESCRA (5),BNNZ,LB,
* LDB,LDC,LWORK
```



```
COM PLEX ALPHA,BETA
COM PLEX VAL (LB*LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINECBCOMM_64(TRANSA,MB,N,KB,ALPHA,DESCRA,
* VAL,BINDX,BJNDX,BNNZ,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,MB,N,KB,DESCRA (5),BNNZ,LB,
* LDB,LDC,LW ORK
INTEGER*8 BINDX (BNNZ),BJNDX (BNNZ)
COM PLEX ALPHA,BETA
COM PLEX VAL (LB*LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
```


## F95 INTERFACE

SUBROUTINE BCOMM (TRANSA, MB,N, $K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D, B J N D$, * BNNZ,LB,B,[LDB],BETA,C,[LDC],[W ORK], [LW ORK])
$\mathbb{N} T E G E R$ TRANSA, MB,N,KB,BNNZ,LB
$\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \quad D E S C R A, B \mathbb{N} D, B \operatorname{ND} D$
COMPLEX ALPHA,BETA
COMPLEX,D $\mathbb{M}$ ENSION (:) ::VAL
COM PLEX,D $\mathbb{I}$ ENSION (: : : :: B, C

* BNNZ,LB,B,[LDB],BETA,C,[LDC],[WORK],[LWORK])
$\mathbb{N} T E G E R * 8$ TRANSA, MB,N,KB, BNNZ,LB
$\mathbb{N} T E G E R * 8, D \mathbb{I}$ ENSION (:) :: DESCRA,B $\mathbb{N} D X, B J N D X$
COMPLEX ALPHA,BETA
COM PLEX,D $\mathbb{M}$ ENSION (:) ::VAL
COM PLEX,D $\mathbb{I}$ ENSION (: : : :: B, C


## DESCRIPTION

$$
C \text { <-alpha op (A) B + beta C }
$$

where A LPHA and BETA are scalar, $C$ and $B$ are dense $m$ atrioes, $A$ is a m atrix represented in block coordinate form at and op(A) is one of $o p(A)=A$ or $o p(A)=A^{\prime}$ or $o p(A)=\operatorname{conjg}\left(A^{\prime}\right)$.
( 'indicates m atrix transpose)

## ARGUMENTS

| TRANSA | A Indicates how to operate w th the sparse $m$ atrix |
| :---: | :---: |
|  | 0 : operate w th m atrix |
|  | 1 : operate w ith transpose m atrix |
|  | 2 : operate $w$ th the conjugate transpose ofm atrix. 2 is equivalent to 1 if the $m$ atrix is real. |
| M B | N um berofblock row s in m atrix A |
| N | $N$ um berof colum $n s$ in $m$ atrix $C$ |
| K B | $N$ um ber ofblock colum ns in m atrix A |
| A LPH A | Scalar param eter |
| DESCRA | () D escriptor argum ent. Five elem ent integer amay |
|  | DESCRA (1) m atrix structure |
|  | 0 : general |
|  | 1 : symmetric ( $\mathrm{A}=\mathrm{A}$ ) |
|  | 2: Herm itian ( $\mathrm{A}=\mathrm{CONJG}(\mathrm{A})$ ) |
|  | 3 :Triangular |
|  | 4 : Skew (A nti)-Symm etric ( $\mathrm{A}=-\mathrm{A}$ ) |
|  | 5 :D iagonal |
|  | 6 : Skew Herm itian ( $\mathrm{A}=-\mathrm{CONJG}(\mathrm{A})$ ) |
|  | D ESCRA (2) upper/low er triangular indicator |
|  | 1 : low er |
|  | 2 : upper |
|  | D ESCRA (3) m ain diagonaltype |

0 :non-unit
1 : unit
DESCRA (4) A ray base $\mathbb{N} O T \mathbb{I}$ PLEM ENTED )
0 :C C++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL ( scalar array of length LB *LB *BNN Z consisting of the non-zero block entries of $A$, in any order. Each block is stored in standard colum n-m ajor form .
$B \mathbb{N} D X($ integer array of length $B N N Z$ consisting of the block row indiaes of the block entries of .

B JND X 0 integer anray of length BNNZ consisting of the block colum $n$ indiges of the block entries of $A$.

BNNZ num berofblock entries

LB dim ension of dense blocks com posing A.
B 0 rectangular array with first dim ension LD B .
LD B leading dim ension ofB
BETA Scalarparam eter

C 0 rectangular array w ith firstdim ension LD C .

LD C leading dim ension ofC
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the currentversion.

LW ORK length ofW ORK array.LW ORK is not referenced in the currentversion.

## SEE ALSO

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov $\mathrm{m}_{\mathrm{c}}$ csd/Staffk Rem ington/tspoblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

codim m -block diagonal form atm atrix-m atrix m ultiply

## SYNOPSIS

```
SUBROUT\mathbb{NECBD IMM(TRANSA,M B,N,KB,ALPHA,DESCRA,}
* VAL,BLDA, \mathbb{BDIAG,NBDIAG,LB,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,MB,N,KB,DESCRA (5),BLDA,NBDIAG,LB,}
* LDB,LDC,LW ORK
INTEGER \mathbb{BDIAG NBDIAG)}
COM PLEX ALPHA,BETA
COM PLEX VAL (LB *LB*BLDA *NBD IAG),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SU BROUTINE CBD IMM_64(TRANSA,M B ,N,KB,ALPHA,DESCRA,
* VAL,BLDA,\mathbb{BDIAG,NBDIAG,LB,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,M B,N,KB,DESCRA (5),BLDA,NBD IAG,LB,}
* LDB,LDC,LW ORK
```



```
COM PLEX ALPHA,BETA
COM PLEX VAL (LB *LB*BLDA *NBD IAG),B (LDB,*),C (LDC,*),W ORK (LW ORK)
```


## F95 INTERFACE

SUBROUTINEBD $\mathbb{I} M$ (TRANSA, MB, $\mathbb{N}], K B, A L P H A, D E S C R A, V A L, B L D A$, * $\mathbb{B D} \mathbb{I} G, N B D \mathbb{I} G, L B, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])$
$\mathbb{N} T E G E R$ TRANSA, MB,KB,BLDA,NBD $\mathbb{I A} G, L B$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}$ (:) :: DESCRA, $\mathbb{B D} \mathbb{I} G$
COMPLEX ALPHA,BETA
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::VAL
COM PLEX,D $\mathbb{I}$ ENSION (: : : :: B,C
SUBROUTINEBD $\mathbb{M} M \_64$ (TRANSA, MB, $\left.\mathbb{N}\right], K B, A L P H A, D E S C R A, V A L, B L D A$,

* $\mathbb{B D} \mathbb{I} G, N B D \mathbb{I A}, \mathrm{LB}, \mathrm{B},[\mathrm{LD} \mathrm{B}], \mathrm{BETA}, \mathrm{C},[\mathrm{LDC}],[\mathbb{W} O R K],[L W O R K])$
$\mathbb{N} T E G E R * 8$ TRANSA, MB,KB,BLDA,NBD $\mathbb{I A} G, L B$
$\mathbb{N} T E G E R * 8, D \mathbb{I M} E N S \mathbb{O N}(:):: \quad$ ESCRA, $\mathbb{B D} \mathbb{I A}$
COM PLEX ALPHA,BETA
COMPLEX,D $\mathbb{M}$ ENSION (:) ::VAL
COM PLEX,D $\mathbb{I M}$ ENSION (:, :) :: B,C


## DESCRIPTION

$$
C \text { <-alpha op (A) B + beta C }
$$

where A LPHA and BETA are scalar, $C$ and $B$ are dense $m$ atrioes, $A$ is a m atrix represented in block diagonal form at and op(A) is one of $o p(A)=A$ or $o p(A)=A^{\prime}$ or $o p(A)=\operatorname{conjg}\left(A^{\prime}\right)$.
('indicates m atrix transpose)

## ARGUMENTS

| TRANSA | A Indicates how to operate w th the sparse $m$ atrix |
| :---: | :---: |
|  | 0 : operate w th m atrix |
|  | 1 : operate w ith transpose m atrix |
|  | 2 : operate $w$ th the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real. |
| M B | N um berofblock row s in m atrix A |
| N | $N$ um berof colum $n s$ in $m$ atrix $C$ |
| K B | $N$ um ber ofblock colum ns in m atrix A |
| A LPH A | Scalar param eter |
| DESCRA | () D escriptor argum ent. Five elem ent integer amay |
|  | DESCRA (1) m atrix structure |
|  | 0 : general |
|  | 1 : symmetric ( $\mathrm{A}=\mathrm{A}$ ) |
|  | 2: Herm itian ( $\mathrm{A}=\mathrm{CONJG}(\mathrm{A})$ ) |
|  | 3 :Triangular |
|  | 4 : Skew (A nti)-Symm etric ( $\mathrm{A}=-\mathrm{A}$ ) |
|  | 5 :D iagonal |
|  | 6 : Skew Herm itian ( $\mathrm{A}=-\mathrm{CONJG}(\mathrm{A})$ ) |
|  | D ESCRA (2) upper/low er triangular indicator |
|  | 1 : low er |
|  | 2 : upper |
|  | D ESCRA (3) m ain diagonaltype |

0 : non-unit
1 : unit
DESCRA (4) A ray base $\mathbb{N} O T \mathbb{I}$ PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? $\mathbb{N} O T \mathbb{I}$ PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL $0 \quad$ tw o-dim ensionalLB *LB *BLD A -by-N BD IA G scalar anay consisting of the NBD IA G nonzero block diagonal in any order. Each dense block is stored in standard colum n.m ajor form .

BLD A leading block dim ension ofV A L ( ).
IBD IA G 0 integer amay of length N BD IA G consisting of the corresponding diagonaloffsets of the non-zero block diagonals ofA in VA L. Low ertriangular block diagonals have negative offsets, the $m$ ain block diagonal has offset 0, and uppertriangular block diagonals have positive offset.

NBD IA G the num berofnon-zero block diagonals in A.
LB dim ension of dense blocks com posing A.
B 0 rectangular array w ith first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w th first dim ension LD C.

LD C leading dim ension ofC
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is not referenced in the currentversion.

## SEE ALSO

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib org/utk/papers/sparse .ps

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

cbdism - block diagonal form at triangular solve

## SYNOPSIS

```
SUBROUTINECBD ISM (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,
* VAL,BLDA, \(\mathbb{B D} \mathbb{A} G, N B D \mathbb{I A}, \mathrm{LB}\),
* B,LDB,BETA, C,LDC,W ORK,LW ORK )
\(\mathbb{N} T E G E R\) TRANSA, MB,N,UNITD,DESCRA (5), BLDA,NBD \(\mathbb{I A} G, L B\),
* LDB,LDC,LWORK
\(\mathbb{N} T E G E R \quad \mathbb{B D} \mathbb{I} G \mathbb{N} B D \mathbb{I} G)\)
COMPLEX ALPHA,BETA
COM PLEX DV M B*LB*LB),VAL (LB*LB*BLDA,NBD IAG),B(LDB,*),C (LDC,*),
* WORK (LW ORK)
SUBROUTINECBD ISM _64 (TRANSA, M B,N,UN ITD,DV,ALPHA,DESCRA,
* VAL,BLDA, \(\mathbb{B D} \mathbb{A} G, N B D \mathbb{I} G, L B\),
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,N,UNITD,DESCRA (5), BLDA,NBD IAG,LB,
* LDB,LDC,LWORK
\(\mathbb{N} T E G E R * 8 \mathbb{B D} \mathbb{A} G \mathbb{N} B \mathbb{A} G)\)
COM PLEX ALPHA,BETA
COM PLEX DV M B*LB*LB),VAL (LB*LB*BLDA,NBD IA G),B(LDB,*),C (LDC,*),
* WORK (LWORK)
```


## F95 INTERFACE

SUBROUTINEBD ISM (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,VAL,BLDA, * $\mathbb{B D} \mathbb{I} G, N B D \mathbb{I} G, L B, B,[L D B], B E T A, C,[L D C],[W O R K],[L W$ ORK]) $\mathbb{N} T E G E R$ TRANSA, MB,N,UNITD,BLDA,NBDIAG,LB $\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \quad \mathrm{DESCRA}, \mathbb{B D} \mathbb{I} G$
COM PLEX ALPHA,BETA
COMPLEX,D $\mathbb{M} E N S I O N(:):: V A L, D V$
COMPLEX,D $\mathbb{I M}$ ENSION (: : : :: B, C

SUBROUTINE BD ISM _64 (TRANSA , M B , N , UN ITD , DV, ALPHA, DESCRA, VAL, BLDA,

* $\mathbb{B D} \mathbb{I A G}, N B D \mathbb{I A G}, \mathrm{LB}, \mathrm{B},[\mathrm{LDB}], \mathrm{BETA}, \mathrm{C},[\mathrm{LD} \mathrm{C}],[\mathrm{O} O R K],[L W O R K])$
$\mathbb{N} T E G E R * 8$ TRANSA, M B , N , UNITD , BLDA, NBD IA G , LB
$\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O}(:):: \quad \mathrm{DESCRA}, \mathbb{B D} \mathbb{I} G$
COMPLEX ALPHA,BETA
COM PLEX,D $\operatorname{IM}$ ENSION (:) ::VAL, DV
COM PLEX , D $\mathbb{M}$ ENSION (:, :) :: B , C


## DESCRIPTION

$$
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { Op (A) B + BETA C } \\
& C<-A L P H A \text { Op (A)D B + BETA } C
\end{aligned}
$$

where ALPHA and BETA are scalar, $C$ and $B$ are m by $n$ dense $m$ atrices, $D$ is a block diagonalm atrix, $A$ is a unit, ornon-unit, upperor low ertriangularm atrix represented in block diagonal form at and op (A) is one of op (A) $)=\operatorname{inv}(A)$ or op (A $)=\operatorname{inv}(A)$ or op (A) $=\operatorname{inv}\left(\operatorname{cong}\left(A^{\prime}\right)\right)$ (inv denotesm atrix inverse, 'indicates m atrix transpose)

## ARGUMENTS

TRAN SA Indicates how to operate $w$ th the sparse $m$ atrix 0 : operate $w$ ith $m$ atrix 1 : operate $w$ th transpose $m$ atrix 2 : operate $w$ ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M B $\quad$ Num ber ofblock row $s$ in $m$ atrix $A$

N $\quad$ um berof colum $n s$ in $m$ atrix $C$

UN ITD Type of scaling:
1 : Identity matrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum $n$ scaling)

DV () A rray of length M B *LB *LB containing the elem ents of the diagonalblocks of them atrix $D$. The size of each square block is LB -by-LB and each block is stored in standard colum n-m ajor form .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay DESCRA (1) m atrix structure
0 : general
1 : symm etric ( $\mathrm{A}=\mathrm{A}$ )
2 : Herm itian ( $A=\operatorname{CONJG}(A))$
3 :Triangular
4 : Skew (A nti)-Symm etric ( $\mathrm{A}=-\mathrm{A}$ )
5 :D iagonal
6 : Skew Herm itian ( $A=-C O N J(A)$ )
N ote: For the routine, D ESCRA (1)=3 is only supported.
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonaltype
0 : non-identity blocks on the $m$ ain diagonal
1 : identity diagonalblocks
2 : diagonalblocks are dense $m$ atrices
DESCRA (4) A may base $\mathbb{N O T} \mathbb{I}$ PLEM ENTED)
0 :C C ++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? $\mathbb{N O T} \mathbb{I}$ PLEM ENTED)
0 : unknown
1 : no repeated indices
VAL () Two-dim ensionalLB *LB *B LD A -by-N BD IA G scalaranay
consisting of the N BD IA G non-zero block diagonal.
Each dense block is stored in standard colum n-m ajor form .
B LD A Leading block dim ension ofV A L (). Should be greater
than orequal to M B .
IBD IA G 0 integer amay of length NBD IA G consisting of the corresponding diagonal offsets of the non-zero block diagonals ofA in VA L. Low ertriangularblock diagonals have negative offsets, them ain block diagonalhas offset 0 , and upper triangularblock diagonals have positive offset. Elem ents of IBD IA G M UST be sorted in increasing order.
NBD IA G The num berofnon-zero block diagonals in A.
LB D im ension of dense blocks com posing A.
B 0 Rectangular aray with firstdim ension LD B .
LD B Leading dim ension of B .
BETA Scalarparam eter.
C 0 Rectangular array w ith first dim ension LD C .
LD C Leading dim ension of C .

W ORK 0 scratch array of length LW ORK. On exit, ifLW ORK=-1,W ORK (1) retums the optim um size ofLW ORK.

LW ORK length ofW ORK amay.LW ORK should be at least M B *LB.

Forgood penform ance, LW ORK should generally be larger. Foroptim um perform ance on $m$ ultiple processors, LW ORK $>=M B * L B * N \_C P U S$ where $N$ _CPU $S$ is the $m$ axim um num berof processors available to the program .

IfLW ORK $=0$, the routine is to allocate $w$ orkspace needed.

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no enrorm essage related to LW ORK is issued by X ERBLA.

## SEE ALSO

N IST FORTRA N Sparse B las U sers G uide available at: http://m ath nistgov/n csol/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

## NOTES/BUGS

1. N o test for singularity ornear-singularity is included in this routine. Such tests $m$ ustbe perform ed before calling this routine.
2. If $D E S C R A$ (3)=0, the low erorupper triangularpart ofeach diagonalblock is used by the routine depending on DESCRA (2).
3. If $D E S C R A$ (3)=1, the unit diagonalblocksm ightorm ight notbe referenced in the BD I representation of a sparse $m$ atrix. They are notused anyw ay.
4. If $D E S C R A$ (3)=2, diagonalblocks are considered as dense $m$ atrices and the LU factorization $w$ th partialpivoting is used by the routine.

WORK (1)=0 on retum if the factorization foralldiagonal
blocks has been com pleted successfiully, otherw ise wORK $(1)=$ -iw here $i$ is the block num ber forw hich the LU
factorization could not.be com puted.
5.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse m atrix A is used. H ow erver DESCRA (1) m ustbe equalto 3 in this case.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cbdsqr-com pute the singularvalue decom position (SVD ) of a realN -by -N (upper or low er) bidiagonalm atrix B.

## SYNOPSIS

```
SU BROUT\mathbb{NE CBD SQR (UPLO,N,NCVT,NRU,NCC,D,E,VT,LDVT,U,LDU,C,}
    LDC,W ORK,\mathbb{NFO)}
CHARACTER * 1 UPLO
COM PLEX VT (LDVT,*),U (LDU,*),C (LDC,*)
INTEGERN,NCVT,NRU,NCC,LDVT,LDU,LDC,INFO
REALD (*),E (*),W ORK (*)
SUBROUTINE CBDSQR_64 (UPLO,N,NCVT,NRU,NCC,D,E,VT,LDVT,U,LDU,
        C,LDC,WORK,\mathbb{NFO)}
```

CHARACTER * 1 UPLO
COM PLEX VT (LDVT,*), U (LDU, , $), C(\mathbb{L D} C, \star)$
$\mathbb{N} T E G E R * 8 N, N C V T, N R U, N C C, L D V T, L D U, L D C, \mathbb{N} F O$
REALD (*), E (*), WORK (*)

## F95 INTERFACE

SU BROUTINE BDSQR (UPLO, $\mathbb{N}], \mathbb{N C V T ] , ~} \mathbb{N} R U], \mathbb{N C C}], D, E, V T,[L D V T]$, U, [LD U ], C, [LD C ], [W ORK], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1) ::UPLO
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::VT,U,C
$\mathbb{N} T E G E R:: N, N C V T, N R U, N C C, L D V T, L D U, L D C, \mathbb{N F O}$
REAL,D $\mathbb{M}$ ENSION (:) ::D ,E,W ORK
SUBROUTINE BDSQR_64 (UPLO, $\mathbb{N}], \mathbb{N} C V T], \mathbb{N} R U], \mathbb{N C C}], D, E, V T,[L D V T]$, $\mathrm{U},[\mathrm{LD} \mathrm{U}], \mathrm{C},[\mathrm{LD} \mathrm{C}],[\mathrm{W}$ ORK ], [ $\mathbb{N F O}])$

CHARACTER (LEN=1) ::UPLO
COMPLEX, D $\mathbb{M} E N S I O N(:,:):: V T, U, C$
$\mathbb{N}$ TEGER (8) :: N , NCVT, NRU, NCC, LDVT, LDU, LD C , $\mathbb{N} F O$
REAL,D $\mathbb{I}$ ENSION (:) :: D ,E,W ORK

## C INTERFACE

\#include <sunperfh>
void dodsqr(charuplo, intn, intncvt, int nnu, int ncc, float*d, float*e, com plex *vt, int ldvt, com plex *u, int ldu, com plex *C, int ldc, int *info);
void dodsqr_64 (charuplo, long n, long ncvt, long nnu, long ncc, float *d, float *e, com plex *vt, long ldvt, com plex *u, long ldu, com plex *c, long ldc, long *info);

## PURPOSE

codsqr com putes the singularvahue decom position (SV D) of a realN -by-N (upper orlow er) bidiagonalm atrix $B: B=Q * S$ * $P^{\prime}\left(P^{\prime}\right.$ denotes the transpose of $\left.P\right)$, w here $S$ is a diagonal $m$ atrix $w$ ith non-negative diagonal elem ents the singular values of $B$ ), and $Q$ and $P$ are orthogonalm atrices.

The routine com putes $S$, and optionally com putes $U * Q, P^{\prime} \star$ $\mathrm{V} T$, or $Q^{\prime *} \mathrm{C}$, forgiven com plex inputm atrices $\mathrm{U}, \mathrm{V} T$, and C.

See "C om puting Sm allSingularV alues ofB idiagonalM atrioes W th G uaranteed H igh Relative A ccuracy," by J. D em m eland W . K ahan, LAPA CK W orking N ote \#3 (orSIAM J.Sci. Statist. C om put.vol.11, no.5, pp. 873-912, Sept1990) and
"A ccurate singular values and differential qd algorithm $s$, "
by B. Parlett and V.Femando, TechnicalReportCPAM -554, $M$ athem atics D epartm ent, U niversity of C alifomia at Berkeley, July 1992 for a detailed description of the algorithm .

## ARGUMENTS

```
UPLO (input)
\(=\mathrm{U}\) : B is upperbidiagonal;
\(=1 \mathrm{~L}\) ': B is low erbidiagonal.
```

N (input) The order of them atrix $\mathrm{B} . \mathrm{N}>=0$.

NCVT (input)
The num berof colum ns of the m atrix V T.NCV T >=0.

NRU (input)
The num ber of row s of the $m$ atrix $U . N R U>=0$.

NCC (input)
The num ber of colum ns of the m atrix C . N CC $>=0$.

D (input/output)
O n entry, the n diagonalelem ents of the bidiago-
nal $m$ atrix $B$. On exit, if $\mathbb{N F O}=0$, the singular
values ofB in decreasing order.
E (input/output)
On entry, the elem ents of E contain the offdiagonalelem ents of of the bidiagonalm atrix w hose SV D is desired. O norm alexit ( $\mathbb{N} \mathrm{FO}=0$ ), E is destroyed. If the algorithm does not converge ( $\mathbb{N}$ FO > 0), D and E w illcontain the diagonal and superdiagonal elem ents of a bidiagonalm atrix orthogonally equivalent to the one given as input. E N ) is used forw orkspace.

## VT (input/output)

On entry, an N-by-N CV T m atrix VT. On exit, V T is overw rilten by $\mathrm{P}^{\prime * V T . V T}$ is notreferenced if $\mathrm{NCVT}=0$.

LDVT (input)
The leading dim ension of the array VT . LDV T >= $\max (1, N)$ ifnCVT >0;LDVT >=1 ifNCVT = 0 .

U (input/output)
On entry, an NRU -by $N$ m atrix $U$. On exit, $U$ is overw ritten by $U$ * $Q$. $U$ is not referenced if $N R U$ $=0$.

LD U (input)
The leading dim ension of the array $U$. LDU >= max (1,NRU).

C (input/output)
On entry, an N -by -NCC matrix C. On exit, C is overw ritten by $Q$ '* C . C is not referenced if NCC $=0$.

LD C (input)
The leading dim ension of the array C. LD C >= $\max (1, N)$ if $N C C>0 ; L D C>=1$ if $N C C=0$.

W ORK (w orkspace)
dim ension ( $4 * N$ )

IN FO (output)
= 0: successfulexit
$<0:$ If $\mathbb{N} F O=-i$, the $i$ th argum enthad an illegalvalue
$>0$ : the algorithm did notconverge; D and E contain the elem ents of a bidiagonalm atrix w hich is orthogonally sim ilarto the input matrix B; if $\mathbb{N F O}=i$, ielem ents ofe have notconverged to zero.

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

cbelm m-block Ellpack form atm atrix-m atrix m ultiply

## SYNOPSIS

```
SUBROUTINE CBELMM(TRANSA,M B,N,KB,ALPHA,DESCRA,
* VAL,B\mathbb{NDX,BLDA,MAXBNZ,LB,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,MB,N,KB,DESCRA (5),BLDA,MAXBNZ,LB,
* LDB,LDC,LW ORK
INTEGER BINDX (BLDA,MAXBNZ)
COM PLEX ALPHA,BETA
COM PLEX VAL (LB*LB*BLDA*M AXBNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINE CBELMM_64(TRANSA,M B,N,KB,ALPHA,DESCRA,
* VAL,BINDX,BLDA,MAXBNZ,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,MB,N,KB,DESCRA (5),BLDA,MAXBNZ,LB,}
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 B\mathbb{NDX (BLDA,MAXBNZ)}}\mathbf{M}\mathrm{ (M,}
COM PLEX ALPHA,BETA
COM PLEX VAL (LB*LB*BLDA *M AXBNZ),B (LDB ,*),C (LDC,*),W ORK (LW ORK)
```


## F95 INTERFACE

```
SUBROUT\mathbb{NE BELMM (TRANSA,MB,N ],KB,ALPHA,DESCRA,VAL,B INDX,}
* BLDA,MAXBNZ,LB,B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])
\mathbb{NTEGER TRANSA,MB,KB,BLDA,MAXBNZ,LB}
```



```
COM PLEX ALPHA,BETA
COM PLEX,D IM ENSION (:) ::VAL
COM PLEX,D IM ENSION (:,:):: B,C
```

SUBROUTINEBELMM_64(TRANSA, MB, $\mathbb{N}], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D X$,

BLDA,MAXBNZ,LB,B,[LDB],BETA,C,[LDC], $\mathbb{W}$ ORK], [LW ORK])
$\mathbb{N} T E G E R * 8$ TRANSA, MB,KB,BLDA,MAXBNZ,LB
$\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{I} N(:):: \quad D E S C R A, B \mathbb{N} D X$
COMPLEX ALPHA,BETA
COMPLEX,D $\mathbb{M}$ ENSION (:) ::VAL
COM PLEX,D IM ENSION (: : : :: B,C

## DESCRIPTION

$$
C \text { <-alpha op (A) B + beta C }
$$

where A LPHA and BETA are scalar, $C$ and $B$ are dense $m$ atrioes, $A$ is a m atrix represented in block Ellpack form at and op(A) is one of

```
op(A )=A or op(A )= A' or op(A )= conjg(A').
```

( 'indicates m atrix transpose)

## ARGUMENTS

| TRANSA | A Indicates how to operate w ith the sparse m atrix |
| :---: | :---: |
|  | 0 : operate w ith m atrix |
|  | 1 : operate w ith transpose $m$ atrix |
|  | 2 : operate $w$ ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real. |
| M B | N um berofblock row s in m atrix A |
| N | $N$ um berof colum ns in m atrix C |
| KB | $N$ um ber ofblock 00 lum ns in m atrix A |
| A LPH A | Scalar param eter |
| DESCRA | () D escriptor argum ent. Five elem ent integer amay |
|  | DESCRA (1) m atrix structure |
|  | 0 : general |
|  | 1 : symm etric ( $\mathrm{A}=\mathrm{A}$ ) |
|  | $2:$ Herm itian ( $\mathrm{A}=\mathrm{CONJ}$ ( A ) ) |
|  | 3 :Triangular |
|  | 4 : Skew (Anti)-Symm etric ( $\mathrm{A}=-\mathrm{A}$ ) |
|  | 5 :D iagonal |
|  | 6 : Skew Herm itian ( $\mathrm{A}=-\mathrm{CONJ}$ ( A ) ) |
|  | D ESCRA (2) upper/low er triangular indicator |
|  | 1 : low er |
|  | 2 :upper |
|  | D ESCRA (3) m ain diagonal type |

0 : non-unit
1 : unit
DESCRA (4) A ray base NOT IM PLEM ENTED )
0 :C C++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL 0 scalar array of length LB *LB *BLDA *M AXBN Z containing $m$ atrix entries, stored 00 lum $n-m$ ajorw thin each dense block.
$B \mathbb{N} D X_{0} \quad$ tw o-dim ensional integerBLD A -by $-M A X B N Z$ aray such B IND X (i,:) consists of the block colum $n$ indices of the nonzero blocks in block row i, padded by the integer value i if the num ber of nonzero blocks is less than MAXBNZ.

BLDA leading dim ension of $\operatorname{INDX(:,:).}$

M A X BN Z max num berof nonzerosblocks per row .
LB row and colum $n$ dim ension of the dense blocks com posing VAL.

B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension ofB
BETA Scalarparam eter

C 0 rectangular array w th first dim ension LD C .
LD C leading dim ension of $C$
W ORK () scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK aray. LW ORK is not referenced in the cumentversion.

## SEE ALSO

N IST FO RTRA N Sparse B las U sers G uide available at:
http://m ath nist.gov/n csd/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)

Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

cbelsm -block Ellipack form at triangular solve

## SYNOPSIS

```
SUBROUTINE CBELSM (TRANSA,MB,N,UNITD,DV,ALPHA,DESCRA,
* VAL,BINDX,BLDA,MAXBNZ,LB,
* B,LDB,BETA, C,LDC,W ORK,LW ORK )
\(\mathbb{N} T E G E R\) TRANSA,MB,N,UNITD,DESCRA (5),BLDA,MAXBNZ,LB,
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R \quad B \mathbb{N D X}(B L D A, M A X B N Z)\)
COM PLEX ALPHA,BETA
COM PLEX DV M B*LB*LB),VAL (LB*LB*BLDA*MAXBNZ),B(LDB,*),C(LDC,*),
* WORK (LW ORK)
SUBROUTINECBELSM_64(TRANSA,MB,N,UNITD,DV,ALPHA,DESCRA,
* VAL,BINDX,BLDA,MAXBNZ,LB,
* B,LDB,BETA, C,LDC,W ORK,LW ORK )
\(\mathbb{N} T E G E R * 8\) TRANSA, M B,N,UNITD,DESCRA (5),BLDA,MAXBNZ,LB,
* LDB,LDC,LWORK
\(\mathbb{N} T E G E R * 8\) B \(\mathbb{N} D X(B L D A, M A X B N Z)\)
COMPLEX ALPHA,BETA
COM PLEX DV M B*LB*LB),VAL (LB*LB*BLDA*MAXBNZ),B(LDB,*),C(LDC,*),
* WORK (LWORK)
```


## F95 INTERFACE

SUBROUTINE BELSM (TRANSA, MB, $\mathbb{N}], \operatorname{UN} I T D, D V, A L P H A, D E S C R A, V A L, B \mathbb{N} D X$,

* BLDA,MAXBNZ,LB,B,[LDB],BETA,C,[LDC], [W ORK], [LW ORK])
$\mathbb{N} T E G E R$ TRANSA,MB,UNITD, BLDA,MAXBNZ,LB
$\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \quad D E S C R A, B \mathbb{N} D X$
COMPLEX ALPHA,BETA
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::VAL,DV
COMPLEX,D $\mathbb{M}$ ENSION (: : : :: B, C

SUBROUT $\mathbb{N} E \operatorname{BELSM}$ _64 (TRANSA, MB, $\mathbb{N}], U N T D, D V, A L P H A, D E S C R A, V A L, B \mathbb{N} D X$,

* $B L D A, M A X B N Z, L B, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])$
$\mathbb{N} T E G E R * 8$ TRANSA, $M B$, UNITD, BLDA, MAXBNZ,LB
$\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O}(:):: D E S C R A, B \mathbb{N} D X$
COMPLEX ALPHA,BETA
COM PLEX,D $\operatorname{IM}$ ENSION (:) ::VAL, DV
COM PLEX , D $\mathbb{M} E N S I O N(:,:$ : $\quad$, C


## DESCRIPTION

$$
\begin{aligned}
& C<-A L P H A \text { op (A) B + BETA } C \quad C<-A L P H A D \text { Op (A) B + BETA C } \\
& C<-A L P H A \text { Op (A) D B + BETA } C
\end{aligned}
$$

where ALPHA and BETA are scalar, $C$ and $B$ are m by $n$ dense $m$ atrices, $D$ is a block diagonalm atrix, $A$ is a unit, ornon-unit, upperor low ertriangularm atrix represented in block Elhoack form at and $o p(A)$ is one of
 (inv denotesm atrix inverse, 'indicates m atrix transpose)

## ARGUMENTS

TRANSA Indicates how to operate $w$ th the sparse $m$ atrix 0 : operate $w$ ith $m$ atrix 1 : operate $w$ th transpose $m$ atrix 2 : operate $w$ ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M B $\quad$ Num ber ofblock row $s$ in $m$ atrix $A$

N $\quad$ Num berof colum ns in matrix C

UN ITD Type of scaling:
1 : Identity matrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)

DV () A may of the length M B *LB *LB consisting of the block entries ofblock diagonalm atrix D w here each block is stored in standard colum n-m ajor form .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general

1 : symm etric ( $\mathrm{A}=\mathrm{A}$ )
2: Herm itian ( $A=\operatorname{CONJ}(A)$ )
3 :Triangular
4 : Skew (A nti)-Symm etric ( $\mathrm{A}=-\mathrm{A}$ )
5 :D iagonal
6 : Skew Herm titian ( $A=-\operatorname{CON}$ J ( $A$ ) )
N ote: For the routine, D ESCRA (1)=3 is only supported.
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 : upper
DESCRA (3) m ain diagonal type
0 : non-identity blocks on the $m$ ain diagonal
1 : identity diagonalblocks
2 : diagonalblocks are dense $m$ atrices
DESCRA (4) A ray base $\mathbb{N O T} \mathbb{M}$ PLEM ENTED )
0 : C C C+ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? $\mathbb{N} O T \mathbb{M}$ PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () scalar array of length LB *LB *BLD A *M A X BN Z containing $m$ atrix entries, stored colum $n-m$ ajorw thin each dense block.

B $\mathbb{N}$ D X () tw o-dim ensionalintegerB LD A boy-M A X BN Z array such B IND X ( $i$, : ) consists of the block colum $n$ indices of the nonzero blocks in block row i, padded by the integer value iif the num ber ofnonzero blocks is less than M A X BN Z. The block colum $n$ indioesM U ST be sorted in increasing order foreach block row.

BLDA leading dim ension ofB INDX (:,:).

M AXBNZ max num berofnonzerosblocks per row .
LB row and colum $n$ dim ension of the dense blocks com posing A.

B 0 rectangular array w ith first dim ension LD B .

LD B leading dim ension of $B$

BETA Scalarparam eter

C 0 rectangular aray w ith first dim ension LD C .

LD C leading dim ension of C

W ORK () scratch array of length LW ORK.
On exit, ifLW ORK $=-1, W$ ORK (1) retums the minim um
size ofLW ORK.

LW ORK length ofW ORK anay.LW ORK should be at least M B *LB.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on $m$ ultiple processors, LW ORK $>=\mathrm{M} \mathrm{B} * \mathrm{LB} * \mathrm{~N}$ _CPU $S$ where $\mathrm{N} \_$CPUS is the $m$ axim um num berof processors available to the program .

If LW ORK $=0$, the routine is to allocate $w$ orkspace needed.

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W O RK array, and no errorm essage related to LW ORK is issued by XERBLA .

## SEE ALSO

## N IST FO RTRA N Sparse B las U ser's G uide available at:

 http:/m ath nist.gov/m cso/Staff/K Rem ington/Espblas/"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib org/utk/papers/sparse _ps

## NOTES /BUGS

1.N o test for singularity ornear-singularity is included in this routine. Such tests m ust.be perform ed before calling this routine.
2. If $D E S C R A(3)=0$, the low er or upper triangularpart of each diagonalblock is used by the routine depending on DESCRA (2) .
3. If $D E S C R A(3)=1$, the unitdiagonalblocksm ightorm ight notbe referenced in the B EL representation of a sparse $m$ atrix. They are notused anyw ay .
4. If $D E S C R A(3)=2$, diagonalblocks are considered as dense m atrices and the LU factorization w ith partialpivoting is used by the routine. WORK (1)=0 on retum if the factorization for alldiagonalblocks has been com pleted successfully, otherw ise WORK (1) = -iw here i is the block num ber forw hich the LU factorization could notbe com puted.
5.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse
m atrix $A$ is used. H ow erver DESCRA (1) m ust.be equalto 3 in this case.

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

cbscm m -block sparse colum $n m$ atrix-m atrix $m$ ultiply

## SYNOPSIS

```
SUBROUT\mathbb{NE CBSCMM(TRANSA,M B,N,KB,ALPHA,DESCRA,}
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,MB,N,KB,DESCRA (5),LB,
* LDB,LDC,LWORK
\mathbb{NTEGER B INDX (BNNZ),BPNTRB (KB),BPNTRE (KB)}
COMPLEX ALPHA,BETA
COM PLEX VAL (LB*LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINECBSCMM_64(TRANSA,M B,N,KB,ALPHA,DESCRA,
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,MB,N,KB,DESCRA (5),LB,
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 B\mathbb{NDX (BNNZ),BPNTRB (KB),BPNTRE (KB)}}\mathbf{(K)}
COM PLEX ALPHA,BETA
COM PLEX VAL (LB*LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
```

where: $\operatorname{BNNZ}=\operatorname{BPNTRE}(\mathbb{K} B)$ BPNTRB (1)

## F95 INTERFACE

SUBROUTINE BSCMM (TRANSA, MB, $\mathbb{N}], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D X$, * BPNTRB,BPNTRE, LB, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
$\mathbb{N} T E G E R$ TRANSA, MB, KB,LB
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \quad D E S C R A, B \mathbb{N} D X, B P N T R B, B P N T R E$
COM PLEX ALPHA,BETA
COM PLEX,D $\mathbb{M}$ ENSION (:) ::VAL
COMPLEX,D $\mathbb{I M}$ ENSION (: : : :: B, C

SUBROUT $\mathbb{N} E \operatorname{BSCM} M \_64(T R A N S A, M B, \mathbb{N}], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D X$,

* BPNTRB,BPNTRE,LB,B, [LDB],BETA,C,[LDC], [WORK], [LWORK])
$\mathbb{I N T E G E R * 8 ~ T R A N S A , ~ M B , K B , L B ~}$
$\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: \quad \mathrm{DESCRA}, \mathrm{B} \mathbb{N} D \mathrm{X}, \mathrm{BPNTRB}, \mathrm{BPN} T R E$
COMPLEX ALPHA,BETA
COMPLEX, D $\mathbb{M}$ ENSION (:) ::VAL
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : B , C


## DESCRIPTION

C <-aloha op (A ) B + beta C
where A LPHA andBETA are scalar, $C$ and $B$ are dense $m$ atrices, $A$ is a m atrix represented in block sparse colum n form at and op (A ) is one of $o p(A)=A$ or $o p(A)=A^{\prime}$ or $o p(A)=\operatorname{conjg}\left(A^{\prime}\right)$.
( 'indicates m atrix transpose)

## ARGUMENTS

TRA N SA Indicates how to operate $w$ ith the sparse $m$ atrix
0 : operate $w$ ith $m$ atrix
1 : operate $w$ th transpose $m$ atrix
2 : operate $w$ th the conjugate transpose ofm atrix.
2 is equivalentto 1 ifm atrix is real.

M B $\quad$ Num ber ofblock row s in m atrix A

N $\quad \mathrm{N}$ um berof colum ns in $m$ atrix $C$

K B $\quad$ Number ofblock colum ns in m atrix A

ALPHA Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( $A=A$ )
2: Herm itian ( $\mathrm{A}=\mathrm{CONJG}(\mathrm{A})$ )
3 :Triangular
4 : Skew (A nti)-Symm etric ( $A=-A$ )
5 :D iagonal
6 : Skew Herm itian ( $A=-C O N J$ ( $A$ ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper

DESCRA (3) m ain diagonaltype
0 : non-unit
1 :unit
DESCRA (4) A ray base NOT $\mathbb{M}$ PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 :unknown
1 : no repeated indices
VAL () scalar array of length $\mathrm{LB} * \mathrm{LB} * \mathrm{BNN} Z$ consisting of the block entries stored collm n-m ajorw thin each dense block .
$B \operatorname{IND}$ X (integer array of length BNNZ consisting of the block row indioes of the block entries ofA .

BPN TRB 0 integer aray of length $K B$ such that BPN TRB (J) BPN TRB (1)+1 points to location in B IND X of the firstblock entry of the J-th block colum n of A.
BPNTRE ( integeramay of length $K B$ such that BPN TRE (J) BPN TRB (1) points to location in B IN D X of the last.block entry of the J-th block colum n of A.

LB dim ension of dense blocks com posing A.
B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension of $B$
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension ofC
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the current version.

LW ORK length ofW ORK array. LW ORK is notreferenced in the currentversion.

## SEE ALSO

N IST FO RTRA N Sparse B las U sers G uide available at:
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse .ps

## NOTES/BUGS

It is know $n$ that there exists another representation of the block sparse colum $n$ form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, $\mathbb{I A}$, containing the pointers to the beginning of each block colum $n$ in the arrays VAL and B INDX is used instead of tw o arrays BPN TRB and BPN TRE. To use the routine $w$ ith this kind ofblock sparse colum $n$ form at the follow ing calling sequence should be used

CALL SBSCMM (TRANSA, MB,N,KB,ALPHA,DESCRA, * $\quad V A L, B \mathbb{N D}, \mathbb{A}, \mathbb{A}(2), L B$, * B,LDB,BETA, C,LDC,WORK,LWORK )

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

cbscsm -block sparse colum $n$ form at triangular solve

## SYNOPSIS

```
SUBROUT\mathbb{NE CBSCSM(TRANSA,M B,N,UNITD,DV,ALPHA,DESCRA,}
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,MB,N,UNITD,DESCRA (5),LB,
* LDB,LDC,LWORK
INTEGER BINDX (BNNZ),BPNTRB MB),BPNTRE MB)
COMPLEX ALPHA,BETA
COM PLEX DV M B &LB*LB),VAL (LB*LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINE CBSCSM_64(TRANSA,M B,N,UNITD,DV,A LPHA,DESCRA,
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,MB,N,UNITD,DESCRA (5),LB,
* LDB,LDC,LW ORK
INTEGER*8 BINDX (BNNZ),BPNTRB MB),BPNTREMB)
COM PLEX ALPHA,BETA
COM PLEX DV M B*LB*LB),VAL (LB*LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
where:BNNZ = BPNTRE M B -BPNTRB (1)
```


## F95 INTERFACE

SUBROUTINEBSCSM (TRANSA, MB,N,UNTTD,DV,ALPHA,DESCRA,VAL,B $\mathbb{N} D X$, * BPNTRB,BPNTRE, LB, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
$\mathbb{N} T E G E R \quad$ TRANSA, MB,N,UNITD,LB
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: D E S C R A, B \mathbb{N} D X, B P N T R B, B P N T R E$
COMPLEX ALPHA,BETA
COMPLEX,D $\mathbb{M}$ ENSION (:) ::VAL,DV
COM PLEX,D $\operatorname{IM}$ ENSION (:, :):: B,C

SU BROUTINE BSCSM_64 (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,VAL,BINDX,

* BPNTRB,BPNTRE,LB,B, [LDB],BETA,C, [LDC], [W ORK], [LWORK])
$\mathbb{I N T E G E R * 8}$ TRANSA, MB,N,UNITD, LB
$\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: \quad \mathrm{DESCRA}, \mathrm{B} \mathbb{N} D X, B P N T R B, B P N T R E$
COMPLEX ALPHA,BETA
COM PLEX, D $\mathbb{I M}$ ENSION (:) ::VAL, DV
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : B , C


## DESCRIPTION

$$
\begin{aligned}
& C<-A L P H A \text { OP (A) B + BETA } C \quad C<-A L P H A D \text { Op }(A) B+B E T A C \\
& C<-A L P H A \text { Op }(A) D B+B E T A C
\end{aligned}
$$

where ALPHA and BETA are scalar, $C$ and $B$ are m by $n$ dense $m$ atrices, $D$ is ablock diagonalm atrix, $A$ is a unit, ornon-unit, upperor low er triangularm atrix represented in block sparse colum $n$ form at and op (A ) is one of op (A) $)=\operatorname{inv}(A)$ or op (A $)=\operatorname{inv}(A)$ or op (A) $=\operatorname{inv}\left(\operatorname{cong}\left(A^{\prime}\right)\right)$ (inv denotesm atrix inverse, 'indicates m atrix transpose)

## ARGUMENTS

TRANSA Indicates how to operate $w$ th the sparse $m$ atrix 0 : operate $w$ ith $m$ atrix 1 : operate $w$ th transpose $m$ atrix 2 : operate $w$ ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M B $\quad$ Num ber ofblock row $s$ in $m$ atrix $A$
$N \quad N$ um berof colum $n s$ in $m$ atrix $C$

UN ITD Type of scaling:
1 : Identity matrix (argum entD V [] is ignored)
2 : Scale on left (row block scaling)
3 : Scale on right (colum $n$ block scaling)
DV () A ray of the length M B *LB *LB consisting of the block entries ofblock diagonalm atrix $D$ w here each block is stored in standard colum n-m ajor form .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 : general

1 : symmetric ( $A=A$ )
2: Herm itian ( $\mathrm{A}=\mathrm{CONJG}(\mathrm{A})$ )
3 :Triangular
4 : Skew (A nti)-Symm etric ( $A=-A$ )
5 :D iagonal
6 : Skew Herm itian ( $\mathrm{A}=-\operatorname{CONJG}(\mathrm{A})$ )
N ote: For the routine, D ESCRA $(1)=3$ is only supported.
D ESCRA (2) upper/low er triangular indicator
1: low er
2 : upper
DESCRA (3) m ain diagonal type
0 : non-identity blocks on the $m$ ain diagonal
1 : identily diagonalblocks
2 : diagonalblocks are densem atrices
DESCRA (4) A ray base NOT IM PLEM ENTED )
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () scalar aray of length LB *LB *BNN Z consisting of the block entries stored colum $n$-m ajorw thin each dense block.
$B \mathbb{N}$ D X ( integer array of length BNN Z consisting of the block row indices of the block entries of $A$.
The block row indicesM U ST be sorted
in increasing order foreach block colum $n$.
BPN TRB () integer aray of length $M B$ such that
BPN TRB (J) BPN TRB (1)+1 points to location in B IN D X of the first.block entry of the J-th block colum n of A.

BPN TRE 0 integer array of length $M B$ such that BPN TRE (J) BPN TRB (1) points to location in B IND X of the lastblock entry of the $J$ th block colum n of A.

LB dim ension of dense blocks com posing A.
B 0 rectangular array w th first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C.
LD C leading dim ension of $C$

W ORK 0 scratch array of length LW ORK. On exit, if LW ORK = $-1, W$ ORK (1) retums the optim um size ofLW ORK.

LW ORK length ofW ORK aray. LW ORK should be at least M B*LB.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on $m$ ultiple processors, LW ORK $>=\mathrm{M} \mathrm{B} * \mathrm{LB}$ *N_CPUS where N_CPUS is the $m$ axim um num berof processors available to the program .

If LW ORK $=0$, the routine is to allocate $w$ orkspace needed.

If LW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

## SEE ALSO

N IST FORTRAN Sparse B las U sers G uide available at: http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
htyp://w w w netlib .org/utk/papers/sparse.ps

## NOTES/BUGS

1. No test for singularity ornear-singularity is included in this routine. Such tests $m$ ustbe perform ed before calling this routine.
2. If $D E S C R A$ ( 3 )=0, the low erorupper triangularpart of each diagonalblock is used by the routine depending on DESCRA (2).
3. If $\operatorname{DESCRA}(3)=1$, the unit diagonalblocksm ightorm ight notbe referenced in the BSC representation of a sparse $m$ atrix. They are notused anyw ay .
4. If $D E S C R A(3)=2$, diagonalblocks are considered as dense m atrices and the LU factorization w th partialpivoting is used by the routine. WORK (1)=0 on retum if the factorization for alldiagonalblocks has been com pleted
successfully, otherw ise wORK (1) = -iw here is the block
num ber forw hich the LU factorization could notbe com puted.
5. The routine can be applied for solving triangular system $s$ w hen the upper or low er triangle of the general sparse m atrix A is used. H ow erver DESCRA (1) m ustbe equal to 3 in this case.
6. It is know $n$ that there exists another representation of the block sparse colum n form at (see for exam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three anray instead of the fourused in the current im plem entation. Them ain difference is that only one array, $\mathbb{I A}$, containing the pointers to the beginning ofeach block colum $n$ in the arrays VAL and B $\mathbb{N D}$ D is used instead of tw o arrays BPN TRB and BPN TRE. To use the routine $w$ ith this kind ofblock sparse colum $n$ form at the follow ing calling sequence should be used

CALL SBSCSM (TRANSA,MB,N,UNITD,DV,ALPHA,DESCRA,

* $V A L, B \mathbb{N} D, \mathbb{A}, \mathbb{A}(2), L B$,
* B,LDB,BETA, C,LDC,WORK,LWORK)


## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

cbsmm -block sparse row form atm atrix-m atrix m ultiply

## SYNOPSIS

```
SUBROUT\mathbb{NE CBSRMM(TRANSA,M B,N,KB,ALPHA,DESCRA,}
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,MB,N,KB,DESCRA (5),LB,
* LDB,LDC,LW ORK
INTEGER B\mathbb{NDX (BNNZ),BPNTRB(MB),BPNTREMB)}
COM PLEX ALPHA,BETA
COM PLEX VAL (LB*LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINECBSRMM_64(TRANSA,M B,N,KB,ALPHA,DESCRA,
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,MB,N,KB,DESCRA (5),LB,
* LDB,LDC,LW ORK
INTEGER*8 B\mathbb{NDX (BNNZ),BPNTRB MB),BPNTREMB)}
COMPLEX ALPHA,BETA
COM PLEX VAL (LB*LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
```

where: $\operatorname{BNN} Z=B P N T R E M B)-B P N T R B(1)$

## F95 INTERFACE

SUBROUTINE BSRMM (TRANSA, MB, $\mathbb{N}], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D X$, * BPNTRB,BPNTRE, LB, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
$\mathbb{N}$ TEGER TRANSA, MB, KB, LB
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \quad D E S C R A, B \mathbb{N} D X, B P N T R B, B P N T R E$
COM PLEX ALPHA,BETA
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::VAL
COM PLEX,D $\mathbb{I M}$ ENSION (: :) :: B,C

SUBROUT INE BSRMM_64 (TRANSA, MB, $\mathbb{N}], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D X$,

* BPNTRB,BPNTRE,LB,B, [LDB],BETA,C, [LDC], [W ORK], [LWORK])
$\mathbb{I N T E G E R * 8 ~ T R A N S A , ~ M B , K B , L B ~}$
$\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: \quad \mathrm{DESCRA}, \mathrm{B} \mathbb{N} D \mathrm{X}, \mathrm{BPNTRB}, \mathrm{BPN} T R E$
COMPLEX ALPHA,BETA
COMPLEX, D $\mathbb{M}$ ENSION (:) ::VAL
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : B , C


## DESCRIPTION

C <-aloha op (A ) B + beta C
where A LPHA andBETA are scalar, $C$ and $B$ are dense $m$ atrices, A is a m atrix represented in block sparse row form at and op (A ) is one of $o p(A)=A$ or $o p(A)=A^{\prime}$ or $o p(A)=\operatorname{conjg}\left(A^{\prime}\right)$.
( 'indicates m atrix transpose)

## ARGUMENTS

TRANSA Indicates how to operate $w$ ith the sparse $m$ atrix
0 : operate $w$ ith $m$ atrix
1 : operate $w$ th transpose $m$ atrix
2 : operate $w$ th the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix $A$ is real.

M B $\quad$ Num ber ofblock row s in m atrix A
$N \quad N$ um berof colum $n s$ in $m$ atrix $C$

K B $\quad$ Number ofblock colum ns in m atrix A

ALPHA Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( $A=A$ )
2: Herm itian ( $\mathrm{A}=\mathrm{CONJG}(\mathrm{A})$ )
3 :Triangular
4 : Skew (A nti)-Symm etric ( $A=-A$ )
5 :D iagonal
6 : Skew Herm itian ( $A=-C O N J$ ( $A$ ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper

DESCRA (3) main diagonal type
0 : non-unit
1 :unit
DESCRA (4) A ray base NOT $\mathbb{M}$ PLEM ENTED)
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () scalar array of length LB *LB*BNNZ consisting of the block entries stored colum n-m ajorw thin each dense block .
$B \operatorname{IND}$ X (integer array of length BNNZ consisting of the block colum n indices of the block entries of A.

BPN TRB () integeramay of length $M B$ such that BPN TRB (J) BPN TRB (1)+1 points to location in B IND X of the first.block entry of the $J$-th block row of A.
BPN TRE () integer array of length $M B$ such that BPN TRE (J)-BPN TRB (1) points to location in B IND X of the lastblock entry of the $J$ th block row ofA.

LB dim ension of dense blocks com posing A.
B 0 rectangular array w th first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w th first dim ension LD C.
LD C leading dim ension of $C$
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the currentversion.

LW ORK length ofW ORK array. LW ORK is not referenced in the currentversion.

## SEE ALSO

N IST FO RTRA N Sparse B lasU ser's G uide available at:
htep://m ath nist.gov/m csd/Staff/k Rem ington/Aspblas/
"D ocum ent forthe B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse дps

## NOTES /BUGS

It is know $n$ that there exists another representation of the block sparse row form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem s",W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. The $m$ ain difference is that only one array, $\mathbb{A}$, containing the pointers to the beginning of each block row in the amays $V A L$ and $B \mathbb{N D X}$ is used instead of tw o arrays BPN TRB and BPN TRE. To use the routine w ith this kind ofblock sparse row form at the follow ing calling sequence should be used

CALL SBSRMM (TRANSA, MB,N,KB,ALPHA,DESCRA, * $V A L, B \mathbb{N} D X, \mathbb{I}, \mathbb{I A}(2), L B$,

* $\quad \mathrm{B}, \mathrm{LD} B, B E T A, C, L D C, W$ ORK,LWORK )


## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

cbsrsm -block sparse row form at triangular solve

## SYNOPSIS

```
SUBROUT\mathbb{NE CBSRSM(TRANSA,M B,N,UNITD,DV,ALPHA,DESCRA,}
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,MB,N,UNITD,DESCRA (5),LB,
* LDB,LDC,LW ORK
INTEGER B\mathbb{NDX (BNNZ),BPNTRB(MB),BPNTREMB)}
COM PLEX ALPHA,BETA
COM PLEX DV M B*LB*LB),VAL (LB*LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINE CBSRSM_64(TRANSA,M B,N,UNITD,DV,A LPHA,DESCRA,
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,MB,N,UNITD,DESCRA (5),LB,
* LDB,LDC,LW ORK
INTEGER*8 B INDX (BNNZ),BPNTRB MB),BPNTREMB)
COM PLEX ALPHA,BETA
COM PLEX DV M B*LB*LB),VAL (LB*LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
```

where: $\operatorname{BNN} Z=B P N T R E M B)-B P N T R B(1)$

## F95 INTERFACE

```
SUBROUT\mathbb{NE BSRSM (TRANSA,M B,N,UNITD,DV,ALPHA,DESCRA,VAL,BINDX,}
```

* BPNTRB,BPNTRE, LB, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
$\mathbb{N} T E G E R \quad$ TRANSA, MB,N,UNITD,LB
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: D E S C R A, B \mathbb{N} D X, B P N T R B, B P N T R E$
COM PLEX ALPHA,BETA
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::VAL,DV
COM PLEX,D $\mathbb{I M}$ ENSION (: :) :: B,C

SU BROUTINE BSRSM_64 (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,VAL,BINDX,

* BPNTRB,BPNTRE,LB,B, [LDB],BETA,C, [LDC], [W ORK], [LW ORK])
$\mathbb{I N T E G E R * 8}$ TRANSA, MB,N,UNITD, LB
$\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: \quad \mathrm{DESCRA}, \mathrm{B} \mathbb{N} D X, B P N T R B, B P N T R E$
COMPLEX ALPHA,BETA
COM PLEX, D $\mathbb{I M}$ ENSION (:) ::VAL, DV
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : B , C


## DESCRIPTION

$$
\begin{aligned}
& C<-A L P H A \text { OP (A) B + BETA } C \quad C<-A L P H A D \text { Op }(A) B+B E T A C \\
& C<-A L P H A \text { Op }(A) D B+B E T A C
\end{aligned}
$$

where ALPHA and BETA are scalar, $C$ and $B$ are m by $n$ dense $m$ atrices, $D$ is ablock diagonalm atrix, $A$ is a unit, ornon-unit, upperor low ertriangularm atrix represented in block sparse row form at form atand op (A) is one of op (A) $)=\operatorname{inv}(A)$ or op (A $)=\operatorname{inv}(A)$ or op (A) $=\operatorname{inv}\left(\infty n \dot{g}\left(A^{\prime}\right)\right)$ (inv denotesm atrix inverse, 'indicates m atrix transpose)

## ARGUMENTS

TRA N SA Indicates how to operate $w$ th the sparse $m$ atrix 0 : operate $w$ ith $m$ atrix 1 : operate $w$ th transpose $m$ atrix 2 : operate $w$ ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M B $\quad$ Num ber ofblock row $s$ in $m$ atrix $A$
$N \quad N$ um berof colum $n s$ in $m$ atrix $C$

UN ITD Type of scaling:
1 : Identity matrix (argum entD V [] is ignored)
2 : Scale on left (row block scaling)
3 : Scale on right (colum $n$ block scaling)

DV () A mray of the length M B *LB *LB consisting of the block entries ofblock diagonalm atrix $D$ w here each block is stored in standard colum n-m ajor form .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem ent integer array
DESCRA (1) m atrix structure
0 : general

1 : symm etric ( $\mathrm{A}=\mathrm{A}$ )
2: Herm itian ( $\mathrm{A}=\mathrm{CONJG}(\mathrm{A})$ )
3 :Triangular
4 : Skew (A nti)-Symm etric ( $A=-A$ )
5 :D iagonal
6 : Skew Herm itian ( $\mathrm{A}=-\operatorname{CONJG}(\mathrm{A})$ )
N ote: For the routine, D ESCRA $(1)=3$ is only supported.
D ESCRA (2) upper/low er triangular indicator
1: low er
2 : upper
DESCRA (3) m ain diagonal type
0 : non-identity blocks on the $m$ ain diagonal
1 : identily diagonalblocks
2 : diagonalblocks are densem atrices
DESCRA (4) A ray base NOT IM PLEM ENTED )
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () scalar aray of length LB *LB *BNN Z consisting of the block entries stored colum $n$-m ajorw thin each dense block.
$B \operatorname{IN}$ X ( integer array of length BNNZ consisting of the block colum n indices of the block entries of A. The block colum n indices M U ST be sorted in increasing order foreach block row .

BPN TRB () integer array of length $M B$ such that BPN TRB (J) BPN TRB (1)+1 points to location in B IND X of the firstblock entry of the J-th block row of A.

BPN TRE ( integer array of length $M B$ such that BPN TRE (J)-BPN TRB (1) points to location in B IND X of the last.block entry of the J-th block row of A.

LB dim ension of dense blocks com posing A.
B 0 rectangular array w th first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C.
LD C leading dim ension of $C$

W ORK 0 scratch array of length LW ORK. On exit, if LW ORK $=-1, W$ ORK (1) retums the optim um size ofLW ORK.

LW ORK length ofW ORK array. LW ORK should be at least M B*LB.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on $m$ ultiple processors, LW ORK $>=\mathrm{M} \mathrm{B} * \mathrm{LB} * \mathrm{~N}$ _CPU $S$ where $N$ _CPUS is the $m$ axim um num berof processors available to the program .

If LW ORK $=0$, the routine is to allocate $w$ orkspace needed.

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

## SEE ALSO

N IST FORTRAN Sparse B las U sers G uide available at: http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
htyp://w w w netlib .org/utk/papers/sparse.ps

## NOTES/BUGS

1. No test for singularity ornear-singularity is included in this routine. Such tests $m$ ustbe perform ed before calling this routine.
2. If $D E S C R A$ ( 3 )=0, the low erorupper triangularpart of each diagonalblock is used by the routine depending on DESCRA (2).
3. If $\operatorname{DESCRA}$ (3)=1, the unitdiagonalblocksm ightorm ight not.be referenced in the BSR representation of a sparse $m$ atrix. They are notused anyw ay .
4. If $D E S C R A(3)=2$, diagonalblocks are considered as dense m atrices and the LU factorization w th partialpivoting is used by the routine. WORK (1)=0 on retum if the factorization for alldiagonalblocks has been com pleted
successfully, otherw ise wORK (1) = -iw here is the block
num ber forw hich the LU factorization could notbe com puted.
5. The routine can be applied for solving triangular system $s$ w hen the upper or low er triangle of the general sparse m atrix A is used. H ow erver DESCRA (1) m ustbe equal to 3 in this case.
6. It is know $n$ that there exists another representation of the block sparse row form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem S", W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, $\mathbb{I A}$, containing the pointers to the beginning of each block row in the amays VAL and B $\mathbb{N D}$ D is used instead of tw o arrays BPN TRB and BPN TRE. To use the routine w ith this kind ofblock sparse row form at the follow ing calling sequence should be used

CALL SBSRSM (TRANSA,MB,N,UNITD,DV,ALPHA,DESCRA,

* $V A L, B \mathbb{N} D, \mathbb{A}, \mathbb{A}(2), L B$,
* B,LDB,BETA, C,LDC,WORK,LWORK)


## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ccnvoor-com pute the convolution or comelation of com plex vectors

## SYNOPSIS

SUBROUTINE CCNVCOR (CNVCOR,FOUR,NX,X, $\mathbb{F} X, \mathbb{N C X}, N Y, N P R E, M, Y$, $\mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N} C 2 Y, N Z, K, Z, \mathbb{F Z}, \mathbb{N} C 1 Z, \mathbb{N} C 2 Z, W$ ORK,LW ORK)

CHARACTER * 1 CNVCOR,FOUR
COM PLEX X (*), Y (*), Z (*), W ORK (*)
$\mathbb{N} T E G E R N X, \mathbb{F} X, \mathbb{N} C X, N Y, N P R E, M, \mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N} C 2 Y, N Z$, $\mathrm{K}, \mathbb{F} Z, \mathbb{N} C 1 Z, \mathbb{N} C 2 Z, L W$ ORK

SU BROUTINE CCNVCOR_64 CNVCOR,FOUR,NX,X, $\mathbb{F} X, \mathbb{N} C X, N Y, N P R E, M, Y$, $\mathbb{F Y}, \mathbb{N C} 1 \mathrm{Y}, \mathbb{N} C 2 Y, N Z, K, Z, \mathbb{F} Z, \mathbb{N C} 1 Z, \mathbb{N} C 2 Z, W$ ORK,LW ORK)

CHARACTER * 1 CNVCOR,FOUR
COM PLEXX (*), Y (*), Z (*), W ORK (*)
$\mathbb{N} T E G E R * 8 N X, \mathbb{F} X, \mathbb{N} C X, N Y, N P R E, M, \mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N} C 2 Y, N Z$,
$\mathrm{K}, \mathbb{F} Z, \mathbb{N} \mathrm{C} 1 \mathrm{Z}, \mathbb{N} \mathrm{C} 2 \mathrm{Z}, \mathrm{LW}$ ORK

## F95 INTERFACE

SU BROUTINE CNVCOR (CNVCOR,FOUR,NX,X, $\mathbb{F} X,[\mathbb{N C X}], N Y, N P R E, M, Y$, $\mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N} C 2 Y, N Z, K, Z, \mathbb{F} Z, \mathbb{N C} 1 Z, \mathbb{N C} 2 Z, W$ ORK, [LW ORK])

CHARACTER (LEN=1) ::CNVCOR,FOUR
COM PLEX,D $\mathbb{M}$ ENSION (:) :: X,Y,Z,W ORK
$\mathbb{N} T E G E R:: N X, \mathbb{F} X, \mathbb{N} C X, N Y, N P R E, M, \mathbb{F} Y, \mathbb{N C} 1 Y, \mathbb{N C} 2 Y$,
$\mathrm{NZ}, \mathrm{K}, \mathbb{F} Z, \mathbb{N} \mathrm{C} 1 \mathrm{Z}, \mathbb{I N C} 2 \mathrm{Z}, \mathrm{LW}$ ORK
SU BROUTINE CNVCOR_64 (CNVCOR,FOUR,NX,X, $\mathbb{F} X,[\mathbb{N} C X], N Y, N P R E, M$, $Y, \mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N} C 2 Y, N Z, K, Z, \mathbb{F} Z, \mathbb{N} C 1 Z, \mathbb{N} C 2 Z, W$ ORK, (LW ORK ])

CHARACTER (LEN=1) ::CNVCOR,FOUR
COM PLEX,D $\mathbb{I}$ ENSION (:) :: X,Y,Z,W ORK
$\mathbb{N} \operatorname{TEGER}(8):: N X, \mathbb{F} X, \mathbb{N} C X, N Y, N P R E, M, \mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N} C 2 Y$, NZ,K, $\mathbb{F} Z, \mathbb{N} C 1 Z, \mathbb{N} C 2 Z, L W O R K$

## C INTERFACE

\#include <sunperfh>
void ccnvcor(char cnvcor, char four, intnx, com plex *x, int
ifx, int incx, intny, intnpre, intm, com plex

* $y$, int ify, int incly, intinc2y, int $n z$, int $k$, com plex ${ }^{\star} z$, int ifz, intinclz, intinc $2 z$, com plex *W Ork, intlw ork);
void ccnvcor_64 (char cnvcor, char four, long nx, com plex *x, long ifx, long incx, long ny, long npre, long m, com plex *y, long ify, long inc1y, long inc2y, long
$n z$, long k, com plex *z, long ifz, long inclz, long inc2z, com plex *w ork, long lw ork);


## PURPOSE

convoor com putes the convolution or correlation of com plex vectors.

## ARGUMENTS

CNVCOR (input)
CHARACTER
$V$ 'or $V^{\prime}$ if convolution is desired, $R^{\prime}$ or $r^{\prime}$ if comelation is desired.

FOUR (input)
CHARACTER
T'or t'ifthe Fourier transform $m$ ethod is to be used, D 'or d'if the com putation should be done directly from the definition. The Fourier transform $m$ ethod is generally faster, but itm ay introduce noticeable errors into certain results, notably w hen both the real and im aginary parts of the filter and data vectors consist entirely of integers or vectors w here elem ents of either the filter vector or a given data vectordiffer significantly in $m$ agnitude from the 1-norm of the vector.

Length of the filtervector. $\mathrm{NX}>=0$. CCNVCOR w ill retum im m ediately if $\mathrm{NX}=0$.

X (input) dim ension (*)
Filtervector.
IFX (input)
Index of the firstelem entofX. $\mathrm{NX}>=\mathbb{F} \mathrm{X}>=1$.
$\mathbb{N C X}$ (input)
Stride betw een elem ents of the filtervector in $X$. $\mathbb{N} C X>0$.
NY (input)
Length of the inputvectors. NY >= 0. CCNVCOR w ill retum im m ediately if $\mathrm{N} \mathrm{Y}=0$.

NPRE (input)
The num ber of im plicit zeros prepended to the $Y$ vectors. NPRE $>=0$.

M (input)
Num berof inputvectors. M >= 0. CCNVCOR will retum im $m$ ediately if $M=0$.

Y (input) dim ension ( ${ }^{*}$ )
Inputvectors.
IFY (input)
Index of the firstelem entof Y . $\mathrm{N} Y>=\mathbb{F Y}>=1$.
$\mathbb{N} C 1 Y$ (input)
Stride betw een elem ents of the inputvectors in $Y$. $\mathbb{N} C 1 Y>0$.
$\mathbb{N} C 2 Y$ (input)
Stride betw een the inputvectors in $Y . \mathbb{N} C 2 Y>0$.

NZ (input)
Length of the output vectors. $\mathrm{NZ}>=0$. CCNVCOR w ill retum im mediately if $\mathrm{N}=0$. See the N otes section below for inform ation abouthow this argu$m$ ent interacts $w$ ith $N X$ and NY to control circular versus end-off shifting.

K (input)
$N$ um berof $Z$ vectors. $K>=0$. If $K=0$ then
$C C N V C O R \mathrm{w}$ ill retum immediately. If $K$ $M$ then
only the firstK inputvectors $w$ ill be processed.
If $K>M$ then $M$ inputvectors $w$ illbe processed.

Z (output)
dim ension (*)
Result vectors.
FZ (input)
Index of the firstelem entof $Z . N Z>=\mathbb{F Z}>=1$.
$\mathbb{N C 1 7}$ (input)
Stride betw een elem ents of the output vectors in Z. $\mathbb{N} C 1 Z>0$.
$\mathbb{N} C 2 Z$ (input)
Stride betw een the output vectors in $\mathrm{Z} . \mathbb{N N C}^{2 Z}>$ 0.

W ORK (input/output)
(input/scratch) dim ension (LW ORK)
Scratch space. Before the firstcall to CCNVCOR w ith particular values of the integer argum ents the firstelem entofW ORK m ustbe set to zero. If W ORK is written betw een calls to CCNVCOR or if CCNVCOR is calledw ith different values of the integer argum ents then the firstelem entofW ORK m ustagain be set to zero before each call. If W ORK has notbeen w rilten and the sam e values of the integer argum ents are used then the firstele$m$ entofW ORK to zero. This can avoid certain initializations that store their results into W ORK, and avoiding the initialization can makeCCNVCOR run faster.

LW ORK (input)
Length ofW ORK. LW ORK >= 2*M AX NX,NY+NPRE,NZ)+8.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ccnvcor2 - com pute the convolution or comelation of com plex m atrices

## SYNOPSIS

```
SUBROUTINE CCNVCOR2 (CNVCOR,METHOD,TRANSX,SCRATCHX,TRANSY,
    SCRATCHY,M X,NX,X,LDX,MY,NY,M PRE,NPRE,Y,LDY,M Z,NZ,Z,
    LD Z,W ORK,LW ORK)
CHARACTER * 1 CNVCOR,METHOD, TRANSX, SCRATCHX, TRANSY,
SCRATCHY
COM PLEX X (LDX,*),Y (LDY,*),Z (LD Z,*),W ORK (*)
\mathbb{N TEGER M X,NX,LDX,M Y,NY,M PRE,NPRE,LDY,M Z,NZ, LD Z,}
LW ORK
SUBROUTINE CCNVCOR2_64 CNVCOR,M ETHOD,TRANSX,SCRATCHX,TRANSY,
    SCRATCHY,M X,NX,X,LDX,M Y,NY,MPRE,NPRE,Y,LDY,M Z,NZ,Z,
    LD Z,W ORK,LW ORK)
```

    CHARACTER * 1 CNVCOR, METHOD, TRANSX, SCRATCHX, TRANSY,
    SCRATCHY
COM PLEX X (LDX,*), Y (LDY,*), Z (LD Z, *), W ORK (*)
$\mathbb{I N}$ TEGER*8 M X,NX,LDX,MY,NY,MPRE,NPRE,LDY,MZ,NZ,LDZ,
LW ORK

## F95 INTERFACE

SU BROUTINE CNVCOR2 (CNVCOR,METHOD,TRANSX,SCRATCHX,TRANSY, SCRATCHY, MX], $\mathbb{N} X], X,[L D X], \mathbb{M} Y], \mathbb{N} Y], M P R E, N P R E, Y,[L D Y]$, M Z ], $\mathbb{N} Z], Z,[L D Z], W$ ORK, [LW ORK ])

CHARACTER (LEN=1) :: CNVCOR, METHOD, TRANSX, SCRATCHX, TRANSY,SCRATCHY

COM PLEX,D $\mathbb{I M} E N S I O N(:):$ W ORK
COM PLEX , D $\mathbb{M}$ ENSION (: :: : : X , Y , Z
$\mathbb{N} T E G E R:: M X, N X, L D X, M Y, N Y, M P R E, N P R E, L D Y, M Z, N Z$, LD Z, LW ORK

SU BROUTINE CNVCOR2_64 CNVCOR,METHOD,TRANSX,SCRATCHX,TRANSY, SCRATCHY, $\mathbb{M} X], \mathbb{N} X], X,[L D X], \mathbb{M} Y], \mathbb{N} Y], M P R E, N P R E, Y,[L D Y]$, [M Z], [NZ], Z, [LD Z ], W ORK, [LW ORK ])

CHARACTER (LEN=1) :: CNVCOR, METHOD, TRANSX, SCRATCHX, TRANSY, SCRATCHY
COM PLEX , D $\mathbb{M}$ ENSION (:) ::W ORK
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : X , Y , Z
$\mathbb{N}$ TEGER (8) :: $M \mathrm{X}, \mathrm{NX}, \mathrm{LD} \mathrm{X}, \mathrm{M} Y, N Y, M P R E, N P R E, L D Y, M Z, N Z$,
LD Z , LW ORK

## C INTERFACE

\#include <sunperfh>
void ccnvcor2 (char cnvcor, charm ethod, char transx, char scratchx, chartransy, char scratchy, intm $x$, int $n x$, com plex * $x$, int ldx, intm $y$, intny, intm pre, int npre, complex *y, int ldy, intm $z$, intnz, com plex *z, int ldz, com plex *w ork, int lw ork);
void ccnvcor2_64 (charcnvcor, charm ethod, chartransx, char scratchx, char transy, char scratchy, long $m x$, long $n x$, com plex *x, long ldx, long m y, long ny, long m pre, long npre, com plex *y, long ldy, long m z, long nz, com plex *z, long ldz, com plex *w ork, long lw ork);

## PURPOSE

ccnvcor2 com putes the convolution or conrelation of com plex m atrices.

## ARGUMENTS

CNVCOR (input)
V 'or $\mathrm{V}^{\prime}$ to com pute convolution, $R$ 'or $I$ ' to com pute correlation.

METHOD (input)
T'or t'if the Fouriertransform $m$ ethod is to be used, D 'or d'to com pute directly from the definition.

TRANSX (input)
$N$ 'or h'ifX is the filterm atrix, $T$ ' or $t^{\prime}$
iftranspose ( $X$ ) is the filterm atrix.

SCRATCHX (input)
N 'or h'ifX m ustbe preserved, S'or s 'if X
can be used as scratch space. The contents of $X$
are undefined after retuming from a callin w hich
$X$ is allow ed to be used for scratch.

TRANSY (input)
N 'or h'ify is the inputm atrix, T'or t'if
transpose $(Y)$ is the inputm atrix .

SCRATCHY (input)
N 'or h'ifY mustbe preserved, S'or S'ifY
can be used as scratch space. The contents ofy are undefined after retuming from a callin w hich Y is allow ed to be used for scratch.

M X (input)
N um ber of row s in the filterm atrix. $\mathrm{M} \mathrm{X}>=0$.

NX (input)
N um ber of colum ns in the filterm atrix. NX $>=0$.

X (input)
O n entry, the filterm atrix. U nchanged on exitif
SCRATCHX is $N^{\prime}$ or $h$ ', undefined on exitif SCRATCHX is S'or $s^{\prime}$.

LD X (input)
Leading dim ension of the array that contains the filterm atrix.

M Y (input)
$N$ um ber of row $S$ in the inputm atrix. $\mathrm{M} \mathrm{Y}>=0$.

NY (input)
Num ber of colum ns in the inputm atrix. $\mathrm{N} Y>=0$.

M PRE (input)
$N$ um ber of im plicit zeros to prepend to each row of the inputm atrix. M PRE $>=0$.

NPRE (input)
N um ber of im plicit zeros to prepend to each colum n of the inputm atrix. NPRE $>=0$.

Y (input)

Inputm atrix. U nchanged on exit if SCRATCHY is $N^{\prime}$ or $h^{\prime}$, undefined on exitifSCRATCHY is $S^{\prime}$ or $\mathrm{S}^{\prime}$.

LD Y (input)
Leading dim ension of the array that contains the inputm atrix.
M Z (input)
N um ber of row s in the output m atrix. $\mathrm{M} \mathrm{Z} \mathrm{>=0}$. CCNVCOR2 will retum im m ediately ifM $Z=0$.

NZ (input)
N um ber of colum ns in the outputm atrix. $\mathrm{N} Z>=0$. $C C N V C O R 2 \mathrm{w}$ ill retum im m ediately if $\mathrm{Z}=0$.

Z (output)
dim ension (LD Z,*)
Resultm atrix.
LD Z (input)
Leading dim ension of the array that contains the resultm atrix. LD Z >= M AX ( $1, \mathrm{M} \mathrm{Z}$ ).

W ORK (input/output)
(input/scratch) dim ension (LW ORK )
On entry for the first call to CCNVCOR2, W ORK (1)
$m$ ust contain CM PLX $(0.0,0.0)$. A fter the first
call, w ORK (1) m ustibe set to CM PLX ( $0.0,0.0$ ) iff
W ORK has been altered since the last call to this subroutine or if the sizes of the arrays have changed.

LW ORK (input)
Length of the w ork vector. The upperbound of the w orkspace length requirem ent is 2 * M Y C + NYC) + 15, where M YC = MAX MAX MX,NX), MAX MY,NY)+NPRE)
and NYC $=M A X M A X(M X X), M A X M Y, N Y)+M P R E)$. If
LW ORK indicates a w orkspace that is too sm all, the routine will allocate its ow n w orkspace. If the
FFT is notused, the value of LW ORK is unim portant.

## Contents

- NAME
- SYNOPSIS


# - F95 INTERFACE 

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

```
ccoomm -coordinatem atrix-m atrix m ultiply
```


## SYNOPSIS

```
SUBROUT\mathbb{NECCOOMM(TRANSA,M,N,K,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,JNDX,NNZ,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,M,N,K,DESCRA (5),NNZ
* LDB,LDC,LW ORK
\mathbb{NTEGER INDX NNZ),JNDX NNZ)}
COM PLEX ALPHA,BETA
COM PLEX VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NECCOOMM_64(TRANSA,M,N,K,ALPHA,DESCRA,}
* VAL, \mathbb{NDX,JNDX,NNZ,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,M,N,K,DESCRA (5),NNZ
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 INDX NNZ),NNDX NNZ)}
COMPLEX ALPHA,BETA
COM PLEX VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
```


## F95 INTERFACE

```
SUBROUT\mathbb{NECOOMM(TRANSA,M, N ],K,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,JNDX,NNZ,B,[LDB],BETA,C,[LDC],}
* [W ORK], [LW ORK])
INTEGER TRANSA,M,K,NNZ
```



```
COMPLEX ALPHA,BETA
COM PLEX,D IM ENSION (:) ::VAL
COM PLEX,D IM ENSION (:,:):: B,C
```

SUBROUTINECOOMM_64 (TRANSA, M, $\mathbb{N}], K, A L P H A, D E S C R A$,

* $\quad V A L, \mathbb{N} D X, J N D X, N N Z, B,[L D B], B E T A, C,[L D C]$,
* [W ORK], [LW ORK])
$\mathbb{N} T E G E R * 8$ TRANSA, M, K, NNZ
$\mathbb{N} \operatorname{TEGER*}, \mathrm{D} \mathbb{M} \operatorname{ENS} \mathbb{I} N(:):: \mathrm{DESCRA}, \mathbb{N} D \mathrm{X}, \mathbb{N} D \mathrm{X}$
COMPLEX ALPHA,BETA
COMPLEX, D $\mathbb{M}$ ENSION (:) ::VAL
COM PLEX , D $\mathbb{M}$ ENSION (: :) :: B , C


## DESCRIPTION

C <-alpha op (A) B + beta C
where A LPHA andBETA are scalar, $C$ and $B$ are dense $m$ atrices, $A$ is a $m$ atrix represented in coordinate form at and op (A) is one of $o p(A)=A$ or $o p(A)=A^{\prime}$ or $o p(A)=\operatorname{conjg}\left(A^{\prime}\right)$.
( 'indicates m atrix transpose)

## ARGUMENTS

TRANSA Indicates how to operate $w$ ith the sparse $m$ atrix
0 : operate w ith m atrix
1 : operate $w$ th transpose $m$ atrix
2 : operate $w$ ith the conjugate transpose ofm atrix.
2 is equivalentto 1 ifm atrix is real.

M $\quad N$ um ber of row $s$ in $m$ atrix A

N $\quad \mathrm{N}$ um berof colum ns in $m$ atrix $C$

K $\quad N$ um berof colum $n s$ in $m$ atrix $A$

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 : general
1 : symm etric ( $A=A$ )
2: Herm Itian ( $A=C O N J(A)$ )
3 :Triangular
4 : Skew (A nti)-Symm etric ( $A=-A$ )
5 :D iagonal
6 : Skew Herm itian ( $A=-C O N J$ ( $A$ ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper

DESCRA (3) main diagonal type
0 : non-unit
1 : unit
DESCRA (4) A ray base NOT IM PLEM ENTED)
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 :no repeated indices
VAL () scalar array of length NNZ consisting of the non-zero entries of $A$, in any order.
$\mathbb{I N D X}$ () integer array of length NNZ consisting of the comesponding row indices of the entries of A.

JND X () integer amray of length NNZ consisting of the corresponding colum $n$ indioes of the entries of A.

NN Z number of non-zero elem ents in A.
B 0 rectangular array w th first dim ension LD B.
LD B leading din ension ofB

BETA Scalarparam eter
C 0 rectangular anray with firstdim ension LD C.

LD C leading dim ension of $C$
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is notreferenced in the current version.

## SEE ALSO

N IST FORTRA N Sparse B las U sers G uide available at:
http://m ath nistgov/m csd/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib.org/utk/papers/sparse.ps

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
coopy -C opy x to y
```


## SYNOPSIS



```
COM PLEX X (*),Y (*)
INTEGERN,\mathbb{NCX,INCY}
```



```
COM PLEX X (*),Y (*)
INTEGER*8N,\mathbb{NCX,INCY}
```


## F95 INTERFACE

SU BROUTINE COPY ( $\mathbb{N}], X,[\mathbb{N C X}], Y,[\mathbb{N C Y}])$
COM PLEX,D $\mathbb{I}$ ENSION (:) :: X,Y $\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y$

SU BROUTINE COPY_64 (N ],X, [ $\mathbb{N} C X], Y,[\mathbb{N} C Y])$
COM PLEX,D $\mathbb{I M}$ ENSION (:) :: X,Y
$\mathbb{N}$ TEGER (8) :: $\mathrm{N}, \mathbb{I N C X , ~} \mathbb{N} C Y$

## C INTERFACE

\#include < sunperfh>
void ccopy (intn, com plex *x, int incx, com plex *y, int incy);
void ccopy_64 (long n, com plex *x, long incx, com plex *y,

## PURPOSE

coopy C opy $x$ to $y w h e r e x$ and $y$ are $n$-vectors.

## ARGUMENTS

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N m ustbe at leastone for the subroutine to have any visible effect. U nchanged on exit.
X (input)
OfD $\mathbb{M}$ ENSION at least ( $1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C X)$
). Before entry, the increm ented array $\mathrm{X} m$ ust contain the vectorx. U nchanged on exit.
$\mathbb{N C X}$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of $\mathrm{X} . \mathbb{N} C X$ m ustnotbe zero. U nchanged on exit.

Y (output)
ofD $\mathbb{I M}$ ENSION at least ( $1+(\mathrm{m}-1) \star a b s(\mathbb{N C Y})$
). On entry, the increm ented array $Y$ m ust contain the vectory. On exit, $Y$ is overw rilten by the vectorx.
$\mathbb{N C Y}$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $Y . \mathbb{I N C Y}$ m ustnotbe zero. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS


# - F95 INTERFACE 

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

ccsam m - com pressed sparse colum $n$ form atm atrix-m atrix m ultiply

## SYNOPSIS

```
SUBROUT\mathbb{NECCSCMM(TRANSA,M,N,K,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,M,N,K,DESCRA (5),
* LDB,LDC,LWORK
INTEGER INDX NNZ),PNTRB(K),PNTRE (K)
COM PLEX ALPHA,BETA
COMPLEX VALNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NECCSCMM_64(TRANSA,M,N,K,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,M,N,K,DESCRA (5),
* LDB,LDC,LWORK
\mathbb{NTEGER*8 }\mathbb{N}DX(NNZ),PNTRB(K),PNTRE (K)
COMPLEX ALPHA,BETA
COM PLEX VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
where NN Z = PN TRE (K )PN TRB (1)
```


## F95 INTERFACE

SUBROUTINECSCMM (TRANSA, M, $\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N} D X$, * PNTRB, PNTRE, B, [LDB],BETA, C, [LDC], [WORK], [LWORK])
$\mathbb{N}$ TEGER TRANSA, M, K
$\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \quad$ ESCRA, $\mathbb{N} D X, P N T R B, P N T R E$
COMPLEX ALPHA,BETA
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::VAL

SUBROUTINE CSCMM_64(TRANSA, M, $\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N D X}$, * PNTRB,PNTRE, B, [LDB],BETA,C, [LDC], [WORK], [LW ORK])
$\mathbb{N}$ TEGER*8 TRANSA, $\mathrm{M}, \mathrm{K}$
$\mathbb{N} T E G E R * 8, D \mathbb{I} \operatorname{ENS} \mathbb{I} N(:)::$ DESCRA, $\mathbb{N} D \mathrm{X}, \operatorname{PNTRB}, \operatorname{PNTRE}$
COMPLEX ALPHA,BETA
COMPLEX,D $\mathbb{M}$ ENSION (:) ::VAL
COMPLEX,D $\mathbb{I M}$ ENSION (: : : :: B,C

## DESCRIPTION

$$
C \text { <-alpha op (A) B + beta C }
$$

where A LPHA and BETA are scalar, $C$ and $B$ are dense $m$ atrices, $A$ is a $m$ atrix represented in com pressed sparse colum $n$ form at and op(A) is one of $o p(A)=A$ or $o p(A)=A^{\prime}$ or $o p(A)=c o n j\left(A^{\prime}\right)$.
( 'indicates m atrix transpose)

## ARGUMENTS

TRANSA Indicates how to operate $w$ ith the sparse $m$ atrix
0 : operate $w$ ith $m$ atrix
1 : operate w th transpose m atrix
2 : operate $w$ ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M $\quad N$ um berof row $s$ in $m$ atrix A

N $\quad N$ um berof colum $n s$ in $m$ atrix $C$
K $\quad N$ um berof colum $n s$ in $m$ atrix A

A LPH A Scalarparam eter

DESCRA ( D escriptor argum ent. Fíve elem ent integer anay
DESCRA (1) m atrix structure
0 : general
1 : symmetric ( $A=A$ )
2 : Herm Aian ( $\mathrm{A}=\mathrm{CONJG}$ (A))
3 :Triangular
4 : Skew (A nti)-Symm etric ( $A=-A$ )
5 :D iagonal
6 : Skew Herm itian ( $\mathrm{A}=-\mathrm{CONJ}$ ( A ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er

2 :upper
DESCRA (3) main diagonal type
0 :non-unit
1 : unit
DESCRA (4) A ray base $\mathbb{N} O T \mathbb{I}$ PLEM ENTED)
0 :C C ++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM EN TED )
0 : unknown
1 : no repeated indices

VAL () scalar array of length NN Z consisting of nonzero entries ofA.

IND X $0 \quad$ integer array of length NN Z consisting of the row indices of nonzero entries ofA .

PN TRB 0 integer amray of length $K$ such thatPN TRB (J)-PN TRB (1)+1 points to location in VA L of the firstnonzero elem ent in colum n J .
PN TRE 0 integer array of length $K$ such thatPN TRE (J)-PN TRB (1) points to location in V A L of the lastnonzero elem ent in colum n J .

B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension ofB

BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C.
LD C leading dim ension of $C$

W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the currentversion.

LW ORK length ofW ORK array.LW ORK is notreferenced in the currentversion.

## SEE ALSO

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov fin csd/Staffk Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee,
http://w w w netlib .org/utk/papers/sparse .ps

## NOTES/BUGS

It is know $n$ that there exists another representation of the com pressed sparse colum n form at (see forexam ple Y Saad, "IterativeM ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, $\mathbb{I A}$, containing the pointers to the beginning of each colum $n$ in the arrays VA L and $\mathbb{N} D \mathrm{X}$ is used instead oftw o arraysPN TRB and PN TRE.To use the routine $w$ th this kind of sparse colum $n$ form at the follow ing calling sequence should be used

SUBROUTINE SCSCMM (TRANSA, M,N,K,ALPHA,DESCRA,

* $V A L, \mathbb{N} D X, \mathbb{A}, \mathbb{I}(2), B, L D B, B E T A$,
* C,LDC,WORK,LWORK)


## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

```
ccscsm -com pressed sparse colum n form at triangular solve
```


## SYNOPSIS

```
SUBROUT\mathbb{NECCSCSM(TRANSA,M ,N,UNITD,DV,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,M,N,UNITD,DESCRA (5),
* LDB,LDC,LW ORK
\mathbb{NTEGER INDX NNZ),PNTRB (M),PNTREM)}
COM PLEX ALPHA,BETA
COM PLEX DV M),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINECCSCSM_64(TRANSA,M,N,UNITD,DV,A LPHA,DESCRA,
* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,M,N,UNITD,DESCRA (5),}
* LDB,LDC,LW ORK
```



```
COM PLEX ALPHA,BETA
COM PLEX DV M),VAL NNZ),B (LDB,*),C (LDC,\star),W ORK (LW ORK)
where NN Z = PNTRE M )PNTRB (1)
```


## F95 INTERFACE

SUBROUTINECSCSM (TRANSA, M, $\mathbb{N}], U N \mathbb{T} D, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X$, * PNTRB,PNTRE, B, [LDB],BETA,C, [LDC], [W ORK], [LW ORK]) $\mathbb{I N}$ TEGER TRANSA, M, UN ITD
$\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: D E S C R A, \mathbb{N} D \mathrm{X}, \mathrm{PN} T R B, \operatorname{PNTRE}$
COM PLEX ALPHA,BETA
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::VAL,DV
COM PLEX,D $\operatorname{IM}$ ENSION (:, :):: B,C

SUBROUTINECSCSM _64 (TRANSA, M, $\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X$,

* PNTRB,PNTRE, B , [LDB],BETA,C, [LDC], [W ORK], [LW ORK])
$\mathbb{N} T E G E R * 8$ TRANSA, M , UN ITD
$\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: \operatorname{DESCRA}, \mathbb{N} D X, \operatorname{PNTRB}, \operatorname{PNTRE}$
COM PLEX ALPHA,BETA
COM PLEX,D $\operatorname{IM}$ ENSION (:) ::VAL, DV
COM PLEX , D $\mathbb{M}$ ENSION (:, :) :: B , C


## DESCRIPTION

$$
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { Op (A) B + BETA C } \\
& C<-A L P H A \text { Op (A)D B + BETA } C
\end{aligned}
$$

where ALPHA and BETA are scalar, $C$ and $B$ are $m$ by $n$ dense $m$ atrices, $D$ is a diagonalscaling $m$ atrix, $A$ is a unit, ornon-unit, upper or low ertriangularm atrix represented in com pressed sparse colum n form atand op (A ) is one of op (A) $)=\operatorname{inv}(A)$ or op (A $)=\operatorname{inv}(A)$ or op (A) $)=\operatorname{inv}\left(\operatorname{con} \dot{g}\left(A^{\prime}\right)\right)$ (inv denotesm atrix inverse, 'indicates $m$ atrix transpose)

## ARGUMENTS

TRANSA Indicates how to operate $w$ th the sparse $m$ atrix 0 : operate $w$ ith $m$ atrix
1 : operate $w$ th transpose $m$ atrix
2 : operate $w$ ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M $\quad N$ um berof row $s$ in $m$ atrix A
$N \quad N$ um berof colum $n s$ in $m$ atrix $C$

UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum $n$ scaling)
4 : A utom atic colum n scaling (see section N O TES for further details)

DV () A rray of length M containing the diagonalentries of the scaling diagonalm atrix D .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay DESCRA (1) m atrix structure

$$
\begin{aligned}
& 0 \text { : general } \\
& 1 \text { : sym m etric ( } A=A \text { ) } \\
& 2: \text { H erm itian ( } A=\operatorname{CON} \text { JG (A }) \text { ) } \\
& 3 \text { : Triangular } \\
& 4 \text { : Skew (A nti)-Sym m etric ( } A=-A \text { ) } \\
& 5 \text { : D iagonal } \\
& 6 \text { : Skew Herm itian ( } A=-\operatorname{CON} \text { JG (A ) ) }
\end{aligned}
$$

N ote: For the routine, D ESCRA (1)=3 is only supported.

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonaltype
0 : non-unit
1 :unit
DESCRA (4) A may base $\mathbb{N O T} \mathbb{M}$ PLEM ENTED)
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? $N$ OT $\mathbb{M}$ PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () scalar array of length N N Z consisting of nonzero entries ofA.
$\mathbb{N} D \mathrm{X}$ () integer array of length N N Z consisting of the row indices of nonzero entries of . (R ow indigesM UST be sorted in increasing order for each colum n).

PNTRB () integer amay of length $M$ such thatPN TRB (J) PN TRB (1)+1 points to location in VA L of the firstnonzero elem ent in colum $n \mathrm{~J}$.

PN TRE () integer array of length $M$ such thatPN TRE (J)-PN TRB (1) points to location in VA L of the lastnonzero elem ent in colum $n \mathrm{~J}$.

B 0 rectangular array w ith first dim ension LD B .

LD B leading dim ension ofB

BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch array of length LW ORK.

On exit, if LW ORK $=-1, W$ ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK array. LW ORK should be at leastM .
Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on $m$ ultiple processors, LW ORK $>=M$ *N_CPUS where N_CPUS is the m axim um num berof processors available to the program .

If LW ORK $=0$, the routine is to allocate $w$ orkspace needed.
IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

## SEE ALSO

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
htep://w w w netlib .org/utk/papers/sparse.ps

## NOTES/BUGS

1. N o test for singularity ornear-singularity is included in this routine. Such tests $m$ ust.be perform ed before calling this routine.
2. If UN ITD $=4$, the routine scales the colum ns of A such that their 2 -norm s are one. The scaling $m$ ay im prove the accuracy of the com puted solution. C orresponding entries of VA L are changed only in the particular case. On retum D V $m$ atrix stored as a vector contains the diagonalm atrix by which the colum ns have been scaled. UN ITD = 3 should be used for the next calls to the routine $w$ ith overw ritten VA L and DV.

WORK (1)=0 on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here i is the colum n num berw hich 2 -norm is exactly zero.
3. If $D E S C R A(3)=1$ and $U N$ ITD < 4, the unitdiagonalelem ents m ightorm ightnotbe referenced in the C SC representation
of a sparse $m$ atrix. They are notused anyw ay in these cases. ButifU N ITD = 4, the unitdiagonalelem ents M U ST be referenced in the CSC representation.
4.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general.sparse $m$ atrix A is used. H ow ever DESCRA (1) m ustbe equal to 3 in this case.
5. It is know $n$ that there exists another representation of the com pressed sparse colum n form at (see for exam ple Y Saad, "Iterative M ethods forSparse LinearSystem $s$ ", W PS, 1996). Its data structure consists of three anray instead of the fourused in the cumentim plem entation. Them ain difference is thatonly one amray, IA , containing the pointers to the beginning ofeach colum $n$ in the arrays VA L and $\mathbb{I N D X}$ is used instead of tw o amaysPNTRB and PN TRE.To use the routine $w$ th this kind of sparse colum $n$ form at the follow ing calling sequence should be used

SUBROUTINESCSCSM (TRANSA, M,N,UNITD,DV,ALPHA,DESCRA,

* $V A L, \mathbb{N} D X, \mathbb{A}, \mathbb{A}(2), B, L D B, B E T A$,
* $\quad$, LDC,$W$ ORK,LW ORK )


## Contents

- NAME
- SYNOPSIS

> - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

```
ccsmm -com pressed sparse row form atm atrix-m atrix m ultiply
```


## SYNOPSIS

```
SUBROUT\mathbb{NECCSRMM(TRANSA,M,N,K,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,M,N,K,DESCRA (5),
* LDB,LDC,LW ORK
\mathbb{NTEGER INDX NNZ),PNTRB (M),PNTREM)}
COM PLEX ALPHA,BETA
COM PLEX VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINE CCSRMM_64(TRANSA,M ,N,K,ALPHA,DESCRA,
* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,M,N,K,DESCRA (5),
* LDB,LDC,LW ORK
INTEGER*8 INDX (NNZ),PNTRB(M),PNTRE M)
COMPLEX ALPHA,BETA
COM PLEX VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
where NN Z = PN TRE M )PNTRB (1)
```


## F95 INTERFACE

SUBROUTINECSRMM (TRANSA, M, $\mathbb{N}], K, A L P H A, D E S C R A, V A L, ~ \mathbb{N D X}$,

* PNTRB, PNTRE, B, [LDB],BETA, C, [LDC], [WORK],[LW ORK])
$\mathbb{N}$ TEGER TRANSA, M, K
$\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: D E S C R A, \mathbb{N} D X, P N T R B, P N T R E$
COM PLEX ALPHA,BETA
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::VAL
COMPLEX,D IM ENSION (: : : :: B, C

SUBROUTINE CSRMM_64 (TRANSA, M, $\mathbb{N}], K, A L P H A, D E S C R A, V A L, ~ \mathbb{N D X}$,

* PNTRB,PNTRE,B, [LDB],BETA,C,[LDC],[WORK],[LWORK])
$\mathbb{N}$ TEGER*8 TRANSA, M, K
$\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: \operatorname{DESCRA}, \mathbb{N} D X, \operatorname{PNTRB}, \operatorname{PNTRE}$
COMPLEX ALPHA,BETA
COMPLEX, D $\mathbb{M}$ ENSION (:) ::VAL
COMPLEX,D $\mathbb{M}$ ENSION (: : : : : B , C


## DESCRIPTION

C <-aloha op (A ) B + beta C
where A LPHA andBETA are scalar, $C$ and $B$ are dense $m$ atrices, A is a m atrix represented in com pressed sparse row form at and op (A) is one of $o p(A)=A \quad$ or $o p(A)=A^{\prime}$ or op $(A)=\operatorname{conjg}\left(A^{\prime}\right)$.
( 'indicates m atrix transpose)

## ARGUMENTS

TRANSA Indicates how to operate $w$ ith the sparse $m$ atrix
0 : operate w ith m atrix
1 : operate $w$ th transpose $m$ atrix
2 : operate $w$ th the conjugate transpose ofm atrix.
2 is equivalentto 1 ifm atrix is real.

M $\quad N$ um ber of row $s$ in $m$ atrix A

N $\quad \mathrm{N}$ um berof colum ns in $m$ atrix $C$

K $\quad$ um berof colum ns in matrix A

ALPHA Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( $A=A$ )
2: Herm itian ( $\mathrm{A}=\mathrm{CONJG}(\mathrm{A})$ )
3 :Triangular
4 : Skew (A nti)-Symm etric ( $A=-A$ )
5 : D iagonal
6 : Skew Herm itian ( $A=-C O N J$ ( $A$ ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper

DESCRA (3) main diagonal type
0 : non-unit
1 : unit
DESCRA (4) A ray base NOT $\mathbb{M}$ PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 :no repeated indices
VAL ( scalar array of length NN Z consisting of nonzero entries ofA.
$\mathbb{I N D X} 0 \quad$ integer array of length NN Z consisting of the colum $n$ indioes of nonzero entries of $A$.

PN TRB () integer array of length $M$ such thatPN TRB (J) PN TRB (1)+1
points to location in VA L of the firstnonzero elem ent in row J .
PNTRE ( integerarray of length $M$ such thatPNTRE (J)-PNTRB (1) points to location in V A L of the lastnonzero elem ent in row J .

B 0 rectangular array w th first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C $0 \quad$ rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC
W ORK () scratch array of length LW ORK.W ORK is not referenced in the curment version.

LW ORK length ofW ORK aray. LW ORK is not referenced in the currentversion.

## SEE ALSO

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/n cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:

## NOTES/BUGS

It is know $n$ that there exists another representation of the com pressed sparse row form at (see forexam ple Y Saad, "Herative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three anray instead of the fourused in the currentim plem entation. Them ain difference is that only one array, $\mathbb{I A}$, containing the pointers to the beginning of each row in the arrays VA L and $\mathbb{N D} \mathrm{X}$ is used instead of tw o arrays PN TRB and PN TRE. To use the routine w th this kind of com pressed sparse row form at the follow ing calling sequence should be used

SUBROUTINESCSRMM (TRANSA, M, N, K,ALPHA,DESCRA,

* $\quad V A L, \mathbb{N} D X, \mathbb{A}, \mathbb{I A}(2), B, L D B, B E T A$,
* C,LDC,WORK,LWORK)


## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

```
ccsrsm - com pressed sparse row form at triangular solve
```


## SYNOPSIS

```
SUBROUTINECCSRSM(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,
* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,M,N,UNITD,DESCRA (5),
* LDB,LDC,LW ORK
\mathbb{NTEGER INDX NNZ),PNTRB (M),PNTREM)}
COM PLEX ALPHA,BETA
COM PLEX DV M),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINECCSRSM_64(TRANSA,M,N,UNITD,DV,A LPHA,DESCRA,
* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,M,N,UNITD,DESCRA (5),}
* LDB,LDC,LW ORK
INTEGER*8 \mathbb{NDX (NNZ),PNTRB M),PNTRE M)}
COM PLEX ALPHA,BETA
COM PLEX DV M),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
where NN Z = PNTRE M )PNTRB (1)
```


## F95 INTERFACE

SUBROUTINECSRSM (TRANSA, M, $\mathbb{N}], U N \mathbb{T} D, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X$, * PNTRB,PNTRE, B, [LDB],BETA,C, [LDC], [W ORK], [LW ORK])
$\mathbb{N}$ TEGER TRANSA, M, UN ITD
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: D E S C R A, \mathbb{N} D X, P N T R B, P N T R E$
COMPLEX ALPHA,BETA
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::VAL,DV
COM PLEX,D $\mathbb{I M}$ ENSION (: :) :: B,C

SUBROUTINECSRSM _64 (TRANSA, M, $\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X$,

* PNTRB,PNTRE,B, [LDB],BETA,C, [LDC], [W ORK], [LW ORK])
$\mathbb{N}$ TEGER*8 TRANSA, M,UNITD
$\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: \operatorname{DESCRA}, \mathbb{N} D X, \operatorname{PNTRB}, \operatorname{PNTRE}$
COM PLEX ALPHA,BETA
COM PLEX, D $\mathbb{I M}$ ENSION (:) ::VAL, DV
COMPLEX,D $\mathbb{M}$ ENSION (: : : : : B , C


## DESCRIPTION

$$
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { op (A)B+BETA C } \\
& C<-A L P H A \text { OP }(A) D B+B E T A C
\end{aligned}
$$

where ALPHA and BETA are scalar, $C$ and $B$ are $m$ by $n$ dense $m$ atrices, $D$ is a diagonalscaling $m$ atrix, $A$ is a unit, ornon-unit, upper or low ertriangularm atrix represented in com pressed sparse row form atand op (A) is one of $\operatorname{op}(A)=\operatorname{inv}(A)$ or op (A) $=\operatorname{inv}(A)$ or op (A) $)=\operatorname{inv}\left(\operatorname{conjg}\left(A^{\prime}\right)\right)$ (inv denotesm atrix inverse, 'indicates m atrix transpose)

## ARGUMENTS

TRANSA Indicates how to operate $w$ th the sparse $m$ atrix 0 : operate $w$ ith $m$ atrix
1 : operate w ith transpose $m$ atrix
2 : operate $w$ ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M $\quad N$ um berof row $s$ in $m$ atrix A
$N \quad N$ um berof colum $n s$ in $m$ atrix $C$

UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum $n$ scaling)
4 : A utom atic row scaling (see section N O TES for further details)

DV () A ray of length M containing the diagonalentries of the scaling diagonalm atrix D.

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay DESCRA (1) m atrix structure

$$
\begin{aligned}
& 0 \text { : general } \\
& 1 \text { : sym m etric ( } A=A \text { ) } \\
& 2: \text { H erm itian ( } A=\operatorname{CON} \text { JG (A }) \text { ) } \\
& 3 \text { : Triangular } \\
& 4 \text { : Skew (A nti)-Sym m etric ( } A=-A \text { ) } \\
& 5 \text { : D iagonal } \\
& 6 \text { : Skew Herm itian ( } A=-\operatorname{CON} \text { JG (A ) ) }
\end{aligned}
$$

N ote: For the routine, only DESCRA (1)=3 is supported.

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 : non-unit
1 :unit
DESCRA (4) A ray base $\mathbb{N O T} \mathbb{M}$ PLEM ENTED )
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? $\mathbb{N O T} \mathbb{I}$ PLEM ENTED)
0 : unknown
1 : no repeated indices
VAL () scalar array of length N N Z consisting of nonzero entries ofA.
$\mathbb{I N D X ( ) \quad i n t e g e r a m a y ~ o f ~ l e n g t h ~ N ~ N ~ Z ~ c o n s i s t i n g ~ o f ~ t h e ~ c o l u m ~ n ~}$ indices of nonzero entries ofA (colum n indices M U ST be sorted in increasing order for each row )

PNTRB () integer amay of length $M$ such thatPN TRB (J) PN TRB (1)+1 points to location in VA L of the firstnonzero elem ent in row J.

PN TRE () integer amay of length M such thatPNTRE (J)-PN TRB (1) points to location in VA L of the lastnonzero elem ent in row J.

B 0 rectangular array w ith first dim ension LD B .

LD B leading dim ension ofB

BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch array of length LW ORK.

On exit, ifLW ORK $=-1, W$ ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK array. LW ORK should be at leastM .
Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on $m$ ultiple processors, LW ORK $>=M$ *N_CPUS where N_CPUS is the m axim um num berof processors available to the program .

If LW ORK $=0$, the routine is to allocate $w$ orkspace needed.
IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

## SEE ALSO

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
htep://w w w netlib .org/utk/papers/sparse.ps

## NOTES/BUGS

1. N o test for singularity ornear-singularity is included in this routine. Such tests $m$ ustbe perform ed before calling this routine.
2. IfUN ITD $=4$, the routine scales the row s of $A$ such that their 2 -nom s are one. The scaling $m$ ay im prove the accuracy of the com puted solution. C orresponding entries ofV A L are changed only in the particular case. On retum D V m atrix stored as a vector contains the diagonalm atrix by which the row s have been scaled. U N ITD = 2 should be used for the next calls to the routine w ith overw ritten VAL and DV.

WORK (1)=0 on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here i is the row num berw hich 2 -norm is exactly zero.
3. If $\operatorname{DESCRA}(3)=1$ and UN ITD < 4, the unitdiagonalelem ents $m$ ightorm ightnotbe referenced in the CSR representation of a sparse m atrix. They are notused anyw ay in these cases.

ButifUN ITD $=4$, the unit diagonalelem ents M U ST be referenced in the CSR representation.
4.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse m atrix $A$ is used. H ow ever $\operatorname{DESCRA}$ (1) m ustbe equal to 3 in this case.
5. It is know $n$ that there exists another representation of the com pressed sparse row form at (see for exam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, IA , containing the pointers to the beginning ofeach row in the amaysVA L and $\mathbb{N} D \mathrm{X}$ is used instead of tw o arrays PN TRB and PN TRE. To use the routine $w$ ith this kind of com pressed sparse row form at the follow ing calling sequence should be used

SUBROUTINESCSRSM (TRANSA,M,N,UNTID,DV,ALPHA,DESCRA,

* $\quad V A L, \mathbb{N} D X, \mathbb{A}, \mathbb{I A}(2), B, L D B, B E T A, C$,
* LDC,WORK,LWORK)


## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

```
cdiam m -diagonal form atm atrix-m atrix m ultiply
```


## SYNOPSIS

```
SUBROUTINE CDIAMM(TRANSA,M,N,K,ALPHA,DESCRA,
* VAL,LDA,\mathbb{DIAG,NDIAG,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,M,N,K,DESCRA (5),LDA,NDIAG,
* LDB,LDC,LWORK
\mathbb{NTEGER IDIAG NDIAG)}
COM PLEX ALPHA,BETA
COM PLEX VAL (LDA,ND IAG),B (LD B ,*),C (LD C,*),W ORK (LW ORK)
SUBROUTINECDIAMM_64(TRANSA,M,N,K,ALPHA,DESCRA,
* VAL,LDA,\mathbb{DIAG,NDIAG,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
NNTEGER*8 TRANSA,M,N,K,DESCRA (5),LDA,NDIAG,
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 \mathbb{D IAG NDIAG)}}\mathbf{N}=(
COMPLEX ALPHA,BETA
COM PLEX VAL (LDA,NDIAG),B (LDB,*),C (LDC,*),W ORK (LWORK)
```


## F95 INTERFACE

SUBROUTINEDIAMM (TRANSA, M, $\mathbb{N}], K, A L P H A, D E S C R A, V A L,[L D A]$, * $\mathbb{D} \mathbb{I} G, N D \mathbb{A}, \mathrm{~B},[\mathrm{LDB}], \mathrm{BETA}, \mathrm{C},[\mathrm{LDC}],[\mathbb{W} O R K],[\mathrm{LW} O R K])$
$\mathbb{N} T E G E R$ TRANSA, $\mathrm{M}, \mathrm{K}, \mathrm{ND} \mathbb{I} G$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \quad D E S C R A, \mathbb{D} \mathbb{A} G$
COMPLEX ALPHA,BETA
COM PLEX,D $\mathbb{M}$ ENSION (:, :) :: VAL,B,C
SUBROUTINEDIAMM_64 (TRANSA, M, $\mathbb{N}], K, A L P H A, D E S C R A, V A L,[L D A]$, * $\mathbb{D} \mathbb{I} G, N D \mathbb{A} G, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])$

## DESCRIPTION

C <-alpha op (A) B + beta C
where A LPH A and BETA are scalar, $C$ and $B$ are dense $m$ atrices, $A$ is a $m$ atrix represented in diagonal form at and op (A ) is one of

$$
o p(A)=A \quad \text { or } \operatorname{op}(A)=A^{\prime} \text { or op }(A)=\operatorname{conjg}\left(A^{\prime}\right) .
$$

( 'indicatesm atrix transpose)
TRANSA Indicates how to operate $w$ ith the sparse $m$ atrix
0 : operate w ith m atrix
1 : operate $w$ ith transpose $m$ atrix
2 : operate w ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M $\quad N$ um berof row $s$ in $m$ atrix A
N $\quad$ um berof colum ns in $m$ atrix $C$
K $\quad$ Num berof colum ns in $m$ atrix A

A LPH A Scalarparam eter
DESCRA () D escriptor argum ent. Five elem ent integer aray
0 :general
1 : symm etric ( $\mathrm{A}=\mathrm{A}$ )
2 : Herm itian ( $\mathrm{A}=\mathrm{CONJG}$ (A))
3 :Triangular
4 : Skew (A nti)-Symm etric ( $A=-A$ )
5 :D iagonal
6 : Skew Herm itian ( $\mathrm{A}=-\mathrm{CONJ}$ ( A ))
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) main diagonal type
0 :non-unit
1 : unit
DESCRA (4) A ray base $\mathbb{N} O T \mathbb{I}$ PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
D ESCRA (5) repeated indices? NOT IM PLEM EN TED )
0 :unknown
1 : no repeated indices

VAL ( ) tw o-dim ensionalLD A boy ND IA G array such thatV A L (:I) consists of non-zero elem ents on diagonal ID IA G (I) of A. D iagonals in the low er triangularpart of A are padded from the top, and those in the upper triangularpart are padded from the bottom.

LDA leading dim ension ofVAL,m ustbe GE.M $\mathbb{N} \mathbb{M}, K$ )
ID IA G () integer array of length ND IA G consisting of the comesponding diagonal offsets of the non-zero diagonals ofA in VA L. Low ertriangular diagonals have negative offsets, them ain diagonal has offset 0 , and upper triangular diagonals have positive offset.

ND IA G num berof non-zero diagonals in A.
B 0 rectangular array with first dim ension LD B .
LD B leading dim ension ofB

BETA Scalarparam eter
C 0 rectangular array w th first dim ension LD C .

LD C leading dim ension ofC
W ORK () scratch array of length LW ORK.W ORK is not referenced in the current version.

LW ORK length ofW ORK array. LW ORK is not referenced in the current version.

## SEE ALSO

N IST FO RTRAN Sparse B las U ser's G uide available at:
http://m ath nistgov/m csd/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib.org/utk/papers/sparse.ps

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

```
cdiasm -diagonal form at triangular solve
```


## SYNOPSIS

```
SUBROUT\mathbb{NECDIASM(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,}
* VAL,LDA,\mathbb{D IAG,NDIAG,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,M,N,UNITD,DESCRA (5),LDA,ND IAG,}
* LDB,LDC,LW ORK
\mathbb{NTEGER IDIAG NDIAG)}
COM PLEX ALPHA,BETA
COM PLEX DV M ),VAL (LDA,ND IAG),B (LDB,*),C (LDC ,*),W ORK (LW ORK)
SUBROUT\mathbb{NECDIASM_64(TRANSA,M ,N,UNITD,DV,A LPHA,DESCRA,}
* VAL,LDA,\mathbb{D IAG,NDIAG,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,M,N,UNITD,DESCRA (5),LDA,NDIAG,}
* LDB,LDC,LW ORK
INTEGER*8 \mathbb{D IAG NDIAG)}
COM PLEX ALPHA,BETA
COM PLEX DV M ),VAL (LDA,NDIAG),B (LDB,*),C (LDC,\star),W ORK (LW ORK)
```


## F95 INTERFACE

```
SUBROUTINEDIASM (TRANSA,M, NN,UNITD,DV,ALPHA,DESCRA,VAL,
* [LDA],\mathbb{D IAG,NDIAG,B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])}
INTEGER TRANSA,M,ND IAG
INTEGER,D IM ENSION (:) :: DESCRA,\mathbb{D IAG}
COM PLEX ALPHA,BETA
COM PLEX,D IM ENSION (:) :: DV
COM PLEX,D IM ENSION (:,:) :: VAL,B,C
```

SUBROUTINEDIASM_64 (TRANSA, M, N ],UNITD,DV,ALPHA,DESCRA,VAL,

* [LDA], $\mathbb{D} \mathbb{I A G}, N D \mathbb{I A G}, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])$
$\mathbb{N} T E G E R * 8$ TRANSA, M,NDIAG
$\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{D} N(:):: \quad$ DESCRA, $\mathbb{D} \mathbb{I} G$
COMPLEX ALPHA,BETA
COM PLEX,D $\mathbb{I M}$ ENSION (:) :: DV
COM PLEX,D $\mathbb{M}$ ENSION (:, :) :: VAL,B,C


## DESCRIPTION

$$
\begin{aligned}
& C<-A L P H A \text { op(A)B + BETA } C \quad C<-A L P H A D \text { op(A)B+BETA C } \\
& C<-A L P H A \text { op(A)D B + BETA } C
\end{aligned}
$$

where A LPHA and BETA are scalar, $C$ and $B$ are $m$ by $n$ dense $m$ atrices, $D$ is a diagonalscaling $m$ atrix, $A$ is a unit, ornon-unit, upper or low ertriangularm atrix represented in diagonal form at and op (A ) is one of
$\mathrm{op}(\mathrm{A})=\operatorname{inv}(\mathrm{A}) \operatorname{or} \operatorname{op}(\mathrm{A})=\operatorname{inv}(\mathrm{A}) \operatorname{or} \operatorname{op}(\mathrm{A})=\operatorname{inv}\left(\operatorname{conjg}\left(\mathrm{A}^{\prime}\right)\right)$ (inv denotesm atrix inverse, 'indicatesm atrix transpose)

## ARGUMENTS

TRANSA Indicates how to operate $w$ ith the sparse $m$ atrix
0 : operate $w$ ith $m$ atrix
1 : operate w th transpose m atrix
2 : operate $w$ ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M $\quad N$ um berof row $s$ in $m$ atrix A

N $\quad N$ um berof colum ns in $m$ atrix $C$

UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 :Scale on left (row scaling)
3 : Scale on right (colum $n$ scaling)
4 :A utom atic row scaling (see section NOTES for furtherdetails)

DV 0 A ray of length M containing the diagonalentries of the scaling diagonalm atrix D .

A LPH A Scalarparam eter
DESCRA 0 D escriptor argum ent. Five elem ent integer amay
DESCRA (1) m atrix structure
0 : general

1 : symm etric ( $\mathrm{A}=\mathrm{A}$ )
2 : Herm itian ( $A=C O N J(A)$ )
3 :Triangular
4 : Skew (A nti)-Symm etric ( $(A=-A$ )
5 :D iagonal
6 : Skew Herm titian ( $A=-\operatorname{CON}$ J ( $A$ ) )
N ote: For the routine, only D ESCRA (1)=3 is supported.

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 :non-unit
1 :unit
DESCRA (4) A ray base $\mathbb{N O T} \mathbb{M}$ PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? (NOT $\mathbb{M}$ PLEM ENTED)
0 : unknown
1 : no repeated indices
VAL () tw o-dim ensionalLD A boy-ND IA G aray such thatVAL(:,I) consists of non-zero elem ents on diagonal ID IA G (I) of A. D iagonals in the low er triangularpart of A are padded from the top, and those in the upper triangularpart are padded from the bottom.

LDA leading dim ension ofVAL, m ustbe GE.M $\mathbb{N}(M, K)$

ID IA G () integer anay of length ND IA G consisting of the corresponding diagonaloffsets of the non-zero diagonals ofA in VAL. Low ertriangular diagonals have negative offsets, them ain diagonalhas offset 0 , and uppertriangular diagonals have positive offset. Elem ents of $\mathbb{D}$ IA G ofM UST be sorted in increasing order.

ND IA G num berofnon-zero diagonals in A.

B 0 rectangular array w ith firstdim ension LD B .

LD B leading dim ension of $B$

BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch amay of length LW ORK. On exit, if LW ORK = -1,W ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK aray. LW ORK should be at leastM.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on $m$ ultiple processors, LW ORK $>=M$ *N_CPU $S$ where $N$ _CPU $S$ is the $m$ axim um num berof processors available to the program .

IfLW ORK $=0$, the routine is to allocate $w$ orkspace needed.
IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK anray, retums this value as the firstentry of theW ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

## SEE ALSO

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov fn csd/Staff/k Rem ington/tspoblas/
"D ocum ent for the Basic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib.org/utk/papers/sparse.ps

## NOTES/BUGS

1. No test for singularity ornear-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.
2. If U N ITD $=4$, the routine scales the row sofA such that their 2 -norm sare one. The scaling $m$ ay im prove the accuracy of the com puted solution. C orresponding entries ofV A L are changed only in the particular case. On retum DV m atrix stored as a vector contains the diagonalm atrix by w hich the row s have been scaled. U N ITD $=2$ should be used for the next calls to the routine w ith overw ritten VA L and DV .

WORK ( 1 )=0 on retum if the scaling has been com pleted successfully, otherw ise $W O R K(1)=-i w$ here $i$ is the row num berw hich 2 -norm is exactly zero.
3. If $D E S C R A(3)=1$ and $U N$ ITD $<4$, the unitdiagonalelem ents m ightorm ightnotbe referenced in the D IA representation of a sparse m atrix. They are notused anyw ay in these cases. ButifUN ITD = 4, the unit diagonalelem ents M U ST be referenced in the $D \mathbb{I A}$ representation.
4.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse
m atrix $A$ is used. H ow ever DESCRA (1) m ustbe equal to 3 in this case.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cdotc -com pute the dotproduct oftw o vectors con $\dot{g}(x)$ and y .

## SYNOPSIS

COM PLEX FUNCTION CDOTC $\mathbb{N}, \mathrm{X}, \mathbb{N} C X, Y, \mathbb{N C Y})$
COM PLEX X (*), Y (*)
$\mathbb{N} T E G E R N, \mathbb{N} C X, \mathbb{N} C Y$

COM PLEX FUNCTION CDOTC_64 $\mathbb{N}, \mathrm{X}, \mathbb{N} C X, Y, \mathbb{N C Y})$

COM PLEX X (*), Y (*)
$\mathbb{N} T E G E R * 8 N, \mathbb{N} C X, \mathbb{N} C Y$

## F95 INTERFACE

COM PLEX FUNCTION DOTC ( $\mathbb{N}], X,[\mathbb{N} C X], Y,[\mathbb{N C Y}])$
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::X,Y
$\mathbb{N} T E G E R:: N, \mathbb{N C X}, \mathbb{N} C Y$
COM PLEX FUNCTION DOTC_64 ( $\mathbb{N}], X,[\mathbb{N C X}], Y,[\mathbb{N C Y}]$ )
COMPLEX,D $\mathbb{I}$ ENSION (:) ::X,Y
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{I N C X}, \mathbb{N} C Y$

## C INTERFACE

\#include <sunperfh>
com plex cdotc (intn, com plex *x, int incx, com plex *y, int incy);

## PURPOSE

codotc com pute the dot product of con $g(x)$ and $y$ where $x$ and y are n -vectors.

## ARGUMENTS

N (input)
O n entry, N specifies the num ber of elem ents in the vector. IfN is notpositive then the function retums the value 0.0. U nchanged on exit.
X (input)
OfD $\mathbb{M}$ ENSION at least ( $1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C X)$
). On entry, the increm ented amay $X$ m ust contain the vectorx. U nchanged on exit.
$\mathbb{N C X}$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of $\mathrm{X} . \mathbb{N} C X$ m ustnotbe zero. U nchanged on exit.

Y (input)
OfD $\mathbb{M}$ ENSION at least ( $1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C Y)$
). On entry, the increm ented amay $Y$ m ust contain the vectory. U nchanged on exit.
$\mathbb{N C Y}$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $Y . \mathbb{N} C Y$ m ustnotbe zero. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS


## NAME

cdotci-C om pute the com plex conjugated indexed dotproduct.

## SYNOPSIS

COM PLEX FUNCTION CDOTCINZ,X, $\mathbb{N} D X, Y)$
COM PLEX X (*), Y (*)
$\mathbb{N}$ TEGER NZ
$\mathbb{N}$ TEGER $\mathbb{I N D X ( * )}$
COM PLEX FUNCTION CDOTCI_64 $\mathbb{N} Z, \mathrm{X}, \mathbb{I N D X , Y )}$
COM PLEX X (*), Y (*)
$\mathbb{N}$ TEGER*8NZ
$\mathbb{N}$ TEGER*8 $\mathbb{N} D \mathrm{X}$ (*)
F95 $\mathbb{N}$ TERFACE
COM PLEX FUNCTION DOTCI(NZ],X, $\mathbb{N} D X, Y)$
COMPLEX,D $\mathbb{I}$ ENSION (:) ::X,Y
$\mathbb{N}$ TEGER ::NZ
$\mathbb{N}$ TEGER,D $\mathbb{I M}$ ENSION (:) :: $\mathbb{N} D \mathrm{X}$
COM PLEX FUNCTION DOTCI_64 (NZ],X, $\mathbb{N} D X, Y$ )
COMPLEX,D $\mathbb{M}$ ENSION (:) ::X,Y
$\mathbb{N}$ TEGER (8) ::NZ
$\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{N D X}$

## PURPOSE

CD O TC IC om pute the com plex conjugated indexed dotproduct of a com plex sparse vectorx stored in com pressed form w ith a
com plex vectory in fullstorage form .

```
dot \(=0\)
do \(i=1, n\)
        dot \(=\operatorname{dot}+\operatorname{con}\) gg \((x\) (i)) * \(y\) (indx (i))
    enddo
```


## ARGUMENTS

N Z (input)
$N$ um ber of elem ents in the com pressed form .
U nchanged on exit.
$X$ (input)
V ector in com pressed form . U nchanged on exit.
$\mathbb{N} D X$ (input)
$V$ ector containing the indioes of the com pressed form. It is assum ed that the elem ents in $\mathbb{N} D \mathrm{X}$ are distinctand greater than zero. U nchanged on exit.

Y (input)
V ector in fullstorage form. O nly the elem ents comesponding to the indices in $\mathbb{N}$ D X w illbe accessed.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cdotu - com pute the dotproductoftw o vectors x and y .

## SYNOPSIS

```
COM PLEX FUNCTION CDOTU N,X,\mathbb{NCX,Y,\mathbb{NCY)}}\mathbf{N}\mathrm{ ( }
```

COM PLEX X (*), Y (*)
$\mathbb{N} T E G E R N, \mathbb{N} C X, \mathbb{N} C Y$

COM PLEX FUNCTION CDOTU_64 $\mathbb{N}, \mathrm{X}, \mathbb{N} C X, Y, \mathbb{N} C Y)$
COM PLEX X (*), Y (*)
$\mathbb{N} T E G E R * 8 N, \mathbb{N} C X, \mathbb{N} C Y$

## F95 INTERFACE

COM PLEX FUNCTION DOT ( $\mathbb{N}], X,[\mathbb{N C X}], Y,[\mathbb{N C Y}])$
COM PLEX,D $\mathbb{I}$ ENSION (:) :: X,Y
$\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N C Y}$
COM PLEX FUNCTION DOT_64 (N ],X, [ $\mathbb{N} C X], Y,[\mathbb{N} C Y])$

COM PLEX,D $\mathbb{I M}$ ENSION (:) ::X,Y
$\mathbb{N}$ TEGER (8) :: $\mathrm{N}, \mathbb{I N C X , ~} \mathbb{N} C Y$
C INTERFACE
\#include <sunperfh>
com plex cdotu (intn, com plex *x, int incx, com plex *y, int incy);
com plex cdotu_64 (long n, com plex *x, long incx, com plex *y,

## PURPOSE

codotu com pute the dotproduct of $x$ and $y w h e r e x$ and $y$ are n-vectors.

## ARGUMENTS

N (input)
O n entry, N specifies the num ber of elem ents in the vector. If N is notpositive then the function retums the value 0.0. U nchanged on exit.
X (input)
ofD $\mathbb{I M}$ ENSION at least ( $1+(\mathrm{n}-1) * a b s(\mathbb{N} C X)$
). On entry, the increm ented array $X$ m ust contain the vectorx. U nchanged on exit.
$\mathbb{N C X}$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of $\mathrm{X} . \mathbb{I N C X}$ m ustnotbe zero. U nchanged on exit.

Y (input)
ofD $\mathbb{I M}$ ENSION at least ( $1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C Y)$
). On entry, the increm ented amay $Y \mathrm{~m}$ ust contain the vectory. U nchanged on exit.
$\mathbb{N C Y}$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $Y . \mathbb{N} C Y$ m ustnotbe zero. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS


## NAME

cdotui-C om pute the com plex unconjugated indexed dot product.

## SYNOPSIS

```
COM PLEX FUNCTION CDOTCINZ,X,NNDX,Y)
COM PLEX X (*),Y (*)
INTEGER NZ
INTEGER \mathbb{NDX (*)}
COM PLEX FUNCTION CDOTCI_64 NZ,X,INDX,Y)
COM PLEX X (*),Y (*)
INTEGER*8NZ
INTEGER*8 \mathbb{NDX (*)}
F95 IN TERFACE
COM PLEX FUNCTION DOTCI(NZ ],X,NNDX,Y)
COM PLEX,D IM ENSION (:) ::X,Y
\mathbb{NTEGER ::NZ}
NNTEGER,D IM ENSION (:) ::\mathbb{NDX}
COM PLEX FUNCTION DOTCI_64(NZ],X,INDX,Y)
COM PLEX,D IM ENSION (:) ::X,Y
INTEGER (8)::N Z
\mathbb{NTEGER (8),D IM ENSION (:)::\mathbb{NDX}}\mathbf{~}=\mp@code{N}
```


## PURPOSE

of a com plex sparse vectorx stored in com pressed form w ith a com plex vectory in fullstorage form .

```
dot=0
do i= 1,n
    dot= dot+x (i) * y (indx (i))
enddo
```


## ARGUMENTS

NZ (input)
N um ber of elem ents in the com pressed form .
U nchanged on exit.

X (input)
V ector in com pressed form. U nchanged on exit.
$\mathbb{I N D X}$ (input)
V ector containing the indioes of the com pressed
form. It is assum ed that the elem ents in $\mathbb{N} D X$ are
distinctand greater than zero. U nchanged on exit.
$Y$ (input)
V ector in fullstorage form. O nly the elem ents corresponding to the indices in $\mathbb{I N D ~ X ~ w ~ i l l b e ~}$ accessed.

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

cellm m -Ellpack form atm atrix-m atrix m ultiply

## SYNOPSIS

```
SUBROUTINE CELLMM (TRANSA,M,N,K,ALPHA,DESCRA,
* VAL,INDX,LDA,MAXNZ,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,M,N,K,DESCRA (5),LDA,MAXNZ,
* LDB,LDC,LWORK
INTEGER INDX (LDA,MAXNZ)
COM PLEX ALPHA,BETA
COM PLEX VAL (LDA,MAXNZ),B (LDB,\star),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NE CELLMM_64(TRANSA,M ,N,K,ALPHA,DESCRA,}
* VAL, \mathbb{NDX,LDA,MAXNZ,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,M,N,K,DESCRA (5),LDA,MAXNZ,}
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 }\mathbb{NDX (LDA,MAXNZ)}
COM PLEX ALPHA,BETA
COM PLEX VAL(LDA,MAXNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
```


## F95 INTERFACE

```
SUBROUT\mathbb{NE ELLMM (TRANSA,M, N ],K,ALPHA,DESCRA,VAL, INDX,}
* [LDA],MAXNZ,B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])
\mathbb{NTEGER TRANSA,M,K,MAXNZ}
INTEGER,D\mathbb{M ENSION (:) :: DESCRA}
INTEGER,D\mathbb{M ENSION (:,:):: \mathbb{NDX}}\mathbf{N}=\mp@code{N}
COM PLEX ALPHA,BETA
COM PLEX,D IM ENSION (:,:) :: VAL,B,C
```

SUBROUTINE ELLMM _64(TRANSA, M, $\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N} D X$,

## DESCRIPTION

where A LPH A and BETA are scalar, $C$ and $B$ are dense $m$ atrioes, A is a m atrix represented in Ellpack form at form at and op (A) is one of
$o p(A)=A$ or $o p(A)=A^{\prime}$ or $o p(A)=\operatorname{conj}\left(A^{\prime}\right)$.
( 'indicates m atrix transpose)

## ARGUMENTS

TRANSA Indicates how to operate $w$ ith the sparse $m$ atrix
0 : operate w ith m atrix
1 : operate $w$ ith transpose $m$ atrix
2 : operate w ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M $\quad$ um berof row $s$ in $m$ atrix A

N $\quad$ Num berof colum ns in $m$ atrix $C$

K $\quad N$ um berof $c o l u m n s$ in $m$ atrix A

A LPH A Scalarparam eter
DESCRA () D escriptor argum ent. Five elem ent integer array
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( $A=A$ )
2 : Herm itian ( $\mathrm{A}=\mathrm{CONJG}(\mathrm{A})$ )
3 :Triangular
4 : Skew (Anti)-Symm etric ( $A=-A$ )
5 :D iagonal
6 : Skew Herm itian ( $\mathrm{A}=-\mathrm{CONJ}$ ( A ))
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonaltype
0 : non-unit
1 :unit

DESCRA (4) A ray base NOT IM PLEM ENTED)
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? (NOT IM PLEM ENTED)
0 : unknown
1 : no repeated indices
VAL ( 0 tw o-dim ensionallD A -by M A XNZ array such thatVA L ( $\mathrm{I}, \mathrm{:}$ ) consists of non-zero elem ents in row IofA, padded by zero values if the row contains less than M AXN Z .
$\mathbb{I N D X} 0 \quad$ tw o-dim ensional integer LD A by -M A XN Z aray such $\mathbb{N} D \mathrm{X}$ ( $I$, :) consists of the colum n indices of the nonzero elem ents in row $I$, padded by the integer value I if the num berof nonzeros is less than M AXNZ.

LD A leading dim ension ofVAL and $\mathbb{N D} X$.

MAXNZ max num berofnonzeros elem ents per row .
B 0 rectangular array w th first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w th first dim ension LD C .

LD C leading dim ension of $C$
W ORK ( scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK anay. LW ORK is not referenced in the currentversion.

## SEE ALSO

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

```
cellsm -Ellpack form attriangular solve
```


## SYNOPSIS

```
SUBROUT\mathbb{NE CELLSM(TRANSA,M ,N,UN ITD,DV,ALPHA,DESCRA,}
* VAL,INDX,LDA,MAXNZ,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,M,N,UNITD,DESCRA (5),LDA,MAXNZ,
* LDB,LDC,LWORK
INTEGER INDX(LDA,MAXNZ)
COMPLEX ALPHA,BETA
COM PLEX DV M),VAL (LDA,MAXNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NE CELLSM_64(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,}
* VAL, \mathbb{NDX,LDA,MAXNZ,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,M,N,UNITD,DESCRA (5),LDA,MAXNZ,
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 INDX (LDA,MAXNZ)}
COM PLEX ALPHA,BETA
COM PLEX DV M),VAL (LDA,MAXNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
```


## F95 INTERFACE

SUBROUTINE ELLSM (TRANSA, M, $\mathbb{N}], U N \mathbb{I T D , D V , A L P H A , D E S C R A , V A L , ~}$ * $\mathbb{N} D X,[L D A], M A X N Z, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])$
$\mathbb{I N T E G E R}$ TRANSA, M, MAXNZ
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: DESCRA
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (: : : : : $\mathbb{N} D X$
COMPLEX ALPHA,BETA
COM PLEX,D $\mathbb{I M}$ ENSION (:) :: DV
COMPLEX,D $\mathbb{M} \operatorname{ENSION(:,~:)::VAL,B,C}$

SUBROUTINE ELLSM _64(TRANSA, M, $\mathbb{N}], U N \mathbb{T} D, D V, A L P H A, D E S C R A, V A L$, * $\mathbb{N} D \mathrm{X},[\mathrm{LDA}], \mathrm{MAXNZ}, \mathrm{B},[\mathrm{LDB}], B E T A, \mathrm{C},[\mathrm{LDC}],[\mathbb{O}$ ORK], [LW ORK])
$\mathbb{I N T E G E R * 8 ~ T R A N S A , M , M A X N Z ~}$
$\mathbb{N} T E G E R * 8, D \mathbb{M}$ ENSION (:) :: DESCRA
$\mathbb{N} T E G E R * 8, D \mathbb{M}$ ENSION (: : : : : $\mathbb{N} D X$
COMPLEX ALPHA,BETA
COM PLEX,D $\mathbb{M}$ ENSION (:) :: DV
COMPLEX,D $\mathbb{M}$ ENSION (: : : :: VAL,B,C

## DESCRIPTION

$$
\begin{aligned}
& C<-A L P H A \text { op (A)B + BETA } C \quad C<-A L P H A D \text { op (A)B + BETA C } \\
& C<-A L P H A \text { op } A) D \text { B + BETA } C
\end{aligned}
$$

where A LPHA and BETA are scalar, C and B are m by n dense $m$ atrices, $D$ is a diagonalscaling $m$ atrix, $A$ is a unit, ornon-unit, upper or low ertriangularm atrix represented in Ellpack form at and op (A) is one of $\mathrm{op}(\mathrm{A})=\operatorname{inv}(\mathrm{A})$ or $\mathrm{op}(\mathrm{A})=\operatorname{inv}\left(\mathrm{A}^{\prime}\right)$ or $\mathrm{op}(\mathrm{A})=\operatorname{inv}\left(\mathrm{conjg}\left(\mathrm{A}^{\prime}\right)\right)$ (inv denotesm atrix inverse, 'indicatesm atrix transpose)

## ARGUMENTS

TRANSA Indicates how to operate $w$ ith the sparse $m$ atrix 0 : operate with m atrix 1 : operate $w$ th transpose m atrix
2 : operate $w$ ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M $\quad N$ um berof row $s$ in $m$ atrix A
$\mathrm{N} \quad \mathrm{N}$ um berof colum ns in $m$ atrix C

UN ITD Type of scaling:
1 : Identity $m$ atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)
4 :A utom atic row scaling (see section N O TES for furtherdetails)

DV 0 A ray of length M containing the diagonalentries of the scaling diagonalm atrix D .

A LPH A Scalarparam eter
DESCRA 0 D escriptor argum ent. Five elem ent integer aray DESCRA (1) m atrix structure

```
    0 :general
    1 : symm etric ( }A=A\mathrm{ ) )
    2:H erm Itian (A = CON JG (A ))
    3:Triangular
    4 :Skew (A nti)-Symm etric (A=-A )
    5 :D iagonal
    6:Skew Herm titian (A= CON JG (A ) )
    N ote:For the routine, only D ESCRA (1)=3 is supported.
    D ESCRA (2) upper/low er triangular indicator
        1 : low er
        2 :upper
    DESCRA (3) m ain diagonaltype
        0:non-unit
        1 :unit
    DESCRA (4) A ray base NOT IM PLEM ENTED )
        0 : C C ++ com patible
        1 :Fortran com patible
    DESCRA (5) repeated indices? (NOT IM PLEM ENTED)
        0 :unknow n
        1: no repeated indices
VAL () tw o-dim ensionalLD A foy M A X N Z array such thatV A L (I,:)
    consists of non-zero elem ents in row IofA, padded by
    zero values if the row contains less than M AXN Z .
INDX () tw o-dim ensionalintegerLD A boy-M A XN Z array such
    \mathbb{N D X (I,:) consists of the colum n indiges of the}
    nonzero elem ents in row I, padded by the integer
    value I if the num berofnonzeros is less than M A XN Z .
    The colum n indices M U ST be sorted in increasing order
    foreach row .
    LDA leading dim ension ofV A L and \mathbb{ND X .}
    M A X N Z m ax num ber ofnonzeros elem ents per row .
    B 0 rectangular array w ith first dim ension LD B .
    LD B leading dim ension ofB
    BETA Scalarparam eter
    C 0 rectangular array w ith first dim ension LD C .
    LD C leading dim ension ofC
```

    W ORK () scratch amay of length LW ORK.
        On exit, ifLW ORK \(=-1, W\) ORK (1) retums the optim um LW ORK.
    LW ORK length ofW ORK array. LW ORK should be at leastM .
Forgood penform ance, LW ORK should generally be larger. Foroptim um perform ance on $m$ ultiple processors, LW ORK $>=\mathrm{M}$ *N_CPUS where N_CPUS is the m axim um num berof processors available to the program .

IfLW ORK $=0$, the routine is to allocate $w$ orkspace needed.
IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of theW ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

## SEE ALSO

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
htep://w w w netlib .org/utk/papers/sparse.ps

## NOTES/BUGS

1. N o test for singularity ornear-singularity is included in this routine. Such tests $m$ ustbe perform ed before calling this routine.
2. IfUN ITD $=4$, the routine scales the row s of $A$ such that their 2 -nom s are one. The scaling $m$ ay im prove the accuracy of the com puted solution. C orresponding entries ofV A L are changed only in the particular case. On retum DV m atrix stored as a vector contains the diagonalm atrix by w hich the row s have been scaled. U N ITD = 2 should be used for the next calls to the routine w ith overw ritten VAL and DV.

WORK ( 1 )=0 on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here i is the row num berw hich 2 -norm is exactly zero.
3. If $\operatorname{DESCRA}(3)=1$ and U N ITD < 4, the unitdiagonalelem ents $m$ ightorm ightnotbe referenced in the ELL representation of a sparse m atrix. They are notused anyw ay in these cases.

ButifUN ITD $=4$, the unit diagonalelem ents M U ST be referenced in the ELL representation.
4.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse m atrix A is used. H ow ever DESCRA (1) m ustbe equal to 3 in this case.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

cfft2b - com pute a periodic sequence from its Fourier coefficients. The xFFT operations are unnorm alized, so a call ofxFFT2F followed by a callof xFFT2B will multiply the input sequence by $\mathrm{M} * \mathrm{~N}$.

## SYNOPSIS

SU BROUTINE CFFT2B $M, N, A, L D A, W$ ORK, LW ORK)
COM PLEX A (LDA,*)
$\mathbb{N}$ TEGERM,N,LDA,LW ORK
REALWORK (*)
SU BROUTINE CFFT2B_64 M,N,A,LDA,W ORK,LW ORK)

COM PLEX A (LDA,*)
$\mathbb{I N T E G E R * 8 M , N , L D A , L W O R K}$
REALWORK (*)

## F95 INTERFACE

SUBROUTINE FFT2B (M ], $\mathbb{N}], A,[L D A], W$ ORK,LW ORK)
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{I N} T E G E R:: M, N, L D A, L W$ ORK
REAL,D $\mathbb{I M}$ ENSION (:) ::W ORK
SU BROUTINE FFT2B_64 (M ], $\mathbb{N}], A,[L D A], W$ ORK,LW ORK)

COM PLEX,D $\mathbb{I}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER (8) ::M , N, LDA,LW ORK
REAL,D IM ENSION (:) ::W ORK
\#include <sunperfh>
void cffl2b (intm , intn, com plex *a, int lda, float *w ork, intlw ork);
void cffi2b_64 (long m, long n, com plex *a, long lda, float *W ork, long lw ork);

## ARGUMENTS

$M$ (input) N um ber of row s to be transform ed. These subroutines are $m$ ost efficientw hen $M$ is a product of sm allprim es. $\mathrm{M}>=0$.
N (input) N um ber of colum ns to be transform ed. These subroutines are $m$ ostefficientw hen $N$ is a product of sm allprim es. $\mathrm{N}>=0$.

A (input/output)
O n entry, a tw o-dim ensional array $A(M, N)$ thatcontains the sequences to be transform ed.

LD A (input)
Leading dim ension of the array containing the data to be transform ed. LD A >=M.

W ORK (input)
On input, w orkspace W ORK m usthave been initialized by CFFT2I.

LW ORK (input)
The dim ension of the array W ORK. LW ORK >= (4* M $+\mathrm{N})+30$ )

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

cfft2f-com pute the Fourier coefficients of a periodic sequence. The xFFT operations are unnorm alized, so a call of xFFT2F follow ed by a callof xFFT2B w ill multiply the input sequence by M * N .

## SYNOPSIS

```
SUBROUT\mathbb{NE CFFT2F M,N,A,LDA,W ORK,LW ORK)}
```

COM PLEX A (LDA,*)
$\mathbb{N}$ TEGERM,N,LDA,LW ORK
REALWORK (*)
SU BROUTINE CFFT2F_64 M, N,A,LDA,W ORK,LW ORK)

COM PLEX A (LDA,*)
$\mathbb{I N T E G E R * 8 M , N , L D A , L W O R K}$
REALWORK (*)

## F95 INTERFACE

SU BROUTINE FFT2F ( $\mathbb{M}$ ], $\mathbb{N}], A,[L D A], W$ ORK, LW ORK)
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{I N} T E G E R:: M, N, L D A, L W$ ORK
REAL,D $\mathbb{I M}$ ENSION (:) ::W ORK
SU BROUTINE FFT2F_64 (M ], N ],A, [LDA ],W ORK,LW ORK)

COM PLEX,D $\mathbb{I}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER (8) ::M , N, LDA,LW ORK
REAL,D IM ENSION (:) ::W ORK
\#include <sunperfh>
void cfft2f(intm, intn, com plex *a, intlda, float *w ork, intlw ork);
void cfft2f_64 (long m, long n, com plex *a, long lda, float *W Ork, long lw ork);

## ARGUMENTS

$M$ (input) N um ber of row s to be transform ed. These subroutines are most efficientw hen M is a product of sm allprim es. $\mathrm{M}>=0$.
N (input) N um ber of colum ns to be transform ed. These subroutines are $m$ ostefficientw hen N is a product of sm allprim es. $\mathrm{N}>=0$.

A (input/output)
O n entry, a tw o-dim ensionalaray A $M, N$ ) thatcontains the sequences to be transform ed.

LD A (input)
Leading dim ension of the array containing the data to be transform ed. LD A >=M.

W ORK (input)
On input, w orkspace W ORK m usthave been initialjzed by CFFT2I.

LW ORK (input)
The dim ension of the anray W ORK. LW ORK >= (4 * M $+\mathrm{N})+30$ )

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

cfftei-initialize the array W SAVE, which is used in both the forw ard and backw ard transform s.

## SYNOPSIS

```
SU BROUTINE CFFT2IM,N,W ORK)
```

$\mathbb{N}$ TEGER $\mathrm{M}, \mathrm{N}$
REALW ORK (*)
SU BROUTINE CFFT2I_64 (M,N,WORK)
$\mathbb{N}$ TEGER*8 M , N
REALW ORK (*)

F95 INTERFACE
SUBROUTINE CFFT2IM ,N,W ORK)
$\mathbb{N} T E G E R:: M, N$
REAL,D IM ENSION (:) ::W ORK

SU BROUTINE CFFT2I_64 (M,N W ORK)
$\mathbb{N} T E G E R(8):: M, N$
REAL,D $\mathbb{I M}$ ENSION (:) ::W ORK

## C INTERFACE

\#include <sunperfh>
void cfflei(intm , intn, float*W ork);
void cfft2i_64 (long m, long n, float *w ork);

## ARGUMENTS

M (input) N um ber of row s to be transform ed. $\mathrm{M}>=0$.

N (input) N um ber of colum ns to be transform ed. $\mathrm{N}>=0$.

W ORK (input/output)
O n entry, an array ofdim ension ( $4 * M+N$ ) + 30) or greater. CFFT2Ineeds to be called only once to initialize array $W$ ORK before calling CFFT2F and/or CFFT2B if $\mathrm{M}, \mathrm{N}$ andW ORK rem ain unchanged betw een these calls. Thus, subsequent transform s or inverse transform $s$ of sam e size can be obtained faster than the firstsince they do not require initialization of the w orkspace.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

Cffl3b - com pute a periodic sequence from its Fourier coefficients. The FFT operations are unnorm alized, so a callof CFFT 3F follow ed by a callof CFFT 3B w illm ultiply the input sequence by $M{ }^{*} N *$ K .

## SYNOPSIS

```
SUBROUTINE CFFT3B M,N,K,A,LDA,LD 2A,W ORK,LW ORK)
COM PLEX A (LDA,LD 2A,*)
INTEGERM,N,K,LDA,LD 2A,LW ORK
REALWORK (*)
SUBROUT\mathbb{NE CFFT3B_64 M,N,K,A,LDA,LD 2A,W ORK,LW ORK)}
COM PLEX A (LDA ,LD 2A,*)
INTEGER*8M,N,K,LDA,LD 2A,LW ORK
REALW ORK (*)
F95 INTERFACE
    SUBROUT\mathbb{NE FFT3B (M ], N ], [K],A,[LDA ],LD 2A ,W ORK,LW ORK)}
    COM PLEX,D IM ENSION (:r:%) ::A
    INTEGER ::M,N,K,LDA,LD 2A,LW ORK
    REAL,DIM ENSION (:) ::W ORK
```



```
    COM PLEX,D IM ENSION (:r:%)::A
    INTEGER (8)::M ,N,K,LDA,LD 2A,LW ORK
    REAL,DIM ENSION (:) ::W ORK
```

void cffl3b (intm , intn, intk, com plex *a, int lda, int ld2a, float *w ork, int lw ork);
void cffl3b_64 (long m, long n, long k, com plex *a, long lda, long ld2a, float *w ork, long lw ork);

## ARGUMENTS

M (input) $N$ um ber of row s to be transform ed. These subroutines are most efficientw hen $M$ is a productof sm allprim es. M >=0.
N (input) N um ber of C lum ns to be transform ed. These subroutines are $m$ ostefficientw hen $N$ is a product of sm allprim es. $\mathrm{N}>=0$.
$K$ (input) $N$ um ber of planes to be transform ed. These subroutines are m ost efficientw hen K is a product of sm allprim es. $K>=0$.

A (input/output)
On entry, a three-dim ensionalamay A (LD A , LD 2A , K ) that contains the sequences to be transform ed.

LD A (input)
Leading dim ension of the amay containing the data to be transform ed. LD A >=M.

LD 2A (input)
Second dim ension of the array containing the data to be transform ed. LD 2A >=N .

W ORK (input)
On input, w orkspace W ORK musthave been initialized by CFFT3I.

LW ORK (input)
The dim ension of the array $W$ ORK. LW ORK >= $(4 * M+$ $\mathrm{N}+\mathrm{K})+45$ ).

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

cfft3f-com pute the Fourier coefficients of a periodic sequence. The FFT operations are unnorm alized, so a callof CFFT 3F follow ed by a callof CFFT3B w illm ultiply the input sequence by $M * N * K$.

## SYNOPSIS

```
SU BROUTINE CFFT3F M ,N,K,A,LDA,LD 2A,W ORK,LW ORK)
COM PLEX A (LDA,LD 2A,*)
INTEGERM,N,K,LDA,LD 2A,LW ORK
REALWORK (*)
SUBROUT\mathbb{NE CFFT3F_64M,N,K,A,LDA,LD 2A,W ORK,LW ORK)}
COM PLEX A (LDA,LD 2A,*)
\mathbb{NTEGER*8M,N,K,LDA,LD 2A,LW ORK}
REALW ORK (*)
F95 INTERFACE
    SU BROUT\mathbb{NE FFT3F (M ], N ], [K],A, [LDA ],LD 2A,W ORK,LW ORK )}
    COM PLEX,D IM ENSION (:,:%) ::A
    INTEGER ::M,N,K,LDA,LD 2A,LW ORK
    REAL,DIM ENSION (:) ::W ORK
    SUBROUT\mathbb{NE FFT3F_64(\mathbb{M ], N ], [K ],A,[LDA ],LD 2A,W ORK,LW ORK)}}\mathbf{~}\mathrm{ (L)}
    COM PLEX,D IM ENSION (:r:%)::A
    INTEGER (8)::M ,N,K,LDA,LD 2A,LW ORK
    REAL,DIM ENSION (:) ::W ORK
```

\#include <sunperfh>
void cffl3f(intm, intn, intk, com plex *a, int lda, int ld2a, float *w ork, int lw ork);
void cffi3f_64 (long m, long n, long k, com plex *a, long lda, long ld2a, float *w ork, long lw ork);

## ARGUMENTS

M (input) N um ber of row s to be transform ed. These subroutines are $m$ ost efficientw hen $M$ is a product of sm allprim es. $\mathrm{M}>=0$.
N (input) N um ber of colum ns to be transform ed. These subroutines are m ostefficientw hen N is a product of sm allprim es. $\mathrm{N}>=0$.
$K$ (input) $N$ um ber ofplanes to be transform ed. These subroutines are most efficientw hen $K$ is a product of sm allprim es. K >=0.

A (input/output)
O n entry, a three-dim ensional array $A(M, N, K)$ that contains the sequences to be transform ed.

LDA (input)
Leading dim ension of the amay containing the data to be transform ed. LD A >=M.

LD 2A (input)
Second dim ension of the array containing the data to be transform ed. LD 2A >=N .

W ORK (input)
On input, w orkspace W ORK m usthave been initialjzed by CFFT3I.

LW ORK (input)
The dim ension of the aray W ORK. LW ORK >= (4* M + $\mathrm{N}+\mathrm{K})+45$ )

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

cfflizi-initialize the array W SAVE, which is used in both CFFT 3F and CFFT3B.

## SYNOPSIS

SU BROUTINE CFFT3IM, N, K, W ORK)
$\mathbb{N}$ TEGER $\mathrm{M}, \mathrm{N}$, K
REALW ORK (*)
SU BROUTINE CFFT3I_64 M,N,K,W ORK)
$\mathbb{N}$ TEGER*8 M , N , K
REALW ORK (*)
F95 INTERFACE
SU BROUTINE CFFT3IM, $N$, K, W ORK)
$\mathbb{N} T E G E R:: M, N, K$
REAL,D IM ENSION (:) ::W ORK

SU BROUTINE CFFT3I_64M,N,K,WORK)
$\mathbb{N} T E G E R(8):: M, N, K$
REAL,D $\mathbb{I M}$ ENSION (:) ::W ORK
C INTERFACE
\#include <sunperfh>
void cfflili(intm, intn, int k, float *w ork);
void cfft3i_64 (long m, long n, long k, float *w ork);

## ARGUMENTS

M (input) N um ber of row s to be transform ed. $\mathrm{M}>=0$.

N (input) N um ber of colum ns to be transform ed. $\mathrm{N}>=0$.

K (input) N um ber of planes to be transform ed. $\mathrm{K}>=0$.

W ORK (input/output)
O n entry, an array ofdim ension ( $4 *(M+N+K$ ) +
45) or greater. CFFT3Ineeds to be called only once to initialize array $W$ ORK before calling CFFT3F and/or CFFT 3B ifM, N, K andW ORK rem ain unchanged betw een these calls. Thus, subsequent transform s or inverse transform s of sam e size can be obtained fasterthan the first since they do not require initialization of the w orkspace.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

cfftb -com pute a periodic sequence from its Fourier coefficients. The FFT operations are unnorm alized, so a call of CFFTF follow ed by a callofCFFTB w ill multiply the input sequence by N .

## SYNOPSIS

```
SUBROUTINE CFFTB N,X,W SAVE)
COM PLEX X (*)
\mathbb{NTEGERN}
REALW SAVE (*)
SUBROUT\mathbb{NE CFFTB_64 N,X,W SAVE)}
COM PLEX X (*)
\mathbb{NTEGER*8N}
REALW SAVE (*)
```

F95 INTERFACE
SU BROUTINE FFTB ( $\mathbb{N}$ ],X,W SAVE)
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::X
$\mathbb{N} T E G E R:: N$
REAL,D $\mathbb{M}$ ENSION (:) ::W SAVE
SU BROUTINE FFTB_64 ( $\mathbb{N}$ ],X,W SAVE)
COMPLEX,D $\mathbb{I}$ ENSION (:) ::X
$\mathbb{N} T E G E R(8):: N$
REAL,D $\mathbb{M}$ ENSION (:) ::W SAVE
\#include <sunperfh>
void cfftb (intn, com plex *x, float *w save);
void cfftb_64 (long n, com plex *x, float *w save);

## ARGUMENTS

N (input) Length of the sequence to be transform ed. These subroutines are $m$ ostefficientw hen $N$ is a product of sm allprim es. $\mathrm{N}>=0$.

X (input) On entry, an amay of length N containing the sequence to be transform ed.

W SAVE (input/output)
O n entry, W SAVE m ust.be an array ofdim ension (4* $\mathrm{N}+15$ ) orgreater and m usthave been initialized by CFFTI.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO


## NAME

cfftc - initialize the trigonom etric weight and factor tables or com pute the FastFouriertransform (forw ard or inverse) of a com plex sequence.

## SYNOPSIS


$\mathbb{N}$ TEGER $\mathbb{I O P T}, \mathrm{N}, \mathbb{F} A C(*)$, LW ORK, $\mathbb{E R R}$
COM PLEX X (*), Y (*)
REALSCALE,TRIGS (*),W ORK (*)

SU BROUTINE CFFTC_64 (TOPT,N,SCALE,X,Y,TRIGS, FAC,W ORK,LWORK,ERR)
$\mathbb{N} T E G E R * 8 \mathbb{I} P T, N, \mathbb{F A C}(*), L W O R K, \mathbb{E R R}$
REAL SCALE,TRIGS (*),WORK (*)
COM PLEXX (*), Y (*)

## F95 INTERFACE

SU BROUTINE FFT (TOPT, $\mathbb{N}$ ], [SCALE],X,Y,TRIGS, $\mathbb{F A C}, W$ ORK, [LW ORK ], $\mathbb{E R R}$ )
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT $(\mathbb{N}):: \mathbb{I O P T}$
$\mathbb{N} T E G E R * 4, \mathbb{N} T E N T(\mathbb{N})$,OPTIONAL ::N,LW ORK
REAL, $\mathbb{N} T E N T(\mathbb{N})$, OPTIONAL::SCALE
COM PLEX, $\mathbb{N}$ TENT ( $\mathbb{N}$ ), D $\mathbb{I M}$ ENSION (:) ::X
COM PLEX, $\mathbb{N}$ TENT (OUT),D $\mathbb{M}$ ENSION (:) ::Y
REAL, $\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{I M}$ ENSION (:) ::TRIGS
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT ( $\mathbb{N} O U T$ ), D $\mathbb{M}$ ENSION (:) :: $\mathbb{F A C}$
REAL, $\mathbb{I N T E N T}$ (OUT),D $\mathbb{M}$ ENSION (:) ::W ORK
$\mathbb{N} T E G E R * 4, \mathbb{I N}$ TENT (OUT) :: $\mathbb{E R R}$

SU BROUTINE FFT_64 (IOPT, $\mathbb{N}]$, [SCALE ],X,Y,TRIGS, $\mathbb{F A C}, \mathrm{W}$ ORK, [LW ORK ], $\mathbb{E R R}$ )
$\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N}):: \operatorname{IOPT}$
$\mathbb{N}$ TEGER (8), $\mathbb{N}$ TENT ( $\mathbb{N}$ ), OPTIONAL ::N,LWORK
REAL, $\mathbb{N} T E N T(\mathbb{N})$, OPTIONAL :: SCALE
COM PLEX, $\mathbb{N}$ TENT ( $\mathbb{N}$ ), D $\mathbb{M} \operatorname{ENSION(:):~:X~}$
COM PLEX, $\mathbb{I N T E N T}(O U T), D \mathbb{M} E N S \mathbb{O N}(:):: Y$
REAL, $\mathbb{N}$ TENT ( $\mathbb{N} O U T$ ), D $\mathbb{M}$ ENSION (:) ::TRIGS
$\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N O U T}), \mathrm{D} \mathbb{M} \operatorname{ENSION(:):~\mathbb {FAC}}$
REAL, $\mathbb{N}$ TENT (OUT),D $\mathbb{M} E N S I O N(:):: W O R K$
$\mathbb{N}$ TEGER (8), $\mathbb{N}$ TENT (OUT) :: $\mathbb{E R R}$

## C INTERFACE

\#include <sunperfh>
void cfftc_ (int*iopt, int*n, float *scale, com plex *x, com plex *y, float *trigs, int*ifac, float *w ork, int *lw ork, int *ienr);
void cfftc_64_ (long *iopt, long *n, float *scale, com plex *x, complex *y, float*trigs, long *ifac, float *W Ork, long *lw ork, long *ien);

## PURPOSE

cfftc initializes the trigonom etric w eight and factortables or com putes the FastFouriertransform (forw ard or inverse) of a com plex sequence as follow s:

## N-1

$Y(k)=$ scale * SUM W *X ( $)$
$\dot{F}$
where
k ranges from 0 to $\mathrm{N}-1$
$i=\operatorname{sqnt}(-1)$
isign $=1$ for inverse transform or -1 for forw ard transform
$W=\exp \left(i s i g n * i^{\star} j^{\star} k \star 2 \star \mathrm{piN}\right)$

## ARGUMENTS

IO PT (input)
Integer specifying the operation to be perform ed:
IO PT $=0$ com putes the trigonom etric $w$ eight table
and factortable
IO PT $=-1$ com putes forw ard FFT
IO PT $=+1$ com putes inverse FFT
$N$ (input)
Integer specifying length of the input sequence $X$.
N is mostefficientw hen it is a productof sm all prim es. $\mathrm{N}>=0$. U nchanged on exit.

SCALE (input)
Real scalarby w hich transform results are scaled.
U nchanged on exit. SCA LE is defaulted to 1.0 for F95 $\mathbb{I N}$ TERFACE.

X (input) On entry, X is a complex array of dimension at least N that contains the sequence to be transform ed.

Y (output)
C om plex array of dim ension at least N that contains the transform results. $X$ and $Y m$ ay be the sam e array starting at the sam e $m$ em ory location. O therw ise, it is assum ed that there is no overlap betw een $X$ and $Y$ in $m$ em ory.

TRIGS (input/output)
Realarray of length $2 * \mathrm{~N}$ that contains the trigonom etric w eights. The w eights are com puted w hen the routine is called w ith IO PT = 0 and they are used in subsequent calls when IO PT $=1$ orIOPT $=$ -1. U nchanged on exit.

IFAC (input/output)
Integer array ofdim ension at least 128 that contains the factors of N . The factors are com puted when the routine is called w ith IO PT = 0 and they are used in subsequent calls w here $10 P T=1$ or IO PT $=-1$. U nchanged on exit.

W ORK (w orkspace)
Realarray of dim ension at least $2 \star \mathrm{~N}$. The user can also choose to have the routine allocate its own w orkspace (see LW ORK).

LW ORK (input)
Integer specifying w orkspace size. If LW O RK = 0, the routine w illallocate its ow n w orkspace.
$\mathbb{E R R}$ (output)
On exit, integer $\mathbb{E} R R$ has one of the follow ing
values:
0 = norm alretum
$-1=10$ PT is not 0,1 or -1
$-2=\mathrm{N}<0$
$-3=(L W O R K$ is not 0 ) and (LW ORK is less than $2 \star \mathrm{~N}$ )
$-4=m$ em ory allocation forw orkspace failed

## SEE ALSO

ffl

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO
- CAUTIONS


## NAME

cfftc2 -initialize the trigonom etric weight and factor
tables or com pute the tw o-dim ensionalFastFourier $T$ ransform
(forw ard or inverse) of a tw o-dim ensionalcom plex amay.

## SYNOPSIS

SU BROUTINE CFFTC2 (IOPT,N1,N2,SCALE,X,LDX,Y,LDY,TRIGS, FFAC,W ORK, LW ORK, $\mathbb{E R R}$ )
$\mathbb{N}$ TEGER $\mathbb{I O P T}, N 1, N 2, L D X, L D Y, \mathbb{F A C}(*), L W$ ORK, $\mathbb{E R R}$
COM PLEX X (LDX,*), Y (LDY,*)
REAL SCALE,TRIGS (*), W ORK (*)

SU BROUTINE CFFTC2_64 (IOPT,N1,N2,SCALE,X,LDX,Y,LDY,TRIGS, FFAC,W ORK, LW ORK, $\mathbb{E R R}$ )
$\mathbb{N} T E G E R * 8 \mathbb{I} P T, N 1, N 2, L D X, L D Y, \mathbb{F A C}(*), L W$ ORK, $\mathbb{E R R}$
REAL SCALE,TRIGS (*), W ORK (*)
COM PLEX X (LDX,*), Y (LDY,*)

## F95 INTERFACE

SU BROUTINE FFT2 (IO PT, $\mathbb{N} 1], \mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S$, FAC,W ORK, [LW ORK], ERR)
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT $(\mathbb{N})::$ IO PT
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT $(\mathbb{N})$, OPTIONAL ::N $1, N 2, L D X, L D Y, L W$ ORK
REAL, $\mathbb{N}$ TENT $(\mathbb{N})$, OPTIONAL ::SCALE
COM PLEX, $\mathbb{N}$ TENT ( $\mathbb{N}$ ), D $\mathbb{M}$ ENSION (:,:) ::X
COM PLEX, $\mathbb{N}$ TENT (OUT), D $\mathbb{M}$ ENSION (:,:) ::Y
REAL, $\mathbb{N}$ TENT ( $\mathbb{N O U T}$ ), D $\mathbb{M}$ ENSION (:) ::TRIGS
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT ( $\mathbb{N} O U T$ ), D $\mathbb{I M}$ ENSION (:) :: $\mathbb{F A C}$

REAL, $\mathbb{N}$ TENT (OUT),D $\mathbb{M}$ ENSION (:) ::W ORK
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT (OUT) :: $\mathbb{E R R}$
SU BROUTINE FFT2_64 (DPPT, $\mathbb{N} 1], \mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S$, تAC,W ORK, [LW ORK], ERR)
$\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N}):: \mathbb{I O P T}$
$\mathbb{N}$ TEGER (8), $\mathbb{N}$ TENT $(\mathbb{N})$, OPTIONAL ::N 1,N2,LDX,LDY,LW ORK
REAL, $\mathbb{N}$ TENT $(\mathbb{N})$, OPTIONAL :: SCALE
COMPLEX, $\mathbb{N}$ TENT ( $\mathbb{N}$ ), D $\mathbb{M}$ ENSION (: : : : : X
COM PLEX, $\mathbb{N}$ TENT (OUT),D $\mathbb{M}$ ENSION (:,:) ::Y
REAL, $\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{I M}$ ENSION (:) ::TRIGS
$\mathbb{N} T E G E R(8), \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M}$ ENSION (:) :: $\mathbb{F} A C$
REAL, $\mathbb{N}$ TENT (OUT),D $\mathbb{M}$ ENSION (:) ::W ORK
$\mathbb{N}$ TEGER (8), $\mathbb{N}$ TENT (OUT) :: $\mathbb{E R R}$

## C INTERFACE

\#include < sunperfh>
void cfftc2_ (int*iopt, int *n1, int *n2, float *scale, com plex *x, int*ldx, com plex *y, int *ldy, float
*trigs, int*ifac, float*w ork, int *lw ork, int
*ienr);
void cfftc2_64_ (long *iopt, long *n1, long *n2, float *scale, com plex *x, long *ldx, com plex *y, long *ldy, float *trigs, long *ifac, float *w ork, long *lw ork, long *ient);

## PURPOSE

cfftc2 initializes the trigonom etric weight and factor tables or computes the two-dim ensional Fast Fourier T ransform (forw ard or inverse) of a tw o-dim ensional com plex array. In computing the two-dimensional FFT, one-dim ensionalFFT s are com puted along the colum ns of the input array. O ne-dim ensionalFFTs are then com puted along the row s of the interm ediate results.

> N2-1 N 1-1
$Y(k 1, k 2)=$ scale $*$ SUM SUM W 2*W 1*X (1, 12)
j2=0 $\mathfrak{j}=0$
where
k 1 ranges from 0 to $\mathrm{N} 1-1$ and k 2 ranges from 0 to $\mathrm{N} 2-1$
$i=\operatorname{sqrt}(-1)$
isign $=1$ for inverse transform or -1 for forw ard transform
W $1=\exp \left(\right.$ isign*i* $\left.{ }^{1} * k 1 * 2 * p i / N 1\right)$
W $2=\exp ($ isign*i* $2 * k 2 * 2 * p i N 2)$

## ARGUMENTS

IOPT (input)
Integer specifying the operation to be perform ed:
IO PT $=0$ com putes the trigonom etric weight table and factortable
IO PT = -1 com putes forw ard FFT
IO PT $=+1$ com putes inverse FFT
N 1 (input)
Integer specifying length of the transform in the first dim ension. N 1 is m ostefficientw hen it is a productofsm allprin es. N $1>=0$. U nchanged on exit.

N 2 (input)
Integer specifying length of the transform in the second dim ension. N 2 is m ostefficientw hen it is a productofsm allprim es. N $2>=0$. U nchanged on exit.
SCALE (input)
Real scalarby which transform results are scaled. U nchanged on exit. SCA LE is defaulted to 1.0 for F95 $\mathbb{I N}$ TERFACE.

X (input) X is a com plex array ofdim ensions (LD $\mathrm{X}, \mathrm{N} 2$ ) that contains inputdata to be transform ed.

LD X (input)
Leading dim ension of X . LD X >= N 1 U nchanged on exit.

Y (output)
Y is a com plex array ofdim ensions (LD Y ,N 2) that contains the transform results. $X$ and $Y$ can be the sam e anray starting at the sam $e m$ em ory location, in which case the input data are overw rilten by their transform results. O therw ise, it is assum ed that there is no overlap betw een $X$ and $Y$ in $m$ em ory.

LD Y (input)
Leading dim ension of $Y$. If $X$ and $Y$ are the same array, LD Y = LD X Else LD Y >= N 1 U nchanged on exit.

TRIGS (input/output)
Realarray of length $2 *(\mathbb{N} 1+\mathrm{N} 2$ ) that contains the trigonom etric weights. The weights are com puted
w hen the routine is called w ith IO PT = 0 and they are used in subsequent calls w hen IO PT $=1$ or IO PT $=-1$. U nchanged on exit.

IFAC (input/output)
Integeranay ofdim ension at least $2 * 128$ that contains the factors of 1 and N 2 . The factors are com puted when the routine is called w ith IO PT
$=0$ and they are used in subsequent calls $w$ hen IO PT = 1 or IO PT = -1 . U nchanged on exit.

W ORK (w orkspace)
Real array of dimension at least
$2 * \mathrm{M} A X(\mathbb{N} 1, N 2) * N C P U S$ where $\operatorname{NCPUS}$ is the num berof threads used to execute the routine. The user can
also choose to have the routine allocate its ow $n$ w orkspace (see LW ORK).

LW ORK (input)
Integer specifying w orkspace size. If LW ORK $=0$, the routine w illallocate its ow n w orkspace.
$\mathbb{E R R}$ (output)
On exit, integer $\mathbb{E R R}$ has one of the follow ing
values:
0 = norm alretum
$-1=10$ PT is not 0,1 or -1
$-2=\mathrm{N} 1<0$
$-3=N 2<0$
$-4=(\mathbb{L D X}<\mathrm{N} 1)$
$-5=(\mathbb{L D} Y<N 1)$ or (LD Y notequalld $X$ when $X$ and $Y$
are sam e array)
$-6=(L W$ ORK not equal 0) and (LWORK <
$2 * M A X(N 1, N 2) * N C P U S)$
$-7=m$ em ory allocation failed

## SEE ALSO

fft

## CAUTIONS

On exit, entire outputanay Y ( $1: L D Y, 1 \mathbb{N} 2$ ) is overw rilten.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO
- CAUTIONS


## NAME

cfftc3-initialize the trigonom etric weight and factor tables or compute the three-dim ensional Fast Fourier $T$ ransform (forw ard or inverse) of a three-dim ensional com plex array.

## SYNOPSIS

SU BROUTINE CFFTC 3 (IOPT,N1,N2,N3,SCALE,X,LDX1,LDX2,Y,LDY 1,LD Y 2, TRIGS, $\mathbb{F A C}, \mathrm{W}$ ORK, LW ORK, $\mathbb{E R R}$ )
$\mathbb{N}$ TEGER IOPT,N1,N2,N3,LDX1,LDX2,LDY1, LDY2, $\mathbb{F}$ AC (*),
LW ORK, $\mathbb{E R R}$
COM PLEXX (LDX1,LDX2,*), Y (LDY1,LDY2,*)
REAL SCALE,TRIGS (*),WORK (*)
SU BROUTINE CFFTC 3_64 (IOPT,N 1,N2,N 3, SCALE, X,LDX 1,LD X 2, Y, LD Y 1, LD Y 2, TRIGS, $\mathbb{F} A C, W$ ORK, LW ORK, $\mathbb{E R R}$ )
$\mathbb{N}$ TEGER*8 IOPT,N1,N2,N3,LDX1,LDX2,LDY1,LD Y 2, FFAC (*), LW ORK, $\mathbb{E R R}$
COM PLEX X (LDX1,LDX2,*), Y (LDY 1,LDY2,*)
REAL SCALE,TRIGS (*),WORK (*)

## F95 INTERFACE

SUBROUTINE FFT3 (TOPT, $\mathbb{N} 1], \mathbb{N} 2], \mathbb{N} 3],[S C A L E], X,[L D X 1], L D X 2, Y,[L D Y 1]$, LDY 2, TRIGS,

FAC, W ORK, [LW ORK], ERR)
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT $(\mathbb{N})::$ IOPT, LD X 2 , LD Y 2
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT ( $\mathbb{N}$ ), OPTIONAL ::N1,N2,N3, LDX1, LDY1, LW ORK

REAL, $\mathbb{N} T E N T(\mathbb{N})$, OPTIONAL :: SCALE
COM PLEX, $\mathbb{N} T E N T(\mathbb{N}), D \mathbb{M}$ ENSION (:,:) ::X
COMPLEX, $\mathbb{N} T E N T(O U T), D \mathbb{M} E N S \mathbb{O N}(:,:):: Y$
REAL, $\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M}$ ENSION (:) ::TRIGS
$\mathbb{N} T E G E R * 4, \mathbb{N}$ TENT ( $\mathbb{N} O U T$ ), D $\mathbb{I M} E N S I O N(:):: \mathbb{F A C}$
REAL, $\mathbb{N} T E N T$ (OUT),D $\mathbb{I}$ ENSION (:) ::W ORK $\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT(OUT) :: $\mathbb{E R R}$

SU BROU T $\mathbb{N} E$ FFT3_64 ( $\mathbb{O} P \mathrm{P}$, $\mathbb{N} 1], \mathbb{N} 2], \mathbb{N} 3],[S C A L E], X,[L D X 1], L D X 2, Y$, [LD Y 1], LD Y 2, TR IG S,平AC,WORK,[LWORK], 正RR)
$\mathbb{N}$ TEGER (8), $\mathbb{N}$ TENT( $\mathbb{N}$ ) :: IOPT,LDX2,LDY2
$\mathbb{N} \operatorname{TEGER}(8), \mathbb{N}$ TENT $(\mathbb{N})$, OPTIONAL : $\mathrm{N} 1, \mathrm{~N} 2, N 3, L D X 1, L D Y 1$, LW ORK
REAL, $\mathbb{N} T E N T(\mathbb{N})$, OPTIONAL : SCALE
COMPLEX, $\mathbb{N}$ TENT ( $\mathbb{N}$ ), D $\mathbb{M} \operatorname{ENSION(:r:)~::X~}$
COM PLEX, $\mathbb{N}$ TENT (OUT),D $\mathbb{M}$ ENSION (:,:) ::Y
REAL, $\mathbb{N}$ TENT ( $\mathbb{N O U T}$ ), D $\mathbb{M}$ ENSION (:) ::TRIGS
$\mathbb{N} T E G E R(8), \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{F A C}$
REAL, $\mathbb{N} T E N T(O U T), D \mathbb{M} E N S I O N(:):: W O R K$
$\mathbb{N}$ TEGER (8), $\mathbb{N}$ TENT (OUT) :: $\mathbb{E R R}$
C INTERFACE
\#include <sunperfh>
void cfftc3_ (int*iopt, int*n1, int**n2, int *n3, float *scale, com plex *x, int*ldx1, int *ldx2, com plex *y, int *ldy1, int*ldy2, float *trigs, int*ifac, float*w ork, int *lw ork, int *ien);
void cfftc3_64_ (long *iopt, long *n1, long *n2, long *n3, float *scale, com plex *x, long *ldx1, long *ldx2, com plex *y, long *ldy1, long *ldy2, float *trigs, long *ifac, float *w ork, long *lw ork, long *ien);

## PURPOSE

cfftc3 initializes the trigonom etric weight and factor tables or computes the three-dim ensional Fast Fourier T ransform (forw ard or inverse) of a three-dim ensional com plex anray.

N 3-1 N 2-1 N 1-1


$$
\mathfrak{j}=0 \quad \mathfrak{2}=0 \quad \mathfrak{j}=0
$$

where
k 1 ranges from 0 to $\mathrm{N} 1-1 ; k 2$ ranges from 0 to $\mathrm{N} 2-1$ and $k 3$ ranges from 0 to $\mathrm{N} 3-1$
$i=\operatorname{sqrt}(-1)$
isign $=1$ for inverse transform or -1 for forw ard transform
W $1=\exp (i s i g n * i * j * k 1 * 2 * p i N 1)$
W $2=\exp \left(i s i g n * i * \sum^{*} * 2 * 2 *\right.$ piN 2$)$
W $3=\exp (i s i g n * i * j 3 * k 3 * 2 * p i N 3)$

## ARGUMENTS

IO PT (input)
Integer specifying the operation to be perform ed:
IO PT $=0$ com putes the trigonom etric weight table and factor table
IO PT $=-1$ com putes forw ard FFT
IO $\mathrm{PT}=+1$ com putes inverse FFT
N 1 (input)
Integer specifying length of the transform in the first dim ension. N 1 is m ostefficientw hen it is a productofsm allprim es. N $1>=0$. U nchanged on exit.

N 2 (input)
Integer specifying length of the transform in the second dim ension. N 2 is m ostefficientw hen it is a productofsm allprim es. N $2>=0$. U nchanged on exit.

N 3 (input)
Integer specifying length of the transform in the third dim ension. N 3 ism ostefficientw hen it is a productofsm allprim es. N $3>=0$. U nchanged on exit.

SCALE (input)
Realscalarby which transform results are scaled. U nchanged on exit. SCA LE is defaulted to 1.0 for F95 $\mathbb{I N}$ TERFACE.

X (input) X is a com plex array of dim ensions (LD X 1, LD X 2, N 3) that contains inputdata to be transform ed.

LD X 1 (input)
firstdim ension ofX. LD X 1 >= N1 Unchanged on exit.

LD X 2 (input)
second dim ension of X . LD X 2 >= N 2 U nchanged on exit.

Y (output)
Y is a com plex array of dim ensions (LD Y 1, LD Y 2,
N 3 ) that contains the transform results. X and Y
can be the sam e array starting at the sam $e m$ em ory
location, in which case the input data are
overw rilten by their transform results. O therw ise, it is assum ed that there is no overlap
betw een $X$ and $Y$ in $m$ em ory.

## LD Y 1 (input)

firstdin ension of $Y$. If $X$ and $Y$ are the same aray, LD Y 1 = LDX1ElseLD Y 1 >= N 1 U nchanged on exit.

LD Y 2 (input)
second dim ension of $Y$. If $X$ and $Y$ are the sam $e$ anay, LD Y $2=$ LD X 2 Else LD Y $2>=\mathrm{N} 2$ U nchanged on
exit.

TRIGS (input/output)
Realarray of length $2 *(\mathbb{N} 1+\mathrm{N} 2+\mathrm{N} 3$ ) that contains the trigonom etric weights. The w eights are com puted w hen the routine is called w ith IO PT = 0 and they are used in subsequent calls w hen $\mathbb{I D}$ PT $=1$ or IO PT $=-1$. U nchanged on exit.

IFAC (input/output)
Integer array ofdim ension at least $3 * 128$ that
contains the factors of $1, N 2$ and N 3 . The factors are com puted w hen the routine is called with IOPT $=0$ and they are used in subsequent calls when IOPT $=1$ or IO PT $=-1$. U nchanged on exit.

W ORK (w orkspace)
Realarray ofdim ension at least ( $2 * \mathrm{M}$ A X $\mathbb{N}, \mathrm{N} 2, \mathrm{~N} 3$ ) + $32 \star$ N 3) * NCPUS where NCPUS is the num berof threads used to execute the routine. The user can also choose to have the routine allocate its ow $n$ w orkspace (see LW ORK).

LW ORK (input)
Integer specifying workspace size. If LW ORK $=0$, the routine w illallocate its ow n w orkspace.
$\mathbb{E R R}$ (output)
On exit, integer $\mathbb{E R R}$ has one of the follow ing
values:
0 = norm alretum
$-1=10$ PT is not 0,1 or -1
$-2=\mathrm{N} 1<0$
$-3=N 2<0$
$-4=$ N $3<0$
$-5=(\mathbb{L D X} 1<\mathrm{N} 1)$
$-6=(\mathbb{L} D 2<N 2)$
$-7=(\mathbb{L D} Y 1<N 1)$ or (LDY 1 notequal LD X 1 when $X$ and $Y$ are same array)
$-8=(\mathbb{L D} Y 2<N 2)$ or (LDY 2 notequal LD X 2 when $X$ and $Y$ are same array)
$-9=(L W$ ORK not equal 0) and (LWORK <
$(2 * M A X(N, N 2, N 3)+16 * N 3) * N C P U S)$
$-10=m$ em ory allocation failed

## SEE ALSO

fft

## CAUTIONS

This routine uses $Y \mathbb{N} 1+1$ LLD Y $1,:,:$ ) as scratch space. Therefore, the original contents of this subamay w illbe lost upon retuming from routine while subarray $Y(1 \mathbb{N} 1,1 \mathbb{N} 2,1$ :N 3 ) contains the transform results.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO


## NAME

cfftem - initialize the trigonom etric weight and factor tables or com pute the one-dim ensionalF astF ourier $T$ ransform (forw ard or inverse) of a set of data sequenœes stored in a tw o-dim ensional com plex array.

## SYNOPSIS

SU BROUTINE CFFTCM (IOPT,N1,N2,SCALE, X,LDX,Y,LDY,TRIGS, FAC,W ORK, LW ORK, $\mathbb{E R R}$ )
$\mathbb{N}$ TEGER IOPT,N1,N2,LDX,LDY, $\mathbb{F} A C(*), L W O R K, \mathbb{E R R}$
COM PLEX X (LDX,*), Y (LDY,*)
REAL SCALE,TRIGS (*),W ORK (*)
SUBROUTINE CFFTCM_64(1OPT,N1,N2,SCALE,X,LDX,Y,LDY,TRIGS, FAC,WORK, LW ORK, $\mathbb{E R R}$ )
$\mathbb{N} T E G E R * 8 \mathbb{I} P T, N 1, N 2, L D X, L D Y, \mathbb{F A C}(*), L W O R K, \mathbb{E R R}$
REAL SCALE,TRIGS (*), W ORK (*)
COM PLEX X (LDX,*), Y (LDY,*)

## F95 INTERFACE

SU BROUTINE FFTM (IOPT, $\mathbb{N} 1], \mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S$, تAC,W ORK, [LWORK], ERR)
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT $(\mathbb{N})::$ IOPT
$\mathbb{N} T E G E R * 4, \mathbb{N} T E N T(\mathbb{N}), O P T I O N A L:: N 1, N 2, L D X, L D Y, L W$ ORK
REAL, $\mathbb{N} T E N T(\mathbb{N})$, OPTIONAL :: SCALE
COMPLEX, $\mathbb{N}$ TENT ( $\mathbb{N}$ ), D $\mathbb{I M}$ ENSION (:,:) ::X
COM PLEX, $\mathbb{N}$ TENT (OUT),D $\mathbb{M}$ ENSION (:,:) ::Y
REAL, $\mathbb{N}$ TENT ( $\mathbb{N O U T}$ ), D $\mathbb{M}$ ENSION (:) ::TRIGS
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT ( $\mathbb{N} O U T$ ), D $\mathbb{I M}$ ENSION (:) :: $\mathbb{F A C}$

SUBROUTINE FFTM _64 (IOPT, $\mathbb{N} 1], \mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S$, FAC,W ORK, [LW ORK], ERR)
$\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N}):: \mathbb{I O P T}$
$\mathbb{N}$ TEGER (8), $\mathbb{N}$ TENT $(\mathbb{N})$, OPTIONAL ::N 1,N2, LDX,LDY,LW ORK
REAL, $\mathbb{N} T E N T(\mathbb{N})$,OPTIONAL ::SCALE
COM PLEX, $\mathbb{N}$ TENT ( $\mathbb{N}$ ), D $\mathbb{I M}$ ENSION (:,:) ::X
COM PLEX, $\mathbb{N} T E N T$ (OUT),D $\mathbb{M}$ ENSION (: : : : ::Y
REAL, $\mathbb{N} T E N T(\mathbb{N O U T})$, D $\mathbb{I M}$ ENSION (:) ::TRIGS
$\mathbb{N}$ TEGER (8), $\mathbb{N}$ TENT ( $\mathbb{N} O U T$ ), D $\mathbb{M}$ ENSION (:) :: $\mathbb{F A C}$
REAL, $\mathbb{N} T E N T(O U T), D \mathbb{M} E N S I O N(:):: W$ ORK
$\mathbb{N}$ TEGER (8), $\mathbb{N}$ TENT (OUT) :: $\mathbb{E R R}$

## C INTERFACE

\#include <sunperfh>
void cfftom _ (int*iopt, int*n1, int *n2, float *scale, com plex *x, int*ldx, com plex *y, int*ldy, float *trigs, int *ifac, float *w ork, int *lw ork, int *ient);
void cfftom _64_ (long *iopt, long *n1, long *n2, float
*scale, com plex *x, long *ldx, com plex *y, long
*ldy, float *trigs, long *ifac, float *w ork, long
*lw ork, long *ienr);

## PURPOSE

cfftem initializes the trigonom etric weight and factor
tables or computes the one-dim ensional Fast Fourier
$T$ ransform (forw ard or inverse) of a set of data sequences
stored in a tw o-dim ensionalcom plex array:

N 1-1
Y ( $k, 1)=S U M$ W *X (jl)
$\ddagger$
where
k ranges from 0 to N 1-1 and lranges from 0 to N 2-1
$i=\operatorname{sqrt}(-1)$
isign $=1$ for inverse transform or -1 for forw ard transform
$W=\exp (i s i g n * i \star j * k * 2 \star p i N 1)$

## ARGUMENTS

IOPT (input)
Integer specifying the operation to be perform ed:
IO PT $=0$ com putes the trigonom etric weight table
and factor table
IO PT $=-1$ com putes forw ard FFT
IO $\mathrm{PT}=+1$ com putes inverse FFT
N 1 (input)
Integer specifying length of the input sequences. N 1 is m ostefficientw hen it is a product of sm all prim es. N $1>=0$. U nchanged on exit.

N 2 (input)
Integer specifying num ber of input sequences. N 2 $>=0$. U nchanged on exit.

SCALE (input)
Realscalarby which transform results are scaled. U nchanged on exit. SCA LE is defaulted to 1.0 for F95 $\mathbb{I N}$ TERFACE.
$X$ (input) $X$ is a com plex array ofdim ensions (LD X ,N2) that contains the sequences to be transform ed stored in its colum ns.

LD X (input)
Leading dim ension of X . LD $\mathrm{X}>=\mathrm{N} 1$ U nchanged on
exit.
Y (output)
$Y$ is a com plex array ofdim ensions (LD Y ,N 2) that contains the transform results of the input sequences. $X$ and $Y$ can be the sam e array starting at the sam e mem ory location, in which case the input sequences are overw rilten by their transform results. O therw ise, it is assum ed that there is no overlap betw een $X$ and $Y$ in $m$ em ory.

LD Y (input)
Leading dimension of $Y$. If $X$ and $Y$ are the same array, LD Y = LD X E lse LD Y >= N 1 U nchanged on exit.

TRIGS (input/output)
Real array of length $2 * \mathrm{~N} 1$ that contains the trigonom etric w eights. The w eights are com puted w hen the routine is called with IOPT = 0 and they are used in subsequent calls when IO PT = 1 or IO PT = -1. U nchanged on exit.

FAC (input/output)

Integer array ofdim ension at least 128 that contains the factors of N 1 . The factors are com puted w hen the routine is called w ith IO PT $=0$ and they are used in subsequent calls w hen IO PT $=1$ or IO PT $=-1$. U nchanged on exit.

W ORK (w orkspace)
Real array ofdim ension at least $2 * \mathrm{~N}$ 1*N CPUS where
N CPUS is the num ber of threads used to execute the
routine. The usercan also choose to have the
routine allocate its ow $n$ w orkspace (see LW ORK).

LW ORK (input)
Integer specifying w orkspace size. If LW ORK $=0$, the routine w illallocate its ow n w orkspace.

正RR (output)
O n exit, integer $\mathbb{E} R R$ has one of the follow ing
values:
0 = norm alretum
$-1=10 P T$ is not 0,1 or -1
$-2=\mathrm{N} 1<0$
$-3=N 2<0$
$-4=(\mathbb{L D X}<\mathrm{N} 1)$
$-5=(L D Y<N 1)$ or (LD Y notequalLD $X$ when $X$ and $Y$
are sam e array)
$-6=(\mathbb{L W}$ ORK notequal0) and (LW ORK $<2 \star \mathrm{~N} 1 * \mathrm{~N}$ CPUS)
$-7=m$ em ory allocation failed

## SEE ALSO

fft

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

cffff-com pute the Fourier coefficients of a periodic
sequence. The FFT operations are unnorm alized, so a callof CFFTF follow ed by a callof CFFTB w ill m ultiply the input sequence by N .

## SYNOPSIS

```
SUBROUTINE CFFTF N,X,W SAVE)
COM PLEX X (*)
\mathbb{NTEGERN}
REAL W SAVE (*)
SUBROUT\mathbb{NE CFFTF_64 N,X,W SAVE)}
COM PLEX X (*)
\mathbb{NTEGER*8N}
REALW SAVE (*)
F95 INTERFACE
    SUBROUT\mathbb{NE FFTF (N ],X,W SAVE)}
    COMPLEX,D IM ENSION (:) ::X
    \mathbb{NTEGER ::N}
    REAL,D IM ENSION (:) ::W SAVE
    SUBROUT\mathbb{NE FFTF_64 (N ],X,W SAVE)}
    COMPLEX,D IM ENSION (:) ::X
    \mathbb{NTEGER (8)::N}
    REAL,D IM ENSION (:) ::W SAVE
```

\#include <sunperfh>
void cfflff(intn, com plex *x, float*w save);
void cfflf_64 (long n, com plex *x, float *w save);

## ARGUMENTS

N (input) Length of the sequence to be transform ed. These subroutines are $m$ ostefficientw hen $N$ is a product of sm allprim es. $\mathrm{N}>=0$.

X (input) On entry, an amay of length N containing the sequence to be transform ed.

W SAVE (input)
O n entry, W SAVE m ustbe an array ofdim ension (4* $\mathrm{N}+15$ ) orgreater and m usthave been initialized by CFFTI.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

cffti-initialize the anay $W$ SAVE, which is used in both CFFTF and CFFTB .

## SYNOPSIS

## SU BROUTINE CFFTIN,W SAVE)

IN TEGER N
REALW SAVE (*)
SU BROUTINE CFFTI_64N,W SAVE)
$\mathbb{N}$ TEGER*8 N
REALW SAVE (*)
F95 INTERFACE
SU BROUTINE CFFTIN,W SAVE)
$\mathbb{N} T E G E R:: N$
REAL,D $\mathbb{I M}$ ENSION (:) ::W SAVE

SU BROUTINECFFTI_64 N, W SAVE)
$\mathbb{N}$ TEGER (8) :: N
REAL,D $\mathbb{M}$ ENSION (:) ::W SAVE

## C INTERFACE

\#include <sunperfh>
void cffti(intn, float *W save);
void cffti_ 64 (long n , float *w save);

## ARGUMENTS

N (imput) Length of the sequence to be transform ed. $\mathrm{N}>=0$.

W SAVE (input/output)
O n entry, an array ofdim ension ( $4 * N+15$ ) or greater. CFFTI needs to be called only once to initialize array $W$ ORK before calling CFFTF and/or CFFTB if $N$ and W SAVE rem ain unchanged betw een these calls. Thus, subsequent transform $s$ or inverse transform s of sam e size can be obtained faster than the first since they do not require in itialization of the w orkspace.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE


## NAME

cfflopt-com pute the length of the closest fastFFT

## SYNOPSIS

INTEGER FUNCTION CFFTOPT (LEN)
$\mathbb{N}$ TEGER LEN
$\mathbb{N} T E G E R * 8$ FUNCTION CFFTOPT_64 (LEN)
$\mathbb{N}$ TEGER*8 LEN

F95 INTERFACE
$\mathbb{N}$ TEGER FUNCTION CFFTOPT (LEN)
$\mathbb{N}$ TEGER ::LEN
$\mathbb{N} T E G E R$ (8) FUNCTION CFFTOPT_64 (LEN)
$\mathbb{N} T E G E R(8):: L E N$
C INTERFACE
\#include < sunperfh>
int cffltopt(int len);
long cfftopt 64 (long len);

## PURPOSE

Fourier transform algorithm s , including those used in Perform ance L ibrary, w ork bestw ith vector lengths that are products of sm all prim es. Forexam ple, an FFT of length $32=2 * * 5 \mathrm{w}$ ill run fasterthan an FFT of prime length 31 because 32 is a product ofsm allprim es and 31 is not. If your application is such that you can taper or zero pad your vector to a larger length then this function $m$ ay help you select a better length and run your FFT faster.

CFFTOPT w ill retum an integerno sm aller than the input argum entN that is the closestnum ber that is the product of sm allprim es. CFFTO PT w ill retum 16 for an input of $\mathrm{N}=16$ and retum 18=2*3*3 foran inputof $\mathrm{N}=17$.

N ote that the length com puted here is not guaranteed to be optim al, only to be a productofsm allprim es. A lso, the value retumed $m$ ay change as the underlying FFTs become capable of handling largerprim es. Forexam ple, passing in $\mathrm{N}=51$ today w ill retum $52=2 * 2 * 13$ rather than 51=3*17 because the FFTs in Perform ance Library do nothave fast radix 17 code. In the future, radix 17 code $m$ ay be added and then $\mathrm{N}=51 \mathrm{w}$ ill retum 51 .

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO


## NAME

cffts - in inialize the trigonom etric weight and factor tables or com pute the inverse FastFourier T ransform of a com plex sequence as follow s.

## SYNOPSIS


$\mathbb{N}$ TEGER IOPT,N, $\mathbb{F} A C(*)$,LW ORK, $\mathbb{E R R}$
COM PLEX X ${ }^{(*)}$
REALSCALE, Y (*), TRIGS (*),WORK (*)
SU BROUTINE CFFTS_64 (IOPT,N,SCALE,X,Y,TRIGS, $\mathbb{F} A C, W$ ORK,LW ORK, $\mathbb{E R R}$ )
$\mathbb{N} T E G E R * 8 \mathbb{I O P T}, \mathrm{~N}, \mathbb{F A C}(*), L W$ ORK, $\mathbb{E R R}$
REAL SCALE, Y (*), TRIGS (*),WORK (*)
COM PLEX X ${ }^{*}$ )

## F95 INTERFACE

SU BROUTINE FFT (IOPT,N, [SCALE],X,Y,TRIGS, $\mathbb{F A C}, \mathrm{W}$ ORK, [LW ORK ], $\mathbb{E R R}$ )
$\mathbb{N} \operatorname{TEGER} * 4, \mathbb{N} T E N T(\mathbb{N}):: \mathbb{I O P T}, N$
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT ( $\mathbb{N}$ ), OPTIONAL ::LW ORK
REAL, $\mathbb{N} T E N T(\mathbb{N})$,OPTIONAL ::SCALE
COM PLEX, $\mathbb{N}$ TENT ( $\mathbb{N}$ ), D $\mathbb{M}$ ENSION (:) ::X
REAL, $\mathbb{N}$ TENT (OUT),D $\mathbb{M}$ ENSION (:) ::Y
REAL, $\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{I M}$ ENSION (:) ::TRIGS
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT ( $\mathbb{N} O U T$ ), D $\mathbb{M}$ ENSION (:) :: $\mathbb{F A C}$
REAL, $\mathbb{I N T E N T}$ (OUT),D $\mathbb{M}$ ENSION (:) ::W ORK
$\mathbb{N} T E G E R * 4, \mathbb{N} T E N T(O U T):: \mathbb{E R R}$

SU BROUTINE FFT_64 (IOPT,N, [SCALE],X,Y,TRIGS, IFAC,W ORK, [LW ORK], ERR)
$\mathbb{N}$ TEGER (8), $\mathbb{N}$ TENT $(\mathbb{N}):: \mathbb{I O P T}, \mathrm{N}$
$\mathbb{N}$ TEGER (8), $\mathbb{N}$ TENT ( $\mathbb{N}$ ), OPTIONAL ::LW ORK
REAL, $\mathbb{N} T E N T(\mathbb{N})$, OPTIONAL ::SCALE
COM PLEX , $\mathbb{N}$ TENT ( $\mathbb{N}$ ), D $\mathbb{M}$ ENSION (:) :: X
REAL, $\mathbb{N}$ TENT (OUT), D $\mathbb{M}$ ENSION (:) ::Y
REAL, $\mathbb{N}$ TENT ( $\mathbb{N O O U T ) , D \mathbb { I M } E N S I O N ( : ) : : T R I G S ~}$
$\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N O U T}), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{F A C}$
REAL, $\mathbb{N}$ TENT (OUT),D $\mathbb{M} E N S I O N(:):: W O R K$
$\mathbb{N}$ TEGER (8), $\mathbb{N}$ TENT (OUT) :: $\mathbb{E R R}$

## C INTERFACE

\#include <sunperfh>
void cffts_ (int*iopt, int*n, float *scale, com plex *x, float *y, float *trigs, int*ifac, float*w ork, int*lw ork, int *ient);
void cffts_64_ (long *iopt, long *n, float *scale, com plex ${ }^{*} \mathrm{x}$, float *y, float *trigs, long *ifac, float *W Ork, long *lw ork, long *ien);

## PURPOSE

cffts initializes the trigonom etric w eight and factor tables or com putes the inverse FastFourier T ransform of a com plex sequence as follow s:

N-1
Y (k) $=$ scale * SUM W *X ( $)$
$=0$
where
k ranges from 0 to $\mathrm{N}-1$
$i=\operatorname{sqnt}(-1)$
isign $=1$ for inverse transform or -1 for forw ard transform
$W=\exp \left(i s i g n \star i^{\star} j^{\star} k * 2 * p i N N\right)$
In com plex-to-realtransform of length $N$, the $(\mathbb{N} / 2+1)$ com plex input data points stored are the positive-frequency half of the spectrum of the $D$ iscrete Fourier T ransform. The other half can be obtained through com plex conjugation and therefore is notstored. Furtherm ore, due to sym $m$ etries the im aginary of the com ponentof $\mathrm{X}(0)$ and $\mathrm{X} \mathbb{N} / 2$ ) (if N is even in the latter) is assum ed to be zero and is notreferenced.

## ARGUMENTS

IOPT (input)
Integer specifying the operation to be perform ed:
IO PT $=0$ com putes the trigonom etric weight table and factortable
IO PT = 1 com putes inverse FFT
N (input)
Integer specifying length of the input sequence $X$.
N is mostefficientw hen it is a product of sm all prim es. $\mathrm{N}>=0$. U nchanged on exit.

SCALE (input)
Real scalarby which transform results are scaled. U nchanged on exit. SCA LE is defaulted to 1.0 for F95 $\mathbb{N}$ TERFACE.
$X$ (input) $O n$ entry, $X$ is a com plex array whose first $\mathbb{N} / 2+1$ )
elem ents are the input sequence to be transform ed.
Y (output)
Realaray ofdim ension at least $N$ that contains the transform results. $X$ and $Y m$ ay be the sam $e$ array starting at the sam emem ory location. O therw ise, it is assum ed that there is no overlap betw een $X$ and $Y$ in $m$ em ory.

## TR IG S (input/output)

Real array of length $2 * \mathrm{~N}$ that contains the trigonom etric w eights. The w eights are com puted w hen the routine is called w ith IOPT $=0$ and they are used in subsequent calls w hen ID PT $=1$. U nchanged on exit.

IFAC (input/output)
Integer array of dim ension at least 128 that contains the factors of N . The factors are com puted when the routine is called w ith $\mathbb{I O P T}=0$ and they are used in subsequent calls where $\mathrm{IOPT}=1$. U nchanged on exit.

W ORK (w orkspace)
Realarray ofdim ension at leastN. The user can also choose to have the routine allocate its ow $n$ w orkspace (see LW ORK).

LW ORK (input)
Integer specifying workspace size. If LW ORK $=0$, the routine w illallocate its ow n w orkspace.

ERR (output)
On exit, integer $\mathbb{E} R R$ has one of the follow ing
values:
0 = norm alretum
$-1=10 P T$ is not 0 or 1
$-2=\mathrm{N}<0$
$-3=(L W O R K$ is not 0) and (LW ORK is less than N)
$-4=m$ em ory allocation forw orkspace failed

## SEE ALSO

ffl

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO
- CAUTIONS


## NAME

Cffls2 -initialize the trigonom etric weight and factor
tables or com pute the tw o-dim ensional inverse FastFourier
$T$ ransform of a tw o-dim ensional com plex aray.

## SYNOPSIS

SU BROUTINE CFFTS2 (IOPT,N1,N2,SCALE, X,LDX,Y,LDY,TRIGS, FAC,W ORK, LW ORK, $\mathbb{E R R}$ )
$\mathbb{N}$ TEGER IOPT,N1,N2,LDX,LDY, $\mathbb{F} A C(*), L W O R K, \mathbb{E R R}$
COM PLEX X (LDX,*)
REAL SCALE, Y (LDY,*), TRIGS (*),W ORK (*)

SU BROU T $\mathbb{N} E$ CFFTS2_64 (IO PT,N1,N2,SCALE, X,LDX,Y,LDY,TRIGS, $\mathbb{F} A C, W$ ORK, LW ORK, $\mathbb{E R R}$ )
$\mathbb{N} T E G E R * 8 \mathbb{I} P T, N 1, N 2, L D X, L D Y, \mathbb{F A C}(*), L W$ ORK, $\mathbb{E R R}$
COM PLEX X (LDX,*)
REAL SCALE, Y (LDY,*), TRIGS (*),W ORK (*)

## F95 INTERFACE

SU BROUTINE FFT2 (IOPT,N1, N2], [SCALE],X, [LDX],Y, [LDY],TRIGS, \& $\quad \mathbb{F A C}, \mathrm{W} O R K$, [LW ORK], $\mathbb{E R R})$
$\mathbb{N} T E G E R * 4, \mathbb{N} T E N T(\mathbb{N}):: \operatorname{IOPT}, N 1$
$\mathbb{N} T E G E R * 4, \mathbb{N} T E N T(\mathbb{N})$,OPTIONAL ::N2,LDX,LDY,LW ORK
REAL, $\mathbb{N}$ TENT $(\mathbb{N})$,OPTIDNAL ::SCALE
COM PLEX, $\mathbb{N}$ TENT ( $\mathbb{N}$ ), D $\mathbb{M}$ ENSION (: : : : : : X
REAL, $\mathbb{N} T E N T(O U T), D \mathbb{I M}$ ENSION (:,:) :: Y
REAL, $\mathbb{N}$ TENT ( $\mathbb{N} O U T$ ), D $\mathbb{I M}$ ENSION (:) ::TRIGS
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT ( $\mathbb{N} O U T$ ),D $\mathbb{I M}$ ENSION (:) :: $\mathbb{F A C}$

REAL, $\mathbb{I N T E N T ( O U T ) , D \mathbb { M } E N S I O N ( : ) : : W O R K}$
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT(OUT) :: $\mathbb{E R R}$

SU BROUTINE FFT2_64 (IOPT,N1, $\mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S, \mathbb{F A C}$, W ORK, [LW ORK], ERR)
$\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N}):: \operatorname{IOPT}, \mathrm{N} 1$
$\mathbb{N}$ TEGER (8), $\mathbb{N}$ TENT $(\mathbb{N})$, OPTIONAL ::N2,LDX,LDY,LW ORK
REAL, $\mathbb{N} T E N T(\mathbb{N})$, OPTIONAL :: SCALE
COM PLEX, $\mathbb{N}$ TENT ( $\mathbb{N}$ ), D $\mathbb{M}$ ENSION (: : : : : X
REAL, $\mathbb{N} T E N T(O U T), D \mathbb{M} E N S I O N(:,:):: Y$
REAL, $\mathbb{N} T E N T$ ( $\mathbb{N} O U T$ ), D $\mathbb{M}$ ENSION (:) :: TRIGS
$\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N} O U T), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{F A C}$
REAL, $\mathbb{N}$ TENT (OUT), D $\mathbb{M} E N S I O N(:):: W O R K$
$\mathbb{N}$ TEGER (8), $\mathbb{N}$ TENT(OUT) :: $\mathbb{E R R}$

## C INTERFACE

\#include <sunperfh>
void cffts2_ (int*iopt, int*n1, int *n2, float *scale, complex ${ }^{*} x$, int *ldx, float *y, int*ldy, float
*trigs, int*ifac, float*w ork, int *lw ork, int
*ienc);
void cffts2_64_ (long *iopt, long *n1, long *n2, float *scale, com plex *x, long *ldx, float*y, long
*ldy, float *trigs, long *ifac, float *w ork, long
*lw ork, long *ien);

## PURPOSE

cffls2 initializes the trigonom etric w eight and factor tables or com putes the tw o-dim ensional inverse FastF ourier Transform of a tw o-dim ensionalcom plex array. In com puting the tw o-dim ensional FFT, one-dim ensionalFFT s are com puted along the row s of the input array. O ne-dim ensionalFFT s are then com puted along the colum ns of the interm ediate results.

N 1-1 N 2-1
$Y(k 1, k 2)=$ scale * SUM SUM W 2*W 1*X (1, 2 2 )

$$
\mathfrak{j}=0 \quad \mathfrak{2}=0
$$

where
k 1 ranges from 0 to $\mathrm{N} 1-1$ and k 2 ranges from 0 to $\mathrm{N} 2-1$
$i=\operatorname{sqrt}(-1)$
isign $=1$ for inverse transform
W $1=\exp ($ isign *i* $11 * k 1 * 2 * \mathrm{pi} N 1)$
W $2=\exp \left(\right.$ isign $\left.* i^{*} \mathrm{R}^{*} k 2 * 2 * \mathrm{pi} / \mathrm{N} 2\right)$
In com plex-to-real transform of length $N 1$, the $\mathbb{N} 1 / 2+1$ ) com -
plex input data points stored are the positive-frequency
half of the spectrum of the $D$ iscrete Fourier T ransform. The other half can be obtained through com plex conjugation and therefore is not stored.

## ARGUMENTS

IO PT (input)
Integer specifying the operation to be perform ed:
IO PT $=0$ com putes the trigonom etric weight table
and factor table
IO PT = 1 com putes inverse FFT
N 1 (input)
Integer specifying length of the transform in the first dim ension. N 1 is m ostefficientw hen it is a productofsm allprim es. N $1>=0$. U nchanged on exit.

N 2 (input)
Integer specifying length of the transform in the second dim ension. N 2 is m ostefficientw hen it is a productofsm allprim es. N $2>=0$. U nchanged on exit.

SCALE (input)
Real scalarby which transform results are scaled. U nchanged on exit. SCA LE is defaulted to 1.0 for F95 $\mathbb{I N}$ TERFACE.

X (input) X is a com plex array ofdim ensions (LD $\mathrm{X}, \mathrm{N} 2)$ that contains input data to be transform ed.

LD X (input)
Leading dimension of $X . \operatorname{LDX}>=\mathbb{N} 1 / 2+1)$
U nchanged on exit.
Y (output)
$Y$ is a realamay of dim ensions (LD Y, N 2) that contains the transform results. $X$ and $Y$ can be the sam e array starting at the sam e mem ory location, in which case the input data are overw ritten by their transform results. O therw ise, it is assum ed that there is no overlap betw een X and Y in $m$ em ory.

LD Y (input)
Leading dim ension of $Y$. If $X$ and $Y$ are the same anay, LD Y $=2 *$ LD X Else LD $Y>=2 * L D X$ and LD Y m ust
be even. U nchanged on exit.

TRIGS (input/output)
Real array of length $2 *(\mathbb{N} 1+\mathrm{N} 2)$ that contains the trigonom etric w eights. The weights are com puted w hen the routine is called w ith $10 \mathrm{PT}=0$ and they are used in subsequent calls when $\mathrm{IOPT}=1$.
U nchanged on exit.

IFAC (input/output)
Integer array ofdim ension at least $2 * 128$ that
contains the factors ofN 1 and N2. The factors are com puted w hen the routine is called w ith IO PT
$=0$ and they are used in subsequent calls w hen
IO PT = 1. U nchanged on exit.

W ORK (w orkspace)
Real array of dimension at least
$M A X(\mathbb{N} 1,2 * N 2) * N C P U S$, where $N C P U S$ is the num ber of
threads used to execute the routine. The user can
also choose to have the routine allocate its ow $n$
w orkspace (see LW ORK).

LW ORK (input)
Integer specifying w ork.space size. If LW ORK $=0$, the routine w illallocate its ow n w ork.space.

ERR (output)
On exit, integer $\mathbb{E} R \mathrm{R}$ has one of the follow ing
values:
0 = norm alretum
$-1=10 P T$ is not 0,1
$-2=N 1<0$
$-3=N 2<0$
$-4=(\llbracket D X<N 1 / 2+1)$
$-5=L D Y$ notequal $2 \star$ LD $X$ when $X$ and $Y$ are same
aray
$-6=(L D Y<2 \star L D X$ orLD $Y$ odd) $w$ hen $X$ and $Y$ are
sam e array
$-7=(L W O R K$ not equal 0) and (LWORK <
M AX $\mathbb{N} 1,2 * N 2$ ) $\left.{ }^{*} N C P U S\right)$
$-8=m$ em ory allocation failed

## SEE ALSO

fflt

## CAUTIONS

Y $\mathbb{N} 1+1: L D Y,:$ ) is used as scratch space. U pon retuming, the original contents of Y $(\mathbb{N} 1+1: L D Y,:)$ w illbe lost, w hereas $\mathrm{Y}(1 \mathbb{N} 1,1 \mathbb{N} 2)$ contains the transform results.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO
- CAUTIONS


## NAME

cffls3 -initialize the trigonom etric weight and factor
tables or com pute the three-dim ensional inverse FastFourier
T ransform of a three-dim ensionalcom plex array.

## SYNOPSIS

SU BROUTINE CFFTS3 (IO PT,N1,N2,N3,SCALE, X,LDX 1,LD X 2, Y, LD Y 1, LD Y 2, TRIGS, $\mathbb{F} A C, W$ ORK, LW ORK, $\mathbb{E R R}$ )

LW ORK, ERR
COM PLEXX (LDX1,LDX2,*)
REAL SCALE, TRIGS (*), W ORK (*), Y (LDY 1,LD Y 2, *)
SU BROUTINE CFFTS3_64 (TOPT,N1,N2,N 3,SCALE, X,LDX 1,LDX 2, Y,LDY 1,LDY2, TRIGS, $\mathbb{F} A C, W$ ORK, LW ORK, $\mathbb{E R R}$ )
$\mathbb{N}$ TEGER*8 IO PT,N1,N2,N3,LDX 1, LD X 2, LD Y 1, LD Y 2, FAA (*),
LW ORK, $\mathbb{E R R}$
COM PLEX X (LDX1,LDX2,*)
REAL SCALE,TRIGS (*),W ORK (*), Y (LDY 1,LDY 2,*)

## F95 INTERFACE

SU BROUTINE FFT3 (IOPT ,N 1, $\mathbb{N} 2], \mathbb{N} 3]$ [SCALE ], X, [LD X 1], LD X 2, Y, [LD Y 1], LD Y 2 , TRIGS,

FAC, W ORK, [LWORK], ERR)
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT $(\mathbb{N}):: \mathbb{I O} \operatorname{PT}, N 1, L D \times 2, L D Y 2$
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT $(\mathbb{N})$, OPTIONAL ::N $2, N 3, L D X 1, L D Y 1, L W O R K$
REAL, $\mathbb{N} T E N T(\mathbb{N})$, OPTIONAL ::SCALE
COM PLEX, $\mathbb{N}$ TENT ( $\mathbb{N}$ ), D $\mathbb{M}$ ENSION (:,:) ::X

REAL, $\mathbb{I N T E N T}(O U T), D \mathbb{I M} E N S I O N(:,:):: Y$
REAL, $\mathbb{N} T E N T(\mathbb{N O U T})$, D $\mathbb{M}$ ENSION (:) ::TRIGS
$\mathbb{N} T E G E R * 4, \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{I M}$ ENSION (:) :: $\mathbb{F} A C$
REAL, $\mathbb{I N T E N T}(\mathrm{OUT}), \mathrm{D} \mathbb{I}$ ENSION (:) ::W ORK
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT (OUT) :: $\mathbb{E R R}$
SU BROUTINE FFT3_64 (IOPT,N1, $\mathbb{N} 2], \mathbb{N} 3],[S C A L E], X,[L D X 1], L D X 2, Y$, [LD Y 1], LD Y 2, TR IG S, تAC,W ORK, [LW ORK], ERR)
$\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} T E N T(\mathbb{N}):: \mathbb{I O P T}, N 1, L D X 2, L D Y 2$
$\mathbb{N} T E G E R(8), \mathbb{N} T E N T(\mathbb{N}), O P T I D N A L:: N 2, N 3, L D X 1$, LDY1,
LW ORK
REAL, $\mathbb{N} T E N T(\mathbb{N})$, OPTIONAL :: SCALE
COM PLEX, $\mathbb{N}$ TENT ( $\mathbb{N}$ ), D $\mathbb{I M}$ ENSION (:,:) ::X
REAL, $\mathbb{N} T E N T$ (OUT),D $\mathbb{I M}$ ENSION (:,:) :: Y
REAL, $\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{I M}$ ENSION (:) ::TRIGS
$\mathbb{N}$ TEGER (8), $\mathbb{N}$ TENT ( $\mathbb{N O U T}$ ), D $\mathbb{M}$ ENSION (:) :: $\mathbb{F A C}$
REAL, $\mathbb{N}$ TENT (OUT),D $\mathbb{M}$ ENSION (:) ::W ORK
$\mathbb{N} T E G E R(8), \mathbb{N} T E N T(O U T):: \mathbb{E R R}$

## C INTERFACE

\#include <sunperfh>
void cffts3_ (int*iopt, int*n1, int*n2, int *n3, float *scale, com plex *x, int*ldx1, int*ldx2, float *y, int *ldy1, int *ldy 2 , float *trigs, int *ifac, float *w ork, int *lw ork, int *ien);
void cffts3_64_ (long *iopt, long *n1, long *n2, long *n3, float *scale, com plex *x, long *ldx1, long *ldx2, float *y, long *ldy1, long *ldy2, float *trigs, long *ifac, float *w ork, long *lw ork, long *ienr);

## PURPOSE

cffls3 initializes the trigonom etric weight and factor tables or com putes the three-dim ensional inverse Fast Fourier $T$ ransform of a three-dim ensional com plex array.

N 3-1 N2-1 N1-1
$Y(k 1, k 2, k 3)=$ scale $*$ SUM SUM SUM W 3*W 2*W 1*X ( $11, \mathfrak{p}, \mathfrak{j})$

$$
\mathfrak{j}=0 \quad \mathfrak{j}=0 \quad j 1=0
$$

where
k 1 ranges from 0 to $\mathrm{N} 1-1$; $k 2$ ranges from 0 to $\mathrm{N} 2-1$ and $k 3$
ranges from 0 to $N 3-1$
$i=\operatorname{sqrt}(-1)$
isign = 1 for inverse transform
W $1=\exp (i s i g n * i * 1 \times k 1 * 2 * p i N 1)$

W $2=\exp \left(i s i g n * i *{ }^{2} * k 2 * 2 * p i N 2\right)$
W $3=\exp ($ isign*i* $3 * k 3 * 2 * p i N 3$ )

## ARGUMENTS

IO PT (input)
Integer specifying the operation to be perform ed:
IO PT $=0$ com putes the trigonom etric $w$ eight table
and factor table
IO $\mathrm{PT}=+1$ com putes inverse FFT
N 1 (input)
Integer specifying length of the transform in the first dim ension. N 1 ism ostefficientw hen it is a productofsm allprim es. N $1>=0$. U nchanged on exit.

N 2 (input)
Integer specifying length of the transform in the second dim ension. N 2 is m ostefficientw hen it is a productofsm allprim es. N $2>=0$. U nchanged on exit.
N 3 (input)
Integer specifying length of the transform in the third dim ension. N 3 ism ostefficientw hen it is a productofsm allprim es. N $3>=0$. U nchanged on exit.

SCALE (input)
Real scalarby which transform results are scaled. U nchanged on exit. SCA LE is defaulted to 1.0 for F95 $\mathbb{I N T E R F A C E}$.

X (input) X is a com plex array of dim ensions (LD X 1, LD X 2, N 3 ) that contains input data to be transform ed.

## LD X 1 (input)

firstdim ension of X . LD X $1>=\mathrm{N} 1 / 2+1$ U nchanged on exit.

LD X 2 (input)
second dim ension of X . LD X $2>=\mathrm{N} 2$ U nchanged on exit.

Y (output)
Y is a com plex anay of dim ensions (LD Y 1, LD Y 2,
N 3 ) that contains the transform results. X and $Y$
can be the sam e array starting at the sam $e m$ em ory
location, in which case the input data are overw ritten by their transform results. O therw ise, it is assum ed that there is no overlap betw een $X$ and $Y$ in $m$ em ory.

LD Y 1 (input)
firstdin ension of $Y$. If $X$ and $Y$ are the same array, LD Y $1=2 * L D X 1$ Else LD Y $1>=2 * L D X 1$ and LD Y 1 is even U nchanged on exit.

LD Y 2 (input)
second dim ension of $Y$. If $X$ and $Y$ are the same aray, LD Y 2 = LD X 2 Else LD Y 2 >= N 2 U nchanged on exit.

TR IG S (input/output)
Real array of length $2 *(\mathbb{N} 1+\mathrm{N} 2+\mathrm{N} 3$ ) that contains the trigonom etric weights. The w eights are com puted when the routine is called w ith $\mathbb{I D}$ PT $=0$ and they are used in subsequent calls when IOPT $=1$. U nchanged on exit.

IFAC (input/output)
Integer array ofdim ension at least $3 * 128$ that contains the factors of $1, \mathrm{~N} 2$ and N 3 . The factors are com puted w hen the routine is called w ith IOPT = 0 and they are used in subsequent calls w hen $\mathrm{IOPT}=1$. U nchanged on exit.

W ORK (w orkspace)
Realarray ofdim ension at least MAX $\mathbb{N}, 2 * N 2,2 * N 3$ )
$+16 * N 3)$ * NCPUS where NCPUS is the num berof
threads used to execute the routine. The user can
also choose to have the routine allocate its ow $n$
w orkspace (see LW ORK).
LW ORK (input)
Integer specifying w orkspace size. If LW ORK = 0, the routine w illallocate its ow n w orkspace.

ERR (output)
On exit, integer $\mathbb{E R R}$ has one of the follow ing
values:
0 = norm alretum
$-1=10 \mathrm{PT}$ is not 0 or 1
$-2=\mathrm{N} 1<0$
$-3=\mathrm{N} 2<0$
$-4=N 3<0$
$-5=\left(\operatorname{LDX} 1<\mathrm{N}_{1} / 2+1\right)$
$-6=(\operatorname{LDX} 2<\mathrm{N} 2)$
$-7=\mathrm{LD} Y 1$ notequal $2 *$ LD X 1 w hen X and Y are same anay
$-8=(L D Y 1<2 * L D X 1)$ or (LDY 1 is odd) $w$ hen $X$ and $Y$ are not sam e array
$-9=(L D Y 2<N 2)$ or (LD Y 2 notequalLD X 2) when $X$ and $Y$ are sam e array $-10=(L W$ ORK not equal 0) and ( (LW ORK < $M A X(\mathbb{N}, 2 * N 2,2 * N 3)+16 * N 3) * N C P U S)$
$-11=m$ em ory allocation failed

## SEE ALSO

fft

## CAUTIONS

This routine uses $Y \mathbb{N} 1+1$ LD Y $1,: \%$ : as scratch space. Therefore, the original contents of this subarray $w$ illbe lost upon retuming from routine while subarray $Y(1 \mathbb{N} 1,1 \mathbb{N} 2,1 \mathbb{N} 3)$ contains the transform results.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO


## NAME

cfftem -initialize the trigonom etric weight and factor tables or com pute the one-dim ensional inverse FastFourier T ransform of a setof com plex data sequences stored in a tw o-dim ensional amray.

## SYNOPSIS

SUBROUTINE CFFTSM (IOPT,N1,N2,SCALE,X,LDX,Y,LDY,TRIGS, $\mathbb{F} A C, W$ ORK, LW ORK, $\mathbb{E R R}$ )
$\mathbb{N}$ TEGER $\mathbb{I O P T}, N 1, N 2, L D X, L D Y, \mathbb{F} A C(*), L W O R K, \mathbb{E R R}$
COM PLEX X (LDX,*)
REAL SCALE, Y (LDY, *), TRIGS (*),W ORK (*)
SU BROUTINE CFFTSM _64 (IOPT,N1,N2,SCALE, X,LDX,Y,LDY,TRIGS, FFAC,W ORK, LW ORK, $\mathbb{E R R}$ )
$\mathbb{N} T E G E R * 8 \mathbb{I O P T}, \mathrm{~N} 1, N 2, L D X, L D Y, \mathbb{F} A C(*), L W O R K, \mathbb{E R R}$
REAL SCALE, Y (LDY, ${ }^{\star}$ ), TRIGS (*), W ORK (*)
COM PLEXX (LDX,*)

## F95 INTERFACE

SU BROUTINE FFTM (IOPT,N1, N2], [SCALE],X, [LDX],Y, [LDY],TRIGS, تAC,W ORK, [LW ORK], ERR)
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT ( $\mathbb{N}$ ) :: IOPT,N1
$\mathbb{N} T E G E R * 4, \mathbb{N} T E N T(\mathbb{N})$,OPTIONAL ::N2,LDX,LDY,LW ORK
REAL, $\mathbb{N} T E N T(\mathbb{N})$, OPTIONAL ::SCALE
COM PLEX, $\mathbb{N}$ TENT ( $\mathbb{N}$ ), D $\mathbb{I M}$ ENSION (:,:) ::X
REAL, $\mathbb{I N T E N T}$ (OUT),D $\mathbb{I M}$ ENSION (:,:) ::Y
REAL, $\mathbb{N}$ TENT ( $\mathbb{N O U T}$ ), D $\mathbb{M}$ ENSION (:) ::TRIGS
$\mathbb{N}$ TEGER*4, $\mathbb{N}$ TENT ( $\mathbb{N} O U T$ ), D $\mathbb{I M}$ ENSION (:) :: $\mathbb{F A C}$

SUBROUTINE FFTM _64 (IOPT,N1, $\mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R \mathbb{I} S, \mathbb{F} A C$, W ORK, [LW ORK], $\mathbb{E R R}$ )

```
\mathbb{NTEGER (8), \mathbb{NTENT (N ) :: IOPT,N1}}\mathbf{N}=1
```



```
REAL,\mathbb{NTENT (\mathbb{N}),OPTIONAL ::SCALE}
COM PLEX,\mathbb{NTENT (\mathbb{N}),D IM ENSION (:,:) ::X}
REAL, \mathbb{NTENT (OUT),D IM ENSION (:,:)::Y}
REAL, \mathbb{NTENT (NOUT),D IM ENSION (:) ::TRIGS}
```



```
REAL, INTENT (OUT),D\mathbb{M ENSION (:) ::W ORK}
```



## C INTERFACE

\#include <sunperfh>
void cfftsm _ (int*iopt, int*n1, int *n2, float *scale, com plex *x, int *ldx, float*y, int*ldy, float *trigs, int *ifac, float *w ork, int *lw ork, int *ient);
void cfftsm_64_ (long *iopt, long *n1, long *n2, float
*scale, com plex *x, long *ldx, float*y, long
*ldy, float *trigs, long *ifac, float *w ork, long
*lw ork, long *ienr);

## PURPOSE

cfftem initializes the trigonom etric weight and factor tables or com putes the one-dim ensional inverse FastFourier T ransform of a set of com plex data sequences stored in a tw o-dim ensionalamay:

N 1-1
Y ( $k, 1)=$ scale * SUM W *X (jl)

$$
\dot{j} 0
$$

where
k ranges from 0 to N 1-1 and lranges from 0 to N 2-1
$i=\operatorname{sqrt}(-1)$
isign $=1$ for inverse transform
$W=\exp \left(i s i g n * i^{\star} j^{\star} k * 2 \star \mathrm{piN} 1\right)$
In com plex-to-real transform of length N 1 , the $(\mathbb{N} 1 / 2+1$ ) com -
plex input data points stored are the positive-frequency half of the spectrum of the $D$ iscrete Fourier $T$ ransform. The other half can be obtained through com plex conjugation and therefore is not stored. Furthem ore, due to sym $m$ etries the
im aginary of the com ponent of $\mathrm{X}(0,0 \mathbb{N} 2-1)$ and $\mathrm{X}(\mathbb{N} 1 / 2,0$ N 2-1) (ifN 1 is even in the latter) is assum ed to be zero and is not referenced.

## ARGUMENTS

IO PT (input)
Integer specifying the operation to be perform ed:
IO PT $=0$ com putes the trigonom etric weight table and factor table
IO PT = 1 com putes inverse FFT

N 1 (input)
Integer specifying length of the input sequences.
N 1 ism ostefficientw hen it is a product ofsm all prim es. N $1>=0$. U nchanged on exit.

N 2 (input)
Integer specifying num ber of input sequences. N 2 $>=0$. U nchanged on exit.
SCALE (input)
Real scalarby w hich transform results are scaled.
U nchanged on exit. SCA LE is defaulted to 1.0 for F95 $\mathbb{I N}$ TERFACE.

X (input) X is a com plex aray ofdim ensions (LD $\mathrm{X}, \mathrm{N} 2$ ) that contains the sequences to be transform ed stored in its colum ns in $\mathrm{X}(0 \mathrm{~N} 1 / 2,0 \mathbb{N} 2-1)$.

LD X (input)
Leading dim ension of $\mathrm{X} . \operatorname{LD} \mathrm{X}>=\mathbb{N} 1 / 2+1$ ) U nchanged on exit.

Y (output)
$Y$ is a realarray of dim ensions (LD Y, N 2) that contains the transform results of the input sequences in $Y(0: N 1-1,0 \mathbb{N} 2-1)$. $X$ and $Y$ can be the sam e array starting at the sam em em ory location, in which case the input sequences are overw rilten by their transform results. O therw ise, it is assum ed that there is no overlap betw een X and Y in $m$ em ory.

LD Y (input)
Leading dim ension of $Y$. If $X$ and $Y$ are the same array, LDY $=2 * L D X$ Else LD Y >= N 1 U nchanged on exit.

TR IG S (input/output)
Realaray of length $2 * \mathrm{~N} 1$ that contains the trigonom etric w eights. The w eights are com puted w hen the routine is called w ith IOPT $=0$ and they are used in subsequent calls w hen IO PT $=1$. U nchanged on exit.

IFAC (input/output)
Integer array ofdim ension at least 128 that contains the factors of 1 . The factors are com puted when the routine is called w ith IO PT $=0$ and they are used in subsequent calls when IOPT $=1$.
U nchanged on exit.
W ORK (w orkspace)
Real anray of dim ension at least 1 1. The user can also choose to have the routine allocate its ow $n$ w orkspace (see LW ORK).

LW ORK (input)
Integer specifying workspace size. IfLW ORK $=0$, the routine w illallocate its ow n w orkspace.

ERR (output)
On exit, integer $\mathbb{E R R}$ has one of the follow ing
values:
0 = norm alretum
$-1=10 \mathrm{PT}$ is not 0 or 1
$-2=\mathrm{N} 1<0$
$-3=N 2<0$
$-4=(\operatorname{LDX}<\mathrm{N} 1 / 2+1)$
$-5=(\operatorname{LD} Y<N 1)$ or (LD Y notequal2*LD w wen X and
Y are sam e array)
$-6=(\mathbb{L W}$ ORK notequal0) and (LW ORK < N 1)
$-7=m$ em ory allocation failed

## SEE ALSO

fft

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgbbrd - reduce a com plex generalm -by-n band $m$ atrix $A$ to real upperbidiagonal form $B$ by a unitary transform ation

## SYNOPSIS

```
SUBROUTINE CGBBRD NECT,M,N,NCC,KL,KU,AB,LDAB,D,E,Q,LDQ,
    PT,LDPT,C,LDC,W ORK,RW ORK,INFO)
CHARACTER * 1VECT
COM PLEX AB (LDAB,*),Q (LDQ ,*),PT (LDPT,*),C (LDC ,*),W ORK (*)
INTEGERM,N,NCC,KL,KU,LDAB,LDQ,LDPT,LDC,INFO
REALD (*),E (*),RWORK (*)
SU BROUT\mathbb{NE CGBBRD_64NECT,M,N,NCC,KL,KU,AB,LDAB,D,E,Q,LDQ,}
    PT,LDPT,C,LDC,W ORK,RW ORK,INFO)
```

CHARACTER * 1 VECT
COM PLEX AB (LDAB,*), Q (LDQ ,*), PT (LDPT,*), C (LDC,*), W ORK (*)
$\mathbb{N} T E G E R * 8 M, N, N C C, K L, K U, L D A B, L D Q, L D P T, L D C, I N F O$
REALD (*), E (*), RW ORK (*)

## F95 INTERFACE

SU BROUTINE GBBRD $N E C T, M, \mathbb{N}], \mathbb{N C C}], K L, K U, A B,[L D A B], D, E, Q$, [LD Q ], PT, [LDPT], C, [LDC], [W ORK ], RW ORK], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::VECT
COM PLEX,D IM ENSION (:) ::W ORK
COMPLEX,D $\mathbb{M}$ ENSION (:,:) ::AB, Q,PT,C
$\mathbb{N} T E G E R:: M, N, N C C, K L, K U, L D A B, L D Q, L D P T, L D C, \mathbb{N} F O$
REAL,D IM ENSION (:) ::D,E,RW ORK

SU BROUTINE GBBRD_64 NECT, M, $\mathbb{N}], \mathbb{N C C}], K L, K U, A B,[L D A B], D, E$,

Q, [LD Q ], PT, [LDPT],C, [LDC], [W ORK], RW ORK], [ $\mathbb{N F O}])$

CHARACTER (LEN=1) ::VECT
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::AB, Q,PT,C
$\mathbb{N} T E G E R(8):: M, N, N C C, K L, K U, L D A B, L D Q, L D P T, L D C, \mathbb{N} F O$
REAL,D IM ENSION (:) ::D,E,RW ORK

## C INTERFACE

\#include < sunperfh>
void cgbbrd (char vect, intm, intn, intncc, int kl, int ku, com plex *ab, int ldab, float*d, float*e, com plex *q, int ldq, com plex *pt, int ldpt, com plex *c, intldc, int*info);
void cgbbrd_64 (charvect, long m, long n, long ncc, long kl, long ku, com plex *ab, long ldab, float *d, float
*e, com plex *q, long ldq, com plex *pt, long ldpt, com plex *c, long ldc, long *info);

## PURPOSE

cgbbrd reduces a com plex generalm -by-n band $m$ atrix $A$ to real upperbidiagonal form $B$ by a unitary transform ation: Q '

* $A * P=B$.

The routine com putes B, and optionally form s Q or P', or com putes $Q$ "C for given $m$ atrix $C$.

## ARGUMENTS

```
VECT (input)
    Specifies w hetherornot the m atrices Q and P 'are
    to be form ed. = N ': do not form Q orP ';
    = Q': form Q only;
    = P':form P'only;
    = B':form both.
```

M (input) The num ber of row s of the m atrix $\mathrm{A} . \mathrm{M}>=0$.
N (input) The num ber of colum ns of the m atrix $\mathrm{A} . \mathrm{N}>=0$.
NCC (input)
The num berof colum ns of the m atrix C. NCC >=0.

KL (input)
The num ber of subdiagonals of the m atrix A.K L >=
0.

KU (input)
The num berof superdiagonals of the m atrix A. KU $>=0$.

AB (input/output)
On entry, the $m$ by $-n$ band $m$ atrix $A$, stored in row $s$ 1 to $\mathrm{KL}+\mathrm{KU}+1$. The $j$ th column of is stored in the $j$ th column of the array $A B$ as follows: $A B(k u+1+i-j)=A(i, j)$ for $\max (1, j$ $\mathrm{ku})<=i<=m$ in $(m, j+k l)$. On exit, $A$ is overw rilten by values generated during the reduction.
LDAB (input)
The leading dim ension of the anay A. LD A B >= $K L+K U+1$.

D (output)
The diagonalelem ents of the bidiagonalm atrix B .

E (output)
The superdiagonal elem ents of the bidiagonal $m$ atrix B.

Q (output)
IfVECT = $Q$ 'or $B$ ', the $m$-by $m$ unitary $m$ atrix $Q$. If $V E C T=N$ 'or $P$ ', the array $Q$ is not referenced.

LD Q (input)
The leading dim ension of the array $Q$. LDQ >= $m$ ax (1, M) ifVECT = Q 'or B'; LD Q >= 1 otherw ise.

PT (output)
IfVECT $=P$ 'or $B^{\prime}$ ', the $n-b y-n$ unitary matrix
$P^{\prime}$. If VECT $=N$ 'or $Q$ ', the amay PT is not referenced.

LDPT (input)
The leading dim ension of the array PT. LD PT >= $\max (1, \mathbb{N})$ if $\mathrm{VECT}=\mathrm{P}$ 'or $\mathrm{B} ; \operatorname{LDPT}>=1$ otherw ise.

C (input/output)
On entry, an $m$-by-ncc m atrix C. On exit, $C$ is overw rilten by $\mathrm{Q}{ }^{*} \mathrm{C} . \mathrm{C}$ is not referenced if $\mathrm{NCC}=$ 0 .

LD C (input)
The leading dim ension of the array C. LD C >=
$\max (1, M)$ if $N C C>0 ; L D C>=1$ if $N C C=0$.
W ORK (w orkspace)
dim ension (MAX $M, N$ ))
RW ORK (w orkspace)
dim ension (MAX $M, N$ ))
$\mathbb{I N} F O$ (output)
= 0: successfulexit.
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvahue.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgbcon -estim ate the reciprocal of the condition num ber of a com plex generalband $m$ atrix $A$, in either the 1 -norm or the infinity-norm,

## SYNOPSIS

```
SUBROUT\mathbb{NECGBCON NORM,N,NSUB,NSUPER,A,LDA, \mathbb{PIVOT,ANORM,}}\mathbf{N},\textrm{N},\textrm{N}
    RCOND,W ORK,W ORK2,INFO)
CHARACTER * 1 NORM
COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGERN,NSUB,NSUPER,LDA,INFO}
INTEGER \mathbb{PIVOT (*)}
REAL ANORM,RCOND
REALW ORK2(*)
SU BROUT\mathbb{NE CGBCON_64 NORM,N,NSUB,NSUPER,A,LDA,}\mathbb{N}IVOT,ANORM,
    RCOND,W ORK,WORK2,\mathbb{NFO)}
CHARACTER * 1 NORM
COM PLEX A (LDA,*),W ORK (*)
INTEGER*8N,NSUB,NSUPER,LDA, INFO
INTEGER*8 \mathbb{PIVOT (*)}
REAL ANORM,RCOND
REALW ORK2 (*)
```


## F95 INTERFACE

SU BROUTINE GBCON $\mathbb{N} O R M, \mathbb{N}], N S U B, N S U P E R, A,[L D A], \mathbb{P} \mathbb{I} O T, A N O R M$, RCOND, [W ORK], [W ORK2], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1) ::NORM
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK

COM PLEX, D IM ENSION (:,:) ::A
$\mathbb{N}$ TEGER :: N,N SUB,N SUPER, LDA, $\mathbb{N}$ FO
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{I} O T$
REAL ::ANORM,RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::W ORK 2

SU BROUTINE GBCON_64 $\mathbb{N} O R M,[N], N S U B, N S U P E R, A,[L D A], \mathbb{P} \mathbb{I} O T, A N O R M$, RCOND, [WORK], [WORK2], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) ::NORM
COMPLEX,D $\mathbb{I M} E N S I O N$ (:) ::W ORK
COM PLEX , D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER (8) :: N , N SUB, N SUPER,LDA , $\mathbb{N} F O$
$\mathbb{N}$ TEGER (8), D $\mathbb{I M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$
REAL ::ANORM,RCOND
REAL,D $\mathbb{I}$ ENSION (:) ::W ORK2

## C INTERFACE

\#include <sunperfh>
void ogboon (charnorm, intn, intnsub, intnsuper, com plex
*a, int lda, int *ipivot, float anorm, float
*rcond, int*info);
void ogbcon_64 (char norm , long n, long nsub, long nsuper, complex *a, long lda, long *ípivot, floatanorm , float *roond, long *info);

## PURPOSE

cgbcon estim ates the reciprocal of the condition num ber of a com plex general band $m$ atrix $A$, in either the 1 -norm or the infinity-norm, using the LU factorization com puted by CGBTRF.

A $n$ estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as
$\operatorname{RCOND}=1 /(\operatorname{norm}(A) * \operatorname{norm}(\operatorname{inv}(A)))$.

## ARGUMENTS

## NORM (input)

Specifies w hether the 1-norm condition num ber or the infinity-norm condition num ber is required:
= 1 'or $0^{\prime}$ : 1-norm;
$=1 ': \quad$ Infinity-norm .

N (input) The order of the m atrix A. $\mathrm{N}>=0$.

N SUB (input)
The num ber of subdiagonals w ithin the band of A. N SUB $>=0$.

## N SU PER (input)

The num ber of superdiagonals w ithin the band of A. N SU PER $>=0$.

A (input) D etails of the LU factorization of the band $m$ atrix
A, as com puted by CGBTRF. U is stored as an upper triangularband $m$ atrix $w$ ith N SU B +N SU PER superdiagonals in rows 1 to NSUB+NSUPER+1, and them ultipliers used during the factorization are stored in row SN SU B + N SU PER +2 to $2 *$ N SU B + N SU PER +1 .

## LDA (input)

The leading dim ension of the array A . LDA >= $2 * N$ SU B + N SU PER +1 .

PIVOT (input)
The pivot indices; for $1<=i<=N$, row $i$ of the $m$ atrix $w$ as interchanged w ith row $\mathbb{P} \mathbb{I V}$ T (i).

ANORM (input)
IfNORM = I' or $\mathrm{D}^{\prime}$ ', the 1 -norm of the original $m$ atrix $A$. IfNORM = 'I', the infinity-norm of the originalm atrix A.

RCOND (output)
The reciprocal of the condition number of the $m$ atrix $A$, computed as RCOND $=1 /($ norm (A) * norm (inv (A))).

W ORK (w orkspace)
dim ension ( $2 * \mathrm{~N}$ )
W ORK 2 (w orkspace)
dim ension $\mathbb{N}$ )
$\mathbb{N} F O$ (output)
= 0 : successfiulexit
$<0:$ if $\mathbb{N F O}=-$ i, the $i$-th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgbequ -com pute row and colum n scalings intended to equilibrate an M -by -N band matrix A and reduce its condition num ber

## SYNOPSIS

```
SUBROUT\mathbb{NE CGBEQU M,N,KL,KU,A,LDA,R,C,ROW CND,}
    COLCND,AMAX,INFO)
COM PLEX A (LDA,*)
\mathbb{NTEGER M,N,KL,KU,LDA, IN FO}
REAL ROW CND,COLCND,AMAX
REALR(*),C (*)
SUBROUT\mathbb{NECGBEQU_64M,N,KL,KU,A,LDA,R,C,}
    ROW CND,COLCND,AMAX,INFO)
COM PLEX A (LDA,*)
\mathbb{NTEGER*8M,N,KL,KU,LDA,}\mathbb{N}FO
REAL ROW CND,COLCND,AMAX
REALR (*),C (*)
```


## F95 INTERFACE

```
SU BROUTINE GBEQU (M) \(\mathbb{N}, \mathbb{N}, K L, K U, A,[L D A], R, C\), ROW CND, COLCND,AMAX, [ \(\mathbb{N F O}]\) )
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, K L, K U, L D A, \mathbb{N} F O\)
REAL ::ROW CND, COLCND,AMAX
REAL,D \(\mathbb{I}\) ENSION (:) ::R,C
SUBROUTINE GBEQU_64 (M) \(\mathbb{M}, \mathbb{N}], K L, K U, A,[L D A], R, C\),
```

COM PLEX, D $\mathbb{M}$ ENSION (: : : : : A
$\mathbb{N} \operatorname{TEGER}$ (8) :: $\mathrm{M}, \mathrm{N}, \mathrm{KL}, \mathrm{KU}, \mathrm{LD} \mathrm{A}, \mathbb{N} \mathrm{FO}$
REAL ::ROW CND, COLCND, AM AX
REAL,D $\mathbb{I}$ ENSION (:) ::R,C

## C INTERFACE

\#include <sunperfh>
void cgbequ (intm, intn, intkl, intku, com plex *a, int lda, float *r, float *c, float*row cnd, float *colend, float *am ax, int*info);
void cgbequ_64 (long m, long n, long kl, long ku, com plex *a, long lda, float *r, float * C , float *row cnd, float
*colcnd, float *am ax, long *info);

## PURPOSE

cgbequ com putes row and colum n scalings intended to equilibrate an M -by-N band $m$ atrix A and reduce its condition num ber. $R$ retums the row scale factors and $C$ the colum $n$ scale factors, chosen to try to $m$ ake the largestelem ent in each row and colum $n$ of the $m$ atrix $B$ ith elem ents $\mathrm{B}(\mathrm{i}, \mathrm{j})=\mathrm{R}(\mathrm{i}) \star \mathrm{A}(\mathrm{i}, \mathrm{j}) * \mathrm{C}(\mathrm{j})$ have absolute value 1.

R (i) and C (j) are restricted to be betw een SM LN UM = sm allest safe num ber and B IG N U M = largest safe num ber. U se of these scaling factors is notguaranteed to reduce the condition num berofA butw orks w ellin practice.

## ARGUMENTS

M (input) The num ber of row s of the m atrix A. M >=0.

N (input) The num ber of colum ns of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

KL (input)
The num ber of subdiagonals $w$ thin the band of $A$. $\mathrm{KL}>=0$ 。

K U (input)
The num ber of superdiagonals $w$ ithin the band of $A$.
$K U>=0$ 。

A (input) The band $m$ atrix $A$, stored in row $s 1$ to $K L+K U+1$.
The $j$ th colum $n$ ofA is stored in the $j$ th colum $n$
of the array A as follow s: A $(k u+1+i-j)=A(i, 7)$
form ax $(1, j \mathrm{jku})<=i<=m$ in $(m, j+k l)$.
LD A (input)
The leading dim ension of the array A. LDA >= K L+KU +1.

R (output)
If $\mathbb{N} F O=0$, or $\mathbb{N} F O>M, R$ contains the row scale factors forA.
C (output)
If $\mathbb{N} F O=0, C$ contains the colum $n$ scale factors
forA.

ROW CND (output)
If $\mathbb{N} F O=0$ or $\mathbb{N} F O>M, R O W C N D$ contains the ratio
of the sm allest $R$ (i) to the largestR (i). If
ROW CND >=0.1 and AM AX is neither too large nor too sm all, it is notw orth scaling by $R$.

COLCND (output)
If $\mathbb{N} F O=0, C O L C N D$ contains the ratio of the sm allest $C$ (i) to the largestC (i). IfC O LCND >= 0.1 , it is notw orth scaling by $C$.

AM AX (output)
A bsolute value of largestm atrix elem ent. IfA M AX
is very close to overflow orvery close to under-
flow , the m atrix should be scaled.
$\mathbb{I N} F O$ (output)
= 0: successfulexit
<0: if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an ille-
galvalue
> 0 : if $\mathbb{N} F O=i$, and $i$ is
$<=\mathrm{M}$ : the i-th row ofA is exactly zero
$>M$ : the ( -H ) -th colum n of A is exactly zero

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgbm $v$-perform one of the $m$ atrix-vectoroperations $y:=$ alpha*A *x + beta*y, ory $:=$ alpha*A *x + beta* $y$, or $y:=$ alpha*cong (A ')*x + beta*y

## SYNOPSIS

```
SUBROUT\mathbb{NE CGBMV (TRANSA,M,N,NSUB,NSUPER,ALPHA,A,LDA,X,INCX,}
    BETA,Y,\mathbb{NCY)}
CHARACTER * 1 TRANSA
COM PLEX ALPHA,BETA
COM PLEX A (LDA,*),X (*),Y (*)
\mathbb{NTEGERM,N,NSUB,NSUPER,LDA, INCX,INCY}
SUBROUT\mathbb{NECGBMV_64(TRANSA,M,N,NSUB,NSUPER,ALPHA,A,LDA,X,}
    INCX,BETA,Y,\mathbb{NCY)}
CHARACTER * 1 TRANSA
COM PLEX ALPHA,BETA
COM PLEX A (LDA,*),X (*),Y (*)
\mathbb{NTEGER*8M,N,NSUB,NSUPER,LDA, INCX,INCY}
```


## F95 INTERFACE

SU BROUTINE GBM V ([TRANSA], $\mathbb{M}], \mathbb{N}], N S U B, N S U P E R, A L P H A, A,[L D A], X$, $[\mathbb{N} C X], B E T A, Y,[\mathbb{N C Y}])$

CHARACTER (LEN=1) ::TRANSA
COMPLEX ::ALPHA,BETA
COM PLEX,D $\mathbb{I M}$ ENSION (:) :: $\mathrm{X}, \mathrm{Y}$
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R:: M, N, N S U B, N S U P E R, L D A, \mathbb{N C X}, \mathbb{N} C Y$

SUBROUTINE GBM V_64 ([TRANSA], M ], $\mathbb{N}], N \operatorname{SUB}, N \operatorname{SUPER}, A L P H A, A,[L D A]$, $\mathrm{X},[\mathbb{N C X}], \mathrm{BETA}, \mathrm{Y},[\mathbb{N C Y}])$

CHARACTER (LEN=1) ::TRANSA
COM PLEX ::ALPHA,BETA
COM PLEX,D IM ENSION (:) :: X,Y
COMPLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER (8) ::M , N , N SUB,$N$ SU PER,LDA, $\mathbb{N} C X, \mathbb{N} C Y$

## C INTERFACE

\#include <sunperfh>
void cgbm v (chartransa, intm , intn, intnsub, int nsuper, com plex *alpha, com plex *a, int lda, com plex *x, int incx, com plex *beta, com plex *y, int incy);
void cgbm v_64 (char transa, long m, long n, long nsub, long nsuper, com plex *alpha, com plex *a, long lda, com plex *x, long incx, com plex *beta, complex *y, long incy);

## PURPOSE

cgbm v perform s one of the $m$ atrix-vector operations $y:=$ alpha*A *x + beta*y, ory : alpha*A *x + beta*y, or y := alpha*conjg (A ')*x + beta*y where alpha and beta are scalars, $x$ and $y$ are vectors and $A$ is an $m$ by $n$ band $m$ atrix, w ith nsub sub-diagonals and nsuper super-diagonals.

## ARGUMENTS

TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow $s$ :
TRANSA $=\mathrm{N}$ 'or h ' $\mathrm{y}:=$ alpha*A *x + beta* y .
TRANSA = ' 'or t' $y=a l p h a * A ~ * x+b e t a * y$.
TRANSA $=$ C'or $\epsilon^{\prime} y:=$ alpha*conǵ (A')*x + beta*y.
U nchanged on exit.

TRANSA is defaulted to $N$ 'forF95 $\mathbb{I N}$ TERFACE.

M (input)
On entry, M specifies the num berof rows of the m atrix A. M mustbe at least zero. U nchanged on exit.

O n entry, $N$ specifies the num ber of colum ns of the $m$ atrix A. $N$ m ustbe at least zero. U nchanged on exit.

NSUB (input)
On entry, NSUB specifies the number of subdiagonals of the matrix A.N SUB m ustsatisfy 0 le. N SU B . U nchanged on exit.

## NSUPER (input)

O n entry, N SU PER specifies the num ber of superdiagonals of them atrix A.N SU PER m ust satisfy 0 le. N SU PER. U nchanged on exit.
ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
Before entry, the leading (nsub + nsuper+1) by $n$ part of the array $A$ ustcontain the $m$ atrix of coefficients, supplied colum $n$ by colum $n$, w th the leading diagonal of the $m$ atrix in row (nsuper+ 1 ) of the array, the first super-diagonal starting at position 2 in row nsuper, the firstsubdiagonalstarting atposition 1 in row (nsuper + 2 ), and so on. Elem ents in the array A that do notcorrespond to elem ents in the band $m$ atrix (such as the top leftnsuperby nsupertriangle) are not referenced. The follow ing program segm ent $w$ illtransfera band $m$ atrix from conventional full $m$ atrix storage to band storage:

$$
\begin{aligned}
& \text { DO } 20, \mathrm{~J}=1, \mathrm{~N} \\
& \mathrm{~K}=\mathrm{N} \text { SUPER }+1-\mathrm{J} \\
& \text { DO } 10, \mathrm{I}=\mathrm{M} \text { AX }(1, \mathrm{~J}-\mathrm{NSUPER}), \mathrm{M} \mathbb{I}(\mathrm{M}, \mathrm{~J}+ \\
& \text { NSUB }) \\
& \quad \text { A }(\mathrm{K}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \quad \text { CONTINUE } \\
& 20 \text { CONTINUE }
\end{aligned}
$$

U nchanged on exit.

## LDA (input)

On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A m ust be at least (nsub + nsuper+ 1 ). Unchanged on exit.

X (input)
$(1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C X))$ when TRANSA $=\mathrm{N}$ 'or
$h^{\prime}$ and at least ( $\left.1+(m-1) * a b s(\mathbb{N} C X)\right)$ otherw ise. Before entry, the increm ented array $X$ $m$ ust contain the vectorx. U nchanged on exit.
$\mathbb{N C X}$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of $\mathrm{X} . \mathbb{N} C X$ m ustnotbe zero. U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then Y need notbe seton input. U nchanged on exit.

Y (input/output)
$(1+(m-1) \star \operatorname{abs}(\mathbb{N} C Y))$ when TRANSA $=\mathrm{N}$ 'or $h^{\prime}$ and at least ( $\left.1+(\mathrm{n}-1)^{\star} \operatorname{abs}(\mathbb{N} C Y)\right)$
otherw ise. Before entry, the increm ented array $Y$ m ust contain the vectory. On exit, Y is overw ritten by the updated vectory.
$\mathbb{N C Y}$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $Y . \mathbb{N} C Y$ m ustnotbe zero. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgbrfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is banded, and provides errorbounds and backw ard error estim ates for the solution

## SYNOPSIS

```
SUBROUT\mathbb{NE CGBRFS (TRANSA,N,KL,KU,NRHS,A,LDA,AF,LDAF,}
    \mathbb{PIVOT,B,LDB,X,LDX,FERR,BERR,WORK,W ORK 2,\mathbb{NFO)}}\mathbf{~}=\textrm{L}
CHARACTER * 1 TRANSA
COM PLEX A (LDA,*),AF (LDAF,*),B (LDB,*),X (LDX,*),W ORK (*)
\mathbb{NTEGERN,KL,KU,NRHS,LDA,LDAF,LDB,LDX,}\mathbb{N}FO
INTEGER \mathbb{PIVOT (*)}
REAL FERR (*),BERR (*),W ORK ( (*)
SU BROUT\mathbb{NE CGBRFS_64 (TRANSA,N,KL,KU,NRHS,A,LDA,AF,LDAF,}
    \mathbb{PIVOT,B,LDB,X,LDX,FERR,BERR,W ORK,W ORK 2, INFO)}
```

CHARACTER * 1 TRANSA
COM PLEX A (LDA,*),AF (LDAF,*), B (LDB,*), X (LDX,*), W ORK (*)
$\mathbb{N} T E G E R * 8 N, K L, K U, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O$
$\mathbb{N}$ TEGER*8 $\mathbb{P} \mathbb{I V O T}$ ( )
REAL FERR (*), BERR (*), WORK 2 (*)

## F95 INTERFACE

SUBROUTINE GBRFS ([TRANSA], $\mathbb{N}], K L, K U, \mathbb{N} R H S], A,[L D A], A F$,
 [ $\mathbb{N} F \mathrm{FO}$ ])

CHARACTER (LEN=1) ::TRANSA
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK

COM PLEX, D $\mathbb{M}$ ENSION (: : : : : A , AF, B , X
$\mathbb{N} T E G E R:: N, K L, K U, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{I} O T$
REAL,D $\mathbb{M}$ ENSION (:) ::FERR,BERR,WORK2

SU BROUTINE GBRFS_64 ([TRANSA ], $\mathbb{N}], K L, K U,[N R H S], A,[L D A]$, $A F,[L D A F], \mathbb{P} \mathbb{I} O T, B,[L D B], X,[L D X], F E R R, B E R R,[W O R K]$, [W ORK 2], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1) ::TRANSA
COMPLEX,D $\mathbb{M}$ ENSION (:) ::W ORK
COM PLEX , D $\mathbb{M}$ ENSION (:,:) ::A,AF,B,X
$\mathbb{N} T E G E R(8):: N, K L, K U, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N F O}$
$\mathbb{N} \operatorname{TEGER}$ (8), D $\mathbb{M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT}$
REAL,D $\mathbb{M}$ ENSION (:) ::FERR,BERR,W ORK2

## C INTERFACE

\#include <sunperfh>
void ogbrfs (chartransa, intn, intkl, int ku, int nrhs, com plex *a, int lda, com plex *af, int ldaf, int *ípi̇vot, com plex *b, int ldb, com plex *x, int ldx, float *ferr, float *berr, int*info);
void ogbrfs_64 (chartransa, long n, long kl, long ku, long nrhs, complex *a, long lda, complex *af, long ldaf, long *ípivot, com plex *b, long ldb, com plex *x, long ldx, float *ferr, float *berr, long *info);

## PURPOSE

cgbrfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is banded, and provides errorbounds and backw ard error estim ates for the solution.

## ARGUMENTS

TRANSA (input)
Specifies the form of the system ofequations:
$=\mathrm{N}: A * X=B \quad$ (Notranspose)
$=T$ ': $\mathrm{A} * * \mathrm{~T} * \mathrm{X}=\mathrm{B} \quad$ ( T ranspose)
$=C$ ': $A * * H * X=B \quad$ (C onjugate transpose)

TRANSA is defaulted to N 'forF95 $\mathbb{I N T E R F A C E .}$

N (input) The order of them atrix A. N $>=0$.

KL (input)
The num berof subdiagonals w thin the band of A. $\mathrm{KL}>=0$.

KU (input)
The num ber of superdiagonals w ithin the band of A. $K U>=0$.

NRHS (input)
The num berof righthand sides, i.e., the num ber of collm ns of the m atrioes B and X. NRH S >=0.

A (input) The originallband $m$ atrix $A$, stored in row $s$ to $K L+K U+1$. The $j$ th column of $A$ is stored in the $j$ th colum $n$ of the array A as follow s: A (ku+1+i$j \gg)=A(i, j)$ form ax $(1, j k u)<=i<=m$ in $(n, j+k l)$.

LD A (input)
The leading dim ension of the anay A. LDA >= $K L+K U+1$.

AF (input)
D etails of the LU factorization of the band $m$ atrix A, as com puted by CGBTRF. U is stored as an upper triangularband $m$ atrix $w$ ith $\mathrm{KL}+\mathrm{KU}$ superdiagonals in row $s 1$ to $K L+K U+1$, and the $m$ ultipliers used during the factorization are stored in row S $K L+K U+2$ to $2 * K L+K U+1$.

LDAF (input)
The leading dim ension of the array AF. LDAF >= $2 * K L * K U+1$.

PIVOT (input)
The pivotindices from CGBTRF; for $1<=i<=N$, row $i$ of the $m$ atrix $w$ as interchanged $w$ th row $\mathbb{P}$ IV OT (i).
$B$ (input) The righthand side $m$ atrix $B$.
LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.

X (input/output)
On entry, the solution $m$ atrix $X$, as com puted by CGBTRS. On exit, the im proved solution $m$ atrix $X$.

## LD X (input)

The leading dim ension of the array X . LD X >=
$\max (1, \mathbb{N})$.
FERR (output)
The estim ated forw ard errorbound for each solution vector $X()$ ) the $j$ th column of the solution $m$ atrix X). If XTRUE is the true solution comesponding to $X(\mathcal{O})$, $\operatorname{FERR}(\mathcal{)}$ is an estim ated upperbound for the $m$ agnitude of the largest ele$m$ ent in ( $X(\mathcal{\nu})-X$ TRU $E$ ) divided by the $m$ agnitude of the largestelem ent in $X(\mathcal{j})$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

BERR (output)
The com ponentw ise relative backw ard emor of each
solution vectorX (j) (ie., the sm allest relative change in any elem entofA orB thatm akes X ( $)$ ) an exactsolution).

W ORK (w orkspace)
dim ension $\left(2{ }^{*} \mathrm{~N}\right)$
W ORK 2 (w orkspace)
dim ension (N)
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
< 0 : if $\mathbb{N} F O=-i$, the $i$-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgbsv - com pute the solution to a com plex system of linear equations $A * X=B$, where $A$ is a band $m$ atrix of order $N$ $w$ th $K L$ subdiagonals and $K U$ superdiagonals, and $X$ and $B$ are N -by-N R H S m atrices

## SYNOPSIS



```
COM PLEX A (LDA,*),B (LDB,*)
INTEGER N,KL,KU,NRHS,LDA,LDB, INFO
INTEGER \mathbb{PIVOT (*)}
SUBROUT\mathbb{NECGBSV_64 N,KL,KU,NRHS,A,LDA,\mathbb{P IVOT,B,LDB,}}\mathbf{N},\textrm{N},\textrm{K}
    \mathbb{NFO)}
COM PLEX A (LDA,*),B (LDB,*)
NNTEGER*8N,KL,KU,NRHS,LDA,LDB,NNFO
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
```


## F95 INTERFACE

SU BROUTINE GBSV ( $\mathbb{N}], K L, K U, \mathbb{N} R S], A,[L D A], \mathbb{P} \mathbb{I} O T, B,[L D B]$, [ $\mathbb{N}$ FO ])

COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A, B
$\mathbb{N} T E G E R:: N, K L, K U, N R H S, L D A, L D B, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T$
 [LDB], [ $\mathbb{N F O}$ ])

## C INTERFACE

\#include <sunperfh>
void cgbsv (intn, int kl, int ku, intnrhs, com plex *a, int lda, int *ịívot, com plex *b, int ldb, int *info);
void cgbsv_64 long n, long kl, long ku, long nrhs, com plex
*a, long lda, long *ipívot, com plex *b, long ldb, long *info);

## PURPOSE

cgbsv com putes the solution to a com plex system of linear equations $A$ * $X=B$, where $A$ is a band $m$ atrix oforder $N$ $w$ th $K L$ subdiagonals and $K U$ superdiagonals, and $X$ and $B$ are N -by-N R H S m atrices.

The LU decom position $w$ ith partialpivoting and row interchanges is used to factorA asA $=\mathrm{L} * \mathrm{U}$, where L is a productof perm utation and unit low er triangularm atrices $w$ th $K L$ subdiagonals, and $U$ is uppertriangularw ith K L+K U superdiagonals. The factored form of $A$ is then used to solve the system of equations $A * X=B$.

## ARGUMENTS

N (input) The num ber of linearequations, i.e., the order of them atrix $A . N>=0$.

KL (input)
The num ber of subdiagonals w ithin the band of A. $K L>=0$.

KU (input)
The num ber of superdiagonals $w$ ithin the band of $A$. $K U>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS >=0.

A (input/output)
On entry, the m atrix A in band storage, in row s
$K L+1$ to $2 * K L+K U+1$; row s 1 to $K L$ of the anay need notbe set. The $j$ th column of A is stored in the $j$ th collumn of the array A as follows: $A(K L+K U+1+i-j, j)=A(i, j)$ for $\max (1, j$ $K U)<=i<=m$ in ( $N, j+K L$ ) On exit, details of the factorization: $U$ is stored as an upper triangular band $m$ atrix $w$ ith $K L+K U$ superdiagonals in row $s 1$ to $K L+K U+1$, and the $m$ ultipliers used during the factorization are stored in rows $K L+K U+2$ to $2 \star K L+K U+1$. See below for further details.

LD A (input)
The leading dim ension of the array A. LDA >= $2 * K L+K U+1$.
$\mathbb{P I V O T}$ (output)
The pivot indices that define the perm utation $m$ atrix $P$; row i of the $m$ atrix $w$ as interchanged w ith row $\mathbb{P I V O T}$ (i).

B (input/output)
On entry, the $N-b y-N R H S$ righthand sidem atrix $B$. On exit, if $\mathbb{N} F O=0$, the $N$ by $-N R H S$ solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the aray $B$. LD B >= $\max (1, N)$.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum ent had an illegalvalue
$>0:$ if $\mathbb{N} F O=i, U(i, i)$ is exactly zero. The factorization has been com pleted, but the factor U is exactly singular, and the solution has notbeen com puted.

## FURTHER DETAILS

The band storage scheme is illustrated by the follow ing exam ple, when $M=N=6, K L=2, K U=1$ :

On entry: Onexit:
u36

*     *         +             +                 +                     +                         *                             * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55 u66
a21 a32 a43 a54 a65 * m 21 m 32 m 43 m 54 m 65 * a31 a42 a53 a64 * * m31 m 42 m 53 m 64 * *

A rray elem entsm arked * are notused by the routine; ele$m$ entsm arked + need notbe seton entry, but are required by the routine to store elem ents of $U$ because of fill-in resulting from the row interchanges.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgbsvx -use the LU factorization to com pute the solution to a complex system of linearequations $\mathrm{A} * \mathrm{X}=\mathrm{B}, \mathrm{A} * * \mathrm{~T} * \mathrm{X}=$ $B$, orA ${ }^{* *} H * X=B$,

## SYNOPSIS

```
SUBROUT\mathbb{NE CGBSVX EACT,TRANSA,N,KL,KU,NRHS,A,LDA,AF,}
    LDAF,\mathbb{PIVOT,EQUED,R,C,B,LDB,X,LDX,RCOND,FERR,}
    BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 FACT,TRANSA,EQUED
COM PLEX A (LDA,*),AF (LDAF,*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,KL,KU,NRHS,LDA,LDAF,LDB,LDX,INFO
\mathbb{NTEGER \mathbb{PIVOT (*)}}\mathbf{(})
REALRCOND
REALR (*),C (*),FERR (*),BERR (*),WORK 2 (*)
SUBROUT\mathbb{NECGBSVX_64&ACT,TRANSA,N,KL,KU,NRHS,A,LDA,AF,}
    LDAF,\mathbb{PIVOT,EQUED,R,C,B,LDB,X,LDX,RCOND,FERR,}
    BERR,W ORK,W ORK2,INFO)
CHARACTER * 1FACT,TRANSA,EQUED
COM PLEX A (LDA,*),AF (LDAF,*),B (LD B,*),X (LDX,*),W ORK (*)
INTEGER*8N,KL,KU,NRHS,LDA,LDAF,LDB,LDX,INFO
NNTEGER*8 \mathbb{PIVOT (*)}
REALRCOND
REALR (*),C (*),FERR (*),BERR (*),W ORK 2 (*)
```


## F95 INTERFACE

SU BROUTINE GBSVX $\mathbb{E A C T},[T R A N S A], \mathbb{N}], K L, K U, \mathbb{N R H S}], A,[L D A]$, AF, [LDAF], $\mathbb{P} \mathbb{V} O T, E Q U E D, R, C, B,[L D B], X,[L D X]$, RCOND,FERR,BERR, [W ORK], [W ORK 2], [ $\mathbb{N} F O]$ )

CHARACTER ( $\amalg E N=1):: F A C T, T R A N S A, E Q U E D$
COM PLEX , D $\mathbb{M}$ ENSION (:) ::W ORK
COM PLEX, D $\mathbb{M}$ ENSION (: : : : : A, AF, B, X
$\mathbb{N}$ TEGER :: N, KL, KU, NRHS,LDA,LDAF,LDB,LDX, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T$
REAL ::RCOND
REAL,D $\mathbb{I M} E N S I O N(:):: R, C, F E R R, B E R R, W$ ORK 2

SU BROUTINE GBSVX_64 (FACT, [TRANSA], $\mathbb{N}], K L, K U, N R H S], A$, $[L D A], A F,[L D A F], \mathbb{P} \mathbb{I} O T, E Q U E D, R, C, B,[L D B], X,[L D X]$, RCOND, FERR, BERR, $\mathbb{W}$ ORK], $\mathbb{W} O R K 2],[\mathbb{N F O}])$

CHARACTER (LEN=1) ::FACT,TRANSA,EQUED
COM PLEX,DIM ENSION (:) ::WORK
COM PLEX, D $\mathbb{M} E N S I O N(:,:): A, A F, B, X$
$\mathbb{N}$ TEGER (8) :: N , KL, KU ,NRHS,LDA, LDAF,LDB,LDX, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$
REAL :: RCOND
REAL,D $\mathbb{I}$ ENSION (:) ::R, C,FERR,BERR,W ORK 2

## C INTERFACE

\#include <sunperfh>
void ogbsvx (char fact, chartransa, intn, intkl, int ku, int nuhs, complex *a, int lda, com plex *af, int ldaf, int*ipivot, char equed, float * $r$, float * c, com plex *b, int ldb, com plex *x, int ldx, float *rcond, float *ferr, float *berr, int*info);
void cgbsvx_64 (char fact, chartransa, long n, long kl, long ku, long nrhs, com plex *a, long lda, com plex *af, long ldaf, long *íivot, char equed, float *r, float *${ }^{\text {C }}$, com plex ${ }^{*} \mathrm{~b}$, long ldlo, com plex ${ }^{*}$ x, long ldx, float *rcond, float *ferr, float*berr, long *info);

## PURPOSE

cgbsvx uses the LU factorization to com pute the solution to a complex system of linearequations $A * X=B, A * * T * X=$ $B$, orA $* * H * X=B, w h e r e A$ is a band $m$ atrix of order $N w$ ith $K L$ subdiagonals and KU superdiagonals, and $X$ and $B$ are $N$ -by-N R H S m atrices.

Emrorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed by this subroutine:

1. IfFACT = E', real scaling factors are com puted to equilibrate
the system :
TRANS $=N^{\prime}: \operatorname{diag}(\mathbb{R}) * A * \operatorname{diag}(C) \quad * \operatorname{inv}(\operatorname{diag}(C)) * X=$ diag (R)*B

TRANS $=T$ ': $(\operatorname{diag}(R) * A * \operatorname{diag}(C)) * * T * \operatorname{inv}(\operatorname{diag}(R)) * X=$ diag (C ) *B

TRANS = C': (diag $(\mathbb{R}) * A * \operatorname{diag}(C)) * * H * \operatorname{inv}(\operatorname{diag}(\mathbb{R})) * X=$ diag (C)*B
W hether or not the system w illbe equilibrated depends on the scaling of the m atrix A, but if equilibration is used, A is overw rilten by diag $(\mathbb{R}) * A$ *diag ( $C$ ) and $B$ by diag $(\mathbb{R}) * B$ (if TRANS = N ) ordiag (C)*B (if TRANS = 'T'orC). 2. IfFACT $=\mathrm{N}$ 'or E ', the LU decom position is used to factor the m atrix A (afterequilibration ifFACT =E) as $A=L * U$, where $L$ is a product of perm utation and unit low er triangular
$m$ atrices $w$ ith KL subdiagonals, and U is upper triangular w ith K L+KU superdiagonals.
3. If som e $U(i, i)=0$, so that $U$ is exactly singular, then the routine
retums w ith $\mathbb{N} F O=$ i. O therw ise, the factored form of A is used
to estim ate the condition num ber of the $m$ atrix $A$. If the reciprocal of the condition num ber is less than $m$ achine precision,
$\mathbb{N} F O=N+1$ is retumed as a waming, but the routine stillgoes on
to solve for $X$ and com pute emor bounds as described below .
4.The system of equations is solved for $X$ using the factored form
of A.
5. Iterative refinem ent is applied to im prove the com puted solution
$m$ atrix and calculate error bounds and backw ard error estim ates
for it.
6. Ifequilibration w as used, the $m$ atrix $X$ is prem ultiplied
by
diag $(C)$ (ifTRANS $=N$ ) ordiag $(\mathbb{R})$ (ifTRANS $=T$ 'or C) so
that itsolves the originalsystem before equilibration.

## ARGUMENTS

FACT (input)
Specifies w hether ornotthe factored form of the $m$ atrix $A$ is supplied on entry, and ifnot, whether the $m$ atrix A should be equilibrated before it is factored. = F ': On entry, AF and $\mathbb{P}$ IV O T contain the factored form of . IfEQUED is not $N$ ', the $m$ atrix A has been equilibrated $w$ ith scaling factors given by R and $\mathrm{C} . \mathrm{A}, \mathrm{AF}$, and $\mathbb{P}$ IV OT are not m odified. $=\mathrm{N}$ ': Them atrix A w ill.be copied to A F and factored.
= E ': The matrix A will be equilibrated if necessary, then copied to A F and factored.

TRANSA (input)
Specifies the form of the system of equations. $=$
$\mathrm{N}: A * X=B \quad$ N o transpose)
$=T$ ': A**T * $\mathrm{X}=\mathrm{B} \quad$ ( T ranspose)
$=C: A * * H * X=B \quad$ (C onjugate transpose)

TRANSA is defaulted to $N$ 'forF95 $\mathbb{I N T E R F A C E .}$

N (input) The num ber of linearequations, ie., the order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

K L (input)
The num ber of subdiagonals $w$ thin the band of $A$. $\mathrm{K} L>=0$ 。

K U (input)
The num ber of superdiagonals $w$ ithin the band of $A$. $K U>=0$ 。

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the m atrioes B and X. NRHS >=0.

A (input/output)
On entry, the m atrix A in band storage, in row s 1 to $K L+K U+1$. The $j$ th colum $n$ of $A$ is stored in the $j$ th colum $n$ of the array A as follow s: A (KU+1+i$j, j)=A(i, j)$ form $a x(1, j K U)<=i<=m$ in $(\mathbb{N}, j+k l)$

IfFACT = F'and EQUED is not $N^{\prime}$, then $A$ must
have been equilibrated by the scaling factors in $R$ and/orC. A is notm odified ifFACT = F'or N',
orifFACT = E'andEQUED $=\mathrm{N}$ 'on exit.
Onexit, ifEQUED ne. $N$ ', A is scaled as follow s: $E Q U E D=R$ ': A : $=\operatorname{diag}(\mathbb{R}) * A$
EQUED = C': A : A * diag (C)
EQUED = B ': A := diag (R)*A * diag (C ).
LD A (input)
The leading dim ension of the array A. LDA >= $K L+K U+1$.
AF (input/output)
If FACT = $F$ ', then AF is an input argum entand on entry contains details of the LU factorization of the band $m$ atrix $A$, as com puted by CGBTRF. U is stored as an upper triangularband $m$ atrix $w$ ith
$K L+K U$ superdiagonals in row $s 1$ to $K L+K U+1$, and the m ultipliers used during the factorization are stored in row sKL+KU+2 to $2 \star \mathrm{~K} L+K U+1$. If EQUED ne. $N$ ', then $A F$ is the factored form of the equilibrated $m$ atrix $A$.

IfFACT $=\mathrm{N}$ ', then $A F$ is an output argum ent and on exit retums details of the LU factorization of A.

IfFACT $=\mathrm{E}$ ', then AF is an output argum ent and on exit retums details of the LU factorization of the equilibrated $m$ atrix $A$ (see the description of $A$ for the form of the equilibrated $m$ atrix).

LD AF (input)
The leading dim ension of the array AF. LDAF >= $2 * K L+K U+1$.
$\mathbb{P} \mathbb{V} O T$ (input)
IfFACT=F', then $\mathbb{P I V O T}$ is an input argum ent and on entry contains the pivot indioes from the factorization $\mathrm{A}=\mathrm{L} * \mathrm{U}$ as com puted by CGBTRF ; row i of the $m$ atrix $w$ as interchanged $w$ ith row $\mathbb{P}$ IV OT (i).

IfFACT $=\mathrm{N}^{\prime}$, then $\mathbb{P} \mathbb{I V} \operatorname{T}$ is an output argum ent and on exit contains the pivot indioes from the factorization $A=L * U$ of the originalm atrix $A$.

IfFACT = E ', then $\mathbb{P}$ IV OT is an output argum ent and on exit contains the pivot indices from the factorization $A=L * U$ of the equilibrated $m$ atrix
A.

EQUED (input)
Specifies the form of equilibration thatw as done.
$=\mathrm{N}^{\prime}$ : N o equilibration (alw ays true ifFACT =
N 7 。
$=R$ ': R ow equilibration, i.e., A has been
prem ultiplied by diag $(\mathbb{R}) .=\mathrm{C}$ : C olum $n$ equilibration, ie., A has been postm ultiplied by diag $(C)$. $=B$ ': B oth row and colum $n$ equilibration, ie., A has been replaced by diag $(R)$ * A * diag (C). EQUED is an inputargum entifFACT= $F$ '; otherw ise, it is an output argum ent.

R (input/output)
The row scale factors forA. IfEQUED $=R^{\prime}$ or $B$ ', A is multiplied on the leftby diag $(\mathbb{R})$; if $E Q U E D=N$ 'or $C$ ', $R$ is notaccessed. $R$ is an input argum ent iffACT = F ; otherw ise, R is an outputargum ent. IfFACT $=\mathrm{F}^{\prime}$ and EQUED $=\mathrm{R}^{\prime}$ or B ', each elem entofR m ustbe posilive.

C (input/output)
The colum n scale factors forA. IfEQUED = C 'or $B$ ', A ism ultiplied on the rightby diag (C); if EQUED $=N$ 'or $R$ ', $C$ is notaccessed. $C$ is an input argum ent ifFACT $=\mathrm{F}$; otherw ise, C is an outputargum ent. IfFACT = $\mathrm{F}^{\prime}$ and EQUED $=\mathrm{C}^{\prime}$ or B ', each elem entofC mustbe positive.

B (input/output)
On entry, the righthand side m atrix B. On exit, ifEQUED $=N^{\prime}$ ', $B$ is notm odified; ifTRANSA $=N^{\prime}$ and EQUED $=R^{\prime}$ or $B^{\prime}, B$ is overw ritten by $\operatorname{diag}(\mathbb{R}) * B$; if TRANSA $=T^{\prime}$ or $C^{\prime}$ and EQUED $=C^{\prime}$ or $B \prime, B$ is overw rilten by diag $(C) * B$.

LD B (input)
The leading dim ension of the array B . LD B >= $\max (1, N)$.

X (output)
If $\mathbb{N}$ FO $=0$ or $\mathbb{N} F O=N+1$, the $N$ boy-NRHS solution $m$ atrix $X$ to the original system of equations. $N$ ote that $A$ and $B$ arem odified on exit if EQUED ne. $N$ ', and the solution to the equilibrated system is inv (diag (C))*X ifTRANSA = N 'and EQUED $=C$ 'or $B$ ', orinv (diag $(\mathbb{R})) * X$ ifTRANSA $=T$ 'or C'andEQUED = R 'or $\mathrm{B}^{\prime}$.

LD $X$ (input)
The leading dim ension of the array X . LD X >= $\max (1, N)$.

RCOND (output)
The estim ate of the reciprocal condition num ber of the $m$ atrix $A$ afterequilibration (if done). If
RCOND is less than them achine precision (in particular, ifRCOND $=0$ ), the $m$ atrix is singular to w orking precision. This condition is indicated by a retum code of $\mathbb{N}$ FO $>0$.

FERR (output)
The estim ated forw ard errorbound for each solution vector $X(\mathcal{I})$ the $j$ th column of the solution matrix X). If XTRUE is the true solution corresponding to $X(\mathcal{H}), \operatorname{FERR}(\mathcal{H})$ is an estim ated upperbound for the $m$ agnitude of the largest ele$m$ ent in ( $\mathrm{X}(\mathcal{i})-\mathrm{X}$ TRUE) divided by the m agnitude of the largestelem ent in $X(\mathcal{J})$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

## BERR (output)

The com ponentw ise relative backw ard error of each solution vector X (i) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( $\mathcal{j}$ ) an exactsolution).

W ORK (w orkspace)
dim ension ( $2 * \mathrm{~N}$ )
W ORK 2 (w orkspace)
dimension (N) On exit, WORK2(1) contains the reciprocal pivot grow th factornorm (A)/norm (U). The "max absolute element" norm is used. If W ORK2(1) is much less than 1 , then the stability of the LU factorization of the (equilibrated)
$m$ atrix A could be poor. This also $m$ eans that the solution X, condition estim atorRCOND, and forw ard error bound FERR could be unreliable. If factorization fails w ith $0<\mathbb{N} F O<=N$, then $W$ ORK 2 (1) contains the reciprocal pivot grow th factor for the leading $\mathbb{N}$ FO colum ns of .
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
< 0: if $\mathbb{N}$ FO = -i, the i-th argum ent had an illegalvalue
$>0:$ if $\mathbb{N} F O=i$, and $i$ is
$<=\mathrm{N}: \mathrm{U}(\mathrm{i}, \mathrm{i})$ is exactly zero. The factorization has been com pleted, but the factor U is exactly singular, so the solution and emor bounds could not be com puted. R COND $=0$ is retumed. $=\mathrm{N}+1: \mathrm{U}$ is nonsingular, butRCOND is less than machine precision, $m$ eaning that the $m$ atrix is singular to w orking precision. N evertheless, the solution and error bounds are com puted because there are a num ber of situationsw here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgbtfe -com pute an LU factorization of a complex $m$-by-n band $m$ atrix A using partialpivoting $w$ th row interchanges

## SYNOPSIS



```
COM PLEX AB (LDAB,*)
INTEGERM,N,KL,KU,LDAB,INFO
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(*)}
```



```
COM PLEX AB (LDAB,*)
INTEGER*8 M ,N,KL,KU,LDAB,INFO
\mathbb{NTEGER*8 \mathbb{PIV (*)}}\mathbf{(})
F95 INTERFACE
```



```
COM PLEX,D IM ENSION (:,:) ::AB
\mathbb{NTEGER ::M,N,KL,KU,LDAB,NNFO}
INTEGER,D\mathbb{M ENSION (:) :: \mathbb{PIV}}\mathbf{N}/\mp@code{N}
```



```
COM PLEX,D IM ENSION (:,:) ::AB
INTEGER (8)::M,N,KL,KU,LDAB,\mathbb{NFO}
INTEGER (8),D \mathbb{M ENSION (:) ::\mathbb{PIV}}\mathbf{N}=\mp@code{N}
```

\#include <sunperfh>
void cgbtf2 (intm, intn, intkl, intku, com plex *ab, int ldab, int *ịiv, int *info);
void cgbtf2_64 (long m, long n, long kl, long ku, com plex *ab, long ldab, long *ipiv, long *info);

## PURPOSE

cgbtf2 com putes an LU factorization of a com plex $m$-by-n band $m$ atrix A using partialpivoting $w$ ith row interchanges.

This is the unblocked version of the algorithm , calling Level2 BLAS.

## ARGUMENTS

M (input) The num ber of row s of the matrix $A . M>=0$.

N (input) The num ber of colum ns of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

KL (input)
The num ber of subdiagonals w ithin the band of A.
$K L>=0$ 。

KU (input)
The num ber of superdiagonals $w$ thin the band of $A$.
$K U>=0$ 。

A B (input/output)
O n entry, the m atrix A in band storage, in row s $K L+1$ to $2 * K L+K U+1$; row $s 1$ to $K L$ of the anay need notbe set. The $j$ th colum $n$ of $A$ is stored in the jth column of the array $A B$ as follow s: $A B(k l+k u+1+i-j)=A(i, j)$ for $\max (1, j$ $\mathrm{ku})<=\mathrm{i}<=\mathrm{m}$ in $(\mathrm{m}, \mathrm{j}+\mathrm{kl})$

O $n$ exit, details of the factorization: $U$ is stored as an upper triangular band $m$ atrix $w$ th $K L+K U$ superdiagonals in row s 1 to $\mathrm{K} L+K \mathrm{U}+1$, and the m ultipliers used during the factorization are stored in row $s K L+K U+2$ to $2 * K L+K U+1$. See below for furtherdetails.

LDAB (input)
The leading dim ension of the array AB. LDAB >=
$2 * K L+K U+1$.

IPIV (output)
The pivot indices; for $1<=i<=m$ in $M, N$ ), row i of the $m$ atrix $w$ as interchanged $w$ ith row $\mathbb{P} \mathbb{V}$ (i).
$\mathbb{N} F O$ (output)
$=0$ : successfulexit
$<0$ : if $\mathbb{N N F O}=-$ i, the $i$-th argum ent had an illegalvalue $>0$ : if $\mathbb{N} F O=+i, U(i, i)$ is exactly zero. The factorization has been com pleted, but the factor $U$ is exactly singular, and division by zero will occur if it is used to solve a system ofequations.

## FURTHER DETAILS

The band storage scheme is illustrated by the follow ing exam ple, $w$ hen $M=N=6, K L=2, K U=1$ :

On entry: On exit:

```
    * * * + + + * * * u14 u25
u36
    * * + + + + * * u13 u24 u35
u46
    * a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
    a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55
u66
    a21 a32 a43 a54 a65 * m 21 m 32 m 43 m 54 m 65
*
    a31 a42 a53 a64 * * m 31 m 42 m 53 m 64 *
```

* 

A ray elem entsm arked * are notused by the routine; ele$m$ entsm arked + need notbe seton entry, but are required by the routine to store elem ents of $U$, because of fill-in resulting from the row interchanges.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgbtrf-com pute an LU factorization of a complex $m$-by-n band $m$ atrix A using partialpivoting $w$ th row interchanges

## SYNOPSIS

```
SUBROUT\mathbb{NE CGBTRF(M,N,KL,KU,AB,LDAB,\mathbb{PIVOT,INFO)}}\mathbf{N}={
COM PLEX AB (LDAB,N)
INTEGERM,N,KL,KU,LDAB,INFO
INTEGER \mathbb{PIVOTM IN M NN)}
```



```
COM PLEX AB (LDAB,N)
INTEGER*8M,N,KL,KU,LDAB,INFO
\mathbb{NTEGER*8 \mathbb{PIVOTM}\mathbb{N}M,N))}
```


## F95 INTERFACE



```
COM PLEX,D IM ENSION (:,:) ::AB
\mathbb{NTEGER ::M,N,KL,KU,LDAB,NNFO}
\mathbb{NTEGER,D IM ENSION (:) ::\mathbb{PIVOT}}\mathbf{T}\mathrm{ (: }
```



```
COM PLEX,D IM ENSION (:,:) ::AB
\mathbb{NTEGER (8)::M ,N,KL,KU,LDAB,INFO}
\mathbb{NTEGER (8),D IM ENSION (:)::\mathbb{PIVOT}}\mathbf{~}=\mp@code{M}
```

void cgbtrf(intm , intn, int kl, int ku, com plex *ab, int ldab, int *ipivot, int*info);
void cgbtrf_ 64 (long m, long n, long kl, long ku, com plex *ab, long ldab, long *ipívot, long *info);

## PURPOSE

cgbtrf com putes an LU factorization of a com plex $m$-by-n band $m$ atrix A using partialpivoting $w$ ith row interchanges.

This is the blocked version of the algorithm, calling Level 3 BLAS.

## ARGUMENTS

M (input) Integer
The num ber of row sof the m atrix A. M $>=0$.
N (input) Integer
The num berof colum ns of the m atrix A. N $>=0$.

K L (input) Integer
The num berof subdiagonals w thin the band of A. $K L>=0$.

KU (input) Integer
The num ber of superdiagonals $w$ ith in the band of A.
$K U>=0$.
AB (input/output) Com plex array ofdim ension (LDAB,N).
On entry, the $m$ atrix A in band storage, in row $s$ $K L+1$ to $2 * K L+K U+1$; row s 1 to $K L$ of the array need not.be set. The J-th colum n ofA is stored in the $J$ th column of the anay $A B$ as follow $s$ : AB $(\mathbb{K} L+K U+1+I J, J)=A(I, N)$ for MAX $(1, J$ $K U)<=\mathbb{K}=M \mathbb{N}(M, N+K L)$

O n exit, details of the factorization: U is stored as an upper triangular band $m$ atrix $w$ ith $K L+K U$ superdiagonals in row s 1 to $K L+K U+1$, and the $m u l-$ tipliers used during the factorization are stored in row $S K+K U+2$ to $2 * K L+K U+1$. See below for furtherdetails.

LD AB (input) Integer
The leading dim ension of the array AB. LDAB >= $2 * K L+K U+1$.
$\mathbb{P} \mathbb{I V O T}$ (output) Integerarray ofdim ension $M \mathbb{I N} M, N$ )
The pivotindioes; for $1<=I<=M \mathbb{N} M, N$ ), row $I$ of the $m$ atrix $w$ as interchanged $w$ ith row $\mathbb{P I V O T}$ (I).
$\mathbb{N}$ FO (output) Integer
= 0 : successfulexit
$<0:$ if $\mathbb{N} F O=-$ I, the I-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N} F O=+\mathbb{I}, U(I, I)$ is exactly zero. The factorization has been com pleted, but the factorU is exactly singular, and division by zero will occur if it is used to solve a system ofequations.

## FURTHER DETAILS

The band storage scheme is illustrated by the follow ing exam ple, when $M=N=6, K L=2, K U=1$ :

On entry: On exit:

*     *         *             +                 +                     +                         *                             *                                 * u14 u25
u36
*     *         +             +                 +                     +                         *                             * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55
u66
a21 a32 a43 a54 a65 * m 21 m 32 m 43 m 54 m 65
* 

a31 a42 a53 a64 * * m31 m 42 m 53 m 64 *
*

A ray elem entsm arked * are notused by the routine; ele$m$ entsm arked + need notbe seton entry, but are required by the routine to store elem ents of $U$ because of fill-in resulting from the row interchanges.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
cgbtrs-solve a system of linear equations A * X = B,A **T
```

* $\mathrm{X}=\mathrm{B}$, or $\mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B}$ w ith a generalband m atrix A using the LU factorization com puted by CGBTRF


## SYNOPSIS

```
SU BROUT\mathbb{NE CGBTRS (TRANSA,N,NSUB,NSUPER,NRHS,A,LDA, IPIVOT,B,}
    LDB,\mathbb{NFO )}
CHARACTER * 1 TRANSA
COM PLEX A (LDA,*),B (LD B,*)
INTEGERN,NSUB,NSUPER,NRHS,LDA,LDB, INFO
INTEGER \mathbb{PIVOT (*)}
SUBROUT\mathbb{NECGBTRS_64(TRANSA,N,NSUB,NSUPER,NRHS,A,LDA,\mathbb{PIVOT,}}\mathbf{N},\textrm{N},\textrm{N}
    B,LDB,INFO)
CHARACTER * 1 TRANSA
COM PLEX A (LDA,*),B (LDB,*)
NNTEGER*8 N,NSUB,NSUPER,NRHS,LDA,LDB,INFO
\mathbb{NTEGER*8 P\mathbb{IVOT (*)}}\mathbf{(})
```


## F95 INTERFACE

SU BROUTINE GBTRS ([TRANSA], $\mathbb{N}], N S U B, N S U P E R, ~ N R H S], A,[L D A]$, $\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}],[\mathbb{N F O}])$

CHARACTER (LEN=1) ::TRANSA
COM PLEX,D $\mathbb{M}$ ENSION (: : : : ::A, B
$\mathbb{N} T E G E R:: N, N$ SUB,NSUPER,NRHS,LDA,LDB, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V} O T$
SU BROUTINE GBTRS_64 ([TRANSA], $\mathbb{N}], N S U B, N S U P E R, ~ \mathbb{N} R H S], A,[L D A]$,


CHARACTER (LEN=1) ::TRANSA
COM PLEX, D IM ENSION (:,:) ::A,B
$\mathbb{N}$ TEGER (8) :: N , N SUB , N SUPER, NRHS,LDA,LDB, $\mathbb{N}$ FO
$\mathbb{N} \operatorname{TEGER}$ (8), D $\mathbb{M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT}$

## C INTERFACE

\#include <sunperfh>
void ogbtrs (chartransa, intn, intnsub, int nsuper, int nrhs, complex *a, intlda, int*ịivot, com plex *b, int ldb, int *info);
void ogbtrs_64 (chartransa, long n, long nsub, long nsuper, long nrhs, com plex *a, long lda, long *ipivot, com plex *b, long ldb, long *info);

## PURPOSE

cgbtrs solves a system of linear equations
$A * X=B, A * * T * X=B$, or $A * * H * X=B$ w ith $a$ general band $m$ atrix $A$ using the LU factorization com puted by CGBTRF.

## ARGUMENTS

TRANSA (input)
Specifies the form of the system ofequations. =
N : A * $\mathrm{X}=\mathrm{B} \quad$ (Notranspose)
$=T{ }^{\prime}: A * * T * X=B \quad$ (Transpose)
$=C$ ': $A * * H * X=B \quad$ (C onjugate transpose)

TRANSA is defaulted to $N$ 'forF $95 \mathbb{I N}$ TERFACE.

N (input) The order of them atrix A. $\mathrm{N}>=0$.

N SUB (input)
The num ber of subdiagonals w thin the band of A. N SUB $>=0$ 。

## N SU PER (input)

The num ber of superdiagonals $w$ ithin the band of A.
N SU PER > $=0$ 。

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of the m atrix B. NRH S $>=0$.

A (input) $D$ etails of the $L U$ factorization of the band $m$ atrix
A, as com puted by C G B TRF. U is stored as an upper triangularband $m$ atrix $w$ ith $N$ SU B +N SU PER superdiagonals in rows 1 to NSU B+N SU PER+1, and them ultipliers used during the factorization are stored in row SN SU B +N SU PER + 2 to $2 \star$ N SU B + N SU PER +1 .

LD A (input)
The leading dim ension of the anay A. LD A $>=$ 2*N SU B +N SU PER +1 .
$\mathbb{P I V O T}$ (input)
The pívotindiges; for $1<=i<=N$, row $i$ of the $m$ atrix $w$ as interchanged $w$ ith row $\mathbb{P I V O T}$ (i).

B (input/output)
On entry, the right hand side matrix B. On exit, the solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the array $B$. LD B $>=$ $\max (1, N)$.
$\mathbb{I N} F O$ (output)
= 0: successfulexit
<0: if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgebak - form the rightor lefteigenvectors of a com plex general m atrix by backw ard transform ation on the com puted eigenvectors of the balanced $m$ atrix outputby C G EBA L

## SYNOPSIS



```
CHARACTER * 1 JOB,S\mathbb{DE}
COM PLEX V (LDV,*)
\mathbb{N TEGER N, #O, \mathbb{H I,M,LDV ,NNFO}}\mathbf{M},\mp@code{L}
REAL SCALE (*)
```



```
CHARACTER * 1 JOB,SDE
COM PLEX V (LDV,*)
\mathbb{NTEGER*8N,\mathbb{NO,}\mathbb{H}I,M,LDV,INFO}
REALSCALE (*)
F95 INTERFACE
```



```
        [\mathbb{N FO ])}
    CHARACTER (LEN=1) :: DOB,SDE
    COM PLEX,D IM ENSION (:,:) ::V
    \mathbb{NTEGER ::N,\mathbb{LO,\mathbb{H}I,M,LDV,}\mathbb{N}FO}=0
    REAL,DIMENSION (:) ::SCALE
```

    SU BROUTINE GEBAK_64 (JOB,SDE, \(\mathbb{N}], \mathbb{I} O, \mathbb{H} I, S C A L E, ~ M], V,[L D V]\),
        [ \(\mathbb{N} F O\) ])
    CHARACTER (LEN=1):: JOB ,SDE
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::V
$\mathbb{N}$ TEGER (8) ::N, $\mathbb{H} O, \mathbb{H} \mathrm{I}, \mathrm{M}, \mathrm{LDV}, \mathbb{N} F \mathrm{O}$
REAL,D $\mathbb{M}$ ENSION (:) ::SCALE

## C INTERFACE

\#include <sunperfh>
void cgebak (char jo.b, charside, intn, int ilo, int ihi, float *scale, int $m$, com plex *v, int ldv, int *info);
void cgebak_64 (char jंb, char side, long n, long ilo, long ihi, float *scale, long m, com plex *v, long ldv, long *info);

## PURPOSE

cgebak form sthe rightor left eigenvectors of a com plex general $m$ atrix by backw ard transform ation on the com puted eigenvectors of the balanced $m$ atrix outputby CGEBAL .

## ARGUMENTS

JOB (input)
Specifies the type of backw ard transform ation required: = N ', do nothing, retum im $m$ ediately; = P ', do backw ard transform ation for perm utation only; = S', do backw ard transform ation for scaling only; = B ', do backw ard transform ations for both perm utation and scaling. JOB m ustbe the sam e as the argum ent $J 0$ B supplied to CG EBA L .

STDE (input)
= R : : V contains righteigenvectors;
= $\mathrm{L}: \mathrm{V}$ contains lefteigenvectors.

N (input) The num ber of row s of the m atrix $\mathrm{V} . \mathrm{N}>=0$.

ㅍO (input)
The integer ILO determ ined by CGEBAL. $1<=\mathbb{L O}$ <= $\mathbb{H} I<=N$, if $N>0 ; \mathbb{H}=1$ and $\mathbb{H} \mathrm{I}=0$, if $\mathrm{N}=0$.

HH I (input)
The integer $\mathbb{H}$ Ideterm ined by CGEBAL. $1<=\mathbb{L O}$ <= $\mathbb{H} I<=N$, if $N>0 ; \mathbb{O}=1$ and $\mathbb{H} \mathrm{I}=0$, if $\mathrm{N}=0$.

SCALE (input)
D etails of the perm utation and scaling factors, as retumed by CGEBAL.

M (input) The num ber of collm ns of the $m$ atrix $V . M>=0$.
V (input/output)
O $n$ entry, the $m$ atrix of right or lefteigenvectors to be transform ed, as retumed by CHSE IN or CTREVC. On ex㝳, $V$ is overw ritten by the transform ed eigenvectors.

LD V (input)
The leading dim ension of the array $V$. LDV >= $\max (1, N)$.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the i-th argum ent had an illegalvalue.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgebal-balance a general com plex m atrix A

## SYNOPSIS

```
SU BROUTINE CGEBAL (JOB,N,A,LDA, \(\mathbb{I} O, \mathbb{H} I, S C A L E, \mathbb{N} F O)\)
```

CHARACTER * 1 Job
COM PLEX A (LDA,*)
$\mathbb{N}$ TEGER $N, L D A, \mathbb{L} O, \mathbb{H} I, \mathbb{N} F O$
REAL SCALE (*)
SU BROUTINE CGEBAL_64 (JOB,N,A,LDA, $\mathbb{I} O, \mathbb{H} I, S C A L E, \mathbb{N} F O$ )
CHARACTER * 1 JOB
COM PLEX A (LDA,*)
$\mathbb{N} T E G E R * 8 N, L D A, \mathbb{L} O, \mathbb{H} I, \mathbb{N} F O$
REAL SCALE (*)
F95 INTERFACE
SU BROUTINE GEBAL (JOB, $\mathbb{N}], A,[L D A], \mathbb{I} O, \mathbb{H} I, S C A L E,[\mathbb{N F O}])$
CHARACTER (LEN=1):: JOB
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : A
$\mathbb{N} T E G E R:: N, L D A, \mathbb{L O}, \mathbb{H} I, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::SCALE
SU BROUTINE GEBAL_64 (OOB, $\mathbb{N}], A,[L D A], \mathbb{L O}, \mathbb{H} I, S C A L E,[\mathbb{N} F O])$
CHARACTER (LEN=1) :: JOB
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : A
$\mathbb{N}$ TEGER (8) :: N, LDA, $\mathbb{L O}, \mathbb{H} \mathrm{I}, \mathbb{N} F O$

## C INTERFACE

\#include <sunperfh>
void cgebal(char j̀b, intn, com plex *a, intlda, int *ilo, int *ini, float *scale, int *info);
void cgebal 64 (char jं.b, long n, com plex *a, long lda, long *ilo, long *ihi, float *scale, long *info);

## PURPOSE

cgebalbalances a general com plex m atrix A. This involves, first, perm uting A by a sim ilarity transform ation to isolate eigenvalues in the first1 to $\mathrm{ILO}-1$ and last $\mathrm{IH} \mathrm{I}+1$ to N elem ents on the diagonal; and second, applying a diagonal sim ilarity transform ation to row s and colum ns $\mathbb{I} O$ to $\mathbb{H} I$ to $m$ ake the rows and colum ns as close in norm as possible. B oth steps are optional.
$B$ alancing $m$ ay reduce the 1 -norm of the $m$ atrix, and im prove the accuracy of the com puted eigenvalues and/oreigenvectors.

## ARGUMENTS

JOB (input)
Specifies the operations to be perform ed on A :
$=\mathrm{N}$ ': none: simply set $\mathbb{H} \mathrm{O}=1, \mathbb{H} I=\mathrm{N}$, SCALE (I) = 1.0 for $i=1, \ldots, N ;=P$ : perm ute only;
= S': scale only;
$=B$ ': both perm ute and scale.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
On entry, the inputm atrix A. On exit, A is overw ritten by the balanced $m$ atrix. If $\mathrm{OB}=\mathrm{N}^{\prime}$, $A$ is not referenced. See FurtherD etails.

LD A (input)
The leading dim ension of the array A. LD A >= $\max (1, N)$.

IO and $\mathbb{H}$ Iare set to integers such thaton exit A $(i, 7)=0$ if $i>$ jand $j=1, \ldots$, ILO-1 or $I=$
 $=\mathrm{N}$.

HI I (output)
IO and $\mathbb{H}$ Iare set to integers such thaton exit A $(i, j)=0$ if $i>j a n d j=1, \ldots$, IL O-1 or $I=$ $\mathrm{HH} \mathrm{I}+1, \ldots, \mathrm{~N}$. If $\mathrm{JOB}=\mathrm{N}$ 'or $\mathrm{S}^{\prime}, \mathrm{LIO}=1$ and H I $=\mathrm{N}$.

SCALE (output)
D etails of the perm utations and scaling factors applied to $A$. IfP $(j)$ is the index of the row and colum $n$ interchanged $w$ th row and colum $n$ jand $D(1)$ is the scaling factor applied to row and column $\mathfrak{j}$ then SCALE $(j)=P(j) \quad$ for $j=1, \ldots, I L O-1=D(j)$ for $j=\mathbb{L O}, \ldots, \mathbb{H} I=P(j) \quad$ for $j=\mathbb{H} I+1, \ldots, N$. The order in which the interchanges are m ade is N to $\mathbb{H} \mathrm{I}+1$, then 1 to $\mathbb{L} O-1$.
$\mathbb{N} F O$ (output)
= 0: successfulexit.
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue.

## FURTHER DETAILS

The perm utations consist of row and column interchanges which put the $m$ atrix in the form

$$
\begin{gathered}
(\mathrm{T} 1 \mathrm{X} \\
\mathrm{PAP}=\left(\begin{array}{ll}
0 & \mathrm{~B}
\end{array}\right) \\
\left(\begin{array}{ll}
0 & \mathrm{O}
\end{array}\right)
\end{gathered}
$$

where T1 and T2 are uppertriangularm atrices whose eigenvalues lie along the diagonal. The colum $n$ indioes $\Pi 0$ and IH Im ark the starting and ending colum ns of the subm atrix B. Balancing consists of applying a diagonal sim ilarity transform ation inv (D) * $B * D$ to $m$ ake the 1 -norm $s$ of each row of $B$ and its comesponding colum n nearly equal. The outputm atrix is
$\left(\begin{array}{lll}T 1 & X * D & Y\end{array}\right)$
$\left(\begin{array}{lll}0 & \operatorname{inv}(D) * B * D & \operatorname{inv}(D) * Z\end{array}\right)$.
$\left(\begin{array}{lll}0 & 0 & T 2\end{array}\right)$

Inform ation about the perm utations $P$ and the diagonalm atrix D is retumed in the vectorSC A LE .

This subroutine is based on the E ISPA CK routine CBAL.
M odified by Tzu-Y iChen, C om puter Science D ìvision, U niversity of
C alifomia atB erkeley, U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgebrd - reduce a general com plex M -by -N m atrix A to upper or low erbidiagonal form B by a unitary transform ation

## SYNOPSIS

```
SUBROUT\mathbb{NE CGEBRD M,N,A,LDA,D,E,TAUQ,TAUP,W ORK,LW ORK,\mathbb{NFO)}}\mathbf{M}\mathrm{ , (T,}
COM PLEX A (LDA,*),TAUQ (*),TAUP (*),W ORK (*)
INTEGERM,N,LDA,LW ORK,INFO
REALD (*),E (*)
SU BROUT\mathbb{NE CGEBRD_64M,N,A,LDA,D,E,TAUQ,TAUP,W ORK,LW ORK,}
        \mathbb{NFO)}
```

COM PLEX A (LDA,*),TAUQ (*),TAUP (*),W ORK (*)
$\mathbb{N} T E G E R * 8 M, N, L D A, L W O R K, \mathbb{N} F O$
REALD ( ${ }^{\star}$ ), E ( ${ }^{*}$ )

## F95 INTERFACE

SU BROUTINE GEBRD ( $\mathbb{M}], \mathbb{N}], A,[L D A], D, E, T A U Q, T A U P,[\mathbb{W} O R K],[L W O R K]$, [ $\mathbb{N}$ FO ])

COM PLEX,D $\mathbb{I M}$ ENSION (:) ::TAUQ,TAUP,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER ::M ,N,LDA,LW ORK, $\mathbb{N} F O$
REAL,D IM ENSION (:) ::D,E

SU BROUTINE GEBRD_64 (M ], $\mathbb{N}], A,[L D A], D, E, T A U Q, T A U P,[W O R K]$, [LW ORK ], [ $\mathbb{N F O}$ ])

COM PLEX,D $\mathbb{I M}$ ENSION (:) ::TAUQ,TAUP,W ORK

## C INTERFACE

\#include <sunperfh>
void cgebrd (intm , intn, com plex *a, int lda, float *d, float *e, complex *tauq, complex *taup, int *info);
void cgebrd_64 (long m, long n, com plex *a, long lda, float *d, float *e, com plex *tauq, com plex *taup, long *info);

## PURPOSE

cgebrd reduces a general com plex M -by-N m atrix A to upper or low er bidiagonal form B by a unitary transform ation: Q **H * $A * P=B$.

Ifm $>=n, B$ is upperbidiagonal; ifm $<n, B$ is low erbidiagonal

## ARGUMENTS

M (input) The num ber of row s in the m atrix $\mathrm{A} . \mathrm{M}>=0$.
N (input) The num ber of colum ns in them atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
O $n$ entry, the $M-b y-N$ generalm atrix to be reduced. On exit, if $m>=n$, the diagonal and the first superdiagonal are overw rilten w ith the upperbidiagonal $m$ atrix B; the elem ents below the diagonal, $w$ ith the amay TAU $Q$, represent the unitary $m$ atrix $Q$ as a product of elem entary reflectors, and the elem ents above the first superdiagonal, w ith the array TAUP, represent the unitary $m$ atrix $P$ as a product ofelem entary reflectors; ifm < n, the diagonaland the first subdiagonal are overw rilten w ith the low erbidiagonalm atrix $B$; the elem ents below the firstsubdiagonal, with the array TAU Q, represent the unitary $m$ atrix $Q$ as a product of elem entary reflectors, and the elem ents above the diagonal, w ith the array TA UP, represent the unitary $m$ atrix $P$ as a productofelem entary reflec-
tors. See FurtherD etails.

LD A (input)
The leading dim ension of the array A. LD A >= $\max (1, M)$.

D (output)
The diagonalelem ents of the bidiagonalm atrix B :
$D(i)=A(i, i)$.

E (output)
The off-diagonalelem ents of the bidiagonalm atrix
$B$ : ifm $>=n, E(i)=A(i, i+1)$ for $i=1,2, \ldots, n-$
1 ; ifm $<n, E(i)=A(i+1, i)$ for $i=1,2, \ldots, m-1$.
TAUQ (output)
The scalar factors of the elem entary reflectors which represent the unitary $m$ atrix $Q$. See Further D etails.

TAUP (output)
The scalar factors of the elem entary reflectors w hich represent the unitary m atrix P. See Further D etails.

W ORK (w orkspace)
On exiv, if $\mathbb{N F O}=0, W$ ORK ( 1 ) retums the optim al LW ORK.

LW ORK (input)
The length of the amay $W$ ORK. LW ORK >= $\max (1, M, N)$. For optim um perform ance LW ORK >= $(M+N) \star N B$, w here N B is the optim alblocksize.

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
$\mathbb{I N F O}$ (output)
= 0: successfillexit.
$<0:$ if $\mathbb{N}$ FO $=-$ i, the i-th argum enthad an illegalvalue.

## FURTHER DETAILS

Them atrices $Q$ and $P$ are represented as products of elem entary reflectors:

Ifm $>=n$,

$$
Q=H(1) H(2) \ldots H(n) \text { and } P=G(1) G(2) \ldots G(n-1)
$$

Each H (i) and G (i) has the form :

$$
H(i)=I-\operatorname{tanq}{ }^{\star} v^{\star} v^{\prime} \text { and } G(i)=I-\operatorname{taup} * u^{\star} u^{\prime}
$$

$w$ here tauq and taup are com plex scalars, and $v$ and $u$ are complex vectors; $v(1: i-1)=0, v(i)=1$, and $v(i+1 m)$ is stored on exitin $A(i+1 m, i) ; u(1: i)=0, u(i+1)=1$, and $u(i+2 m)$ is stored on exitin A $(i, i+2 m)$; tauq is stored in TAUQ (i) and taup in TAUP (i).
Ifm < n,

$$
Q=H(1) H(2) \ldots H(m-1) \text { and } P=G(1) G(2) \ldots G(m)
$$

Each H (i) and G (i) has the form :

$$
H(i)=I-\operatorname{tanq}{ }^{\star} v^{\star} v^{\prime} \text { and } G(i)=I-\operatorname{taup}{ }^{\star} u^{\star} u^{\prime}
$$

$w$ here tauq and taup are com plex scalars, and $v$ and $u$ are complex vectors; $v(1: i)=0, v(i+1)=1$, and $v(i+2 \mathrm{~m})$ is stored on exitin $A(i+2 m, i) ; u(1: i-1)=0, u(i)=1$, and $u(i+1 \mathrm{~m})$ is stored on exitin $A(i, i+1 \mathrm{~m})$; tauq is stored in TAUQ (i) and taup in TA UP (i).

The contents of A on exitare illustrated by the follow ing exam ples:
$m=6$ and $n=5(m>n): \quad m=5$ and $n=6(m<n):$
( d e u1 u1 u1 ) ( d u1 u1 u1 u1
u1)
(v1 d e u2 u2) ( e d u2 u2 u2
u2 )
( v1 v2 d e u3) (v1 e d u3 u3
u3)
( v1 v2 v3 d e ) (v1 v2 e d u4 u4)
( v1 v2 v3 v4 d ) (v1 v2 v3 e d u5 )
( v1 v2 v3 v4 v5 )
where d and e denote diagonal and off-diagonal elem ents of $B$, videnotes an elem ent of the vectordefining $H$ (i), and ui an elem ent of the vectordefining $G$ (i).

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgecon -estim ate the reciprocal of the condition num ber of a general com plex matrix $A$, in either the 1 -norm or the infinity-norm, using the LU factorization com puted by CGETRF

## SYNOPSIS

```
SUBROUT\mathbb{NE CGECON NORM,N,A,LDA,ANORM,RCOND,W ORK,W ORK2,\mathbb{NFO)}}\mathbf{N}\mathrm{ (N,A}
CHARACTER * 1NORM
COM PLEX A (LDA,*),W ORK (*)
INTEGERN,LDA,}\mathbb{N}F
REAL ANORM,RCOND
REAL W ORK2 (*)
SUBROUTINE CGECON_64 NORM,N,A,LDA,ANORM,RCOND,W ORK,W ORK2,
        \mathbb{NFO)}
CHARACTER * 1 NORM
COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGER*8N,LDA,NNFO}
REAL ANORM,RCOND
REALW ORK2 (*)
```

F95 INTERFACE
SU BROUTINE GECON $\mathbb{N} O R M, \mathbb{N}], A,[L D A], A N O R M, R C O N D,[W O R K],[W O R K 2]$,
[ $\mathbb{N} F O$ ])
CHARACTER (LEN=1) ::NORM
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R:: N, L D A, \mathbb{N} F O$
REAL ::ANORM,RCOND

SU BROUTINE GECON_64 $\mathbb{N} O R M, \mathbb{N}], A,[L D A], A N O R M, R C O N D,[W O R K],[\mathbb{W} O R K 2]$, [ $\mathbb{N} \mathrm{FO}]$ )

CHARACTER ( $几 E N=1$ ) :: NORM
COM PLEX,D $\mathbb{I M} E N S I O N(:):$ ORK
COM PLEX, D $\mathbb{M}$ ENSION (: : : : : A
$\mathbb{N}$ TEGER (8) :: N, LD A , $\mathbb{N}$ FO
REAL ::ANORM,RCOND
REAL,D $\mathbb{I M} E N S I O N(:):: W$ ORK2

## C INTERFACE

\#include <sunperfh>
void ogecon (charnorm , intn, com plex *a, int lda, float anorm, float*rcond, int*info);
void cgecon_64 (charnorm, long n, complex *a, long lda, floatanorm , float *rcond, long *info);

## PURPOSE

cgecon estim ates the reciprocal of the condition num ber of a general com plex m atrix $A$, in either the 1 -norm orthe infinity-norm, using the LU factorization com puted by CGETRF.

A $n$ estim ate is obtained fornorm (inv (A ) ), and the reciprocal of the condition num ber is com puted as
$\operatorname{RCOND}=1 /(\operatorname{nom}(A) * \operatorname{norm}(\operatorname{inv}(A)))$.

## ARGUMENTS

NORM (input)
Specifies w hether the 1-norm condition number or the infinity-norm condition num ber is required:
= 1 'or $\mathrm{O}^{\prime}$ : 1 -nom ;
$=$ I': Infinity-norm .

N (input) The order of the matrix A. $\mathrm{N}>=0$.
A (input) The factors $L$ and $U$ from the factorization $A=$ $\mathrm{P} * \mathrm{~L} * \mathrm{U}$ as com puted by CGETRF.

LD A (input)
The leading dim ension of the amay A. LDA >= $\max (1, N)$.

## ANORM (input)

IfNORM = 1 'or 0 ', the 1 -nom of the original
$m$ atrix $A$. IfNORM = $I$ ', the infinity-norm of the originalm atrix A.

RCOND (output)
The reciprocal of the condition num ber of the
$m$ atrix $A$, computed as $R C O N D=1 /($ norm (A) * norm (inv (A))).

W ORK (w orkspace)
dim ension ( $2 \star \mathrm{~N}$ )

W ORK 2 (w orkspace)
dim ension ( $2 * N$ )
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgeequ - com pute row and colum n scalings intended to equili-
brate an $M$-by-N $m$ atrix A and reduce its condition num ber

## SYNOPSIS

```
SUBROUT\mathbb{NE CGEEQUM,N,A,LDA,R,C,ROW CND,COLCND,AMAX,}
    INFO)
COM PLEX A (LDA,*)
\mathbb{NTEGERM,N,LDA,}\mathbb{N}FO
REAL ROW CND,COLCND,AMAX
REALR (*),C (*)
SUBROUT\mathbb{NE CGEEQU_64M,N,A,LDA,R,C,ROW CND,COLCND,}
        AMAX,INFO)
    COM PLEX A (LDA,*)
    \mathbb{NTEGER*8M,N,LDA, INFO}
    REAL ROW CND,COLCND,AMAX
    REALR (*),C (*)
```


## F95 INTERFACE

```
SU BROUTINE GEEQU (M ], \(\mathbb{N}], A,[L D A], R, C, R O W C N D, C O L C N D\), AMAX, [ \(\mathbb{N F O}]\) )
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER ::M,N,LDA, \(\mathbb{N} F O\)
REAL ::ROW CND,COLCND,AMAX
REAL,D IM ENSION (:) :: R, C
SUBROUTINE GEEQU_64 (M) \(\mathbb{M}\) ] \(, A,[L D A], R, C, R O W C N D, C O L C N D\), AMAX, \([\mathbb{N} F O]\) )
```

COM PLEX, D IM ENSION (:,:) ::A
$\mathbb{N}$ TEGER (8) ::M,N,LDA, $\mathbb{N} F O$
REAL ::ROW CND, COLCND, AM AX
REAL,D $\mathbb{I}$ ENSION (:) ::R,C

## C INTERFACE

\#include <sunperfh>
void cgeequ (intm , intn, com plex *a, int lda, float *r, float *C, float *row cnd, float *colend, float
*am ax, int*info);
void ogeequ_64 (long m, long n, com plex *a, long lda, float
*r, float*c, float*row cnd, float*colend, float
*am ax, long *info);

## PURPOSE

cgeequ com putes row and colum n scalings intended to equilibrate an M boy-N m atrix A and reduce its condition num ber. R retums the row scale factors and $C$ the colum $n$ scale factors, chosen to try to $m$ ake the largestelem ent in each row and column of the $m$ atrix $B \quad w$ ith elements $B(i, j)=R(i) * A(i, j) * C(i)$ have absolute value 1.

R (i) and C (i) are restricted to be betw een SM LN UM = sm allest safe num ber and B IG N U M = largest safe num ber. U se of these scaling factors is notguaranteed to reduce the condition num berofA butw orks w ellin practice.

## ARGUMENTS

M (input) The num ber of row s of the m atrix $\mathrm{A} . \mathrm{M}>=0$.
N (input) The num ber of colum ns of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input) The M -by-N m atrix whose equilibration factors are to be com puted.

LD A (input)
The leading dim ension of the array A. LDA >= max (1, M).

R (output)
If $\mathbb{N} F O=0$ or $\mathbb{N} F O>M, R$ contains the row scale factors forA.

C (output)
If $\mathbb{N} F O=0, C$ contains the colum $n$ scale factors forA.

ROW CND (output)
If $\mathbb{N} F O=0$ or $\mathbb{N} F O>M, R O W C N D$ contains the ratio of the sm allest $R$ (i) to the largest $R$ (i). If
ROW CND >=0.1 and AM AX is neither too large nor too sm all, it is notw orth scaling by $R$.

COLCND (output)
If $\mathbb{N F O}=0, \operatorname{COLCND}$ contains the ratio of the sm allest $C$ (i) to the largestC (i). IfCO LCND >= 0.1 , it is notw orth scaling by $C$.

AMAX (output)
A bsolute value of largestm atrix elem ent. IfA M A X
is very close to overflow orvery close to underflow , the m atrix should be scaled.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an illegalvalue
$>0$ : if $\mathbb{N F O}=i$, and $i$ is
$<=\mathrm{M}$ : the i-th row ofA is exactly zero
> M : the ( $\mathrm{i}-\mathrm{M}$ ) -th colum n of A is exactly zero

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgees - com pute foran $N$-by-N com plex nonsym $m$ etric $m$ atrix A, the eigenvalues, the Schur form T, and, optionally, the $m$ atrix of Schurvectors Z

## SYNOPSIS

```
SUBROUT\mathbb{NE CGEES (JOBZ,SORTEV,SELECT,N,A,LDA,NOUT,W,Z,LD Z,}
    W ORK,LDW ORK,W ORK 2,W ORK 3, INFO)
CHARACTER * 1 OOBZ,SORTEV
COM PLEX A (LDA,*),W (*),Z (LDZ,*),W ORK (*)
INTEGERN,LDA,NOUT,LDZ,LDW ORK,INFO
LOG ICAL SELECT
LOG ICAL W ORK 3 (*)
REALW ORK2 (*)
SU BROUT\mathbb{NE CGEES_64(JOBZ,SORTEV,SELECT,N,A,LDA,NOUT,W ,Z,LD Z,}
    W ORK,LDW ORK,W ORK 2,W ORK 3, INFO)
CHARACTER * 1 JOBZ,SORTEV
COM PLEX A (LDA,*),W (*),Z (LDZ,*),W ORK (*)
\mathbb{NTEGER*8N,LDA,NOUT,LD Z,LDW ORK,INFO}
LOG ICAL*& SELECT
LOG ICAL * & W ORK 3 (*)
REALW ORK2 (*)
```


## F95 INTERFACE

SU BROUTINE GEES (JOBZ,SORTEV, [SELECT], $\mathbb{N}], A,[L D A], ~ N O U T], W,[Z],[L D Z]$, [W ORK], [LDW ORK ], [W ORK 2], [W ORK 3], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::JOBZ,SORTEV
COM PLEX,D $\mathbb{M}$ ENSION (:) ::W,W ORK

COM PLEX , D $\mathbb{M}$ ENSION (: : : : : A , Z
$\mathbb{N}$ TEGER :: N, LDA, NOUT,LDZ,LDW ORK, $\mathbb{N}$ FO
LOGICAL :: SELECT
LOG ICAL, D IM EN SION (:) ::W ORK 3
REAL,D $\mathbb{I}$ ENSION (:) ::W ORK2

SU BROU T INE GEES_64 (DOBZ, SORTEV , [SELECT], $\mathbb{N}], A,[L D A], \mathbb{N} O U T], W,[Z]$, [LD Z], [W ORK], [LDW ORK], [W ORK2], [W ORK 3], [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1) :: JOBZ, SORTEV
COM PLEX,D $\mathbb{I M} E N S I O N(:):: W, W$ ORK
COM PLEX,D $\mathbb{I}$ ENSION (:,:) ::A,Z
$\mathbb{N}$ TEGER (8) :: N , LDA , NOUT, LD Z , LDW ORK, $\mathbb{N} F O$
LOG ICAL (8) :: SELECT
LOGICAL (8), D IM ENSION (:) ::W ORK 3
REAL,D $\mathbb{I}$ ENSION (:) ::W ORK2

## C INTERFACE

\#include <sunperfh>
void cgees (char jobz, char sortev, int(*select) (com plex), intn, com plex *a, int lda, int *nout, com plex *w, com plex ${ }^{*} z$, int ldz, int *info);
void ogees_64 (char jobz, char sortev, long (*select) (com plex), long n, com plex *a, long lda, long *nout, com plex *w, com plex *z, long ldz, long *info);

## PURPOSE

cgees com putes for an N -by- N com plex nonsym m etric $m$ atrix A , the eigenvalues, the Schur form T, and, optionally, the $m$ atrix of Schurvectors Z. This gives the Schur factorization $A=Z * T *(Z * * H)$.

Optionally, italso orders the eigenvalues on the diagonal of the Schur form so that selected eigenvalues are at the top left. The leading colum ns of $Z$ then form an orthonorm al basis for the invariant subspace corresponding to the selected eigenvalues.

A com plex $m$ atrix is in Schur form if it is uppertriangular.

## ARGUMENTS

JO B Z (input)
$=\mathrm{N}$ ': Schurvectors are notcom puted;
$=\mathrm{V}$ : Schurvectors are com puted.
SORTEV (input)
Specifies w hether ornot to order the eigenvalues on the diagonal of the Schur form . = N ': Eigenvalues are not ordered:
= $S^{\prime}$ ': E igenvalues are ordered (see SE LEC T ) .

## SELECT (input)

SELECT mustbe declaredEXTERNAL in the calling subroutine. If SORTEV $=S^{\prime}$, SELECT is used to selecteigenvalues to order to the top left of the Schurform. IfSORTEV = N', SELECT is notreferenced. The eigenvalue $W(\mathcal{j})$ is selected if SELECT $(\mathbb{N}(\mathcal{J})$ is true.

N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.
A (input/output)
On entry, the N -by -N m atrix A. On exit, A has been overw rilten by its Schur form T.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, N)$.

NOUT (output)
If SORTEV $=\mathrm{N}$ ', NOUT $=0$. IfSORTEV $=S^{\prime}$, NOUT
= num berofeigenvalues forw hich SELEC $T$ is true.
W (output)
W contains the com puted eigenvalues, in the sam e order that they appearon the diagonal of the outputSchur form T.

Z (output)
If $\mathrm{OOBZ}=\mathrm{V}^{\prime}, \mathrm{Z}$ contains the unitary m atrix Z of Schur vectors. If $J O B Z=N^{\prime}, Z$ is not referenced.

LD $Z$ (input)
The leading $d i m$ ension of the array $Z . L D Z>=1$;
if $\mathrm{JOBZ}=\mathrm{V}$ ', LD Z >= N.
W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the array W ORK. LDW ORK >= $\max (1,2 \star N)$. For good perform ance, LDW ORK must
generally be larger.
IfLDW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 2 (w orkspace)
dim ension (N)
W ORK 3 (w orkspace)
dim ension (N) N ot referenced if SO RTEV = N '.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0$ : if $\mathbb{N N F O}=-$ i, the $i$-th argum ent had an illegalvalue.
$>0:$ if $\mathbb{N F O}=i$, and $i$ is
<= N : the QR algorithm failed to com pute all the eigenvalues; elem ents $1: \mathbb{I}-1$ and i+ $1 \mathbb{N}$ ofW con-
tain those eigenvalues which have converged; if JO BZ $=V$ ', Z contains the $m$ atrix which reduces $A$ to its partially converged Schur form.$=N+1$ : the eigenvalues could notbe reordered because some eigenvalues were too close to separate the problem is very ill-conditioned); $=\mathrm{N}+2$ : after reordering, roundoff changed values of som e com plex eigenvalues so that leading eigenvalues in the Schur form no longer satisfy SELECT = TRUE.. This could also be caused by underflow due to scaling.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgeesx - com pute foran $N$-by -N com plex nonsym $m$ etric $m$ atrix A, the eigenvalues, the Schur form $T$, and, optionally, the $m$ atrix of Schurvectors Z

## SYNOPSIS

```
SUBROUT\mathbb{NE CGEESX (OBZ,SORTEV,SELECT,SENSE,N,A,LDA,NOUT,W ,Z,}
    LDZ,RCONE,RCONV,W ORK,LDW ORK,W ORK2,BW ORK 3, INFO)
```

CHARACTER * 1 JOBZ, SORTEV, SEN SE
COM PLEX A (LDA, *), W (*), Z (LDZ,*), W ORK (*)
$\mathbb{N}$ TEGERN,LDA,NOUT,LDZ,LDW ORK, $\mathbb{N} F O$
LOG ICAL SELECT
LOG ICAL BW ORK 3 (*)
REAL RCONE,RCONV
REALW ORK 2 (*)
SU BROUTINE CGEESX_64 (OOBZ, SORTEV, SELECT, SENSE,N,A,LDA,NOUT,W,
Z,LDZ,RCONE,RCONV,W ORK,LDW ORK,W ORK 2,BW ORK 3, $\mathbb{N} F O$ )
CHARACTER * 1 JOBZ, SORTEV, SENSE
COM PLEXA (LDA, *), W (*), Z (LDZ, $\left.{ }^{*}\right), \mathrm{W} O R K(*)$
$\mathbb{N} T E G E R * 8 N, L D A, N O U T, L D Z, L D W$ ORK, $\mathbb{N} F O$
LOG ICAL*8 SELECT
LOG ICAL*8BW ORK 3 (*)
REAL RCONE,RCONV
REALW ORK 2 (*)

## F95 INTERFACE

SU BROUTINE GEESX (JOBZ, SORTEV, [SELECT],SEN SE, $\mathbb{N}], A,[L D A], N O U T, W$, [Z], [LD Z],RCONE,RCONV, [W ORK], [LDW ORK], [W ORK2], BW ORK 3], [ $\mathbb{N}$ FO ])

CHARACTER ( $几 E N=1$ ) : : JOBZ, SORTEV, SEN SE
COM PLEX , D $\mathbb{M}$ ENSION (:) ::W ,W ORK
COM PLEX, D $\mathbb{I M}$ ENSION (: : : : : A, Z
$\mathbb{N}$ TEGER : : N, LDA, NOUT,LDZ,LDW ORK, $\mathbb{N}$ FO
LOGICAL :: SELECT
LOGICAL,D $\mathbb{I M} E N S I O N(:):$ BW ORK 3
REAL ::RCONE,RCONV
REAL,D $\mathbb{I}$ ENSION (:) ::W ORK2

SU BROU T INE GEESX_64 (OB B , SORTEV, [SELECT],SEN SE, $\mathbb{N}], A,[L D A], N O U T$, W, [Z], [LDZ],RCONE,RCONV, [WORK], [LDW ORK], [WORK2], [BWORK3], [ $\mathbb{N}$ FO ])

CHARACTER ( $\amalg E N=1$ ) : : OBZ , SORTEV, SEN SE
COM PLEX ,D $\mathbb{M}$ ENSION (:) ::W ,W ORK
COM PLEX, D $\mathbb{I M}$ ENSION (: : : : : A, Z
$\mathbb{N}$ TEGER (8) :: N , LDA , NOUT,LD Z, LDW ORK, $\mathbb{N} F O$
LOGICAL (8) :: SELECT
LOGICAL (8), D IM ENSION (:) ::BW ORK 3
REAL ::RCONE,RCONV
REAL,D $\mathbb{M}$ ENSION (:) ::W ORK2

## C INTERFACE

\#include <sunperfh>
void ogeesx (char jobz, char sortev, int(*select) (com plex), char sense, intn, com plex *a, int lda, int *nout, com plex *${ }_{\mathrm{w}}$, com plex ${ }_{\mathrm{z}}$, int ldz, float *roone, float*roonv, int*info);
void cgeesx_64 (char j̀bz, char sortev, long (*select) (com plex), char sense, long n, com plex *a, long lda, long *nout, com plex *w, com plex
*z, long ldz, float *rcone, float*rconv, long
*info);

## PURPOSE

cgeesx com putes for an $N$ toy $N$ com plex nonsym m etric $m$ atrix $A$, the eigenvalues, the Schur form $T$, and, optionally, the $m$ atrix of Schurvectors $Z$. This gives the Schur factorization $A=Z{ }^{*} T^{*}(Z * * H)$.

Optionally, 辻also orders the eigenvalues on the diagonal of the Schur form so that selected eigenvalues are at the top left; com putes a reciprocal condition num ber for the average of the selected eigenvahes (RCONDE); and com putes a reciprocal condition num ber for the right invariant subspace
comesponding to the selected eigenvalues (RCONDV). The leading colum ns of $Z$ form an orthonorm al basis for this invariant subspace.

For furtherexplanation of the reciprocalcondition num bers RCONDE and RCONDV, see Section 4.10 of the LAPACK U sers' G uide (w here these quantities are called s and sep respectively).

A com plex $m$ atrix is in Schur form if it is uppertriangular.

## ARGUMENTS

JO BZ (input)
= N ':Schurvectors are not com puted;
$=\mathrm{V}$ ':Schurvectors are com puted.
SORTEV (input)
Specifies w hether ornot to order the eigenvalues on the diagonal of the Schur form . = N ': Eigenvalues are notordered;
= S ': E igenvalues are ordered (see SE LEC T ) .

## SELECT (input)

SELECT mustbe declared EXTERNAL in the calling subroutine. If SORTEV = S', SELECT is used to select eigenvalues to order to the top left of the Schurform. IfSORTEV = N', SELECT is notreferenced. An eigenvalue $W(\mathcal{)}$ is selected if SELECT $(\mathbb{N}(\mathcal{J})$ is true.

SENSE (input)
D eterm ines which reciprocal condition num bers are com puted. = N ': N one are com puted;
= E ': C om puted for average of selected eigenvalues only;
= V ': C om puted for selected right invariant subspace only;
= B ': C om puted forboth. If SEN $S E=E$ ', $\mathrm{V}^{\prime}$ or B',SORTEV mustequal $S^{\prime}$.

N (input) The order of the m atrix A. $\mathrm{N}>=0$.

A (input/output)
On entry, the N boy -N m atrix A. On exit, A is overw ritten by its Schur form T.

LDA (input)
The leading dim ension of the array A. LDA >=
$\max (1, N)$.

NOUT (output)
If SORTEV $=\mathrm{N}$ ',NOUT $=0$. IfSORTEV $=\mathrm{S}$ ', NOUT
= num ber ofeigenvalues forw hich SELEC $T$ is true.
W (output)
W contains the com puted eigenvalues, in the sam e order that they appearon the diagonal of the outputSchur form T.

Z (output)
If $\mathrm{OBBZ}=\mathrm{V}^{\prime}, \mathrm{Z}$ contains the unitary m atrix Z of Schur vectors. If $J O B Z=N^{\prime}, Z$ is not referenced.

LD Z (input)
The leading din ension of the array $\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD Z >=N.

RCONE (output)
If SEN SE = E 'or B', RCONE contains the reciprocal condition number for the average of the selected eigenvalues. N ot referenced if SEN SE = N 'or V'.

## RCONV (output)

If SEN SE = V 'or B',RCONV contains the reciprocal condition num ber for the selected right invariant subspace. N ot referenced if SEN SE $=\mathrm{N}$ ' or E'.

W ORK (w orkspace)
dim ension (LDW ORK) On exit, if $\mathbb{I N F O}=0, W$ ORK (1) retums the optim allD W ORK .

LDW ORK (input)
The dimension of the amay W ORK. LDW ORK >= max (1,2*N). A lso, if SENSE = E'or V'or B',
LDW ORK >= $2 *$ NOUT* $(\mathbb{N}-N O U T)$, where NOUT is the num ber of selected eigenvalues com puted by this routine. N ote that 2*NOUT* $(\mathbb{N}-\mathrm{NOUT})<=\mathrm{N} * \mathrm{~N} / 2$. For good perform ance, LD W O RK m ust generally be larger.

W ORK 2 (w orkspace)
dim ension $(\mathbb{N})$

BW ORK 3 (w orkspace)
dim ension ( N ) N ot referenced if SO RTEV $=\mathrm{N}^{\prime}$.
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum enthad an illegalvalue.
> 0 : if $\mathbb{N} F O=i$, and $i$ is
<= N : the QR algorithm failed to com pute all the eigenvalues; elem ents $1: \mathbb{I} \mathrm{O}-1$ and i+1 N ofW contain those eigenvalues which have converged; if $\mathrm{JOBZ}=\mathrm{V}$ ', Z contains the transform ation which reduces A to its partially converged Schur form . $=\mathrm{N}+1$ : the eigenvalues could not be reordered because some eigenvahues were too close to separate (the problem is very ill-conditioned); = $\mathrm{N}+2$ : after reordering, roundoff changed values of som e com plex eigenvalues so that leading eigenvalues in the Schur form no longer satisfy SELEC T= TRUE. This could also be caused by underflow due to scaling.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgeev -com pute foran N -by -N com plex nonsym m etric m atrix A , the eigenvalues and, optionally, the left and/orright eigenvectors

## SYNOPSIS

```
SUBROUTINE CGEEV (JOBVL,JOBVR,N,A,LDA,W ,VL,LDVL,VR,LDVR,
    W ORK,LDW ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 JOBVL,JOBVR
COM PLEX A (LDA,*),W (*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
INTEGERN,LDA,LDVL,LDVR,LDW ORK, \mathbb{NFO}
REALW ORK2 (*)
SUBROUT\mathbb{NE CGEEV_64 (JOBVL,JOBVR,N,A,LDA,W ,VL,LDVL,VR,LDVR,}
        W ORK,LDW ORK,W ORK2, INFO)
CHARACTER * 1 JOBVL,NOBVR
COM PLEX A (LDA,*),W (*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
\mathbb{NTEGER*8N,LDA,LDVL,LDVR,LDW ORK, NNFO}
REALW ORK2 (*)
```


## F95 INTERFACE

SU BROUTINE GEEV (JOBVL, JOBVR, $\mathbb{N}$ ],A, [LDA], $\mathrm{W}, \mathrm{VL},[\operatorname{LDVL}], \mathrm{VR},[\operatorname{LDVR}]$, [W ORK ], [LDW ORK], $\mathbb{W}$ ORK2], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1)::JOBVL, JO BVR
COM PLEX,D $\mathbb{M}$ ENSION (:) ::W ,W ORK
COM PLEX,D IM ENSION (: : : : : A, VL,VR
$\mathbb{N}$ TEGER :: N,LDA,LDVL,LDVR,LDW ORK, $\mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::W ORK2
 [LDVR], [W ORK ], [LDW ORK ], [W ORK 2], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1) :: JOBVL, JOBVR
COM PLEX,D $\mathbb{M}$ ENSION (:) ::W ,W ORK
COMPLEX,D IM ENSION (:,:) ::A,VL,VR
$\mathbb{N} T E G E R(8):: N, L D A, L D V L, L D V R, L D W O R K, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::W ORK2

## C INTERFACE

\#include <sunperfh>
void cgeev (char jंbvl, char jंbvr, intn, com plex *a, int lda, com plex ${ }^{\mathrm{w}}$, com plex *vl, int ldvl, com plex *vr, int ldvr, int *info);
void cgeev_64 (char jobvl, char jobvr, long n, com plex *a, long lda, com plex *w , com plex *vl, long ldvl, com plex *vr, long ldvr, long *info);

## PURPOSE

cgeev com putes foran N -by- N com plex nonsym m etric m atrix A , the eigenvalues and, optionally, the left and/or right eigenvectors.

The righteigenvectorv $(\mathcal{J})$ of A satisfies A * $\mathrm{V}(\mathrm{y})=\operatorname{lam} \operatorname{bda}(\mathrm{y}) * \mathrm{~V}(\mathrm{I})$ where lam bda ( $\mathcal{j}$ ) is its eigenvalue. The lefteigenvectoru ( $)$ ) ofA satisfies

where $u(\mathcal{j}) * * H$ denotes the conjugate transpose of $u(j)$.
The com puted eigenvectors are norm alized to have Euclidean norm equalto 1 and largest com ponent real.

## ARGUMENTS

$J 0 \mathrm{BVL}$ (input)
$=\mathrm{N}$ ': lefteigenvectors of A are not com puted;
= V': lefteigenvectors of are com puted.
JOBVR (input)
= N ': righteigenvectors of A are not com puted;
= V ': righteigenvectors of A are com puted.
N (input) The order of the m atrix $\mathrm{A} \cdot \mathrm{N}>=0$.

A (input/output)
On entry, the N -by-N m atrix A. On exit, A has been overw rilten.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, N)$.

W (output)
W contains the com puted eigenvalues.

VL (input)
If $\mathrm{JOBVL}=\mathrm{V}$ ', the left eigenvectors $\mathrm{u}(\mathcal{)}$ are stored one after another in the colum ns of V L, in the sam e order as theireigenvalues. If JOBVL = $N^{\prime}, \mathrm{VL}$ is not referenced. $\mathrm{u}(\mathcal{J})=\mathrm{VL}(:, 7)$, the j-th colum n ofVL.

LDVL (input)
The leading dim ension of the array V L. LD V L >=1; if $\mathrm{JOBVL}=\mathrm{V}, \mathrm{LD} V \mathrm{~L}>=\mathrm{N}$.

VR (input)
If $\mathrm{JO} \mathrm{BVR}=\mathrm{V}$ ', the right eigenvectors $\mathrm{V}(\mathrm{I})$ are stored one after another in the colum ns of V R, in the sam e order as theireigenvalues. If JOBVR = $\mathrm{N}^{\prime}, \mathrm{VR}$ is not referenced. $\mathrm{V}(\mathcal{j})=\mathrm{VR}(:, 7)$, the $j$ th colum n ofVR.

LDVR (input)
The leading dim ension of the array VR. LD V R >= 1; if JO BVR = V', LDVR >= N .

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the amray $W$ ORK. LDW ORK >= $\max (1,2 \star \mathrm{~N})$. For good perform ance, LDW ORK must generally be larger.

If LD W ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA .

W ORK 2 (w orkspace)
dim ension ( $2 \star \mathrm{~N}$ )
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the i-th argum enthad an illegalvałue.
$>0$ : if $\mathbb{N F O}=i$, the QR algorithm failed to com pute all the eigenvalues, and no eigenvectors have been com puted; elem ents and i+1 N of W contain eigenvalues w hich have converged.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgeevx -com pute foran $N$-by -N com plex nonsym $m$ etric $m$ atrix A, the eigenvalues and, optionally, the left and/or right eigenvectors

## SYNOPSIS

```
SUBROUT\mathbb{NE CGEEVX BALANC,NOBVL,JOBVR,SENSE,N,A,LDA,W,VL,}
    LDVL,VR,LDVR,\PsiO,IHI,SCALE,ABNRM,RCONE,RCONV,W ORK,
    LDW ORK,W ORK2,INFO)
CHARACTER * 1 BALANC,JOBVL,JOBVR,SENSE
COM PLEX A (LDA,*),W (*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
INTEGER N,LDA,LDVL,LDVR,\mathbb{LO,}\mathbb{H}I,LDW ORK,\mathbb{NFO}
REAL ABNRM
REALSCALE (*),RCONE (*),RCONV (*),W ORK2 (*)
SU BROUTINE CGEEVX_64(BA LANC,JOBVL,JO BVR,SENSE,N,A LDA ,W ,V L,
    LDVL,VR,LDVR, HO,HI,SCALE,ABNRM,RCONE,RCONV,WORK,
    LDW ORK,WORK2,\mathbb{NFO)}
CHARACTER * 1 BALANC,JOBVL,JOBVR,SENSE
COM PLEXA (LDA,*),W (*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
INTEGER*8 N,LDA,LDVL,LDVR,\mathbb{LO},\mathbb{H}I,LDW ORK,\mathbb{NFO}
REAL ABNRM
REALSCALE (*),RCONE (*),RCONV (*),W ORK2 (*)
```


## F95 INTERFACE

SU BROUTINE GEEVX (BALANC, JOBVL, JO BVR, SENSE, $\mathbb{N}], A,[L D A], W, V L$, $[L D V L], V R,[L D V R], \mathbb{I} O, \mathbb{H} I, S C A L E, A B N R M, R C O N E, R C O N V,[W O R K]$, LDW ORK, $\mathbb{W}$ ORK2], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) ::BALANC, JOBVL, JOBVR,SENSE

COM PLEX,D $\mathbb{M}$ ENSION (:) ::W ,W ORK
COMPLEX,D $\mathbb{I M}$ ENSION (:,:) ::A,VL,VR
$\mathbb{N}$ TEGER :: N,LDA,LDVL,LDVR, $\mathbb{H}$, $\mathbb{H} I, L D W O R K, \mathbb{N} F O$
REAL ::ABNRM
REAL,D $\mathbb{M}$ ENSION (:) ::SCALE,RCONE,RCONV,W ORK 2
SU BROUTINE GEEVX_64 (BALANC, JOBVL, JOBVR,SENSE, $\mathbb{N}], A,[L D A], W$, VL, [LDVL], VR, [LDVR], $\mathbb{I} O, \mathbb{H} I, S C A L E, A B N R M, R C O N E, R C O N V$, [W ORK ],LDW ORK, [W ORK 2], [ $\mathbb{N} F O$ ])

CHARACTER (LEN =1) ::BALANC, JOBVL, JOBVR,SEN SE
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ,W ORK
COMPLEX,D $\mathbb{I M}$ ENSION (:,:) ::A,VL,VR
$\mathbb{N} T E G E R(8):: N, L D A, L D V L, L D V R, \mathbb{I} O, \mathbb{H} I, L D W O R K, \mathbb{N} F O$
REAL ::ABNRM
REAL,D $\mathbb{M}$ ENSION (:) ::SCALE,RCONE,RCONV,W ORK 2

## C INTERFACE

\#include <sunperfh>
void cgeevx (char, char, char, char, int, complex*, int, complex*, complex*, int, com plex*, int, int*, int*, float*, float*, float*, float*, int* );
void cgeevx_64 (char, char, char, char, long, com plex*, long, com plex*, com plex*, long, com plex*, long, long*, long*, float*, float*, float*, float*, long*);

## PURPOSE

cgeevx com putes for an $N$-by-N com plex nonsym m etric $m$ atrix A, the eigenvalues and, optionally, the left and/orright eigenvectors.

O ptionally also, it com putes a balancing transform ation to im prove the conditioning of the eigenvalues and eigenvectors ( $\mathbb{L} \mathrm{O}, \mathbb{H} \mathrm{I}, \mathrm{SCA} L E$, and A BN RM ), reciprocal condition num bers for the eigenvalues (RCONDE), and reciprocal condition num bers for the right eigenvectors (RCONDV).

The righteigenvectorv (i) ofA satisfies

$$
\text { A * } \mathrm{v}(\mathcal{\jmath})=\operatorname{lam} \operatorname{bda}(\mathcal{\jmath}) * v(\mathcal{I})
$$

where $\operatorname{lam} \operatorname{bda}(\mathcal{j})$ is its eigenvalue.
The lefteigenvectoru ( ) ) of A satisfies

where $u(j) * * H$ denotes the conjugate transpose of $u(j)$.
The com puted eigenvectors are norm alized to have Euclidean
norm equalto 1 and largest com ponent real.
B alancing a $m$ atrix $m$ eans perm uting the row $s$ and colum $n s$ to m ake itm ore nearly upper triangular, and applying a diagonalsm ilarity transform ation $D$ * $A$ * $D *(-1)$, where $D$ is a diagonalm atrix, to $m$ ake its row $s$ and colum ns closer in norm and the condition num bers of its eigenvalues and eigenvectors sm aller. The com puted reciprocalcondition num bers comespond to the balanced $m$ atrix. Perm uting row $s$ and colum ns w ill not change the condition num bers (in exact arithm etic) but diagonalscaling w ill. For further explanation of balancing, see section 4.102 of the LA PA CK U sers' Guide.

## ARGUMENTS

BA LANC (input)
Indicates how the input matrix should be diagonally scaled and/orperm uted to im prove the conditioning of its eigenvalues. = N ': D ○ not diagonally scale orperm ute;
$=\mathrm{P}$ ': Perform perm utations to m ake the m atrix $m$ ore nearly upper triangular. D o notdiagonally scale; = S':D iagonally scale the matrix, ie. replace $A$ by $D$ *A *D ** ( -1 ), where $D$ is a diagonal $m$ atrix chosen to $m$ ake the row $s$ and colum ns of $A$ m ore equal in norm .D o notperm ute; = B ':Both diagonally scale and perm ute A .

C om puted reciprocalcondition num bersw illbe for the $m$ atrix afterbalancing and/orperm uting. Per$\mathrm{m} u$ uting does not change condition num bers (in exact arithm etic), butbalancing does.

JOBVL (input)
= N ': lefteigenvectors of A are not com puted;
= V': lefteigenvectors of A are computed. If SEN SE = E 'or B', JO BV L m ust= V'.

JO BV R (input)
= N ': righteigenvectors of A are not com puted;
$=\mathrm{V}$ ': righteigenvectors of A are com puted. If SEN SE = E 'or B', JO BV R m ust= V'.

SENSE (input)
D eterm ines which reciprocal condition num bers are com puted. = N ': N one are com puted;
= E ': C om puted foreigenvalues only;
$=\mathrm{V}$ : $: \mathrm{C}$ om puted for righteigenvectors only;
= B ': C om puted foreigenvalues and right eigenvectors.

If SEN SE = E 'or B ', both leftand right eigenvectors must also be com puted ( $\mathrm{JOBVL}=\mathrm{V}$ 'and $\mathrm{JOBVR}=\mathrm{V}$ ).

N (input) The order of the m atrix A . $\mathrm{N}>=0$.

A (input/output)
On entry, the $\mathrm{N}-$ by -N m atrix A. On exit, A has been overw rilten. If JOBVL = V'orJOBVR = V', $A$ contains the Schur form of the balanced version of the m atrix A.
LD A (input)
The leading din ension of the array A. LD A >= $\max (1, N)$.

W (output)
W contains the com puted eigenvalues.
VL (input)
If $\mathrm{JOBVL}=\mathrm{V}$ ', the left eigenvectors $u(1)$ are stored one after another in the colum ns of V L, in the sam e order as theireigenvahues. If $\mathrm{JOBVL}=$ $\mathrm{N}^{\prime}, \mathrm{VL}$ is not referenced. $\mathrm{u}(\mathcal{j})=\mathrm{VL}(:, 7)$, the $j$ th colum n ofVL.

LDVL (input)
The leading dim ension of the array V L. LD V L >=1; if $\mathrm{JOBVL}=\mathrm{V}$ ', LDVL $>=\mathrm{N}$.

VR (input)
If JOBVR = V', the right eigenvectors $\mathrm{v}(\mathrm{J})$ are stored one after another in the colum ns of VR, in the sam e order as theireigenvalues. If JOBVR $=$ $\mathrm{N}^{\prime}, \mathrm{VR}$ is notreferenced. $\mathrm{V}(\mathrm{O})=\mathrm{VR}(:, 7)$, the $j$ th colum n ofVR.

LDVR (input)
The leading dim ension of the array V R. LD V R >=1; if $J O B V R=V$ ', LDVR $>=N$.

ㅍO (output)
HO and $\mathbb{H}$ I are integervalues determ ined when $A$ w as balanced. The balanced $A(i, 7)=0$ if I> J and $J=1, \ldots, \mathbb{I L} O-1$ or $I=\mathbb{H} \mathrm{I}+1, \ldots, N$.

IH I (output)

ㅍO and $\mathbb{H}$ I are integer values determ ined when $A$ $w$ as balanced. The balanced $A(i, T)=0$ if $I>J$ and $J=1, \ldots, \mathbb{I L} O-1$ or $I=\mathbb{H} I+1, \ldots, N$.

SCALE (output)
D etails of the perm utations and scaling factors applied w hen balancing $A$. IfP $(j)$ is the index of the row and column interchanged $w$ th row and colum $n j$ and $D(j)$ is the scaling factorapplied to row and column $j$ then SCALE $(J)=P(J)$, for $J=1, \ldots, \Pi \circ-1=D(J)$ for $J=\mathbb{L O}, \ldots, \mathbb{H} I=$ P (J) for $J=\mathbb{H} I+1, \ldots, N$. The order in which the interchanges are $m$ ade is $N$ to $\mathbb{H} I+1$, then 1 to ㅍㅇㅇ.

## ABNRM (output)

The one-norm of the balanced $m$ atrix (the $m$ axim um of the sum of absolute values of elem ents of any colum n).

RCONE (output)
RCONE ( ) ) is the reciprocalcondition num ber of the $j$ th eigenvalue.

## RCONV (output)

RCONV ( $j$ ) is the reciprocalcondition num berof the $j$ th righteigenvector.

W ORK (w orkspace)
On exit, if $\mathbb{N F F O}=0, \mathrm{~W}$ ORK (1) retums the optim al LDW ORK.

LDW ORK (output)
The dim ension of the array $W$ ORK. IfSENSE $=N^{\prime}$ or $\mathrm{E}^{\prime}, \mathrm{LDW}$ ORK $>=\mathrm{max}\left(1,2^{*} \mathrm{~N}\right)$, and ifSENSE $=\mathrm{V}^{\prime}$ or B ', LDW ORK $>=\mathrm{N} * \mathrm{~N}+2{ }^{*} \mathrm{~N}$. Forgood perform ance, LD W ORK m ust generally be larger.

If LD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

## W ORK 2 (w orkspace)

dim ension ( $2 * \mathrm{~N}$ )
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an ille-
galvalue.
$>0:$ if $\mathbb{N} F O=$ i, the $Q R$ algorithm failed to com pute all the eigenvalues, and no eigenvectors or condition num bers have been com puted; elem ents 1: IHO-1 and i+1 N ofW contain eigenvalues which have converged.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgegs - routine is deprecated and has been replaced by routine CGGES

## SYNOPSIS



```
    LDVSL,VSR,LDVSR,W ORK,LDW ORK,W ORK 2, INFO)
CHARACTER * 1 JOBVSL, JOBVSR
COM PLEX A (LDA,*),B (LD B,*),ALPHA (*),BETA (*),VSL (LDVSL,*),
VSR (LDVSR,*),W ORK (*)
INTEGERN,LDA,LDB,LDVSL,LDVSR,LDW ORK,INFO
REALW ORK2 (*)
SUBROUTINE CGEGS_64(JOBVSL,JOBVSR,N,A,LDA,B,LDB,ALPHA,BETA,
        VSL,LDVSL,VSR,LDVSR,W ORK,LDW ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 JOBVSL,NOBVSR
COM PLEX A (LDA,*),B (LDB,*),ALPHA (*),BETA (*),VSL (LDVSL,*),
VSR (LDVSR,*),W ORK (*)
\mathbb{NTEGER*8N,LDA,LD B,LDVSL,LDVSR,LDW ORK,INFO}
REALW ORK2 (*)
```


## F95 INTERFACE

SU BROUTINE GEGS (OOBVSL, $\mathfrak{J} 0 \mathrm{BV}$ SR, $\mathbb{N}], A,[L D A], B,[L D B], A L P H A, B E T A$, VSL, [LDVSL],VSR, [LDVSR], [W ORK], [LDW ORK], [W ORK 2], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1)::J0 BVSL, JO BV SR
COM PLEX,D $\mathbb{M}$ ENSION (:) ::ALPHA,BETA,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A, B,VSL,VSR
$\mathbb{N} T E G E R:: N, L D A, L D B, L D V S L, L D V S R, L D W O R K, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::W ORK2

SU BROU T IN E G EG S_64 (JO BV SL , JO BV SR , $\mathbb{N}]$, A, [LD A ], B, [LD B ], A LPH A , BETA, VSL, [LDVSL],VSR, [LDVSR], [W ORK], [LDW ORK], [W ORK2], [ $\mathbb{N}$ FO ])

CHARACTER ( $\llcorner E N=1$ ) :: JOBVSL, JOBV SR
COM PLEX,D $\mathbb{I M} E N S I O N(:):: A L P H A, B E T A, W O R K$

$\mathbb{N}$ TEGER (8) :: N , LD A ,LDB,LDVSL, LDVSR, LDW ORK, $\mathbb{N} F O$
REAL,D $\mathbb{I}$ ENSION (:) ::W ORK2

## C INTERFACE

\#include <sunperfh>
void cgegs (char jobvsl, char j’bvsr, intn, com plex *a, int lda, com plex *b, int ldb, com plex *alpha, com plex *beta, com plex *vsl, int ldvsl, com plex *Vsr, int ldvsr, int *info);
void ogegs_64 (char j̇bvsl, char jobvss, long n, com plex *a, long lda, com plex *b, long ldb, com plex *alpha, com plex *beta, com plex *vsl, long ldvsl, com plex *vsr, long ldvss, long *info);

## PURPOSE

cgegs routine is deprecated and has been replaced by routine CGGES .

CGEGS com putes for a pair of N łoy- N com plex nonsymm etric $m$ atrices A, B: the generalized eigenvalues (alpha, beta), the com plex Schur form (A , B ), and optionally left and/or rightSchurvectors ( V SL and V SR ) .
(If only the generalized eigenvalues are needed, use the driverCGEGV instead.)

A generalized eigenvalue for a pair of $m$ atrices ( $A, B$ ) is, roughly speaking, a scalar w or a ratio alphaßeta $=\mathrm{w}$, such that $A-w * B$ is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation forbeta $=0$, and even forboth being zero. A good beginning reference is the book, "M atrix C om putations", by G.G olub \& C .van Loan (Johns H opkins U .Press)

The (generalized) Schur form of a pair of $m$ atriges is the result ofm ultiplying both $m$ atrices on the leftby one unitary $m$ atrix and both on the rightby another unitary $m$ atrix, these two unitary $m$ atrioes being chosen so as to bring the pair ofm atrioes into upper triangular form w ith the diago-
nal elem ents ofB being non-negative realnum bers this is also called com plex Schur form .)

The left and rightSchurvectors are the colum ns ofV SL and VSR, respectively, where VSL and VSR are the unitary $m$ atrices
which reduce A and B to Schur form :

Schurform of $(A, B)=(N S L) * * H A(N S R),(N S L) * * H B(N S R))$

## ARGUMENTS

JO BVSL (input)
$=N^{\prime}:$ do notcom pute the left:Schurvectors;
= V ': com pute the leftSchurvectors.
$J O B V S R$ (input)
$=N$ : do notcom pute the rightSchurvectors;
= V ': com pute the rightSchurvectors.
N (input) The order of the m atrices A, B, V SL, and VSR. N $>=0$.

A (input/output)
O $n$ entry, the firstof the pairofm atrices whose generalized eigenvalues and (optionally) Schur vectors are to be com puted. On exit, the generalized Schur form of A.

LD A (input)
The leading dim ension ofA. LD A $>=\max (1, N)$.
B (input/output)
O n entry, the second of the pair ofm atrices w hose generalized eigenvalues and (optionally) Schur vectors are to be com puted. O n exit, the general ized Schur form ofB.

LD B (input)
The leading dim ension ofB. LD B $>=m$ ax $(1, N)$.

## A LPHA (output)

On exit, ALPHA ( $\mathfrak{j}$ ) BETA ( $\mathcal{j}$ ) $\mathfrak{j} 1, \ldots, N$, w illbe the generalized eigenvalues. A LPHA ( $\mathcal{\imath}, \dot{于} 1, \ldots, \mathrm{~N}$ and $\operatorname{BETA}(\mathcal{j}) \dot{j} 1, \ldots, N$ are the diagonals of the com plex Schur form ( $A, B$ ) output by CGEGS. The BETA (j) w illbe non-negative real.

N ote: the quotients A LPHA ( $\mathcal{7}$ ) $B E T A(\mathcal{)}$ may easily
over- orunderflow, and BETA ( 7 ) m ay even be zero. Thus, the user should avoid naively com puting the ratio alpha/beta. H ow ever, A LPH A w illbe alw ays less than and usually com parable w ith norm (A) in m agnitude, and BETA alw ays less than and usually com parable w ith norm (B).

BETA (output)
See the description of A LPH A.

VSL (input)
If JOBVSL = V',VSL w illcontain the left Schur vectors. (See "Puppose", above.) N ot referenced if $J O B V S L=N^{\prime}$ 。

LDVSL (input)
The leading dim ension of the m atrix V SL.LDV SL >= 1 , and if $\mathrm{OBVSL}=\mathrm{V}^{\prime}$, LDVSL $>=\mathrm{N}$.

VSR (input)
If OB BVSR $=V$ ',VSR willcontain the right Schur
vectors. (See "Puppose", above.) N ot referenced if JO BV SR = $N^{\prime}$ 。

LDVSR (input)
The leading dim ension of the $m$ atrix $V$ SR.LD V SR $>=$ 1 , and if OB B $S R=V^{\prime}$ ', LD $V S R>=N$.

W ORK (w orkspace)
On exit, if $\mathbb{N F O}=0, W$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dim ension of the anray $W$ ORK. LDW ORK $>=$ $m a x(1,2 * N)$. For good perform ance, LD W ORK must generally be larger. To com pute the optim alvalue of LDW ORK, call ILAENV to getblocksizes (for $C G E Q R F, C U N M Q R$, and $C U N G Q R$.) Then com pute: $N B$ as the MAX of the blocksizes forCGEQRF, CUNM QR, and CUNGQR; the optim alLDW ORK is $N$ * $(\mathbb{B}+1)$.

IfLDW ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK2 (w orkspace)
dim ension $(3 * N)$
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the ith argum enthad an illegalvalue.
$=1, \ldots, N$ : The Q Z iteration failed. ( $\mathrm{A}, \mathrm{B}$ ) are not
in Schur form, butA LPHA ( 1 ) and BETA ( 7 ) should be
correct for $\mathcal{F} \mathbb{N F O}+1, \ldots, N .>N: ~ e m o r s ~ t h a t ~$
usually indicate LA PA C K problem s:
$=\mathrm{N}+1$ : emor retum from CGGBAL
$=\mathrm{N}+2$ : error retum from $C G E Q R F$
$=\mathrm{N}+3$ : emor retum from $C U N M Q R$
$=\mathrm{N}+4$ : error retum from CUNGQR
$=\mathrm{N}+5$ : error retum from CGGHRD
$=\mathrm{N}+6$ : error retum from CHGEQZ (otherthan failed iteration) $=\mathrm{N}+7$ : error retum from CGGBAK (com puting V SL)
$=\mathrm{N}+8$ : error retum from CGGBAK (com puting V SR)
$=\mathrm{N}+9$ : error retum from CLASCL (variousplaces)

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgegv - routine is deprecated and has been replaced by routine CGGEV

## SYNOPSIS

```
SUBROUT\mathbb{NE CGEGV (JOBVL,JOBVR,N,A,LDA,B,LDB,ALPHA,BETA,VL,}
    LDVL,VR,LDVR,W ORK,LDW ORK,W ORK 2, \mathbb{NFO)}
```

CHARACTER * 1 JOBVL, JOBVR

VR (LDVR,*), W ORK (*)
$\mathbb{N}$ TEGER N,LDA,LDB,LDVL,LDVR,LDW ORK, $\mathbb{N} F$ O
REALWORK2 (*)
SU BROUTINE CGEGV_64 (JO BVL, JO BVR,N,A,LDA,B,LDB,ALPHA,BETA,VL,
LDVL,VR,LDVR,W ORK,LDW ORK,W ORK2, $\mathbb{N} F O$ )
CHARACTER * 1 JOBVL, JOBVR
COM PLEX A (LDA,*), B (LD B ,*), ALPHA (*), BETA (*), VL (LDVL,*),
VR (LDVR,*), W ORK ( ${ }^{*}$ )
$\mathbb{N}$ TEGER*8N,LDA,LDB,LDVL,LDVR,LDW ORK, $\mathbb{N} F$ F

REAL W ORK 2 (*)

## F95 INTERFACE

SU BROUTINE GEGV (JOBVL, JOBVR, $\mathbb{N}], A,[L D A], B,[L D B], A L P H A, B E T A$, VL, [LDVL],VR, [LDVR], [W ORK], [LDW ORK], [W ORK2], [ $\mathbb{N F O}])$

CHARACTER (LEN=1) :: JOBVL, JOBVR
COMPLEX,D $\mathbb{M}$ ENSION (:) ::ALPHA,BETA,W ORK
COM PLEX,D $\mathbb{I M}$ ENSION (: : : : : A, B, VL,VR
$\mathbb{N} T E G E R:: N, L D A, L D B, L D V L, L D V R, L D W O R K, \mathbb{N F O}$

SU BROUTINE GEGV_64 (JOBVL, $\operatorname{OOBVR}, \mathbb{N}], A,[L D A], B,[L D B], A L P H A$, BETA, VL, [LDVL], VR, [LDVR], $\mathbb{W} O R K],[L D W O R K],[W O R K 2],[\mathbb{N} F O])$

CHARACTER ( $L E N=1$ ) :: JOBVL, OOBVR
COM PLEX,D $\mathbb{I M} E N S I O N(:):: A L P H A, B E T A, W O R K$
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : A , B , VL, VR
$\mathbb{N} \operatorname{TEGER}$ (8) :: $\mathrm{N}, \mathrm{LD} A, L D B, L D V L, L D V R, L D W O R K, \mathbb{N} F O$
REAL,D $\mathbb{I}$ ENSION (:) ::W ORK2

## C INTERFACE

\#include <sunperfh>
void cgegv (char jंjbvl, char jobvr, intn, com plex *a, int lda, com plex *b, int ldlb, com plex *alpha, com plex *beta, com plex *vl, int ldvl, com plex *vr, int ldvx, int*info);
void ogegv_64 (char jobvl, char jobvr, long n, com plex *a, long lda, com plex *b, long ldl, com plex *alpha, com plex *beta, complex *vl, long ldvl, complex *vr, long ldvr, long *info);

## PURPOSE

cgegv routine is deprecated and has been replaced by routine CGGEV .

CGEGV com putes fora pair of N boy N com plex nonsymm etric $m$ atrices $A$ and $B$, the generalized eigenvalues (alpha, beta), and optionally, the leftand/or rightgeneralized eigenvectors ( VL and $V R$ ).

A generalized eigenvalue for pair of $m$ atrices ( $A, B$ ) is, roughly speaking, a scalar w ora ratio alpha/beta $=\mathrm{w}$, such that $A-w * B$ is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation forbeta $=0$, and even forboth being zero. A good beginning reference is the book, "M atrix C om putations", by G.G olub \& C . van Loan (Johns H opkins U .Press)

A rightgeneralized eigenvector comesponding to a generalized eigenvalue $w$ for pair ofm atrioes $(A, B)$ is a vector $r$ such that ( $A-w B$ ) r=0.A left generalized eigenvector is a vectorlsuch that $l^{* *} H *(A-w B)=0$, where ${ }^{* * *} \mathrm{H}$ is the conjugate-transpose of l.

N ote: this routine perform s "fullbalancing" on A and B. See "FurtherD etails", below .

## ARGUMENTS

JOBVL (input)
$=\mathrm{N}^{\prime}$ : do not com pute the leftgeneralized eigenvectors;
$=\mathrm{V}$ ': com pute the left generalized eigenvectors.
JOBVR (input)
$=\mathrm{N}$ ': do not com pute the right generalized eigenvectors;
$=\mathrm{V}$ : com pute the right generalized eigenvectors.

N (input) The order of the matrices $A, B, V L$, and $V R . N \quad>=$ 0.

A (input/output)
O n entry, the first of the pairofm atrices w hose generalized eigenvalues and (optionally) generalized eigenvectors are to be com puted. On exit, the contents will have been destroyed. (Fora description of the contents of A on exit, see "FurtherD etails", below .)

LD A (input)
The leading dim ension ofA. LD A $>=\max (1, \mathbb{N})$.
B (input/output)
O n entry, the second of the pair ofm atrices w hose generalized eigenvalues and (optionally) generalized eigenvectors are to be com puted. On exit, the contents w ill have been destroyed. (Fora description of the contents of $B$ on exit, see "FurtherD etails", below .)

LD B (input)
The leading dim ension ofB. LD B $>=\mathrm{max}(1, \mathrm{~N})$.
ALPHA (output)
On exit, A LPHA ( ) NL ( $\mathfrak{j}$ ) $\dot{=} 1, \ldots, N$, will be the generalized eigenvalues.

N ote: the quotients ALPHA ( $)$ NL ( $\mathcal{j}$ ) may easily over- or underflow, and VL ( 7 ) m ay even be zero. Thus, the user should avoid naively com puting the
ratio alpha/beta. H ow ever, A LPH A w illbe alw ays less than and usually com parable w th norm (A) in m agnitude, and VL alw ays less than and usually com parable $w$ ith norm (B).

VL (output)
If $\mathrm{JO} \mathrm{BVL}=\mathrm{V}$ ', the left generalized eigenvectors. (See "Purpose", above.) Each eigenvectorw ill.be scaled so the largest com ponentw ill have abs (real part) + abs(mag. part) $=1$, *exœept that for eigenvalues $w$ th alpha=beta=0, a zero vector $w i l l$ be retumed as the corresponding eigenvector. N ot referenced if $\mathrm{JOBVL}=\mathrm{N}^{\prime}$.
BETA (output)
If $\mathrm{JOBVL}=\mathrm{V}$ ', the left generalized eigenvectors.
(See "Pupose", above.) Each eigenvectorw ill.be scaled so the largest com ponentw ill have abs (real part) $+\mathrm{abs}($ ( m ag. part) $=1$, *exœept* that for eigenvalues w ith alpha=beta=0, a zero vector w ill be retumed as the comesponding eigenvector. N ot referenced if $\mathrm{JO} \mathrm{BVL}=\mathrm{N}$ '.

LDVL (input)
The leading dim ension of the $m$ atrix $V \mathrm{~L} . \mathrm{LD} V \mathrm{~L}>=1$, and if $\mathrm{JOBVL}=\mathrm{V}$ ', LDVL >=N.

## VR (output)

If $\mathrm{JOBVR}=\mathrm{V}$ ', the right generalized eigenvectors. (See "Pupose", above.) Each eigenvector w illbe scaled so the largest com ponentw ill have abs(real part) +abs (im ag. part) $=1$,*except* that foreigenvalues $w$ th alpha=beta=0, a zero vector will be retumed as the corresponding eigenvector. N ot referenced if $\mathrm{JO} B V R=\mathrm{N}^{\prime}$.

LDVR (input)
The leading dim ension of the $m$ atrix $V R$.LD $V R>=1$, and if $J O B V R=V$ ', LDVR $>=N$.

W ORK (w orkspace)
On exit, if $\mathbb{N F} F=0, W$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the array W ORK. LDW ORK >= $\max (1,2 \star \mathrm{~N})$. For good perform ance, LDW ORK must generally be larger. To com pute the optim alvalue of LDW ORK, call IIA ENV to getblocksizes (for CGEQRF, CUNM QR, and CUNGQR.) Then com pute: NB as the MAX of the blocksizes forCGEQRF, CUNM QR, and

CUNGQR; The optim alldW ORK is MAX $(2 * N, N * \mathbb{N}+1)$
).
IfLDW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
W ORK 2 (w orkspace)
dim ension ( $8 * \mathrm{~N}$ )
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0$ : if $\mathbb{N}$ FO $=-$ i, the i-th argum enthad an illegalvahue.
$=1, \ldots, N$ : The Q Z iteration failed. N o eigenvectors have been calculated, butA LPH A ( $\mathcal{j}$ ) and VL ( $\mathcal{j}$ )
should be correct for $\mathcal{j} \mathbb{N} F O+1, \ldots, N .>N$ :
enrors that usually indicate LA PACK problem s:
$=\mathrm{N}+1$ : error retum from CGGBAL
$=\mathrm{N}+2$ : error retum from CGEQRF
$=N+3$ : error retum from CUNM $Q R$
$=\mathrm{N}+4$ : error retum from CUNGQR
$=\mathrm{N}+5$ : error retum from CGGHRD
$=\mathrm{N}+6$ : error retum from CHGEQZ (other than failed iteration) $=\mathrm{N}+7$ : emor retum from CTGEVC
$=\mathrm{N}+8$ : enror retum from CGGBAK (com puting VL)
$=\mathrm{N}+9$ : error retum from CGGBAK (com puting VR)
$=\mathrm{N}+10$ : enror retum from CLASCL (various calls)

## FURTHER DETAILS

Balancing

This driver calls C G G B A L to both perm ute and scale row s and colum ns of $A$ and $B$. The perm utationsPL and PR are chosen so that $\mathrm{PL} * \mathrm{~A} * P R$ and $P L * B * R$ w illlbe upper triangular except for the diagonal blocksA ( $i: j i: j$ ) and $B(i: j i: j$, w ith $i$ and jas close together as possible. The diagonal scaling $m$ atrices $D L$ and $D R$ are chosen so that the pair DL*PL*A*PR*DR,DL*PL*B*PR*DR have elem ents close to one (except for the elem ents that start out zero .)

A fter the eigenvalues and eigenvectors of the balanced $m$ atrices have been com puted, C G G B A K transform s the eigenvectors back to $w$ hat they w ould have been (in perfect arithm etic) if they had notbeen balanced.

Contents of $A$ and $B$ on Exit

If any eigenvectors are com puted (either $J 0 B V L=V$ ' or JO $B V R=V$ ' or both), then on exit the arrays $A$ and $B$ will contain the com plex Schur form [*] of the "balanced" versions of $A$ and $B$. If no eigenvectors are com puted, then only the diagonalblocksw illbe conect.
[ $\star$ ] In otherw ords, uppertriangular form .

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgehrd -reduce a com plex generalm atrix A to upper H essen-
berg form $H$ by a unitary sim ilarity transform ation

## SYNOPSIS



```
COM PLEX A (LDA,*),TAU (*),W ORK\mathbb{N (*)}
```




```
COM PLEX A (LDA,*),TAU (*),W ORK\mathbb{N (*)}
```



## F95 INTERFACE

SU BROUT $\mathbb{N} E \operatorname{GEHRD}(\mathbb{N}], \mathbb{L O}, \mathbb{H} I, A,[L D A], T A U,[W O R K \mathbb{N}],[L W O R K \mathbb{N}]$, [ $\mathbb{N}$ FO ])

COM PLEX,D $\mathbb{I}$ ENSION (:) ::TAU,W ORK $\mathbb{N}$
COM PLEX,D $\mathbb{I}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER :: $\mathrm{N}, \mathbb{L} \mathrm{O}, \mathbb{H} \mathrm{I}, \mathrm{LDA}, \mathrm{LW} O R K \mathbb{N}, \mathbb{N} F O$
SU BROUTINE GEHRD_64 ( $\mathbb{N}], \mathbb{I} O, \mathbb{H} I, A,[L D A], T A U,[W O R K \mathbb{N}],[L W O R K \mathbb{N}]$, [ $\mathbb{N}$ FO ])

COM PLEX,D $\mathbb{M}$ ENSION (:) ::TAU,W ORK $\mathbb{N}$
COMPLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER (8) ::N, $\mathbb{N} O, \mathbb{H} I, L D A, L W O R K \mathbb{N}, \mathbb{N} F O$
void ogehrd (intn, intilo, intini, com plex *a, int lda, com plex *tau, int *info);
void ogehrd_64 (long n, long ilo, long ihi, com plex *a, long lda, com plex *tau, long *info);

## PURPOSE

cgehrd reduces a com plex generalm atrix A to upper H essenberg form $H$ by a untary sim ilarity transform ation: $Q^{\prime \prime *} A$

* $\mathrm{Q}=\mathrm{H}$.


## ARGUMENTS

N (input) The order of them atrix $A . N>=0$.

IIO (input)
It is assum ed that A is already upper triangular in row s and colum ns $1: \mathbb{W} 0-1$ and $\mathbb{H} \mathrm{I}+1 \mathbb{N} . \mathbb{I} O$ and
IH I are norm ally setby a previous call to CGEBAL; otherw ise they should be setto 1 and $N$ respectively . See FurtherD etails.

IH I (input)
See the description of $\amalg \mathrm{O}$.

A (input/output)
O n entry, the N boy N generalm atrix to be reduced.
O n exit, the upper triangle and the first subdiagonalofA are overw rilten $w$ ith the upper $H$ essenberg $m$ atrix $H$, and the elem ents below the first subdiagonal, w ith the array TAU, represent the unitary $m$ atrix $Q$ as a product of elem entary reflectors. See FurtherD etails.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, N)$.

TAU (output)
The scalar factors of the elem entary reflectors (see Further Details). Elem ents 1:Ш1O-1 and IH IN -1 ofTAU are setto zero.

W ORK $\mathbb{N}$ (w orkspace)

On exit, if $\mathbb{N F O}=0, W O R K \mathbb{N}$ (1) retums the optim allW ORK $\mathbb{N}$.

LW ORK $\mathbb{N}$ (input)
The length of the array $W O R K \mathbb{N}$. LW ORK $\mathbb{N}>=$ $\max (1, \mathbb{N})$. For optim um perform ance LW ORK $\mathbb{N}>=$ N *N B , w here N B is the optim alblocksize.

If LW ORK $\mathbb{N}=-1$, then a workspace query is assum ed; the routine only calculates the optim al size of the $W$ ORK $\mathbb{N}$ array, retums this value as the firstentry of the $W$ ORK $\mathbb{N}$ array, and no error $m$ essage related to LW ORK $\mathbb{N}$ is issued by XERBLA.
$\mathbb{N F O}$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the i-th argum enthad an illegalvahue.

## FURTHER DETAILS

Them atrix $Q$ is represented as a productof (hi-ib) ele$m$ entary reflectors

Each H (i) has the form

$$
\mathrm{H}(\mathrm{i})=\mathrm{I}-\tan * \mathrm{~V}^{*} \mathrm{~V}^{\prime}
$$

$w$ here tau is a com plex scalar, and $v$ is a com plex vector w ith $\mathrm{v}(1: i)=0, \mathrm{v}(i+1)=1$ and $v(i h i+1 \mathrm{~m})=0 ; \mathrm{v}(i+2$ : ihi $)$ is stored on exitin A (i+2:ihi,i), and tau in TAU (i).

The contents ofA are illustrated by the follow ing exam ple, w ith $\mathrm{n}=7$, $\mathrm{il}=2$ and ihi $=6$ :

```
on entry, on ex斗,
```

(a a a a a a a) (a a h h h h a) ( a a a a a a) ( a h h h $h a)(a \operatorname{a} a \operatorname{a} a)(\mathrm{h} h \mathrm{~h}$ h h h ) ( a a a a a a) ( v2 h h h h h) ( a a a a a a) ( v2 v3 h h h h ) ( a a a a a a) ( v2 v3 v4 h h h ) (a) ( a)
where a denotes an elem ent of the original $m$ atrix $A, h$ denotes a $m$ odified elem ent of the upper $H$ essenberg $m$ atrix $H$, and videnotes an elem ent of the vector defining $H$ (i).

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgelgf-com pute an LQ factorization of a com plex M -by-N $m$ atrix A

## SYNOPSIS

```
SUBROUTINE CGELQF M,N,A,LDA,TAU,W ORK,LDW ORK,\mathbb{NFO)}
```

COM PLEX A (LDA,*),TAU (*),W ORK (*)
$\mathbb{N}$ TEGER M,N,LDA,LDW ORK, $\mathbb{N} F O$
SUBROUTINE CGELQF_64 $M, N, A, L D A, T A U, W O R K, L D W O R K, \mathbb{N} F O)$
COM PLEX A (LDA,*),TAU (*),W ORK (*)
$\mathbb{N} T E G E R * 8 \mathrm{M}, \mathrm{N}, \operatorname{LD} A, L D W$ ORK, $\mathbb{N} F O$

## F95 INTERFACE

SU BROUTINE GELQF (M ], $\mathbb{N}], A,[L D A], T A U, \mathbb{W} O R K],[L D W O R K],[\mathbb{N F O}])$
COM PLEX,D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
COMPLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER ::M , N,LDA,LDW ORK, $\mathbb{N} F O$
SU BROUTINE GELQF_64 ( $\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N} F O])$
COM PLEX,D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
COM PLEX,D $\mathbb{I}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER (8) ::M , N, LDA, LDW ORK, $\mathbb{N}$ FO

## C INTERFACE

\#include < sunperfh>
void cgelqf(intm , intn, com plex *a, int lda, com plex *tau, int*info);
void cgelqf_64 (long m, long n, com plex *a, long lda, com plex
*tau, long *info);

## PURPOSE

cgelqfoom putes an LQ factorization of a complex M -by-N $m$ atrix $A: A=L * Q$.

## ARGUMENTS

M (input) The num ber of row s of the $m$ atrix $\mathrm{A} . \mathrm{M}>=0$.
N (input) The num ber of colum ns of the matrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
On entry, the M -by-N m atrix A. On exit, the ele$m$ ents on and below the diagonal of the array contain the $m$-by-m in $(m, n)$ low er trapezoidalm atrix $L$ ( $L$ is low er triangularifm <= n); the elem ents above the diagonal, w ith the array TA $U$, represent the unitary matrix $Q$ as a productofelem entary reflectors (see FurtherD etails).

LDA (input)
The leading dim ension of the array A. LDA >= max (1, M).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the array $W$ ORK. LDW ORK >= $m a x(1, M)$. Foroptim um perform ance LDW ORK $>=M$ *NB, w here NB is the optim alblocksize.

If LD W ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim al size of
the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LDW ORK is issued by X ERBLA .
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an illegalvalue

## FURTHER DETAILS

Them atrix $Q$ is represented as a product of elem entary reflectors
$Q=H(k)^{\prime} \ldots H(2)$ 'H $(1)$ ', where $k=m$ in $(m, n)$.
Each $H$ (i) has the form
H (i) $=I-\tan * V^{*} V^{\prime}$
$w$ here tau is a com plex scalar, and $v$ is a com plex vector w ith $v(1: i-1)=0$ and $v(i)=1$; con $\dot{g}(v(i+1 n))$ is stored on exitin A ( $i, i+1 \mathrm{n}$ ), and tau in TAU (i).

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
cgels - solve overdeterm ined or underdeterm ined com plex
linear systems involving an M -by }\textrm{N}\mathrm{ N matrix A, or its
conjugate-transpose, using a Q R orLQ factorization ofA
```


## SYNOPSIS

```
SUBROUTINE CGELS (TRANSA,M ,N,NRHS,A,LDA,B,LDB,W ORK,LDW ORK,
    \mathbb{NFO)}
CHARACTER * 1 TRANSA
COM PLEX A (LDA,*),B (LD B,*),W ORK (*)
\mathbb{NTEGERM,N,NRHS,LDA,LDB,LDW ORK,\mathbb{NFO}}\mathbf{N},\mp@code{N},\mp@code{N}
SU BROUTINE CGELS_64 (TRANSA,M ,N,NRHS,A,LDA,B,LDB,W ORK,LDW ORK,
    INFO)
```

CHARACTER * 1 TRANSA
COM PLEX A (LDA, *), B (LD B,*), W ORK (*)
$\mathbb{N}$ TEGER*8M,N,NRHS,LDA,LDB,LDW ORK, $\mathbb{N} F O$

## F95 INTERFACE

SU BROUTINE GELS ([TRANSA], $\mathbb{M}], \mathbb{N}], \mathbb{N R H S}], A,[L D A], B,[L D B],[W$ ORK ], LDW ORK, [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) ::TRANSA
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A,B
$\mathbb{N} T E G E R:: M, N, N R H S, L D A, L D B, L D W O R K, \mathbb{N} F O$

SU BROU T $\mathbb{N} E$ GELS_64 ([TRANSA ], $\mathbb{M}], \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B]$, [ $\mathbb{W}$ ORK ],LDW ORK, [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::TRANSA
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A,B
$\mathbb{N}$ TEGER (8) ::M,N,NRHS,LDA,LDB,LDW ORK, $\mathbb{N} F O$

## C INTERFACE

\#include <sunperfh>
void cgels (char, int, int, int, com plex*, int, com plex*, int, int ${ }^{\star}$ );
void cgels_64 (char, long, long, long, com plex*, long, com plex*, long, long*);

## PURPOSE

cgels solves overdeterm ined or underdeterm ined com plex linear system $s$ involving an $M$ by N matrix $A$, or its conjugate-transpose, using a Q R orLQ factorization of A. It is assum ed thatA has full rank.

The follow ing options are provided:

1. IfTRANS $=N$ 'and $m>=n$ : find the leastsquares solution of
an overdeterm ined system , i.e., solve the least squares problem $m$ inim ize $\|B-A * X\|$.
2. IfTRAN $S=N$ 'and $m<n$ : find the $m$ inim um norm solution of an underdeterm ined system $A * X=B$.
3. IfTRAN $S=C$ 'and $m>=n$ : find the $m$ inim um norm solution of
an undeterm ined system $A * * H * X=B$.
4. IfTRANS = C'and $m<n$ : find the least squares solution of
an overdeterm ined system , ie., solve the least squares problem

$$
\mathrm{m} \text { in.m ize }\|\mathrm{B}-\mathrm{A} * * \mathrm{H} * \mathrm{X}\| .
$$

Several righthand side vectors b and solution vectors $x$ can be handled in a single call; they are stored as the colum ns of the M -by-NRHS righthand side m atrix B and the $N$-by-NRHS solution $m$ atrix $X$.

## ARGUMENTS

TRANSA (input)
$=\mathrm{N}$ : the linearsystem involves A;
$=C$ ': the linear system involves $A * * H$.

TRANSA is defaulted to $N$ 'forF $95 \mathbb{I N}$ TERFACE.

M (input) The num ber of row s of the matrix A. M >=0.

N (input) The num ber of colum ns of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atriges B and X .NRH S >=0.
A (input/output)
On entry, the M boy-N m atrix A. ifM $>=N, A$ is overw ritten by details of its $Q R$ factorization as retumed by CGEQRF; ifM < N, A is overw ritten by details of its LQ factorization as retumed by CGELQF.

LDA (input)
The leading dim ension of the anay A. LDA >= $\max (1, M)$.

B (input/output)
O n entry, the m atrix B of righthand side vectors,
stored colum nw ise; B is M byy-NRHS ifTRANSA = N',
orN byy-NRHS ifTRANSA = C'. On exit, B is
overw rilten by the solution vectors, stored colum nw ise: ifTRANSA $=N$ 'and $m>=n$, row 1 to n ofB contain the least squares solution vectors; the residual.sum ofsquares for the solution in each colum $n$ is given by the sum of squares of ele$m$ ents $N+1$ to $M$ in thatcolum $n$; ifTRA $N S A=N$ 'and $m<n$, row $s 1$ to $N$ ofB contain them inim um norm solution vectors; ifTRANSA $=C$ 'and $m>=n$, row $s$ 1 to M ofB contain them inim um norm solution vectors; ifTRANSA $=C$ 'and $m<n$, row 1 to $M$ of $B$ contain the least squares solution vectors; the residualsum of squares for the solution in each colum $n$ is given by the sum of squares of elem ents $M+1$ to $N$ in that colum $n$.

LD B (input)
The leading dim ension of the amay $\mathrm{B} . \mathrm{LD} \mathrm{B}>=$ M AX ( $1, \mathrm{M}, N$ ) 。

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LDW ORK.

LDW ORK (output)
The dim ension of the amay W ORK. LDW ORK >= $\max ($
1, $\mathrm{M} N+\max (\mathrm{M} N, \mathrm{NRHS})$ ). Foroptim alperfor
$m$ ance, LDW ORK $>=\max (1, M N+\max (M N, N R H S) * N B$
). where $M N=m$ in $M N$ ) and $N B$ is the optim um
block size.

IfLD W ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
theW ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LD W ORK is issued by XERBLA.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the i-th argum enthad an ille-
galvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgelsd -com pute the m inim um -norm solution to a real linear
least squares problem

## SYNOPSIS

```
SUBROUT\mathbb{NE CGELSD M,N,NRHS,A,LDA,B,LDB,S,RCOND,RANK,W ORK,}
    LW ORK,RW ORK,\mathbb{N ORK,\mathbb{NFO)}}\mathbf{N}=(
COM PLEX A (LDA,*),B (LD B,*),W ORK (*)
INTEGERM,N,NRHS,LDA,LDB,RANK,LW ORK, INFO
INTEGER IN ORK (*)
REALRCOND
REALS (*),RW ORK (*)
SUBROUT\mathbb{NECGELSD_64M,N,NRHS,A,LDA,B,LDB,S,RCOND,RANK,}
        W ORK,LW ORK,RW ORK,\mathbb{IN ORK,\mathbb{NFO)}}\mathbf{N}=(
COM PLEX A (LDA,*),B (LD B,*),W ORK (*)
\mathbb{NTEGER*8M,N,NRHS,LDA,LDB,RANK,LW ORK,INFO}
INTEGER*8 IN ORK (*)
REAL RCOND
REALS (*),RW ORK (*)
```


## F95 INTERFACE

```
SU BROUTINE GELSD ( \(\mathbb{M}], \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], S, R C O N D\), RANK, [W ORK ], [LW ORK], RW ORK], [IW ORK], [ \(\mathbb{N F O}\) ])
COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,B
\(\mathbb{N} T E G E R:: M, N, N R H S, L D A, L D B, R A N K, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
```

REAL: RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::S,RW ORK
SU BROUTINE GELSD_64 (M) $\mathbb{M}, \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], S, R C O N D$, RANK, [W ORK ], [LW ORK], RW ORK], [IW ORK], [ $\mathbb{N F O}]$ )

COM PLEX,D $\mathbb{M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A, B
$\mathbb{N}$ TEGER (8) :: M , N,NRHS,LDA,LDB,RANK,LW ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{I W}$ ORK
REAL ::RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::S,RW ORK

## C INTERFACE

\#include < sunperfh>
void cgelsd (intm, intn, intnrhs, com plex *a, int lda, complex *b, int ldb, float *s, float roond, int
*rank, int*info);
void cgelsd_64 (long m, long n, long nrhs, com plex *a, long
lda, com plex *b, long ldb, float *s, float roond, long *rank, long *info);

## PURPOSE

cgelsd com putes the $m$ inim um -norm solution to a real linear least squares problem :
$m$ inim ize 2 -norm (|b-A *x |)
using the singularvalue decom position (SVD ) ofA.A is an M -by-N m atrix which m ay be rank-deficient.

Several righthand side vectors b and solution vectors x can be handled in a single call; they are stored as the colum ns of the M -by-NRHS righthand side m atrix $B$ and the $N$ by-NRHS solution $m$ atrix $X$.

The problem is solved in three steps:
(1) Reduce the coefficientm atrix A to bidiagonal form $w$ ith H ouseholdertranform ations, reducing the original problem
into a "bidiagonal least squares problem " (BLS)
(2) Solve the BLS using a divide and conquer approach.
(3) A pply back all the H ouseholder tranform ations to solve the original least squares problem .

The effective rank of is determ ined by treating as zero those singular values which are less than RCOND tim es the largest singular value.

The divide and conquer algorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on $m$ achines w ith a guard digit in add/subtract, or on those binary $m$ achines $w$ thout guard digits w hich subtract like the $C$ ray $X-M P, C$ ray $Y-M P, C$ ray $C-90$, or C ray-2. It could conceivably fail on hexadecim al or decim al machines w thout guard digits, butw e know of none.

## ARGUMENTS

M (input) The num ber of row sof the $m$ atrix $A . M>=0$.
N (input) The num ber of colum ns of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

NRH S (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the $m$ atrices $B$ and X.NRHS $>=0$.

A (input/output)
On entry, the M -by -N m atrix A. On exit, A has been destroyed.

LD A (input)
The leading dim ension of the array A. LDA >= max (1,M).

B (input/output)
On entry, the M-by-NRHS righthand sidem atrix B . On exit, B is overw ritten by the N -by-NRH S solution $m$ atrix $X$. If $m>=n$ and RANK $=n$, the residual sum -of-squares for the solution in the $i$-th colum $n$ is given by the sum of squares of elem ents $\mathrm{n}+1 \mathrm{~m}$ in thatcolumn.

LD B (input)
The leading dim ension of the amay $B$. LD B >= $\max (1, M, N)$.

S (output)
The singular values ofA in decreasing order. The condition number of $A$ in the 2 -norm $=$ $S(1) / S(m$ in $(m, n))$.

RCOND (input)
RCOND is used to determ ine the effective rank of
A. Singularvalues $S(i)<=$ RCOND *S ( 1 ) are treated as zero. IfRCOND $<0, m$ achine precision is used instead.

RANK (output)
The effective rank of A, ie., the num ber of singular values w hich are greater than RCOND *S (1).

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the anray $W$ ORK. LW ORK $>=1$. The exact $m$ inim um am ount of $w$ orkspace needed depends on $\mathrm{M}, \mathrm{N}$ andNRHS. IfM >= N , LW ORK >= 2*N + $N * N R H S$. If $M<N$, LW ORK $>=2 * M+M * N R H S$. For good penform ance, LW ORK should generally be larger.

IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of theW ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

RW ORK (w orkspace)
IfM $>=\mathrm{N}$, LRW ORK $>=8{ }^{*} \mathrm{~N}+2{ }^{\star} \mathrm{N} *$ SM LSIZ $+8{ }^{*} \mathrm{~N} * \mathrm{NLVL}+$ $\mathrm{N} * \mathrm{~N}$ RHS. If $\mathrm{M}<\mathrm{N}, \mathrm{LRW}$ ORK $>=8 * \mathrm{M}+2 * \mathrm{M} *$ SM LSZ + $8 * M * N L V L+M * N R H S . S M L S Z$ is retumed by HAENV and is equal to the $m$ axim um size of the subproblem sat the bottom of the com putation tree (usually about 25), and NLVL= $\mathbb{N} T\left(L O G \_2(M \mathbb{N}(M, N\right.$
)/(SM LSTZ+1)) ) + 1
IV ORK (w orkspace)
LIV ORK >= $3 * M \mathbb{N} M N * N L V L+11 * M \mathbb{N} M N$, where $M \mathbb{N} M N=M \mathbb{N}(M, N)$.
$\mathbb{I N F O}$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N} F O=-$ i, the $i$-th argum enthad an illegalvalue.
>0: the algorithm forcom puting the SVD failed
to converge; if $\mathbb{N} F O=$ i, ioff-diagonalelem ents of an interm ediate bidiagonal form did not converge to zero.

## FURTHER DETAILS

B ased on contributions by
M ing Gu and Ren-Cang Li, Computer Science D ivision,
U niversity of Califomia atBerkeley, U SA

O sniM arques, LBNLN ER SC , U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgelss-com pute the m inim um norm solution to a complex linear least squares problem

## SYNOPSIS

```
SUBROUT\mathbb{NE CGELSSM,N,NRHS,A,LDA,B,LDB,S\mathbb{NG,RCOND, RRANK,}}\mathbf{N},\mp@code{N},
    W ORK,LDW ORK,W ORK2,INFO)
COM PLEX A (LDA,*),B (LD B,*),W ORK (*)
\mathbb{NTEGERM,N,NRHS,LDA,LDB, \mathbb{RANK,LDW ORK, INFO}}\mathbf{N},\textrm{L}
REAL RCOND
REALSING (*),W ORK2 (*)
```



```
    W ORK,LDW ORK,W ORK2, INFO)
```

COM PLEX A (LDA,*), B (LDB,*),W ORK (*)
$\mathbb{N} T E G E R * 8 M, N, N R H S, L D A, L D B, \mathbb{R A N K}, L D W O R K, \mathbb{N} F O$
REALRCOND
REALSING (*), W ORK 2 (*)

## F95 INTERFACE

SU BROUT $\mathbb{N} E \operatorname{GELSS}(\mathbb{M}], \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], S \mathbb{N} G, R C O N D$, RANK, [W ORK], [LDW ORK ], [W ORK 2], [ $\mathbb{N F O}$ ])

COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A,B
$\mathbb{N} T E G E R:: M, N, N R H S, L D A, L D B, \mathbb{R A N K}, L D W O R K, \mathbb{N} F O$
REAL ::RCOND
REAL,D $\mathbb{I M} E N S I O N(:):: S \mathbb{N} G, W$ ORK 2
SU BROUTINE GELSS_64 (M) $\mathbb{M}], \mathbb{N} R H S], A,[L D A], B,[L D B], S \mathbb{N} G$,

COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A, B
$\mathbb{N} T E G E R(8):: M, N, N R H S, L D A, L D B, \mathbb{R A N K}, L D W O R K, \mathbb{N} F O$
REAL ::RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::SNG,W ORK 2

## C INTERFACE

\#include <sunperfh>
void cogelss (intm, intn, intnrhs, complex *a, int lda, com plex *b, int ldb, float *sing, float roond, int *irank, int*info);
void cgelss_64 (long m, long n, long nrhs, com plex *a, long
lda, com plex *b, long ldb, float*sing, float rcond, long *irank, long *info);

## PURPOSE

cgelss com putes the $m$ inim um norm solution to a com plex linear least squares problem :
$M$ inim ize 2 -norm ( $|\mathrm{b}-\mathrm{A} * \mathrm{x}|$.
using the singularvalue decom position (SVD) ofA.A is an M -by-N $m$ atrix which $m$ ay be rank-deficient.

Several righthand side vectors b and solution vectors x can be handled in a single call; they are stored as the colum ns of the M -by-NRH S righthand sidem atrix $B$ and the $N$ by-NRH S solution $m$ atrix $X$.

The effective rank ofA is determ ined by treating as zero those singular values which are less than RCOND tim es the largest singularvalue.

## ARGUMENTS

M (input) The num ber of row sof the $m$ atrix $\mathrm{A} . \mathrm{M}>=0$.
$N$ (input) The num ber of collm ns of the m atrix $A . N>=0$.

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of the $m$ atrioes $B$ and $X$.NRH $S>=0$.

A (input/output)
On entry, the M -by -N m atrix A. On exit, the first $m$ in $(m, n)$ row $s$ of A are overw ritten $w$ ith its right singularvectors, stored row w ise.

LD A (input)
The leading dim ension of the array A. LDA >= $m a x(1, M)$.

B (input/output)
On entry, the M -by NRHS righthand side m atrix B .
On exit, B is overw ritten by the $N$ by $-\mathrm{NRH} S$ solution $m$ atrix $X$. Ifm $>=n$ and $\mathbb{R A N K}=n$, the residual sum -of-squares for the solution in the i-th colum $n$ is given by the sum of squares of ele$m$ ents $n+1 m$ in that $c o l u m n$.

LD B (input)
The leading dim ension of the anay $B$. LD B >= $\max (1, M, N)$.

## SNN (output)

The singularvalues ofA in decreasing order. The condition number of $A$ in the 2 -norm $=$ $S \mathbb{N} G(1) / S \mathbb{N} G(m$ in $(m, n))$.

RCOND (input)
RCOND is used to determ ine the effective rank of
A. Singular values SING (i) <= RCOND *SING (1) are treated as zero. IfRCOND $<0, \mathrm{~m}$ achine precision is used instead.

RANK (output)
The effective rank of A, i.e., the num ber of singular values which are greater than RCOND *SING (1).

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dim ension of the amray $W$ ORK. LDW ORK $>=1$, and also: LDW ORK $>=2 \star_{m}$ in $\left.(M, N)+m a x M, N, N R H S\right)$ For good perform ance, LDW ORK should generally be larger.

If LDW ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK anay, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 2 (w orkspace)
dim ension $\left(5{ }^{*} \mathrm{~m}\right.$ in $\left.\mathrm{M}, \mathrm{N}\right)$ )
$\mathbb{N F O}$ (output)
$=0$ : successfulexit
< 0: if $\mathbb{N N}$ FO = -i, the i-th argum ent had an illegalvalue.
> 0: the algorithm forcom puting the SVD failed to converge; if $\mathbb{N} F O=$ i, ioff-diagonalelem ents of an interm ediate bidiagonal form did not converge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgelsx -routine is deprecated and has been replaced by routine CGELSY

## SYNOPSIS

```
SU BROUT\mathbb{NE CGELSX M,N,NRHS,A,LDA,B,LDB,JPIVOT,RCOND,\mathbb{RANK,}}\mathbf{N},\textrm{N},\textrm{N}
    W ORK,W ORK2,INFO)
COM PLEX A (LDA,*),B (LDB,*),W ORK (*)
\mathbb{NTEGERM,N,NRHS,LDA,LDB,\mathbb{RANK,INFO}}\mathbf{N},\mp@code{L}
INTEGER JPIVOT (*)
REAL RCOND
REALW ORK2 (*)
SUBROUT\mathbb{NE CGELSX_64M,N,NRHS,A,LDA,B,LDB,JPIVOT,RCOND,}
```



```
COM PLEX A (LDA,*),B (LD B,*),W ORK (*)
NNTEGER*8M,N,NRHS,LDA,LDB,\mathbb{RANK,INFO}
INTEGER*8 \mathbb{PIVOT (*)}
REAL RCOND
REALW ORK2 (*)
```


## F95 INTERFACE

SU BROUTINE GELSX (M) $\mathbb{M}], \mathbb{N} R H S], A,[L D A], B,[L D B], J P I V O T, R C O N D$, $\mathbb{R A N K},[\mathbb{W}$ ORK], [W ORK2], [ $\mathbb{N} F O])$

COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A,B
$\mathbb{N}$ TEGER ::M,N,NRHS,LDA,LDB, $\mathbb{R} A N K, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:)::$ JPIVOT
REAL ::RCOND

SU BROUT $\mathbb{N} E$ GELSX_64 ( $\mathbb{M}], \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], \mathbb{P} \mathbb{I} O T$, $R C O N D, \mathbb{R} A N K,[W O R K],[W O R K 2],[\mathbb{N} F O])$

COM PLEX,D $\mathbb{M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{I}$ ENSION (: : : : : A , B
$\mathbb{N}$ TEGER (8) :: M , N , NRHS,LDA $, \operatorname{LDB}, \mathbb{R} A N K, \mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T$
REAL: RCOND
REAL,D $\mathbb{I M}$ ENSION (:) ::W ORK2

## C INTERFACE

\#include < sunperfh>
void cgelsx (intm, intn, intnrhs, complex *a, int lda, com plex *b, int ldb, int * jivivot, floatrcond, int *irank, int *info);
void cgelsx_64 (long m, long n, long nrhs, com plex *a, long lda, com plex *b, long ldb, long * jpivot, float rcond, long *irank, long *info);

## PURPOSE

cgelsx routine is deprecated and has been replaced by routine CGELSY .

CGELSX com putes the $m$ inim um-norm solution to a complex linear least squares problem :
$m$ inim ize $\| A$ * $X-B \|$
using a com plete orthogonal factorization of A. A is an $M-$ by-N m atrix w hich m ay be rank-deficient.

Several righthand side vectors b and solution vectors x can be handled in a single call; they are stored as the colum ns of the M -by - N R H S righthand side m atrix B and the N boy-NRHS solution m atrix X .

The routine firstcom putes a QR factorization with colum $n$ pivoting:
$A * P=Q *[R 11 R 12]$
[ 0 R22]
w ith R 11 defined as the largest leading subm atrix w hose estim ated condition num ber is less than $1 \notin C O N D$. The order of $\mathrm{R} 11, \mathrm{RANK}$, is the effective rank ofA.

Then, R 22 is considered to be negligible, and R 12 is annihilated by unitary transform ations from the right, arriving at the com plete orthogonal factorization:
$A * P=Q *[T 110] * Z$
[ 0 0]
Them inim um norm solution is then
$\mathrm{X}=\mathrm{P} * \mathrm{Z}$ ' [inv (T11)*Q 1 *B ]
[ 0 ]
where Q 1 consists of the firstRANK colum ns of Q .

## ARGUMENTS

M (input) The num ber of row s of the m atrix $\mathrm{A} . \mathrm{M}>=0$.
N (input) The num ber of colum ns of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, ie., the num ber of collm ns ofm atrices B and X.NRHS $>=0$.

A (input/output)
On entry, the $M$-by -N matrix A. On exit, A has been overw rilten by details of its complete orthogonal factorization.

LD A (input)
The leading dim ension of the array A. LDA >= max (1,M).

B (input/output)
On entry, the M-by-NRHS righthand sidem atrix B . On exit, the N -by-N RH S solution $m$ atrix X . Ifm >= n and $\mathbb{R} A N K=n$, the residual sum-of-squares for the solution in the $i$-th colum $n$ is given by the sum of squares of elem ents $N+1 \mathrm{M}$ in thatcolum $n$.

LD B (input)
The leading dim ension of the array $\mathrm{B} . \operatorname{LD} \mathrm{B}>=$ $\max (1, M, N)$.

JPIVOT (input/output)
On entry, if $\mathbb{P} \mathbb{I V O T}$ (i) ne.0, the i-th column of $A$ is an initial colum $n$, otherw ise it is a free colum $n$. Before the $Q R$ factorization of $A$, all initial colum ns are perm uted to the leading positions; only the rem aining free colum ns are m oved as a resultof colum $n$ pivoting during the factorization. On exit, if $\mathbb{P} \operatorname{IVOT}(i)=k$, then the $i$-th colum $n$ of A *P was the $k$-th collm $n$ of A.

RCOND (input)

RCOND is used to determ ine the effective rank of A, which is defined as the order of the largest leading triangular subm atrix R 11 in the $Q$ R factorization w ith pivoting ofA , whose estim ated condition num ber $<1$ RCOND.

RANK (output)
The effective rank of A, i.e., the order of the subm atrix R11. This is the sam e as the orderof the subm atrix T11 in the com plete orthogonal factorization of A.
W ORK (w orkspace)
$(m$ in $M, N)+\max \left(N, 2 \star_{m}\right.$ in $\left.\left.(M, N)+N R H S\right)\right)$,

W ORK 2 (w orkspace)
dim ension ( $2 * \mathrm{~N}$ )
$\mathbb{I N F O}$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the i-th argum enthad an ille-
galvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgelsy - com pute the $m$ inim um -norm solution to a complex linear least squares problem

## SYNOPSIS

```
SUBROUT\mathbb{NE CGELSY M,N,NRHS,A,LDA,B,LDB,JPVT,RCOND,RANK,}
    W ORK,LW ORK,RW ORK,INFO)
COM PLEX A (LDA,*),B (LDB,*),W ORK (*)
\mathbb{NTEGERM,N,NRHS,LDA,LDB,RANK,LW ORK,INFO}
INTEGER JPVT (*)
REALRCOND
REAL RW ORK (*)
SUBROUT\mathbb{NE CGELSY_64M,N,NRHS,A,LDA,B,LDB,UPVT,RCOND,RANK,}
    W ORK,LW ORK,RW ORK,INFO)
COM PLEX A (LDA,*),B (LD B,*),W ORK (*)
INTEGER*8M,N,NRHS,LDA,LDB,RANK,LW ORK,\mathbb{NFO}
INTEGER*8 \PVT (*)
REAL RCOND
REAL RW ORK (*)
```


## F95 INTERFACE

```
SU BROUTINE GELSY (M ], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], J P V T, R C O N D\), RANK, [W ORK], [LW ORK], [RW ORK], [ \(\mathbb{N F O}]\) )
COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : ::A, B
\(\mathbb{N} T E G E R:: M, N, N R H S, L D A, L D B, R A N K, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{J V} T\)
```

REAL ::RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::RW ORK
SUBROUTINE GELSY_64 (M), $\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], \mathbb{P V} T$, RCOND,RANK, [W ORK], [LW ORK], [RW ORK], [ $\mathbb{N F O}]$ )

COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\operatorname{IM}$ ENSION (:,:) ::A,B
$\mathbb{N}$ TEGER (8) :: M , N,NRHS,LDA,LDB,RANK,LW ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{J V T}$
REAL ::RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::RW ORK

## C INTERFACE

\#include <sunperfh>
void cgelsy (intm, intn, intnrhs, com plex *a, int lda, complex *b, int ldb, int *jpvt, float roond, int *rank, int*info);
void cgelsy_64 (long m, long n, long nrhs, com plex *a, long lda, com plex *b, long ldb, long * jpvt, float rcond, long *rank, long *info);

## PURPOSE

cgelsy com putes the m inim um-norm solution to a com plex linear least squares problem :
$m$ inim ize $\|A * X-B\|$
using a com plete orthogonal factorization of . A is an $M$ by -N m atrix w hich m ay be rank-deficient.

Several righthand side vectors b and solution vectors x can be handled in a single call; they are stored as the colum ns of the M by $-N$ RH S righthand side $m$ atrix $B$ and the $N$ by-NRHS solution $m$ atrix $X$.

The routine firstcom putes $a Q R$ factorization with $c o l u m n$ pivoting:

A * $\mathrm{P}=\mathrm{Q}$ * [R11R12]
[ 0 R22]
w ith R 11 defined as the largest leading subm atrix whose estim ated condition num ber is less than $1 \not R C O N D$. The order ofR11,RANK, is the effective rank ofA.

Then, R22 is considered to be negligible, and R 12 is annihilated by unitary transform ations from the right, amiving at the com plete orthogonal factorization:
$A * P=Q *[T 110] * Z$
[ 0 0]

Them inim um norm solution is then
$\mathrm{X}=\mathrm{P}$ * Z ' [ inv (T11)*Q 1 *B ]
[ 0 ]
where Q 1 consists of the firstRANK colum ns of $Q$.
This routine is basically identical to the original xG ELSX except three differences:

- The perm utation ofm atrix $B$ (the right hand side) is faster and
m ore sim ple.
o The call to the subroutine $x G E Q P F$ has been substituted by the
the call to the subroutine $x G E Q P 3$. This subroutine is a B las-3
version of the $Q R$ factorization $w$ th colum $n$ pivoting. $O M$ atrix $B$ (the righthand side) is updated $w$ ith $B$ las-3.


## ARGUMENTS

M (input) The num ber of row s of the m atrix $\mathrm{A} . \mathrm{M}>=0$.
N (input) The num ber of collm ns of the $m$ atrix $\mathrm{A} . \mathrm{N}>=0$.

NRH S (input)
The num ber of righthand sides, ie., the num ber of colum ns ofm atrices $B$ and $X . N R H S>=0$.

A (input/output)
On entry, the M by -N matrix A. On exit, A has been overw ritten by details of its com plete orthogonal factorization.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, \mathrm{M})$.

B (input/output)
On entry, the M -by - N RH S righthand side m atrix B . On exit, the $N$-by N RH S solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the aray $B . L D B>=$ $\max (1, M, N)$.

JPVT (input/output)
On entry, if JPV T (i) ne. 0, the i-th collm n of A is perm uted to the frontofA $P$, otherw ise colum n i is a free colum $n$. On exit, if JPV $T(i)=k$, then
the i-th column of A *P w as the $k$-th colum $n$ of A.

## RCOND (input)

RCOND is used to determ ine the effective rank of A, which is defined as the order of the largest leading triangular subm atrix R 11 in the $Q R$ factorization $w$ th pivoting ofA , whose estim ated condition num ber $<1$ RCOND.

## RANK (output)

The effective rank ofA, ie., the order of the subm atrix R11. This is the sam e as the orderof the subm atrix T11 in the complete orthogonal factorization of A.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array W ORK. The unblocked strategy requires that: LW ORK >=MN+MAX (2*M N, $\mathrm{N}+1, \mathrm{M} \mathrm{N}+\mathrm{NRH} \mathrm{S}$ ) where $\mathrm{M} \mathrm{N}=\mathrm{m}$ in $\mathrm{M}, \mathrm{N})$. The block algorithm requires that: LW ORK $>=\mathrm{MN}+\mathrm{MAX}(2 * \mathrm{M} N$,
 upper bound on the blocksize retumed by IUA EN V for the routines CGEQP3, CTZRZF, CTZRQF, CUNM QR, and CUNMRZ.

If LW ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
theW ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

RW ORK (w orkspace)
dim ension $(2 * N)$
$\mathbb{N} F O$ (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N} F O=-$ i, the $i$-th argum ent had an illegalvalue

## FURTHER DETAILS

```
B ased on contributions by
    A .Petitet, C om puterScience D ept, U niv . ofTenn., K nox-
ville, U SA
    E.Q uintana-O rti, D epto. de Inform atica,U niversidad Jaim e
I, Spain
```

G. Q uintana-O rti, D epto. de Inform atica, U niversidad Jaim e I, Spain

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgemm -perform one of the $m$ atrix-m atrix operations $C:=$ alpha*op (A ) *op (B ) + beta*C

## SYNOPSIS

```
SUBROUTINE CGEMM (TRANSA,TRANSB,M,N,K,ALPHA,A,LDA,B,LDB,
    BETA,C,LDC)
CHARACTER * 1 TRANSA,TRANSB
COM PLEX ALPHA,BETA
COM PLEX A (LDA,*),B (LD B,*),C (LD C ,*)
INTEGERM,N,K,LDA,LDB,LDC
SUBROUTINE CGEMM _64 (TRANSA,TRANSB,M ,N,K,ALPHA,A,LDA,B,LDB,
    BETA,C,LDC)
```

CHARACTER * 1 TRANSA, TRANSB
COM PLEX ALPHA,BETA
COM PLEX A (LDA,*), B (LD B , *), C (LD C , *)
$\mathbb{N}$ TEGER*8 $\mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LD} A, L D B, L D C$

## F95 INTERFACE

SU BROUTINE GEM M ([TRANSA], [TRANSB], $\mathbb{M}], \mathbb{N}], \mathbb{K}], A L P H A, A,[L D A]$, B, [LD B ], BETA , C , [LD C ])

CHARACTER (LEN=1) ::TRANSA,TRANSB
COMPLEX ::ALPHA,BETA
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A,B,C
$\mathbb{N} T E G E R:: M, N, K, L D A, L D B, L D C$
SU BROUTINE GEMM_64 ([TRANSA], [TRANSB], M ], $\mathbb{N}],[K], A L P H A, A,[L D A]$, B, [LDB],BETA, C, [LDC])

CHARACTER ( $L E N=1$ ) : : TRAN SA, TRAN SB
COM PLEX ::ALPHA,BETA
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : A , B , C
$\mathbb{N}$ TEGER (8) :: $\mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LDA}$, LD B , LD C

## C INTERFACE

\#include <sunperfh>
void ogem m (chartransa, chartransb, intm , int $n$, int $k$, com plex *alpha, com plex *a, int lda, com plex *b, int ldlo, com plex *beta, com plex *c, int ldc);
void cgem m _64 (chartransa, chartransb, long m, long n, long k, com plex *alpha, com plex *a, long lda, com plex *b, long ldb, com plex *beta, com plex *c, long ldc);

## PURPOSE

cgem $m$ perform s one of the $m$ atrix-m atrix operations

C : = alpha*op (A )*op (B) + beta*C
where op (X ) is one of
op $(X)=X$ or $o p(X)=X^{\prime}$ or op $(X)=$ con $\dot{g}(X)$, alpha and beta are scalars, and $A, B$ and $C$ arem atrices, $w$ ith op (A) an $m$ by $k m$ atrix, op (B) a $k$ by $n m$ atrix and $C$ an $m$ by $n m$ atrix.

## ARGUMENTS

TRANSA (input)
O n entry, TRAN SA specifies the form of op (A ) to be used in the $m$ atrix $m$ ultiplication as follow $s$ :

TRANSA $=N$ 'or $h$ ', op $(A)=A$.

TRANSA = T'or $t^{\prime}, o p(A)=A$ '.

TRANSA $=C^{\prime}$ or $E^{\prime}, o p(A)=\operatorname{conjg}\left(A^{\prime}\right)$.

U nchanged on exit.

TRANSA is defaulted to $N$ 'forF $95 \mathbb{I N}$ TERFACE.

TRANSB (input)
On entry, TRAN SB specifies the form of op (B) to be used in the $m$ atrix $m$ ultiplication as follow $s:$

TRANSB $=N^{\prime}$ 'or $h^{\prime}, ~ o p(B)=B$.

TRANSB = T'or $t^{\prime}$, op (B) $)=B^{\prime}$.

TRANSB = C'or $C^{\prime}, ~ o p(B)=$ con $\dot{g}\left(B^{\prime}\right)$.

U nchanged on exit.

TRAN SB is defaulted to $N$ 'forF95 $\mathbb{I N T E R F A C E .}$

M (input)
O n entry, $M$ specifies the num ber of rows of the $m$ atrix op (A ) and of the $m$ atrix C. $M>=$ 0 . U nchanged on exit.

N (input)
O n entry, $N$ specifies the num ber of colum ns of the $m$ atrix op ( B ) and the num ber of colum ns of the $m$ atrix $C . N>=0$. U nchanged on exit.

K (input)
On entry, $K$ specifies the num ber of colum ns of the $m$ atrix op (A) and the num ber of row s of the $m$ atrix op (B).K $>=0$. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
COM PLEX array ofD $\mathbb{M}$ ENSION (LDA, ka), where ka is K when TRANSA $=\mathrm{N}$ 'or h ', and is M otherw ise. Before entry w ith TRANSA $=\mathrm{N}$ 'or h ', the leading $M$ by $K$ part of the array A m ust contain the $m$ atrix A , otherw ise the leading $K$ by $M$ part of the array A m ust contain the m atrix A. U nchanged on exit.

## LD A (input)

O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .W hen TRANSA $=N$ 'or $h$ 'then LDA $>=\max (1, M)$, otherw ise LD $A>=\mathrm{m}$ ax $(1, K)$. U nchanged on exit.

B (input)
COM PLEX aray ofD $\mathbb{I M} E N S I O N$ (LD $\mathrm{B}, \mathrm{kb}$ ), where kb is n when TRANSB $=\mathrm{N}$ 'or h ', and is k oth-
erw ise. Before entry $w$ ith TRANSB $=N^{\prime}$ or $h$ ', the leading $k$ by $n$ part of the anay $B$ must contain the $m$ atrix $B$, otherw ise the leading $n$ by $k$ part of the array $B$ mustcontain the $m$ atrix $B$. U nchanged on exit.

LDB (input)
O n entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program .W hen TRANSB $=N$ 'or h'then LD B $>=m a x(1, k)$, otherw ise LD B $>=\max (1, \mathrm{n})$. U nchanged on exit.
BETA (input)
O n entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then C need not.be set on input. U nchanged on exit.

C (input/output)
COM PLEX aray ofD $\mathbb{I M}$ ENSION (LDC, n ). Before entry, the leading $m$ by $n$ part of the array $C$ $m$ ustcontain the $m$ atrix $C$, exceptw hen beta is zero, in which case $C$ need notbe seton entry. On exit, the array $C$ is overw ritten by the $m$ by n matrix (alpha*op (A )*op (B ) + beta*C ).

LD C (input)
On entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program. LD C $>=\max (1, m)$. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgem $v$-perform one of the $m$ atrix-vectoroperations $y:=$ alpha*A *x + beta*y, ory : alpha*A *x + beta*y, or $y:=$ alpha*con $\dot{g}\left(A^{\prime}\right){ }^{\prime} x+$ beta*y

## SYNOPSIS

```
SUBROUT\mathbb{NE CGEMV (TRANSA,M,N,ALPHA,A,LDA,X, INCX,BETA,Y,INCY)}
CHARACTER * 1 TRANSA
COM PLEX A LPHA,BETA
COM PLEX A (LDA,*),X (*),Y (*)
```



```
SUBROUT\mathbb{NE CGEM V_64(TRANSA,M ,N,ALPHA,A ,LDA,X,INCX,BETA,Y,}
    \mathbb{NCY})
CHARACTER * 1 TRANSA
COM PLEX ALPHA,BETA
COM PLEX A (LDA,*),X (*),Y (*)
\mathbb{NTEGER*8 M ,N,LDA, INCX, INCY}
```


## F95 INTERFACE

SU BROUTINE GEMV ([TRANSA ], $\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A], X,[\mathbb{N C X}], B E T A$, Y, [ $\mathbb{N} C Y])$

CHARACTER (LEN=1) ::TRANSA
COMPLEX ::ALPHA,BETA
COM PLEX,D IM ENSION (:) :: X,Y
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : : A
$\mathbb{N} T E G E R:: M, N, L D A, \mathbb{N} C X, \mathbb{N} C Y$
SUBROUTINE GEM V_64 ([TRANSA], $\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A], X,[\mathbb{N} C X]$,

BETA, Y, [ $\mathbb{N C Y}])$

CHARACTER (LEN=1) ::TRANSA
COM PLEX ::ALPHA,BETA
COM PLEX , D $\mathbb{M}$ ENSION (:) :: X , Y
COMPLEX , D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{LD} A, \mathbb{N C X}, \mathbb{N C Y}$

## C INTERFACE

\#include <sunperfh>
void cgem v (chartransa, intm, intn, com plex *alpha, com plex *a, int lda, com plex *x, int incx, com plex
*beta, com plex *y, int incy);
void ogem v_64 (chartransa, long m, long n, com plex *alpha, com plex *a, long lda, com plex *x, long incx, com plex *beta, com plex *y, long incy);

## PURPOSE

cgem v perform s one of the $m$ atrix-vector operations $y:=$ alpha*A *x + beta*y, ory := alpha*A *x + beta*y, or y := alpha*con $\dot{g}\left(A^{\prime}\right)^{\star} x+b e t a * y ~ w h e r e ~ a l p h a ~ a n d ~ b e t a ~ a r e ~$ scalars, $x$ and $y$ are vectors and $A$ is an $m$ by $n m$ atrix.

## ARGUMENTS

TRANSA (input)
O n entry, TRANSA specifies the operation to be perform ed as follow s:


TRANSA = T'ort' $y=$ alpha*A *x + beta*y.
 beta*y.

U nchanged on exit.

TRANSA is defaulted to $N$ 'forF $95 \mathbb{I N}$ TERFACE.

M (input)
O n entry, $M$ specifies the num ber of row s of the $m$ atrix $A . M>=0$. U nchanged on exit.

O n entry, $N$ specifies the num ber of colum ns of the $m$ atrix $A . N>=0$. U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
Before entry, the leading $m$ by $n$ part of the array A must contain the $m$ atrix of coefficients. U nchanged on exit.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program . LD A >= $\max (1, m)$. U nchanged on exit.
$X$ (input)
$(1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C X))$ when TRANSA $=\mathrm{N}$ 'or
$h^{\prime}$ and at least ( $\left.1+(m-1) * a b s(\mathbb{N} C X)\right)$
otherw ise. Before entry, the increm ented array $X$ $m$ ust contain the vectorx. U nchanged on exit.
$\mathbb{N} C X$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of $\mathrm{X} . \mathbb{N} C X$ m ustnotbe zero. Unchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then $Y$ need notbe seton input. U nchanged on exit.

Y (input/output)
$(1+(m-1) \star \operatorname{abs}(\mathbb{N} C Y))$ when TRANSA $=\mathrm{N}$ 'or
$h^{\prime}$ and at least ( $1+(\mathrm{n}-1)$ *abs( $\left.\mathbb{N} C Y\right)$ )
otherw ise. Before entry $w$ ith BETA non-zero, the increm ented array $Y$ mustcontain the vectory. On exit, $Y$ is overw ritten by the updated vectory.
$\mathbb{N} C Y$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of Y. $\mathbb{N} C Y$ m ustnotbe zero. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgeqlf-com pute $a \operatorname{L}$ factorization of a com plex M -by N m atrix A

## SYNOPSIS


COM PLEX A (LDA,*),TAU (*),W ORK (*)
$\mathbb{N}$ TEGER M,N,LDA,LDW ORK, $\mathbb{N} F O$
SUBROUTINECGEQLF_64 $M, N, A, L D A, T A U, W O R K, L D W O R K, \mathbb{N} F O)$
COM PLEX A (LDA, *),TAU (*),W ORK (*)
$\mathbb{N}$ TEGER*8M,N,LDA,LDW ORK, $\mathbb{N} F O$

## F95 INTERFACE

SU BROUTINE GEQLF (M ], $\mathbb{N}], A,[L D A], T A U, \mathbb{W} O R K],[L D W O R K],[\mathbb{N F O}])$
COM PLEX,D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
COMPLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R:: M, N, L D A, L D W$ ORK, $\mathbb{N} F O$
SU BROUTINE GEQLF_64 ( $\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N} F O])$
COM PLEX,D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
COM PLEX,D $\mathbb{I}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER (8) ::M , N, LDA, LDW ORK, $\mathbb{N}$ FO

## C INTERFACE

\#include <sunperfh>
void cgeqlf(intm , intn, com plex *a, int lda, com plex *tau, int*info);
void cgeqlf_ 64 (long m , long n, com plex *a, long lda, com plex
*tau, long *info);

## PURPOSE

cgeqlf com putes a QL factorization of a complex M -by -N $m$ atrix $A: A=Q * L$.

## ARGUMENTS

M (input) The num ber of row s of the $m$ atrix $\mathrm{A} . \mathrm{M}>=0$.
N (input) The num ber of $c o l u m$ ns of the $m$ atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
On entry, the $\mathrm{M}-\mathrm{by}-\mathrm{N} \mathrm{m}$ atrix A . On exit, if $\mathrm{m}>=$ $n$, the low er triangle of the subarray A ( $m$ $\mathrm{n}+1 \mathrm{~m}, 1 \mathrm{n}$ ) contains the N -by-N low er triangular $m$ atrix $L$; ifm <= $n$, the elem ents on and below the ( $n-m$ )-th superdiagonalcontain the M by -N low er trapezoidalm atrix $L$; the rem aining elem ents, w ith the anray TA $U$, represent the unitary $m$ atrix $Q$ as a product of elem entary reflectors (see Further D etails).

LD A (input)
The leading dim ension of the array A . LD A $>=$ max (1, M).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, \mathrm{~W}$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dim ension of the array W ORK. LDW ORK >= $m$ ax $(1, N)$. Foroptim um perform ance LD W ORK $>=N * N B$, w here NB is the optim alblocksize.

IfLD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
$\mathbb{N F O}$ (output)
= 0 : successfiulexit
<0: if $\mathbb{I N}$ FO $=-$ i, the $i$-th argum enthad an illegalvalue

## FURTHER DETAILS

Them atrix $Q$ is represented as a product of elem entary reflectors

$$
Q=H(k) \ldots H(2) H(1), \text { where } k=m \text { in }(m, n) .
$$

Each $H$ (i) has the form

H (i) $=I-\tan * V^{*} V^{\prime}$
$w$ here tau is a com plex scalar, and $v$ is a com plex vector w ith $v(m-k+i+1 \mathrm{~m})=0$ and $v(m-k+i)=1 ; v(1 m-k+i-1)$ is stored on exitin A ( $1 \mathrm{~m}-\mathrm{k}+\mathrm{i}-1, \mathrm{n}-\mathrm{k}+\mathrm{i}$ ), and tau in TAU (i).

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgeqp3 - com pute a Q R factorization with colum n pivoting of a matrix A

## SYNOPSIS



```
COM PLEX A (LDA,*),TAU (*),W ORK (*)
\mathbb{NTEGERM,N,LDA,LW ORK,\mathbb{NFO}}\mathbf{N}=\mp@code{L}
INTEGER JPVT (*)
REAL RW ORK (*)
SUBROUT\mathbb{NE CGEQP3_64M,N,A,LDA,JPVT,TAU,W ORK,LW ORK,RW ORK,}
        \mathbb{NFO )}
COM PLEX A (LDA,*),TAU (*),W ORK (*)
\mathbb{NTEGER*8M,N,LDA,LW ORK,NNFO}
INTEGER*8 JPVT (*)
REAL RW ORK (*)
```


## F95 INTERFACE

SU BROUTINE GEQP3 (M ], $\mathbb{N}], A,[L D A], J P V T, T A U,[W O R K],[L W ~ O R K]$, [RW ORK], [ $\mathbb{N} F O]$ )

COM PLEX,D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER ::M,N,LDA,LW ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I}$ ENSION (:) :: JPVT
REAL,D $\mathbb{M}$ ENSION (:) ::RW ORK

SU BROUTINE GEQP3_64 (M ], $\mathbb{N}], A,[L D A], J P V T, T A U,[W O R K],[L W$ ORK ],
[RW ORK], [ $\mathbb{N} F O]$ )
COM PLEX,D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : A
$\mathbb{N} T E G E R(8):: M, N, L D A, L W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I M}$ ENSION (:) :: JPVT
REAL,D $\mathbb{M}$ ENSION (:) ::RW ORK

## C INTERFACE

\#include < sunperfh>
void cgeqp3 (intm, intn, com plex *a, int lda, int * jpvt, com plex *tau, int *info);
void ogeqp3_64 (long m, long n, com plex *a, long lda, long * jpvt, com plex *tau, long *info);

## PURPOSE

cgeqp3 com putes a $Q R$ factorization $w$ ith colum $n$ pivoting of a $m$ atrix $A: A * P=Q * R$ using Level3 BLA $S$.

## ARGUMENTS

M (input) The num ber of row s of the $m$ atrix $\mathrm{A} . \mathrm{M}>=0$.
N (input) The num ber of colum ns of the m atrix $\mathrm{A} . \mathrm{N}>=0$.
A (input/output)
On entry, the $M$-by-N m atrix A. On exit, the upper triangle of the array contains the $m$ in $(\mathbb{N}, N)$-by $-N$ upper trapezoidalm atrix $R$; the elem ents below the diagonal, together $w$ th the array TA $U$, represent the unitary $m$ atrix $Q$ as a productofm in $M, N$ ) ele$m$ entary reflectors.

LDA (input)
The leading dim ension of the array A. LDA >= $m a x(1, M)$.

JPVT (input/output)
On entry, if $\mathbb{P V} T(J)$ ne. 0 , the $J$ th colum $n$ of $A$ is perm uted to the frontof $A$ * (a leading colum $n$ ); if JPVT $(J)=0$, the $J$ th column of $A$ is a free column. On exit, if $\mathbb{J P V}(\mathcal{J})=\mathrm{K}$, then the J th colum $n$ of A *P was the the $K$-th colum $n$ of A.

TAU (output)
The scalar factors of the elem entary reflectors.
W ORK (w orkspace)
Onexit, if $\mathbb{N} F O=0, \mathrm{~W}$ ORK (1) retums the optim al
LW ORK.

LW ORK (input)
The dim ension of the aray W ORK. LW ORK >= N+1. For optim alperform ance LW ORK >=( $\mathrm{N}+1$ ) N B , where N B is the optim alblocksize.

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

RW ORK (w orkspace)
dim ension ( $2 \star \mathrm{~N}$ )
$\mathbb{N} F O$ (output)
= 0 : successfiulexit.
$<0:$ if $\mathbb{N F O}=-$ i, the $i$-th argum enthad an illegalvalue.

## FURTHER DETAILS

Them atrix $Q$ is represented as a product of elem entary reflectors

$$
Q=H(1) H(2) \ldots H(k), \text { where } k=m \text { in }(m, n) .
$$

Each $H$ (i) has the form

H (i) $=I-\tan * V^{*} V^{\prime}$
w here tau is a real/com plex scalar, and $v$ is a real/com plex vectorw ith $v(1: i-1)=0$ and $v(i)=1 ; v(i+1 \mathrm{~m})$ is stored on exit in A (i+1 $m, i)$, and tau in TAU (i).

B ased on contributions by
G . Q uintana-O rti, D epto. de Inform atica, U niversidad Jaim e I, Spain
X . Sun, C om puter Science D ept., D uke U niversity, U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgeqpf-routine is deprecated and has been replaced by routine C GEQP3

## SYNOPSIS

```
SUBROUT\mathbb{NE CGEQPF M,N,A,LDA,JPIVOT,TAU,W ORK,W ORK2,\mathbb{NFO)}}\mathbf{N}\mathrm{ (N,N}
COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGERM,N,LDA,INFO
INTEGER JPIVOT (*)
REAL W ORK 2 (*)
SUBROUT\mathbb{NECGEQPF_64M,N,A,LDA,UPIVOT,TAU,WORK,WORK2,INFO)}
COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGER*8M,N,LDA,INFO
INTEGER*8 JPIVOT (*)
REALW ORK2 (*)
```


## F95 INTERFACE

SU BROUTINE GEQPF (M ], $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I} O T, T A U,[\mathbb{W}$ ORK ], [W ORK 2], [ $\mathbb{N}$ FO ])

COM PLEX,D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : ::A
$\mathbb{N}$ TEGER ::M,N,LDA, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: JPIVOT
REAL,D $\mathbb{I M} E N S I O N(:):: W$ ORK2
SU BROUTINE GEQPF_64 (M ], $\mathbb{N}], A,[L D A], J P I V O T, T A U,[\mathbb{W} O R K],[\mathbb{W}$ ORK 2], [ $\mathbb{N}$ FO ])

## C INTERFACE

\#include <sunperfh>
void cgeqpf(intm, intn, com plex *a, int lda, int * jivivot, com plex *tau, int *info);
void cgeqpf_64 (long m, long n, com plex *a, long lda, long

* jpivot, com plex *tau, long *info);


## PURPOSE

cgeqpf routine is deprecated and has been replaced by routine CGEQP3.

CGEQPF com putes $a \mathrm{Q}$ factorization w th colum n pivoting of a com plex $M$-by $-N$ m atrix $A: A * P=Q * R$.

## ARGUMENTS

M (input) The num ber of row s of the $m$ atrix A. $\mathrm{M}>=0$.
N (input) The num ber of colum ns of the $m$ atrix $\mathrm{A} \cdot \mathrm{N}>=0$

A (input/output)
On entry, the M -by-N m atrix A. On exit, the upper triangle of the array contains the $m$ in $(M, N)$-by $-N$ upper triangularm atrix $R$; the elem ents below the diagonal, together $w$ th the array TA $U$, represent the unitary $m$ atrix $Q$ as a productofm in $(m, n)$ ele$m$ entary reflectors.

LDA (input)
The leading dim ension of the array A. LD A >= $m a x(1, M)$.

JPIVOT (input/output)
On entry, if $\mathbb{P} \mathbb{I V O T}$ (i) ne.0, the i-th colum n of $A$ is perm uted to the front of A *P (a leading colum n); if $\mathbb{P} \mathbb{V} O T(i)=0$, the i-th column of $A$ is a free column. On exit, if $\mathbb{P I V O T}(i)=k$, then
the i-th colum $n$ of $A * P$ was the $k$-th colum $n$ of $A$.

TAU (output)
The scalar factors of the elem entary reflectors.

W ORK (w orkspace)
dim ension $\mathbb{N}$ )

W ORK 2 (w orkspace)
dim ension ( $2 \star \mathrm{~N}$ )
$\mathbb{I N} F O$ (output)
= 0: successfulexit
$<0$ : if $\mathbb{N F O}=-\mathrm{i}$, the i -th argum ent had an illegal value

## FURTHER DETAILS

Them atrix $Q$ is represented as a product of elem entary reflectors
$Q=H(1) H(2) \ldots H(n)$
Each $H$ (i) has the form
$\mathrm{H}=\mathrm{I}-\tan { }^{*} \mathrm{~V}^{*} \mathrm{~V}^{\prime}$
$w$ here tau is a com plex scalar, and $v$ is a com plex vector with $v(1: i-1)=0$ and $v(i)=1 ; v(i+1 \mathrm{~m})$ is stored on exit in $A(i+1 m, i)$.

The $m$ atrix $P$ is represented in jpvtas follow s: If put $(\mathcal{j})=i$
then the th colum n ofP is the ith canonicalunitvector.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgeqrf-com pute $a \operatorname{R}$ factorization of a com plex $M$-by -N m atrix A

## SYNOPSIS

```
SUBROUT\mathbb{NE CGEQRFM,N,A,LDA,TAU,W ORK,LDW ORK,INFO)}
```

COM PLEX A (LDA,*),TAU (*),W ORK (*)
$\mathbb{N}$ TEGER M, N,LDA,LDW ORK, $\mathbb{N}$ FO
SU BROUTINE CGEQRF_64 M,N,A,LDA,TAU,WORK,LDW ORK, $\mathbb{N} F O$ )
COM PLEX A (LDA, *),TAU (*),W ORK (*)
$\mathbb{N}$ TEGER*8M,N,LDA,LDW ORK, $\mathbb{N} F O$

## F95 INTERFACE

$\operatorname{SUBROUT\mathbb {NE}GEQRF}(\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N F O}])$
COM PLEX,D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : A
$\mathbb{N}$ TEGER ::M , N,LDA,LDW ORK, $\mathbb{N} F O$
SUBROUTINE GEQRF_64 (M) $\mathbb{N} \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N} F O])$

COM PLEX,D $\mathbb{I M}$ ENSION (:) ::TAU,W ORK
COM PLEX,D $\mathbb{I}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R(8):: M, N, L D A, L D W$ ORK, $\mathbb{N} F O$

## C INTERFACE

\#include <sunperfh>
void cgeqrf(intm, intn, com plex *a, int lda, com plex *tau, int*info);
void cgeqrf_ 64 (long m , long n, com plex *a, long lda, com plex *tau, long *info);

## PURPOSE

cgeqrf com putes a $Q R$ factorization of a complex $M$ by -N $m \operatorname{atrix} A: A=Q * R$.

## ARGUMENTS

M (input) The num ber of row s of the m atrix A. M >=0.
N (input) The num ber of colum ns of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
On entry, the M -by -N m atrix A. On exit, the ele$m$ ents on and above the diagonalof the array contain the $m$ in $M, N$ )-by $-N$ uppertrapezoidalm atrix $R$ $(R$ is upper triangular ifm $>=n$ ); the elem ents below the diagonal, w ith the aray TAU, represent the unitary $m$ atrix $Q$ as a productofm in $(m, n)$ ele$m$ entary reflectors (see FurtherD etails).

LDA (input)
The leading dim ension of the array A. LDA >= max (1, M).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the amay W ORK. LDW ORK >= $\max (1, \mathbb{N})$. Foroptim um perform ance LDW ORK $>=N * N B$, where NB is the optim alblocksize.

IfLDW ORK $=-1$, then aw orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LDW ORK is issued by XERBLA.
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an illegalvalue

## FURTHER DETAILS

Them atrix $Q$ is represented as a product of elem entary reflectors
$Q=H(1) H(2) \ldots H(k)$, where $k=m$ in $(m, n)$.
Each $H$ (i) has the form
H (i) $=I-\tan * V^{*} V^{\prime}$
$w$ here tau is a com plex scalar, and $v$ is a com plex vector w ith $\mathrm{v}(1: i-1)=0$ and $v(i)=1 ; \mathrm{v}(i+1 \mathrm{~m})$ is stored on exit in A (i+1 $\mathrm{m}, \mathrm{i}$ ), and tau in TAU (i).

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgerc-perform the rank 1 operation A := alpha*x*conjg ( $\left.y^{\prime}\right)+A$

## SYNOPSIS

```
SUBROUTINE CGERC M,N,ALPHA,X, INCX,Y, INCY,A,LDA)
COM PLEX ALPHA
COM PLEXX (*),Y (*),A (LDA,*)
\mathbb{NTEGERM ,N,}\mathbb{N}CX,\mathbb{NCY,LDA}
```



```
COM PLEX A LPHA
COM PLEX X (*),Y (*),A (LDA,*)
\mathbb{NTEGER*8M,N,INCX,INCY,LDA}
```


## F95 INTERFACE

SUBROUTINE GERC ( $\mathbb{M}], \mathbb{N}], A L P H A, X,[\mathbb{N C X}], Y,[\mathbb{N C Y}], A,[\operatorname{LDA}])$
COM PLEX ::ALPHA
COM PLEX,D IM ENSION (:) ::X,Y
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R:: M, N, \mathbb{N C X}, \mathbb{N C Y}, L D A$
SUBROUTINE GERC_64 ( $\mathbb{M}], \mathbb{N}], A \operatorname{LPH} A, X,[\mathbb{N} C X], Y,[\mathbb{N} C Y], A,[L D A])$
COMPLEX ::ALPHA
COM PLEX,D IM ENSION (:) :: X,Y
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R(8):: M, N, \mathbb{N C X}, \mathbb{N} C Y, L D A$

## C INTERFACE

\#include <sunperfh>
void ogerc (intm , intn, com plex *alpha, complex *x, int incx, com plex *y, intincy, com plex *a, int lda);
void ogerc_64 (long m, long n, com plex *alpha, com plex *x, long incx, com plex *y, long incy, com plex *a, long lda);

## PURPOSE

cgerc perform sthe rank 1 operation $A:=a ł p h a \star x^{\star} c o n \dot{g}\left(y^{\prime}\right)$
$+A$ where alpha is a scalar, $x$ is an $m$ elem entvector, $y$ is an $n$ elem entvector and $A$ is an $m$ by $n m$ atrix.

## ARGUMENTS

M (input)
O n entry, M specifies the num ber of row s of the $m$ atrix A. M >=0. U nchanged on exit.

N (input)
O n entry, $N$ specifies the num ber of colum ns of the $m$ atrix A. N >=0. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
$(1+(m-1) \star \operatorname{abs}(\mathbb{N C X}))$. Before entry, the increm ented anray $X$ must contain the $m$ elem ent vectorx. U nchanged on exit.
$\mathbb{N} C X$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of X. $\mathbb{N} C X$ m ustnotbe zero. U nchanged on exit.

Y (input)
$(1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C Y))$. Before entry, the increm ented array $Y \mathrm{~m}$ ust contain the n elem ent vectory. U nchanged on exit.
$\mathbb{N} C Y$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the
elem ents of $Y . \mathbb{N} C Y$ m ustnotbe zero. U nchanged on exit.

A (input/output)
Before entry, the leading $m$ by $n$ part of the array
A must contain the matrix ofcoefficients. O n exit, A is overw rilten by the updated $m$ atrix.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A $>=$ $\max (1, m)$. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgerfs -im prove the com puted solution to a system of linear equations and provides errorbounds and backw ard enroresti$m$ ates for the solution

## SYNOPSIS

SU BROUTINE CGERFS (TRANSA,N,NRHS,A,LDA,AF,LDAF, $\mathbb{P} I V O T, B, L D B$,
X,LDX,FERR,BERR,W ORK,W ORK2, $\mathbb{N} F O$ )
CHARACTER * 1 TRANSA
COM PLEX A (LDA,*),AF (LDAF,*), B (LDB,*),X (LDX,*),W ORK (*)
$\mathbb{N}$ TEGERN,NRHS,LDA,LDAF,LDB,LDX, $\mathbb{N} F O$
$\mathbb{N} T E G E R \mathbb{P} \mathbb{V} O T\left({ }^{( }\right)$
REAL FERR (*), BERR (*), W ORK 2 (*)
SU BROUTINE CGERFS_64 (TRANSA,N,NRHS,A,LDA,AF,LDAF, $\mathbb{P} \mathbb{I V O T}, B$,
LD $B, X, L D X, F E R R, B E R R, W$ ORK,W ORK 2, $\mathbb{N} F O$ )
CHARACTER * 1 TRANSA
COM PLEX A (LDA,*),AF (LDAF,*), B (LDB,*),X (LDX,*),W ORK (*)
$\mathbb{N} T E G E R * 8 N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{V} O T(*)$
REAL FERR (*), BERR (*), WORK 2 (*)

## F95 INTERFACE

SU BROUTINE GERFS ([TRANSA], $\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F], \mathbb{P} \mathbb{I V O T}$, B, [LDB], $\mathrm{X},[\operatorname{LD} \mathrm{X}], F E R R, B E R R,[\mathbb{O} O R K],[W O R K 2],[\mathbb{N} F O])$

CHARACTER (LEN=1) ::TRANSA
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : A, AF, B, X
$\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{I} O T$
REAL,D $\mathbb{M}$ ENSION (:) ::FERR,BERR,W ORK2

SU BROUTINE GERFS_64 ([TRANSA ], $\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F]$, $\mathbb{P} \mathbb{V} O T, B,[L D B], X,[L D X], F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N} F O])$

CHARACTER (LEN=1) ::TRANSA
COM PLEX,D $\mathbb{I M} E N S I O N(:):: W O R K$
COM PLEX , D $\mathbb{M}$ ENSION (: ::) ::A,AF,B,X
$\mathbb{N}$ TEGER (8) :: N , NRHS,LDA,LDAF,LDB,LDX, $\mathbb{N}$ FO
$\mathbb{N} \operatorname{TEGER}$ (8), D $\mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{V} O T$
REAL,D $\mathbb{I}$ ENSION (:) ::FERR,BERR,W ORK 2

## C INTERFACE

\#include <sunperfh>
void ogerfs (char transa, intn, intnrhs, com plex *a, int lda, com plex *af, intldaf, int*ipivot, com plex *b, int ldb, com plex *x, int ldx, float *ferr, float*berr, int*info);
void ogerfs_64 (chartransa, long n, long nrhs, com plex *a, long lda, com plex *af, long ldaf, long *ipivot, com plex *b, long ldb, com plex *x, long ldx, float * ferrr, float *berr, long *info);

## PURPOSE

cgerfs im proves the com puted solution to a system of linear equations and provides emorbounds and backw ard erroresti$m$ ates for the solution.

## ARGUMENTS

TRANSA (input)
Specifies the form of the system of equations:
$=N: A * X=B \quad$ (Notranspose)
$=T$ ': $A * * T * X=B \quad$ ( ranspose)
$=C$ ': $A * * H * X=B \quad$ (C onjugate transpose)

TRANSA is defaulted to $N$ 'forF $95 \mathbb{I N}$ TERFACE.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of the m atriges B and X. NRH S $>=0$.

A (input) The originalN by N m atrix A .

LD A (input)
The leading dim ension of the anay A. LDA >= $\max (1, N)$.

AF (input)
The factors L and U from the factorization $\mathrm{A}=$ $\mathrm{P} * \mathrm{~L} * \mathrm{U}$ as com puted by CGETRF.

LDAF (input)
The leading dim ension of the array AF. LDAF >= max ( $1, \mathbb{N}$ ).
$\mathbb{P I V O T}$ (input)
The pivotindices from CGETRF; for $1<=\mathrm{i}<=\mathrm{N}$, row i of the $m$ atrix $w$ as interchanged w ith row $\mathbb{P}$ IV OT (i).
$B$ (input) The righthand side m atrix $B$.
LD B (input)
The leading dim ension of the array $B$. LD B >= max ( $1, N$ ).

X (input/output)
On entry, the solution $m$ atrix $X$, as com puted by CGETRS. On exit, the im proved solution $m$ atrix $X$.

LD X (input)
The leading din ension of the array X . LD X >= $\max (1, N)$.

FERR (output)
The estim ated forw ard errorbound for each solution vector $X()$ ) the $j$ th column of the solution $m$ atrix X). If XTRUE is the true solution conesponding to $X(\mathcal{O})$, FERR ( $)$ ) is an estim ated upperbound for the $m$ agnitude of the largest ele$m$ ent in ( $\mathrm{X}(\mathcal{\nu})-\mathrm{X}$ TRU E ) divided by the m agnitude of the largestelem entin $X(\mathcal{J})$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

BERR (output)
The com ponentw ise relative backw ard error of each solution vectorX ( $\mathcal{j}$ ) (i.e., the sm allest relative change in any elem entof $A$ orB thatm akes $X(\mathcal{J})$ an exactsolution).

W ORK (w orkspace)
dim ension $(2 * N)$

W ORK 2 (w orkspace)
dim ension $(\mathbb{N})$
$\mathbb{N} F O$ (output)
= 0: successfulexit
< 0 : if $\mathbb{N} F O=-i$, the $i$-th argum ent had an illegal value

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgergf-com pute an $R Q$ factorization of a com plex $M$-by $-N$ m atrix A

## SYNOPSIS

```
SUBROUT\mathbb{NE CGERQFM,N,A,LDA,TAU,W ORK,LDW ORK,INFO)}
```

COM PLEX A (LDA,*),TAU (*),W ORK (*)
$\mathbb{N}$ TEGER M, N,LDA,LDW ORK, $\mathbb{N}$ FO
SU BROUTINE CGERQF_64 M,N,A,LDA,TAU,W ORK,LDW ORK, $\mathbb{N} F O$ )
COM PLEX A (LDA, *),TAU (*),W ORK (*)
$\mathbb{N}$ TEGER*8M,N,LDA,LDW ORK, $\mathbb{N} F O$

## F95 INTERFACE

$\operatorname{SUBROUT\mathbb {NE}GERQF}(\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N F O}])$
COM PLEX,D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : A
$\mathbb{N} T E G E R:: M, N, L D A, L D W$ ORK, $\mathbb{N} F O$
SU BROUTINE GERQF_64 (M) $\mathbb{N} \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N} F O])$

COM PLEX,D $\mathbb{I M} E N S I O N(:):: T A U, W$ ORK
COM PLEX,D $\mathbb{I}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER (8) ::M , N,LDA,LDW ORK, $\mathbb{N}$ FO

## C INTERFACE

\#include <sunperfh>
void cgerqf(intm, intn, com plex *a, int lda, com plex *tau, int*info);
void cgerqf_ 64 (long m , long n, com plex *a, long lda, com plex
*tau, long *info);

## PURPOSE

cgergf com putes an RQ factorization of a com plex $M$-by-N $m$ atrix $A: A=R * Q$.

## ARGUMENTS

M (input) The num ber of row s of the $m$ atrix $\mathrm{A} . \mathrm{M}>=0$.
N (input) The num ber of $c o l u m$ ns of the $m$ atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
On entry, the $\mathrm{M}-$ by -N m atrix A. On exit, if $\mathrm{m}<=$ n , the upper triangle of the subarray $A(1 \mathrm{~m}, \mathrm{n}$ $\mathrm{m}+1 \mathrm{~m}$ ) contains the M boy -M upper triangularm atrix $R$; if $m>=n$, the elem ents on and above the $m$ n )-th subdiagonalcontain the M by -N upper trapezoidal $m$ atrix $R$; the rem aining elem ents, $w$ th the anay TA $U$, represent the unitary $m$ atrix $Q$ as a product of $m$ in $(m, n$ ) elem entary reflectors (see FurtherD etails).

LD A (input)
The leading dim ension of the array A . LD A $>=$ max (1, M).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, \mathrm{~W}$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the array W ORK. LDW ORK >= $m$ ax ( $1, M$ ). Foroptim um perform ance LDW ORK $>=M$ *NB, w here NB is the optim alblocksize.

IfLD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
$\mathbb{N F O}$ (output)
= 0 : successfulexit
<0: if $\mathbb{I N}$ FO $=-$ i, the $i$-th argum enthad an illegalvalue

## FURTHER DETAILS

Them atrix $Q$ is represented as a product of elem entary reflectors

$$
Q=H(1) \text { 'H (2)' } \ldots H(k) \text { ', where } k=m \text { in }(m, n) \text {. }
$$

Each $H$ (i) has the form

H (i) $=I-\tan * V^{*} V^{\prime}$
$w$ here tau is a com plex scalar, and $v$ is a com plex vector with $v(n-k+i+1 n)=0$ and $v(n-k+i)=1$; con $\operatorname{jg}(v(1 n-k+i-1))$ is stored on exitin A $(m-k+i, 1 m-k+i-1)$, and tau in TAU (i).

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgenu -penform the rank 1 operation A := alpha*x*y'+A

## SYNOPSIS



```
COM PLEX A LPHA
COM PLEX X (*),Y (*),A (LDA,*)
\mathbb{NTEGERM,N,INCX,INCY,LDA}
```



```
COM PLEX ALPHA
COM PLEX X (*),Y (*),A (LDA,*)
\mathbb{NTEGER*8M,N,INCX,INCY,LDA}
F95 INTERFACE
```



```
    COMPLEX ::ALPHA
    COM PLEX,D IM ENSION (:) ::X,Y
    COM PLEX,D IM ENSION (:r:) ::A
    \mathbb{NTEGER ::M,N,INCX,}\mathbb{NCY,LDA}
```



```
    COMPLEX ::ALPHA
    COM PLEX,DIM ENSION (:) ::X,Y
    COM PLEX,D IM ENSION (:,:) ::A
    \mathbb{NTEGER (8)::M,N,\mathbb{NCX,INCY,LDA}}\mathbf{N},\textrm{N}
```

void cgenu (intm, intn, com plex *alpha, com plex *x, int incx, com plex *y, int incy, com plex *a, int lda);
void cgeru_64 (long m,long n, com plex *alpha, com plex *x, long incx, com plex *y, long incy, com plex *a, long lda);

## PURPOSE

cgeru perform sthe rank 1 operation $A:=a l p h a^{*} x^{*} y^{\prime}+A$ $w$ here alpha is a scalar, $x$ is an $m$ elem entvector, $y$ is an $n$ elem entvectorand $A$ is an $m$ by $n m$ atrix.

## ARGUMENTS

M (input)
O $n$ entry, $M$ specifies the num berof row $s$ of the $m$ atrix $A . M>=0$. U nchanged on exit.

N (input)
O n entry, N specifies the num ber of colum ns of the $m$ atrix $A . N>=0$. U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
$(1+(m-1) * \operatorname{abs}(\mathbb{N C X}))$. Before entry, the increm ented array $X$ ust contain them elem ent vectorx. U nchanged on exit.
$\mathbb{N C X}$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of $\mathrm{X} . \mathbb{I N C X}$ m ustnotbe zero. U nchanged on exit.
$Y$ (input)
$(1+(n-1) * a b s(\mathbb{N} C Y))$. Before entry, the increm ented array $Y \mathrm{~m} u$ ust contain the n elem ent vectory. U nchanged on exit.
$\mathbb{N C Y}$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of Y. $\mathbb{N} C Y$ m ustnotbe zero. U nchanged
on exit.

A (input/output)
B efore entry, the leading $m$ by $n$ part of the aray A must contain the matrix of coefficients. O $n$ exit, $A$ is overw rilten by the updated $m$ atrix.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD $A>=$ $m a x(1, m)$. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgesdd - com pute the singular value decom position (SV D ) of a complex M -by-N matrix A, optionally com puting the left and/orrightsingularvectors, by using divide-and-oonquer $m$ ethod

## SYNOPSIS

```
SU BROUT\mathbb{NE CGESDD (JOBZ,M,N,A,LDA,S,U,LDU,VT,LDVT,W ORK,}
    LW ORK,RW ORK,\mathbb{N ORK,\mathbb{NFO)}}\mathbf{N}=(
CHARACTER * 1 JOBZ
COM PLEX A (LDA,*),U (LDU ,*),VT (LDVT,*),W ORK (*)
\mathbb{NTEGERM,N,LDA,LDU,LDVT,LW ORK, INFO}
INTEGER IN ORK (*)
REAL S (*),RW ORK (*)
SU BROUTINE CGESDD_64(JOBZ,M,N,A,LDA,S,U,LDU,VT,LDVT,W ORK,
        LW ORK,RW ORK,IN ORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ
COM PLEX A (LDA,*),U (LDU ,*),VT (LDVT,*),W ORK (*)
\mathbb{NTEGER*8M,N,LDA,LDU,LDVT,LW ORK, INFO}
INTEGER*8 IN ORK (*)
REALS (*),RW ORK (*)
```


## F95 INTERFACE

SU BROUTINE GESDD (JOBZ, M ], $\mathbb{N}], A,[L D A], S, U,[L D U], V T,[L D V T]$, [W ORK ], [LW ORK ], [RW ORK], [IW ORK ], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1)::JOBZ
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK

COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A, U,VT
$\mathbb{N}$ TEGER ::M ,N,LDA,LDU,LDVT,LW ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{I N}$ ORK
REAL,D $\mathbb{M}$ ENSION (:) ::S,RW ORK

SU BROUTINE GESDD_64 (JOBZ, M ], $\mathbb{N}], A,[L D A], S, U,[L D U], V T,[L D V T]$, $\left[\begin{array}{l}W \\ O R K\end{array}\right]$ [LW ORK ], [RW ORK ], [IW ORK ], [ $\left.\mathbb{N} F O\right]$ )

CHARACTER (LEN=1): : JOBZ
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : A, U,VT
$\mathbb{N}$ TEGER (8) ::M , N,LDA,LDU,LDVT,LW ORK, $\mathbb{N}$ FO
$\mathbb{N} T E G E R(8), D \mathbb{I M}$ ENSION (:) :: $\mathbb{I N}$ ORK
REAL,D $\mathbb{M}$ ENSION (:) ::S,RW ORK

## C INTERFACE

\#include <sunperfh>
void cgesdd (char j̀bz, intm , intn, com plex *a, int lda, float *s, com plex *u, int ldu, com plex *vt, int ldvt, int *info);
void cgesdd_64 (char jobz, long m, long n, com plex *a, long lda, float*s, com plex *u, long ldu, com plex *vt, long ldvt, long *info);

## PURPOSE

cgesdd com putes the singular value decom position (SV D ) of a complex M -by-N matrix A, optionally com puting the left and/orrightsingularvectors, by using divide-and-conquer method.The SVD isw ritten $=U * S \mathbb{I G M A}$ * conjugate-transpose $(N)$
$w$ here SIG M A is an $M$ by $-N \mathrm{~m}$ atrix which is zero except for its $m$ in $(m, n$ ) diagonal elem ents, $U$ is an $M$ by $M$ unitary $m$ atrix, and V is an N -by -N unitary m atrix. The diagonalelem ents of SIGMA are the singularvalues ofA ; they are real and nonnegative, and are retumed in descending order. The first $m$ in $(m, n)$ colum ns of $U$ and $V$ are the left and rightsingular vectors of A.

N ote that the routine retums $\mathrm{V} T=\mathrm{V} * * \mathrm{H}$, not V .

The divide and conquer algorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on $m$ achines w ith a guard digit in add/subtract, or on those binary $m$ achines $w$ ithout guard digits w hich subtract like the $C$ ray X M P , C ray Y M P , C ray C -90, orC ray-2. It could conceivably fail on hexadecim al or decim al machines w thout guard

## ARGUMENTS

JOBZ (input)
Specifies options for com puting allorpart of the
m atrix U :
= $A$ ': allM colum ns of U and all N row sof $\mathrm{V} * * H$ are retumed in the arays $U$ and $V T ;=S$ : the firstm in $(M, N)$ colum ns of $U$ and the firstm in $(M, N)$ row s of V ** H are retumed in the arrays U and VT ; $=\mathrm{O}^{\prime}$ : If $\mathrm{M}>=\mathrm{N}$, the first N colum ns of U are overw rilten on the array $A$ and allrow sofV **H are retumed in the array VT; otherw ise, all colum $n s$ of $U$ are retumed in the anray $U$ and the firstM row sofV ${ }^{* * H}$ are overw rilten in the amay $\mathrm{VT} ;=\mathrm{N}$ ': no colum ns of U or row sof V ** H are com puted.

M (input) The num ber of row s of the inputm atrix $A . M>=0$.

N (input) The num ber of colum ns of the inputm atrix $\mathrm{A} . \mathrm{N}>=$ 0.

A (input/output)
On entry, the M -by-N m atrix A. On exit, if OBB = $O^{\prime}$ ' $A$ is overw rilten $w$ ith the first $N$ colum $n s$ of
U (the leftsingularvectors, stored colum nw ise) if $\mathrm{M}>=\mathrm{N}$; A is overw ritten w ith the firstM row S of $V * * H$ the right singularvectors, stored row wise) otherw ise. if JOBZ ne. O', the contents of A are destroyed.

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, M)$.

S (output)
The singularvalues of $A$, sorted so that $S$ (i) $>=$ $S(i+1)$ 。

U (output)
$\mathrm{UCOL}=\mathrm{M}$ if $\mathrm{OBZ}=\mathrm{A}^{\prime}$ or $\mathrm{OBZ}=\mathrm{D}^{\prime}$ 'and $\mathrm{M}<\mathrm{N}$;
$\mathrm{UCOL}=\mathrm{m}$ in $(\mathrm{M}, \mathrm{N})$ if $J O B Z=S^{\prime}$. If $\mathrm{OOBZ}=A^{\prime}$ or
JOBZ $=0$ 'and $M<N$, U contains the $M$ boy $M$ unitary $m$ atrix $U$; if $J O B Z=S$ ', $U$ contains the first
$m$ in $(M, N)$ colum ns of $U$ (the leftsingular vectors, stored colum nw ise); if $\mathrm{OBBZ}=0$ 'and $\mathrm{M}>=\mathrm{N}$, or
$J O B Z=N ', U$ is not referenced.

LD U (input)
The leading dim ension of the array $U$. LD U >= 1 ; if $\mathrm{OBZ}=\mathrm{S}^{\prime}$ or A 'or $\mathrm{OB} \mathrm{BZ}=\mathrm{O}^{\prime}$ 'and $\mathrm{M}<\mathrm{N}$, LDU $>=M$.

VT (output)
If $\mathrm{JOBZ}=\mathrm{A}$ 'or $\mathrm{JOBZ}=\mathrm{O}$ 'and $\mathrm{M}>=\mathrm{N}, \mathrm{VT}$ contains the N -by N unitary m atrix $\mathrm{V} * * \mathrm{H}$; if $\mathrm{JOBZ}=$ $S ', V T$ contains the firstm in $M, N$ ) row s of $V * * H$ (the right singularvectors, stored row w ise); if $\mathrm{JOBZ}=\mathrm{O}$ 'andM $<\mathrm{N}$, or $\mathrm{OBBZ}=\mathrm{N}$ ', VT is not referenced.

LDVT (input)
The leading dim ension of the array V T. LD V T >=1;
if $J 0 B Z=A$ 'or $O B Z=0$ 'and $M>=N, L D V T>=N$;
if $J O B Z=S \prime, L D V T>=m$ in $M, N)$.
W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array $W$ ORK.LW ORK >= 1. if $\operatorname{JOBZ}=N$ ', LW ORK $>=2 \star \mathrm{~m}$ in $M, N)+m a x(M, N)$. if JOBZ
$=\quad 0!\quad L W O R K \quad>=$

$=S^{\prime}$ or $A^{\prime}$, LWORK >=
$m$ in $(M, N){ }^{*} m$ in $\left.M, N\right)+2 \star_{m}$ in $\left.M, N\right)+m$ ax $(M, N)$. For good perform ance, LW ORK should generally be larger. If LW ORK < 0 but other input argum ents are legal, W ORK (1) retums optim alLW ORK.

RW ORK (w orkspace)
If $\mathrm{JOBZ}=\mathrm{N}$ ', LRW ORK $>=7 * \mathrm{~m}$ in $M, N)$. O therw ise, LRW ORK $>=5 \star^{m}$ in $(M, N) *_{m}$ in $(M, N)+5 \star_{m}$ in $\left.M, N\right)$

IV ORK (w orkspace)
dim ension ( $8 \star \mathrm{M} \mathbb{I} \mathbb{M}, N)$ )
$\mathbb{N} F O$ (output)
= 0 : successfulexit.
< 0: if $\mathbb{N N}$ FO = -i, the i-th argum ent had an illegalvahue.
$>0$ : The updating process of SBD SD C did not converge.

## B ased on contributions by

M ing $G u$ and $H$ uan Ren, $C$ om puterScience D ívision, U niversity of
C alifomia atB erkeley, U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgesv -com pute the solution to a com plex system of linear equations $A * X=B$,

## SYNOPSIS



```
COM PLEX A (LDA,*),B (LDB,*)
INTEGERN,NRHS,LDA,LDB,INFO
INTEGER \mathbb{PIVOT (*)}
```



```
COM PLEX A (LDA,*),B (LDB,*)
INTEGER*8N,NRHS,LDA,LDB,INFO
INTEGER*8 \mathbb{PIVOT (*)}
```

F95 INTERFACE
SUBROUTINE GESV ( $\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I V O T}, \mathrm{B},[\operatorname{LDB}],[\mathbb{N} F \mathrm{O}])$
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A,B
$\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$
SUBROUTINE GESV_64 ( $\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I V O T}, \mathrm{B},[\operatorname{LDB}],[\mathbb{N} F O])$
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::A,B
$\mathbb{N}$ TEGER (8) ::N,NRHS,LDA,LDB, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T$
C INTERFACE
\#include <sunperfh>
void ogesv (int $n$, int nrhs, complex *a, int lda, int *ipivot, com plex *b, int ldlo, int *info);
void cgesv_64 (long n, long nrhs, com plex *a, long lda, long *ípivot, com plex *b, long ldb, long *info);

## PURPOSE

cgesv com putes the solution to a com plex system of linear equations
$A * X=B, w h e r e A$ is an $N$ boy $-N m$ atrix and $X$ and $B$ are N -by -N R H S m atrices.

The LU decomposition w ith partial pivoting and row interchanges is used to factorA as
$A=P * L * U$,
$w$ here $P$ is a perm utation $m$ atrix, $L$ is unit low er triangular, and $U$ is upper triangular. The factored form of $A$ is then used to solve the system ofequations $A * X=B$.

## ARGUMENTS

N (input) The num ber of linear equations, ie., the order of them atrix $A . N>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS $>=0$.

A (input/output)
O n entry, the $N$ boy -N coefficient $m$ atrix A. On exit, the factors $L$ and $U$ from the factorization $A$ $=P * L * U$; the unitdiagonalelem ents of L are not stored.

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, N)$.

IPIVOT (output)
The pivot indices that define the perm utation $m$ atrix $P$; row $i$ of the $m$ atrix $w$ as interchanged w ith row $\mathbb{P}$ IVOT (i).

B (input/output)
On entry, the N -by-N RH S m atrix of righthand side
$m$ atrix $B$. On exit, if $\mathbb{N} F O=0$, the $N$ boy-NRHS solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the aray $B$. LD B $>=$ $\max (1, N)$.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{I N}$ FO = -i, the i-th argum ent had an illegalvalue
$>0:$ if $\mathbb{N F O}=\mathrm{i}, \mathrm{U}(i, i)$ is exactly zero. The factorization has been com pleted, but the factor $U$ is exactly singular, so the solution could not be com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgesvd - com pute the singular value decom position (SV D ) of a com plex M -by -N m atrix A, optionally com puting the left and/or right singularvectors

## SYNOPSIS

```
SUBROUT\mathbb{NE CGESVD(JOBU,NOBVT,M,N,A,LDA,SNNG,U,LDU,VT,LDVT,}
    W ORK,LDW ORK,W ORK2,INFO)
CHARACTER * 1 JOBU,NOBVT
COM PLEX A (LDA,*),U (LDU ,*),VT (LDVT,*),W ORK (*)
INTEGERM,N,LDA,LDU,LDVT,LDW ORK,\mathbb{NFO}
REALSING (*),W ORK2(*)
SU BROUT\mathbb{NE CGESVD_64(JOBU,JOBVT,M ,N,A,LDA,SING,U ,LDU,VT,}
    LDVT,W ORK,LDW ORK,W ORK2,\mathbb{NFO)}
```

CHARACTER * 1 JOBU, JOBVT
COM PLEX A (LDA, *), U (LDU, *), VT (LDVT,*), WORK (*)
$\mathbb{N}$ TEGER*8 M , N,LDA, LDU,LDVT,LDW ORK, $\mathbb{N} F O$
REALSING (*),W ORK 2 (*)

## F95 INTERFACE

SU BROUTINE GESVD (JOBU, JO BVT, $\mathbb{M}], \mathbb{N}], A,[L D A], S \mathbb{N G}, \mathrm{U},[L D U], V T$, [LDVT], [W ORK], [LDW ORK], $\mathbb{W}$ ORK2], [ $\mathbb{N F O}])$

CHARACTER (LEN=1)::JOBU, JO BV T
COMPLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : A, U,VT
$\mathbb{N} T E G E R:: M, N, L D A, L D U, L D V T, L D W O R K, \mathbb{N} F O$
REAL,D $\mathbb{I M} E N S I O N(:):: S \mathbb{N} G, W$ ORK 2
 VT, [LDVT], [W ORK ], [LDW ORK ], [W ORK 2], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1) :: JOBU, JOBVT
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A, U,VT
$\mathbb{N} T E G E R(8):: M, N, L D A, L D U, L D V T, L D W O R K, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::SING,W ORK2

## C INTERFACE

\#include <sunperfh>
void cgesvd (char jobu, char jobvt, intm , intn, com plex *a, int lda, float *sing, com plex *u, int ldu, com plex *vt, int ldvt, int *info);
void cgesvd_64 (char jobu, char jobvt, long m, long n, com plex *a, long lda, float *sing, com plex *u, long ldu, com plex *vt, long ldvt, long *info);

## PURPOSE

cgesvd com putes the singular value decom position (SVD ) of a com plex M -by -N m atrix A , optionally com puting the left and/or rightsingularvectors. The SVD is w ritten
$=U$ * SIGMA* conjugate-transpose $(N)$
where SIG M A is an $M$-by $-\mathrm{N} m$ atrix which is zero except for its $m$ in $(m, n$ ) diagonal elem ents, $U$ is an $M$ by $M$ unitary $m$ atrix, and $V$ is an $N$ by $-N$ unitary $m$ atrix. The diagonalelem ents of SIGMA are the singularvalues ofA; they are realand nonnegative, and are retumed in descending order. The first $m$ in ( $m, n$ ) colum ns of $U$ and $V$ are the left and right singular vectors of A.

N ote that the routine retums V ** H , not V .

## ARGUMENTS

## JO BU (input)

Specifies options for com puting allor part of the m atrix U :
= A : allm colum ns of U are retumed in anay
U :
$=S$ : the firstm in (m,n) colum nsofU the left singular vectors) are retumed in the array $U$; $=$ $O^{\prime}$ : the firstm in $(m, n)$ colum ns of $U$ the left singular vectors) are overw rilten on the array A;
$=\mathrm{N}$ ': no colum ns of U (no left singularvectors) are com puted.
$J 0 \operatorname{BVT}$ (input)
Specifies options for com puting allor part of the m atrix V **H:
$=A$ ': alln rowsofV ${ }^{* *} \mathrm{H}$ are retumed in the array VT;
$=S$ ': the firstm in $(m, n)$ row sofV $* * H$ the right singular vectors) are retumed in the array VT ; $=$ 0 : the firstm in $(m, n)$ row s of $V * * H$ the right singular vectors) are overw ritten on the array A; $=\mathrm{N}$ ': no row sofV **H (no right singular vectors) are com puted.

JO BVT and JOBU cannotboth be $\mathrm{D}^{\prime}$.
$M$ (input) The num ber of row s of the inputm atrix $A . M>=0$.

N (input) The num ber of colum ns of the inputm atrix $\mathrm{A} . \mathrm{N}>=$ 0.

A (input/output)
On entry, the $M$-by $-N m$ atrix A. On exit, if $J O B U=$ $O^{\prime}$ ' A is overw ritten w the firstm in $(m, n)$
collum ns of $U$ (the left singular vectors, stored colum nw ise); if JOBVT = O',A is overw rilten w th the firstm in $(m, n)$ row sof $V * * H$ the right singularvectors, stored row w ise); if JO BU ne. 0 'and JOBVT ne. 0 ', the contents of A are destroyed.

## LD A (input)

The leading dim ension of the array A. LDA >= max (1, M).

## SING (output)

The singularvalues ofA, sorted so that SIN G (i) $>=S \mathbb{N} G(i+1)$.

$S '$. If $\operatorname{OOBU}=A$ ', $U$ contains the $M$-by $-M$ unitary
$m$ atrix $U$; if $J O B U=S^{\prime}, U$ contains the first
$m$ in ( $m, n$ ) colum ns of $U$ (the left singular vectors, stored collm nw ise); if $\mathrm{OOBU}=\mathrm{N}$ 'or $\mathrm{O}^{\prime}$ ' U is not referenced.

LD U (input)
The leading dim ension of the array $U$. LD $U>=1$;
if $J 0 B U=S$ 'or $A$ ', LDU $>=M$.
VT (input)

If $\mathrm{JOBVT}=\mathrm{A}, \mathrm{VT}$ contains the N by -N unitary $m$ atrix $\mathrm{V} * * \mathrm{H}$; if $\mathrm{JOBV} T=\mathrm{S}^{\prime}, \mathrm{VT}$ contains the first m in $(\mathrm{m}, \mathrm{n})$ row s of $\mathrm{V} * * \mathrm{H}$ (the right singularvectors, stored row w ise); if JOBVT = N 'or D',VT is not referenced.

LDVT (input)
The leading dim ension of the array V T. LD V T >=1;
 m in $(\mathrm{M}, \mathrm{N})$.
W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LDW ORK.

LD W ORK (input)
The dim ension of the aray W ORK. LDW ORK >= 1 . LDW ORK >= $2 \star \mathrm{M} \mathbb{N}(M, N)+M A X(M, N)$ Forgood perform ance, LD W ORK should generally be larger.

IfLDW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 2 (w orkspace)
D $\mathbb{I}$ ENSION $(5 * M \mathbb{N} M, N))$. On exit, if $\mathbb{N} F O>0$, W ORK $2(1 \mathbb{M} \mathbb{N} M, N)-1)$ contains the unconverged superdiagonal elem ents of an upper bidiagonal $m$ atrix $B$ whose diagonal is in $S \mathbb{N} G$ (notnecessarily sorted). $B$ satisfies $A=U * B * V T$, so it has the sam e singular values as A, and singular vectors related by U and V T.
$\mathbb{N} F O$ (output)
= 0: successfulexit.
$<0$ : if $\mathbb{N}$ FO $=-$-i, the $i$-th argum enthad an illegalvalue.
> 0: ifCBD SQR did notconverge, $\mathbb{N F O}$ specifies
how $m$ any superdiagonals of an interm ediate bidiagonalform B did not converge to zero. See the description ofW ORK 2 above for details.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgesvx - use the LU factorization to com pute the solution to a com plex system of linear equations $A * X=B$,

## SYNOPSIS

```
SUBROUT\mathbb{NECGESVX (FACT,TRANSA,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,}}\mathbf{N},\textrm{L}
    EQUED,R,C,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,
    W ORK2,\mathbb{NFO)}
CHARACTER * 1 FACT,TRANSA,EQUED
COM PLEX A (LDA,*),AF (LDAF,*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDA,LDAF,LDB,LDX,}\mathbb{N}F
INTEGER \mathbb{PIVOT (*)}
REALRCOND
REALR (*),C (*),FERR (*),BERR (*),W ORK 2 (*)
```



```
    EQUED,R,C,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,
    WORK2, \mathbb{NFO)}
CHARACTER * 1FACT,TRANSA,EQUED
COM PLEX A (LDA,*),AF (LDAF,*),B (LDB,*),X (LDX,*),W ORK (*)
\mathbb{N TEGER*8 N,NRHS,LDA,LDAF,LDB,LDX, IN FO}
INTEGER*8 \mathbb{PIVOT (*)}
REALRCOND
REALR (*),C (*),FERR (*),BERR (*),W ORK 2 (*)
```


## F95 INTERFACE

SU BROUTINE GESVX (FACT, [TRANSA], $\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F]$, IPIVOT,EQUED,R,C,B,[LDB],X, [LDX],RCOND,FERR, BERR, [W ORK], [W ORK2], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1) ::FACT,TRANSA, EQUED
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A,AF,B,X
$\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V O T}$
REAL ::RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::R,C,FERR,BERR,W ORK2

SU BROUTINE GESVX_64 (FACT, [TRANSA], $\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F]$, $\mathbb{P} \mathbb{V} O T, E Q U E D, R, C, B,[L D B], X,[L D X], R C O N D, F E R R$, BERR, [ $\mathbb{W}$ ORK], [W ORK2], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) ::FACT,TRANSA, EQUED
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : A , AF, B, X
$\mathbb{N}$ TEGER (8) ::N,NRHS,LDA,LDAF,LDB,LDX, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$
REAL ::RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::R,C,FERR,BERR,W ORK 2

## C INTERFACE

\#include <sunperfh>
void cgesvx (char fact, chartransa, intn, intnins, com plex
*a, int lda, com plex *af, int ldaf, int *ípivot, char equed, float*r, float*c, com plex *b, int ldb, com plex *x, int ldx, float*rcond, float * ferrs, float *bers, int *info);
void cgesvx_64 (char fact, chartransa, long n, long nihs, com plex *a, long lda, com plex *af, long ldaf, long *ipívot, char equed, float *r, float *c, com plex *b, long ldb, com plex *x, long ldx, float * rcond, float *ferr, float *berr, long *info);

## PURPOSE

cgesvx uses the LU factorization to com pute the solution to a com plex system of linear equations
$A$ * $X=B$, where $A$ is an $N$ boy $N \mathrm{~m}$ atrix and $X$ and $B$ are N -by-N R H S m atrices.

E rorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:

1. IfFACT = E', real scaling factors are com puted to equilibrate
the system :
TRANS $=N^{\prime}: \operatorname{diag}(R) \star A * \operatorname{diag}(C) \quad \star \operatorname{inv}(\operatorname{diag}(C)) \star X=$ $\operatorname{diag}(R) \star B$
$\operatorname{TRANS}=T:(\operatorname{diag}(\mathbb{R}) \star A * \operatorname{diag}(C)) \star * T * \operatorname{inv}(\operatorname{diag}(\mathbb{R})) \star \mathrm{X}=$ diag (C) *B

TRANS $=C^{\prime}:(\operatorname{diag}(\mathbb{R}) * A * \operatorname{diag}(C)) * * H * \operatorname{inv}(\operatorname{diag}(\mathbb{R})) * X=$ diag (C)*B

W hether or not the system w illbe equilibrated depends on the scaling of them atrix A , but ifequilibration is used, A is overw rilten by diag $(\mathbb{R}) * A$ *diag $(C)$ and $B$ by $\operatorname{diag}(\mathbb{R}) * B \quad$ (if TRANS = N ) ordiag (C)*B (if TRANS = T'or C).
2. IfFACT $=N$ 'or $E$ ', the LU decomposition is used to factorthe
$m$ atrix A (afterequilibration ifFACT $=$ E) as

$$
A=P \star L \star U
$$

$w$ here $P$ is a perm utation $m$ atrix, $L$ is a unit low er triangular
$m$ atrix, and $U$ is upper triangular.
3. If som e $U(i, i)=0$, so that $U$ is exactly singular, then the routine
retums $w$ ith $\mathbb{N F O}=i .0$ therw ise, the factored form of $A$ is used
to estim ate the condition num ber of the $m$ atrix $A$. If the reciprocal of the condition num ber is less than $m$ achine precision,
$\mathbb{N} F O=N+1$ is retumed as a $w$ aming, but the routine stillgoes on
to solve for $X$ and com pute error bounds as described below .
4.The system ofequations is solved forX using the factored form of A.
5. Iterative refinem ent is applied to im prove the com puted solution
$m$ atrix and calculate error bounds and backw ard error estim ates
for 五.
6. If equilibration $w$ as used, the $m$ atrix $X$ is prem ultiplied by diag (C) (ifTRANS = N) ordiag $(\mathbb{R})$ (ifTRANS = T' or
C) so that itsolves the originalsystem before equilibration.

## ARGUMENTS

FACT (input)
Specifies w hether ornot the factored form of the $m$ atrix A is supplied on entry, and ifnot, w hether them atrix A should be equilibrated before it is factored. $=\mathrm{F}^{\prime}:$ On entry, AF and $\mathbb{P} \mathbb{I V O T}$ contain the factored form of A. IfEQUED is not $N$ ', the $m$ atrix A has been equilibrated $w$ ith scaling factors given by $R$ and $C . A, A F$, and $\mathbb{P} \mathbb{I V O T}$ are not m odified. $=\mathrm{N}$ ': The m atrix A w ill.be copied to A F and factored.
$=\mathrm{E}$ : The matrix A w ill be equilibrated if necessary, then copied to A F and factored.

TRANSA (input)
Specifies the form of the system of equations:
$=\mathrm{N}$ ': A * $\mathrm{X}=\mathrm{B} \quad$ N o transpose)
$=T: A * * T * X=B \quad$ ( ranspose)
$=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad$ (C onjugate transpose)
TRANSA is defaulted to $N$ 'forF $95 \mathbb{I N}$ TERFACE.

N (input) The num ber of linear equations, i.e., the order of them atrix $A . N>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the $m$ atrices $B$ and X. NRH S $>=0$.

A (input/output)
On entry, the N -by -N m atrix A . IfFA CT $=\mathrm{F}^{\prime}$ and EQUED is not $N$ ', then $A$ musthave been equilibrated by the scaling factors in R and/orC. A is not modified if FACT $=$ F'or $\mathrm{N}^{\prime}$, or if $\mathrm{FACT}=$ E'and EQUED = N 'on exit.

Onexit, ifEQUED ne. $N$ ', A is scaled as follow s: EQUED $=R$ ': $A=\operatorname{diag}(R) * A$
EQUED = C ': A :=A * diag (C)
EQUED = B': A := diag (R)*A * diag (C ).

LD A (input)
The leading dim ension of the anay A. LDA >= $\max (1, \mathbb{N})$.

AF (input/output)
IfFACT = $\mathrm{F}^{\prime}$, then AF is an input argum entand on
entry contains the factors $L$ and $U$ from the factorization $A=P * L * U$ as com puted by CGETRF. If EQUED ne. $N$ ', then $A F$ is the factored form of the equilibrated $m$ atrix $A$.

If $\mathrm{FACT}=\mathrm{N}$ ', then AF is an output argum ent and on exit retums the factors $L$ and $U$ from the factorization $A=P * L * U$ of the originalm atrix $A$.

IfFACT $=\mathrm{E}$ ', then $A F$ is an output argum ent and on exit retums the factors $L$ and $U$ from the factorization $\mathrm{A}=\mathrm{P} * \mathrm{~L} * \mathrm{U}$ of the equilibrated m atrix A (see the description of $A$ for the form of the equilibrated $m$ atrix).

## LDAF (input)

The leading dim ension of the array AF. LD AF >= $\max (1, N)$.

## IPIVOT (inputoroutput)

IfFACT=F', then $\mathbb{P} \mathbb{V} O T$ is an input argum ent and on entry contains the pivot indioes from the factorization $\mathrm{A}=\mathrm{P} * \mathrm{~L} * \mathrm{U}$ as com puted by CGETRF; row $i$ of the $m$ atrix $w$ as interchanged $w$ th row $\mathbb{P} \mathbb{V} O T$ (i).

IfFACT $=N$ ', then $\mathbb{P} \mathbb{I V O T}$ is an output argum ent and on exit contains the pivot indiges from the factorization $\mathrm{A}=\mathrm{P} * \mathrm{~L} * \mathrm{U}$ of the originalm atrix A .

IfFACT=E', then $\mathbb{P I V O T}$ is an output argum ent and on exit contains the pivot indices from the factorization $A=P * L * U$ of the equilibrated $m$ atrix A.

EQUED (input/output)
Specifies the form of equilibration thatw as done.
$=N^{\prime}:$ N o equilibration (alw ays true ifFACT =
$\mathrm{N})$.
$=R$ ': Row equilibration, ie., A has been
prem ultiplied by diag $(R)$ ) = C': Colum n equilibration, ie., A has been postm ultiplied by diag (C ). = B ': B oth row and colum $n$ equilibration, ie., A has been replaced by diag $(\mathbb{R})$ * A * diag (C). EQUED is an inputargum entifFACT= F '; otherw ise, it is an output argum ent.
$R$ (input/output)
The row scale factors forA. IfEQUED = R' or
$B$ ', A is multiplied on the leftby diag $(\mathbb{R})$; if
$E Q U E D=N$ 'or $C$ ', $R$ is notaccessed. $R$ is an
input argum ent if $\mathrm{FACT}=\mathrm{F}$ '; otherw ise, R is an output argum ent. IfFACT = F'andEQUED = R'or $B$ ',each elem entofR $m$ ustbe positive.

C (input/output)
The colum $n$ scale factors for $A$. IfEQUED = C 'or
B ', A is multiplied on the right.by diag ( C ); if EQUED $=N$ 'or $R$ ', $C$ is notaccessed. $C$ is an input argum ent ifFACT = F '; otherw ise, C is an outputargum ent. IfFACT = F 'and $\mathrm{EQUED}=\mathrm{C}$ 'or B ', each elem entofC m ustbe positive.
B (input/output)
On entry, the N -by-NRH S righthand side m atrix B .
On exit, if EQUED $=N$ ', $B$ is notm odified; if TRANSA $=N^{\prime}$ and EQUED $=R$ 'or $B$ ', $B$ is overw rilten by diag $(R) * B$; ifTRANSA $=T^{\prime}$ 'or $C^{\prime}$ and EQUED $=C^{\prime}$ or $B^{\prime}, B$ is overw ritten by diag (C) *B .

LD B (input)
The leading dim ension of the array $B$. LD B >= $\max (1, N)$.

X (output)
If $\mathbb{N} F O=0$ or $\mathbb{N} F O=N+1$, the N -by -N RH $S$ solution
$m$ atrix $X$ to the original system of equations.
$N$ ote that $A$ and $B$ arem odified on exit if EQUED
ne. $N$ ', and the solution to the equilibrated
system is inv (diag (C ))*X ifTRANSA = N 'and EQUED
$=C$ 'or $B^{\prime}$, orinv (diag $\left.(R)\right) * X$ ifTRANSA $=T$ 'or $C^{\prime}$ and EQUED $=R$ 'or $B$ '.

LD X (input)
The leading dim ension of the anay X . LD X >= $\max (1, N)$.

## RCOND (output)

The estim ate of the reciprocal condition num ber of the matrix A afterequilibration (if done). If
RCOND is less than them achine precision (in particular, ifRCOND $=0$ ), the $m$ atrix is singular to w orking precision. This condition is indicated by a retum code of $\mathbb{N} \mathrm{FO}>0$.

FERR (output)
The estim ated forw ard errorbound for each solution vector $X()$ ) the $j$ th column of the solution $m$ atrix X). If XTRUE is the true solution comesponding to $X(\mathcal{O}), \operatorname{FERR}(\mathcal{7})$ is an estim ated upperbound for the $m$ agnitude of the largest ele-
$m$ ent in $(X(\mathcal{O})-X$ TRUE $)$ divided by the $m$ agnitude of the largestelem ent in $X(\mathcal{)}$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

## BERR (output)

The com ponentw ise relative backw ard error of each solution vector $X(\mathcal{)}$ (i.e., the sm allest relative change in any elem entof $A$ orB thatm akes $X(\mathcal{j})$ an exactsolution).

W ORK (w orkspace)
dim ension ( $2 * \mathrm{~N}$ )
W ORK 2 (w orkspace)
dim ension ( $2 *$ N ) On exit, W ORK 2 (1) contains the reciprocal pivot grow th factornorm (A)/norm (U). The "max absolute elem ent" norm is used. If W ORK2(1) is much less than 1, then the stability of the LU factorization of the (equilibrated)
$m$ atrix A could be poor. This also $m$ eans that the solution X, condition estim atorRC OND, and forw ard error bound $F E R R$ could be unreliable. If factorization fails $w$ ith $0<\mathbb{N} F O<=N$, then $W$ ORK 2 (1) contains the reciprocalpivot grow th factor for the leading $\mathbb{N}$ FO colum ns of A.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the i-th argum enthad an illegalvalue
$>0$ : if $\mathbb{N F O}=i$, and $i$ is
<= N : U (i,i) is exactly zero. The factorization
has been com pleted, but the factorU is exactly singular, so the solution and error bounds could not be com puted. RCOND $=0$ is retumed. $=\mathrm{N}+1: \mathrm{U}$ is nonsingular, butRCOND is less than machine precision, $m$ eaning that the $m$ atrix is singular to w orking precision. N evertheless, the solution and error bounds are com puted because there are a num ber of situations w here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgetf2 - com pute an LU factorization of a general $m$-by-n $m$ atrix A using partialpivoting $w$ th row interchanges

## SYNOPSIS

```
SU BROUTINE CGETF2M,N,A,LDA,\mathbb{P}\mathbb{M},\mathbb{NFO)}
COM PLEX A (LDA,*)
INTEGERM,N,LDA,INFO
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(*)}
```



```
COM PLEX A (LDA,*)
\mathbb{NTEGER*8M,N,LDA,INFO}
\mathbb{NTEGER*8 P\mathbb{IV (*)}}\mathbf{*})
F95 INTERFACE
```



```
    COM PLEX,D IM ENSION (:,:) ::A
    \mathbb{NTEGER ::M,N,LDA,NNFO}
    INTEGER,D IM ENSION (:) :: \mathbb{P IV}
```



```
    COM PLEX,D IM ENSION (:,:) ::A
    \mathbb{NTEGER (8)::M,N,LDA, NNFO}
    INTEGER (8),D IM ENSION (:) ::\mathbb{PIV}
```

C INTERFACE
\#include < sunperfh>
void ogetf2 (intm, intn, com plex *a, int lda, int *ịiv, int*info);
void cgetf2_64 (long m, long n, com plex *a, long lda, long *ipiv, long *info);

## PURPOSE

cgetff com putes an LU factorization of a general $m$-by-n $m$ atrix A using partialpivoting $w$ ith row interchanges.

The factorization has the form

$$
A=P * L * U
$$

$w$ here $P$ is a perm utation $m$ atrix, $L$ is low er triangular $w$ ith unit diagonal elem ents (low ertrapezoidalifm > n), and U is uppertriangular (uppertrapezoidalifm < n).

This is the right-looking Level2 B LA S version of the algorithm .

## ARGUMENTS

M (input) The num ber of row s of the $m$ atrix $\mathrm{A} . \mathrm{M}>=0$.

N (input) The num ber of colum ns of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
O n entry, them by $n m$ atrix to be factored. On
exit, the factors $L$ and $U$ from the factorization $A$
$=\mathrm{P} * \mathrm{~L} * \mathrm{U}$; the unitdiagonalelem ents of L are not stored.

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, M)$.
$\mathbb{P} \mathbb{I} V$ (output)
The pivotindioes; for $1<=i<=m$ in $(M)$ ), row i of the $m$ atrix $w$ as interchanged $w$ ith row $\mathbb{P} \mathbb{I V}$ (i).

INFO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-\mathrm{k}$, the k -th argum enthad an illegalvalue
$>0:$ if $\mathbb{N} F O=k, U(k, k)$ is exactly zero. The factorization has been com pleted, but the factor $U$ is
exactly singular, and division by zero will occur
if it is used to solve a system of equations.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgetrf-com pute an LU factorization of a general M by -N $m$ atrix A using partialpivoting $w$ th row interchanges

## SYNOPSIS



```
COM PLEX A (LDA,*)
\mathbb{NTEGERM,N,LDA,\mathbb{NFO}}\mathbf{M}\mathrm{ (N,}
\mathbb{NTEGER \mathbb{PIVOT (*)}}\mathbf{(})
SUBROUT\mathbb{NE CGETRF_64(M,N,A,LDA, \mathbb{PIVOT,INFO)}}\mathbf{M}\mathrm{ (N)}
COM PLEX A (LDA,*)
INTEGER*8M,N,LDA,INFO
INTEGER*8 \mathbb{PIVOT (*)}
F95 INTERFACE
```



```
    COM PLEX,D IM ENSION (:,:) ::A
    \mathbb{NTEGER ::M,N,LDA,INFO}
    \mathbb{NTEGER,D IM ENSION (:) :: \mathbb{PIVOT}}\mathbf{T}\mathrm{ O}
```



```
    COM PLEX,D IM ENSION (:,:) ::A
    \mathbb{NTEGER (8) ::M,N,LDA,INFO}
```


C INTERFACE
\#include < sunperfh>
void ogetrf(intm , intn, com plex *a, int lda, int *ipivot, int*info);
void ogetrf_64 (long m, long n, com plex *a, long lda, long *ípivot, long *info);

## PURPOSE

cgetrf com putes an LU factorization of a general M boy-N $m$ atrix A using partialpivoting $w$ ith row interchanges.

The factorization has the form

$$
A=P * L * U
$$

$w$ here $P$ is a perm utation $m$ atrix, $L$ is low er triangular $w$ ith unit diagonal elem ents (low ertrapezoidalifm > n), and U is upper triangular (uppertrapezoidalifm < n).

This is the right-looking Level3 B LA S version of the algorithm .

## ARGUMENTS

M (input) The num ber of row s of the m atrix $\mathrm{A} . \mathrm{M}>=0$.

N (input) The num ber of colum ns of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
On entry, the $M$ boy $-\mathrm{N} m$ atrix to be factored. On exit, the factors $L$ and $U$ from the factorization $A$
$=\mathrm{P} * \mathrm{~L} * \mathrm{U}$; the unitdiagonalelem ents of L are not stored.

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, M)$.

## IPIVOT (output)

The pivotindioes; for $1<=i<=m$ in $(M)$ ), row i of the $m$ atrix $w$ as interchanged $w$ ith row $\mathbb{P}$ IV OT (i).

INFO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N F O}=-i$, the $i$ th argum enthad an illegalvalue
$>0:$ if $\mathbb{N} F O=i, U(i, i)$ is exactly zero. The factorization has been com pleted, but the factor U
is exactly singular, and division by zero will occur if it is used to solve a system ofequations.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgetri-com pute the inverse of a m atrix using the LU factorization com puted by C G ETRF

## SYNOPSIS



```
COM PLEX A (LDA,*),W ORK (*)
INTEGERN,LDA,LDWORK,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}
```



```
COM PLEX A (LDA,*),W ORK (*)
INTEGER*8N,LDA,LDW ORK, INFO
INTEGER*8 \mathbb{PIVOT (*)}
```


## F95 INTERFACE

SU BROUTINE GETRI( $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V O T}, \mathbb{W}$ ORK $],[L D W O R K],[\mathbb{N F O}])$
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : : A
$\mathbb{N} T E G E R:: N, L D A, L D W$ ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T$
SU BROUTINE GETRI_64 ( $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V O T},[\mathbb{W} O R K],[L D W$ ORK $],[\mathbb{N F O}])$
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER (8) :: N, LDA,LDW ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T$

## C INTERFACE

\#include <sunperfh>
void cgetri(intn, com plex *a, int lda, int *ipivot, int *info);
void ogetri_ 64 (long n, com plex *a, long lda, long *ịivot, long *info);

## PURPOSE

cgetricom putes the inverse of a m atrix using the LU factorization com puted by CGETRF .

Thism ethod inverts $U$ and then com putes inv (A) by solving the system inv (A) ${ }^{(A L}=\operatorname{inv}(\mathbb{U})$ for inv (A).

## ARGUMENTS

N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
O n entry, the factors $L$ and $U$ from the factoriza-
tion $\mathrm{A}=\mathrm{P} * \mathrm{~L} * \mathrm{U}$ as com puted by CGETRF. On ex斗, if $\mathbb{N} F O=0$, the inverse of the originalm atrix $A$.

LD A (input)
The leading dim ension of the array A. LD A >= $\max (1, N)$.

IPIVOT (input)
The pivotindiges from CGETRF ; for $1<=i<=N$, row $i$ of the $m$ atrix $w$ as interchanged $w$ ith row $\mathbb{P I V O T}$ (i).

W ORK (w orkspace)
On exit, if $\mathbb{N F O}=0$, then $W$ ORK (1) retums the optim alLDW ORK.

LDW ORK (input)
The dimension of the array W ORK. LDW ORK >= $\max (1, N)$. Foroptim alperform ance LDW ORK $>=N * N B$, where NB is the optim al blocksize retumed by ILAENV.

If LD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK aray, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N} F O=i, U(i, i)$ is exactly zero; the $m$ atrix is singular and its inverse could notbe com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgetrs - solve a system of linearequations A * $\mathrm{X}=\mathrm{B}, \mathrm{A} * * T$

* $\mathrm{X}=\mathrm{B}$, or $\mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B}$ w th a generaln toy -N m atrix A using the LU factorization com puted by CGETRF


## SYNOPSIS



```
CHARACTER * 1 TRANSA
COM PLEX A (LDA,*),B (LD B,*)
INTEGERN,NRHS,LDA,LDB,INFO
INTEGER \mathbb{PIVOT (*)}
```



```
CHARACTER * 1 TRANSA
COM PLEX A (LDA,*),B (LD B,*)
INTEGER*8N,NRHS,LDA,LDB,INFO
INTEGER *8 \mathbb{PIVOT (*)}
```

F95 INTERFACE
SU BROUTINE GETRS ([TRANSA], $\mathbb{N}], \mathbb{N R H S}], A,[L D A], \mathbb{P} \mathbb{I} O T, B,[L D B]$,
[ $\mathbb{N}$ FO ])
CHARACTER ( $\llcorner E N=1$ ) ::TRANSA
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::A,B
$\mathbb{N}$ TEGER ::N,NRHS,LDA,LDB, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$
SU BROUTINE GETRS_64 ([TRANSA], $\mathbb{N}], \mathbb{N R H S}], A,[L D A], \mathbb{P} \mathbb{I} O T, B,[L D B]$,
[ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::TRANSA
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : A , B
$\mathbb{N}$ TEGER (8) ::N,NRHS,LDA,LDB, $\mathbb{N}$ FO
$\mathbb{N}$ TEGER (8), D $\mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$

## C INTERFACE

\#include <sunperfh>
void cgetrs (chartransa, intn, intnrhs, com plex *a, int lda, int *ipivot, com plex *b, int ldlo, int *info);
void ogetrs_64 (chartransa, long n, long nrhs, com plex *a, long lda, long *ipívot, com plex *b, long ldb, long *info);

## PURPOSE

cgetrs solves a system of linear equations
$A * X=B, A * * T * X=B$, or $A * * H * X=B$ with $a$ general $N$ boy $-N$ m atrix A using the LU factorization com puted by CGETRF.

## ARGUMENTS

TRANSA (input)
Specifies the form of the system ofequations:
$=N: A * X=B \quad$ ( $o$ transpose)
$=T$ ': A **T * $\mathrm{X}=\mathrm{B} \quad$ (Transpose)
$=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad$ (C onjugate transpose)

TRANSA is defaulted to N 'forF95 $\mathbb{I N T E R F A C E .}$

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRH S >=0.

A (input) The factors $L$ and $U$ from the factorization $A=$ $\mathrm{P} * \mathrm{~L} * \mathrm{U}$ as com puted by CGETRF.

LDA (input)
The leading dim ension of the array A. LD A >= $\max (1, N)$.

IPIVOT (input)
The pivotindiaes from CGETRF ; for $1<=i<=N$, row $i$
of the $m$ atrix $w$ as interchanged $w$ ith row $\mathbb{P}$ IV OT (i).

B (input/output)
On entry, the right hand side $m$ atrix $B$. On exit, the solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the array B . LD B >= $\max (1, N)$.

IN FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N N}$ FO $=-i$, the $i$ th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cggbak - form the rightor lefteigenvectors of a com plex
generalized eigenvalue problem $A * x=$ lam boda*B ${ }^{*}$ x, by backw ard
transform ation on the com puted eigenvectors of the balanced
pair ofm atrioes outputby C G GBAL

## SYNOPSIS

```
SUBROUTINE CGGBAK (JOB,S\mathbb{DE,N,}\mathbb{NO,\mathbb{H}I,LSCALE,RSCALE,M,V,LDV,}
    \mathbb{NFO)}
CHARACTER * 1 JOB,SDE
COM PLEX V (LDV,*)
\mathbb{NTEGERN,ILO,\mathbb{H I,M,LDV,INFO}}\mathbf{N},\mp@code{L}
REAL LSCALE (*),RSCALE (*)
```



```
    LDV,\mathbb{NFO)}
CHARACTER * 1 JOB,SIDE
COM PLEX V (LDV,*)
```



```
REAL LSCALE (*),RSCALE (*)
```


## F95 INTERFACE

```
SU BROUTINE GGBAK (JOB,SDE, \(\mathbb{N}], \mathbb{L} O, \mathbb{H} I, L S C A L E, R S C A L E, ~ M], V\), [LDV], [ \(\mathbb{N F O}]\) )
CHARACTER (LEN=1)::JOB,SDE
COM PLEX,D \(\mathbb{I}\) ENSION (: : : : : V
\(\mathbb{N} T E G E R:: N, \mathbb{I} O, \mathbb{H} \mathrm{I}, \mathrm{M}, \mathrm{LDV}, \mathbb{N} F O\)
REAL,D \(\mathbb{I M}\) ENSION (:) ::LSCALE,RSCALE
```

SU BROU T $\mathbb{N E}$ GGBAK_64 (JOB,SDE, $\mathbb{N}], \mathbb{I} O, \mathbb{H} I, L S C A L E, R S C A L E, ~ M ~], V$, [LDV], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1):: JOB ,SDE
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::V
$\mathbb{N} T E G E R(8):: N, \mathbb{L O}, \mathbb{H} I, M, L D V, \mathbb{N F O}$
REAL,D IM ENSION (:) ::LSCALE,RSCALE

## C INTERFACE

\#include <sunperfh>
void cgg.bak (char job, charside, intn, int ilo, int ini, float *lscale, float*rscale, intm, com plex *v, int ldv, int*info);
void cggbak_64 (char job, char side, long n, long ilo, long ihi, float *lscale, float * rscale, long m, com plex * v , long ldv, long *info);

## PURPOSE

cggbak form $s$ the rightor left eigenvectors of a com plex generalized eigenvalue problem $A$ * $x=$ lam bda* $B * x$, by backw ard transform ation on the com puted eigenvectors of the balanced pair ofm atrioes output.by C G G BAL .

## ARGUMENTS

$J O B$ (input)
Specifies the type of backw ard transform ation
required:
$=\mathrm{N}$ ': do nothing, retum im m ediately;
$=\mathrm{P}:$ do backw ard transform ation forperm utation
only;
= S': do backw ard transform ation for scaling
only;
= B ': do backw ard transform ations forboth per$m$ utation and scaling. JO B m ustbe the sam e as the argum ent JO B supplied to C G GBAL.
$S D E$ (input)
= R : : V contains righteigenvectors;
$=\mathrm{L} \cdot: \mathrm{V}$ contains lefteigenvectors.
N (input) The num ber of row s of the m atrix $\mathrm{V} . \mathrm{N}>=0$.

The integers $\mathbb{H O}$ and $\mathbb{H}$ I determ ined by CG GBAL. 1 $<=\mathbb{H O}<=\mathbb{H} I<=N$, if $N>0 ; \mathbb{H}=1$ and $\mathbb{H} \mathrm{I}=0$, if $\mathrm{N}=0$.

IH I (input)
The integers $\mathbb{I L O}$ and $\mathbb{H}$ I determ ined by CG GBAL. 1 $<=\mathbb{L O}<=\mathbb{H} \mathrm{I}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{H} \mathrm{O}=1$ and $\mathbb{H} \mathrm{I}=0$, if $\mathrm{N}=0$.

LSCALE (input)
D etails of the perm utations and/or scaling factors applied to the left side of $A$ and $B$, as retumed by CGGBAL .
RSCALE (input)
D etails of the perm utations and/or scaling factors applied to the right side of $A$ and $B$, as retumed by CGGBAL .

M (input) The num ber of colum ns of the matrix $\mathrm{V} . \mathrm{M}>=0$.
V (input/output)
O $n$ entry, the $m$ atrix of right or lefteigenvectors to be transform ed, as retumed by CTGEVC. On exit, $V$ is overw ritten by the transform ed eigenvectors.

LDV (input)
The leading $d m$ ension of the $m$ atrix $V$. LDV >= $\max (1, N)$.
$\mathbb{N} F O$ (output)
= 0: successfulexit.
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum enthad an illegalvahue.

## FURTHER DETAILS

See R . .W ard, B alancing the generalized eigenvalue problem, SIAM J.Sci.Stat.C omp. 2 (1981),141-152.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

```
cggbal-balance a pairofgeneral com plex m atrices (A ,B )
```


## SYNOPSIS



```
    W ORK,INFO)
CHARACTER * 1 JOB
COM PLEX A (LDA,*),B (LD B,*)
INTEGERN,LDA,LDB, #O, \mathbb{H I, INFO}
REALLSCALE (*),RSCALE (*),W ORK (*)
```



```
        RSCALE,WORK,INFO)
CHARACTER * 1 JOB
COM PLEX A (LDA,*),B (LD B,*)
```



```
REAL LSCALE (*),RSCALE (*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE GGBAL (JOB, $\mathbb{N}$ ], A, [LDA ], B, [LD B], $\mathbb{I} O, \mathbb{H} I, L S C A L E$, RSCALE, [W ORK], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) :: JOB
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::A,B
$\mathbb{N}$ TEGER :: N,LDA,LDB, $\mathbb{L}$, $\mathbb{H} \mathrm{I}, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::LSCALE,RSCALE,W ORK
SU BROUTINE GGBAL_64 (JOB, $\mathbb{N}], A,[L D A], B,[L D B], \mathbb{L O}, \mathbb{H} I, L S C A L E$, RSCALE, [W ORK], [ $\mathbb{N} F O]$ )

CHARACTER ( $L E N=1$ ) :: JOB
COM PLEX, D $\mathbb{I M} E N S I O N(:,:$ : $: A, B$
$\mathbb{N}$ TEGER (8) :: $\mathrm{N}, \mathrm{LD} A, L D B, \mathbb{L}, \mathbb{H} \mathrm{I}, \mathbb{N} \mathrm{FO}$
REAL,D $\mathbb{I M} E N S I O N$ (:) ::LSCALE,RSCALE,W ORK

## C INTERFACE

\#include <sunperfh>
void cggbal(char j.b, intn, com plex *a, int lda, com plex
*b, int ldlo, int*ilo, int*ihi, float*lscale, float*rscale, int*info);
void cggbal 64 (char jंb, long n, com plex *a, long lda, com plex *b, long ldb, long *ilo, long *ihi, float *lscale, float * rscale, long *info);

## PURPOSE

cggbalbalances a pair of general com plex matrices ( $A, B$ ). This involves, first, perm uting $A$ and $B$ by sim ilarity transform ations to isolate eigenvalues in the first 1 to IIO \$-\$1 and last $\mathbb{H}$ I+1 to $N$ elem ents on the diagonal; and second, applying a diagonal sim ilarity transform ation to row $s$ and colum ns $\mathbb{I} O$ to $\mathbb{H}$ Ito $m$ ake the row $s$ and colum ns as close in norm as possible. B oth steps are optional.

B alancing $m$ ay reduce the 1 -norm of the $m$ atrices, and im prove the accuracy of the com puted eigenvalues and/oreigenvectors in the generalized eigenvalue problem $A{ }^{*} x=\operatorname{lam} . b d a * B{ }^{*} \times$.

## ARGUMENTS

JO B (input)
Specifies the operations to be perform ed on A and
B :
$=\mathrm{N}^{\prime}$ : none: simply set $\mathbb{H} 0=1, \mathbb{H} I=\mathrm{N}$, LSCALE $(\mathbb{I})=1.0$ and RSCALE $(\mathbb{I})=1.0$ fori=1,...N ;
$=P^{\prime}$ : perm ute only;
$=\mathrm{S}$ ': scale only;
$=B:$ both perm ute and scale.

N (input) The order of them atriges $A$ and $B . N>=0$.

A (input/output)
On entry, the input $m$ atrix A. On exit, A is overw rilten by the balanced $m$ atrix. If $\mathrm{OB}=\mathrm{N}^{\prime}$,

A is not referenced.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, N)$.
$B$ (input) On entry, the input $m$ atrix $B$. On exit, $B$ is overw rilten by the balanced $m$ atrix. If $J 0 B=N$ ', $B$ is not referenced.

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.

ㅍO (output)
IOO and $\mathbb{H}$ Iare set to integers such thaton exit
$A(i, j)=0$ and $B(i, 1)=0$ if $i>$ jand $j=$ $1, \ldots$, ILO O -1 or $i=\mathbb{H}$ I $+1, \ldots, N$. If $J O B=N$ ' or $S^{\prime}, \mathrm{HO}=1$ and $\mathbb{H} \mathrm{I}=\mathrm{N}$.

IH I (output)
IIO and $\mathbb{H}$ I are set to integers such that on exit
$A(i, j)=0$ and $B(i, 7)=0$ if $i>j a n d i=$


LSCALE (input)
D etails of the perm utations and scaling factors applied to the left side of $A$ and $B$. IfP $(\mathcal{J})$ is the index of the row interchanged $w$ ith row $j$ and D ( $j$ ) is the scaling factor applied to row $j$ then LSCALE ( $)=\mathrm{P}(\mathcal{i}$ ) for $J=1, \ldots$, ILO-1 $=\mathrm{D}(\mathcal{j})$ for $J=\mathbb{L O}, \ldots, \mathbb{H} I=P(\mathcal{O}) \quad$ for $J=\mathbb{H} I+1, \ldots, N$. The order in which the interchanges are m ade is N to $\mathbb{H} \mathrm{I}+1$, then 1 to $\mathbb{H} O-1$.

RSCALE (input)
D etails of the perm utations and scaling factors applied to the right side of $A$ and $B$. If $P(i)$ is the index of the colum $n$ interchanged $w$ ith colum $n$ $j$ and $D(j)$ is the scaling factorapplied to column $j$ then RSCALE $(\mathcal{j})=P(\mathcal{j})$ for $J=$
 for $J=\mathbb{H} I+1, \ldots, N$. The order in which the interchanges are m ade is N to $\mathrm{IH} \mathrm{I}+1$, then 1 to [ H - 1 .

W ORK (w orkspace)
dim ension (6*N )
$\mathbb{N} F O$ (output)
$=0$ : successfulexit
< 0: if $\mathbb{N}$ FO = -i, the i-th argum enthad an illegalvalue.

## FURTHER DETAILS

See R C.W A RD , B alancing the generalized eigenvalue problem, SIAM J.Sci.Stat.Comp. 2 (1981),141-152.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgges - com pute fora pairof $N$ by -N com plex nonsymm etric $m$ atrices $(A, B)$, the generalized eigenvalues, the generalized com plex Schur form ( $\mathrm{S}, \mathrm{T}$ ), and optionally leftand/or right Schurvectors (VSL and V SR)

## SYNOPSIS

```
SU BROUT\mathbb{NE CGGES (JOBVSL,JOBVSR,SORT,SELCTG N,A,LDA,B,LDB,}
    SD IM,ALPHA,BETA,VSL,LDVSL,VSR,LDVSR,W ORK,LW ORK,RW ORK,
    BWORK,\mathbb{NFO)}
CHARACTER * 1 JOBVSL,JOBVSR,SORT
COM PLEX A (LDA,*),B (LDB,*),ALPHA (*),BETA (*),VSL (LDVSL,*),
VSR (LDVSR,*),W ORK (*)
INTEGERN,LDA,LDB,SD IM,LDVSL,LDVSR,LW ORK,INFO
LOGICAL SELCTG
LOG ICAL BW ORK (*)
REAL RW ORK (*)
SU BROUT\mathbb{NE CGGES_64 (JOBVSL,JOBVSR,SORT,SELCTG,N,A,LDA,B,LD B,}
    SD IM,ALPHA,BETA,VSL,LDVSL,VSR,LDVSR,W ORK,LW ORK,RW ORK,
    BW ORK,\mathbb{NFO)}
CHARACTER * 1 JOBVSL,JOBVSR,SORT
COM PLEX A (LDA,*),B (LDB,*),ALPHA (*),BETA (*),VSL (LDVSL,*),
VSR (LDVSR,*),W ORK (*)
```



```
LOGICAL*8 SELCTG
LOG ICAL*8 BW ORK (*)
REAL RW ORK (*)
```

SU BROUTINE GGES (OOBVSL, JOBVSR, SORT, [SELCTG], $\mathbb{N}], A,[L D A], B,[L D B]$, SD $\mathbb{I}$, ALPHA, BETA, VSL, [LDVSL],VSR, [LDVSR], [W ORK], [LW ORK], [RW ORK], [BW ORK], [ $\mathbb{N F O}$ ])

CHARACTER (LEN =1) :: JOBV SL, JO BV SR , SO RT
COM PLEX,D $\mathbb{I M} \operatorname{ENSION~(:)~::ALPHA,BETA,W~ORK~}$
COMPLEX,D $\mathbb{M}$ ENSION (:,:) ::A,B,VSL,VSR
$\mathbb{N}$ TEGER :: N,LDA, LD B, SD $\mathbb{I M}, L D V S L, L D V S R, L W O R K, \mathbb{N} F O$
LOG ICAL :: SELCTG
LOG ICAL,D IM ENSION (:) ::BW ORK
REAL,D $\mathbb{M}$ ENSION (:) ::RW ORK

SU BROUTINE G GES_64 (JO BV SL, JO BV SR , SORT, [SELCTG ], $\mathbb{N}$ ], A, [LDA ],B,
 [LW ORK], $\mathbb{R W}$ ORK ], [BW ORK], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1) :: JOBVSL, JOBVSR , SORT
COMPLEX,D $\mathbb{M}$ ENSION (:) ::ALPHA,BETA,W ORK
COM PLEX,D IM ENSION (:,:) ::A, B,VSL,VSR
$\mathbb{N}$ TEGER (8) :: N, LD A, LD B , SD $\mathbb{I M}, L D V S L, L D V S R, L W O R K, \mathbb{N} F O$
LOG ICAL (8) :: SELCTG
LOG ICAL (8),D IM ENSION (:) ::BW ORK
REAL,D $\mathbb{I M}$ ENSION (:) ::RW ORK

## C INTERFACE

\#include <sunperfh>
void cgges(char jobvsl, char jobvss, char sort, int(*selctg) (com plex,com plex), intn, com plex *a, int lda, com plex *b, int ldb, int *sdim, com plex *alpha, com plex *beta, com plex *vsl, int ldvsl, com plex *vsr, int ldvsr, int*info);
void cgges_64 (char jobvsl, char jojbvsr, char sort, long (*selctg) (com plex,com plex), long n, com plex *a, long lda, com plex *b, long ldb, long *sdim, com plex *alpha, com plex *beta, com plex *vsl, long ldvsl, com plex *vsr, long ldvss, long *info);

## PURPOSE

cgges com putes for a pair of N -by-N com plex nonsym m etric $m$ atrices ( $A, B$ ), the generalized eigenvalues, the generalized com plex Schur form ( $\mathrm{S}, \mathrm{T}$ ), and optionally left and/or right Schur vectors (NSL and VSR). This gives the generalized Schur factorization

$$
(A, B)=(N S L) * S * N S R) * * H,(N S L) * T * N S R) * * H)
$$

where ( NSR$)^{* *} \mathrm{H}$ is the conjugate-transpose of SR .
Optionally, it also orders the eigenvalues so that a selected cluster of eigenvalues appears in the leading diagonalblocks of the upper triangularm atrix $S$ and the upper triangularm atrix T. The leading colum ns of V SL and V SR then form an unitary basis for the comesponding left and right eigenspaces (deflating subspaces).
(Ifonly the generalized eigenvalues are needed, use the driverC G G EV instead, which is faster.)

A generalized eigenvalue fora pairofm atrices $(A, B)$ is a scalar w or a ratio alphałbeta $=\mathrm{w}$, such that $A-\mathrm{w} * \mathrm{~B}$ is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation for beta $=0$, and even forboth being zero.
A pairofm atrices ( $\mathrm{S}, \mathrm{T}$ ) is in generalized complex Schur form if S and T are uppertriangularand, in addition, the diagonalelem ents of $T$ are non-negative realnum bers.

## ARGUMENTS

JO BV SL (input)
$=\mathrm{N}^{\prime}:$ do notcom pute the leftSchurvectors;
$=\mathrm{V}$ : com pute the leftSchurvectors.

JO BV SR (input)
$=\mathrm{N}$ ': do notcom pute the rightSchurvectors;
$=\mathrm{V}$ : com pute the rightSchurvectors.
SORT (input)
Specifies w hether ornot to order the eigenvalues on the diagonal of the generalized Schur form . =
N ': E igenvalues are notordered;
= S': Eigenvalues are ordered (see SELCTG).

SELCTG (input)
SELCTG mustbe declaredEXTERNAL in the calling subroutine. If SORT = N',SELCTG is notreferenced. IfSORT = S',SELCTG is used to select eigenvalues to sort to the top leftof the Schur form. A n eigenvalue A LPHA ( $)$ ( BETA ( $\mathcal{O}$ ) is selected ifSELCTG (ALPHA ( $\mathcal{j}$, BETA ( $\mathbf{j}$ ) is true.
$N$ ote that a selected com plex eigenvalue $m$ ay no longer satisfy SELCTG (ALPHA ( $\mathcal{j}$,BETA ( $\mathcal{j}$ ) = TRUE. afterordering, since ordering $m$ ay change the
value of com plex eigenvalues (especially if the eigenvalue is ill-conditioned), in this case $\mathbb{N N}$ FO is set to $\mathrm{N}+2$ (See $\mathbb{N}$ FO below ).

N (input) The order of the m atrioes A, B, V SL, and V SR. N $>=0$.

A (input/output)
O $n$ entry, the first of the pair of $m$ atrices. On exit, A has been overw ritten by its generalized Schur form $S$.

LD A (input)
The leading dim ension ofA. LD A $>=\max (1, N)$.
B (input/output)
O n entry, the second of the pair ofm atrices. On exit, B has been overw ritten by its generalized Schur form T.

LD B (input)
The leading din ension ofB. LD $B>=m a x(1, N)$.
SD $\mathbb{I M}$ (output)
If $S O R T=N^{\prime}, S D \mathbb{I M}=0$. IfSORT $=S^{\prime}, S D \mathbb{M}=$ num ber of eigenvalues (after sorting) forw hich SELCTG is true.

ALPHA (output)
On exit, A LPHA ( $\mathfrak{j}$ ) BETA ( $\mathcal{j}$ ) $\dot{j} 1, \ldots, N$, w illbe the generalized eigenvalues. ALPHA ()$, j 1, \ldots, N$ and $\operatorname{BETA}(\mathcal{D}, \dot{于} 1, \ldots, N$ are the diagonals of the com plex Schur form (A,B) output by CGGES. The BETA ( $)$ w illbe non-negative real.

N ote: the quotients A LPHA ( $)$ ) BETA ( ) may easily over- orunderflow, and BETA ( $\mathcal{)}$ m ay even be zero. Thus, the user should avoid naively com puting the ratio alpha/beta. H ow ever, A LPH A w illbe alw ays less than and usually com parable w ith norm (A) in m agnitude, and BETA alw ays less than and usually com parable w ith norm (B).

BETA (output)
See description of A LPHA.
VSL (input)
If $J 0 B V S L=V$ ',VSL w illcontain the left Schur
vectors. N ot referenced if $\mathrm{JO} \mathrm{BVSL}=\mathrm{N}$ '.

The leading dim ension of the $m$ atrix V SL. LD V SL >= 1 , and if $J 0$ BV SL $=V$ ', LDV SL $>=N$.

VSR (input)
If $J 0$ BV SR $=V$ ', V SR w illcontain the right Schur vectors. N ot referenced if JO BV SR $=\mathrm{N}$ '.

LDV SR (input)
The leading dim ension of the $m$ atrix $V$ SR .LD V SR >= 1 , and if $J O B V S R=V$ ', LD V SR $>=N$.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the array W ORK. LW ORK >= $\max (1,2 * \mathrm{~N})$. Forgood perform ance, LW ORK m ustgenerally be larger.

IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

RW ORK (w orkspace)
dim ension ( $8 * \mathrm{~N}$ )
BW ORK (w orkspace)
dim ension $(\mathbb{N})$ N ot referenced ifSORT $=N^{\prime}$.
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
< 0: if $\mathbb{N}$ FO = -i, the i-th argum ent had an illegalvalue.
$=1, \ldots, \mathrm{~N}$ : The Q Z teration failed. (A , B) are not in Schur form, butA LPHA ( $)$ and BETA ( $)$ ) should be comect for $\mathcal{j} \mathbb{N} F O+1, \ldots, N .>N$ : $=\mathrm{N}+1$ : other than $\mathrm{Q} Z$ teration failed in CHGEQ Z
$=\mathrm{N}+2$ : after reordering, roundoff changed values of some complex eigenvalues so that leading eigenvalues in the $G$ eneralized Schur form no longer satisfy SELCTG=TRUE. This could also be caused due to scaling. $=\mathrm{N}+3$ : reordering falied in CTGSEN.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cggesx - com pute for a pair of N -by -N com plex nonsym m etric $m$ atrices $(A, B)$, the generalized eigenvalues, the com plex Schurform ( $\mathrm{S}, \mathrm{T}$ ),

## SYNOPSIS

```
SUBROUT\mathbb{NE CGGESX (JO BV SL,JOBVSR,SORT,SELCTG,SENSE,N,A,LDA,B,}
    LDB,SD IM,ALPHA,BETA,VSL,LDVSL,VSR,LDVSR,RCONDE,RCONDV,
    W ORK,LW ORK,RW ORK,\mathbb{N ORK,LIN ORK,BW ORK,\mathbb{NFO)}}\mathbf{N}\mathrm{ (IN}
```

CHARACTER * 1 JOBVSL, JOBVSR, SORT, SEN SE
COM PLEX A (LDA,*), B (LDB,*),ALPHA (*), BETA (*), VSL (LDVSL, $\left.{ }^{\star}\right)$,
VSR (LDVSR, $\left.{ }^{\star}\right)$, W ORK (*)
$\mathbb{N}$ TEGER N,LDA,LDB,SD $\mathbb{I M}, L D V S L, L D V S R, L W O R K, L \mathbb{I N} O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R \mathbb{I N}$ ORK (*)
LOG ICAL SELCTG
LO G ICAL BW ORK (*)
REALRCONDE (*), RCONDV (*), RW ORK (*)
SU BROUTINE CG GESX_64 (JO BV SL, JO BV SR, SORT, SELCTG, SEN SE, N, A, LD A,
B,LDB,SD $\mathbb{I}, A L P H A, B E T A, V S L, L D V S L, V S R, L D V S R, R C O N D E$,
RCONDV,W ORK,LW ORK,RW ORK, IW ORK,LIW ORK,BW ORK, $\mathbb{N} F O$ )
CHARACTER * 1 JOBVSL, JOBVSR, SORT, SENSE
COM PLEX A (LDA, *), B (LDB,*),ALPHA (*), BETA (*), VSL (LDVSL, $\left.{ }^{\star}\right)$,
VSR (LDVSR, *), W ORK (*)
$\mathbb{N}$ TEGER*8 N,LDA,LD B,SD $\mathbb{I}$,LDVSL, LDVSR, LW ORK, LIV ORK,
$\mathbb{N F O}$
$\mathbb{N}$ TEGER * $8 \mathbb{I N}$ ORK ( ${ }^{*}$ )
LOG ICAL*8 SELCTG
LOG ICAL*8BW ORK (*)
REALRCONDE (*), RCONDV (*),RWORK (*)

## F95 INTERFACE

SU BROUTINE G GESX (JO BV SL, JO BV SR, SORT, [SELCTG ], SENSE, $\mathbb{N}]$ ], A, [LD A ], $B,[L D B], S D \mathbb{M}, A L P H A, B E T A, V S L,[L D V S L], V S R,[L D V S R], R C O N D E$,
 [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) :: JOBVSL, JOBVSR,SORT,SENSE
COMPLEX,D $\mathbb{M}$ ENSION (:) ::ALPHA,BETA,W ORK
COM PLEX,D IM ENSION (:,:) ::A,B,VSL,VSR
$\mathbb{N} T E G E R:: N, L D A, L D B, S D \mathbb{M}, L D V S L, L D V S R, L W O R K, L \mathbb{N} O R K$, $\mathbb{N}$ FO
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: $\mathbb{I N}$ ORK
LOG ICAL :: SELCTG
LO G ICAL,D IM ENSION (:) ::BW ORK
REAL,D $\mathbb{M}$ ENSION (:) ::RCONDE,RCONDV,RW ORK
SU BROUTINE G GESX_64 (JOBV SL, JO BV SR, SORT, [SELCTG ], SEN SE, $\mathbb{N}$ ], A, [LDA ], $B,[L D B], S D \mathbb{I}, A L P H A, B E T A, V S L,[L D V S L], V S R,[L D V S R], R C O N D E$, RCONDV, [W ORK ], [LW ORK], RW ORK ], [IW ORK], [LIW ORK], [BW ORK ], [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1) :: JOBVSL, JOBVSR,SORT,SENSE
COMPLEX,D $\mathbb{M}$ ENSION (:) ::ALPHA,BETA,W ORK
COM PLEX,D IM ENSION (:,:) ::A,B,VSL,VSR
$\mathbb{N}$ TEGER (8) :: N, LDA, LDB, SD $\mathbb{M}, ~ L D V S L, ~ L D V S R, ~ L W ~ O R K, ~$
LIN ORK, $\mathbb{N} F O$
$\mathbb{N}$ TEGER (8),D $\mathbb{M}$ ENSION (:) :: $\mathbb{I N}$ ORK
LO G ICAL (8) :: SELCTG
LOG ICAL (8),D IM ENSION (:) ::BW ORK
REAL,D $\mathbb{M}$ ENSION (:) ::RCONDE,RCONDV,RWORK

## C INTERFACE

\#include <sunperfh>
void cggesx (char jobvsl, char jobvsr, char sort, int(*selctg) (com plex,com plex), charsense, intn, com plex *a, int lda, com plex $*$, int ldb , int *sdim, complex *alpha, complex *beta, complex *vsl, int ldvsl, com plex *vsr, int ldvsr, float *rconde, float *rcondv, int *info);
void cggesx_64 (char jobvsl, char jobvssr, char sort, long (*selctg) (com plex,com plex), char sense, long n, com plex *a, long lda, com plex *b, long ldb, long *sdim , com plex *alpha, com plex *beta, com plex *vsl, long ldvsl, com plex *vsr, long ldvss, float * rconde, float *rcondv, long *info);

## PURPOSE

cggesx com putes for a pair of N łoy -N com plex nonsymm etric $m$ atrices ( $A, B$ ), the generalized eigenvalues, the com plex Schurform ( $\mathrm{S}, \mathrm{T}$ ), and, optionally, the left and/or right $m$ atrices of Schur vectors (VSL andVSR). This gives the generalized Schur factorization $A, B)=(N S L) S(V S R) * * H$, (VSL) T (VSR)**H )
where $(\mathrm{VSR}){ }^{\star *} \mathrm{H}$ is the conjugate-transpose of SR .

Optionally, it also orders the eigenvalues so that a selected cluster of eigenvalues appears in the leading diagonalblocks of the uppertriangularm atrix $S$ and the upper triangular m atrix T; com putes a reciprocalcondition num ber for the average of the selected eigenvalues (RCONDE); and com putes a reciprocal condition num ber for the rightand left deflating subspaces corresponding to the selected eigenvalues (RCONDV). The leading colum ns ofV SL and V SR then form an orthonorm albasis for the corresponding left and righteigenspaces (deflating subspaces).

A generalized eigenvalue for a pair ofm atrices ( $A, B$ ) is a scalar $w$ or a ratio alpha/beta $=w$, such that $A-w * B$ is singular. It is usually represented as the pair (alpharbeta), as there is a reasonable interpretation for beta=0 or forboth being zero.

A pairofm atrices ( $\mathrm{S}, \mathrm{T}$ ) is in generalized complex Schur form if $T$ is uppertriangularw ith non-negative diagonal and $S$ is upper triangular.

## ARGUMENTS

JOBVSL (input)
$=\mathrm{N}$ ': do notcom pute the leftSchurvectors;
$=\mathrm{V}$ ': com pute the leftSchurvectors.

JO BV SR (input)
$=\mathrm{N}$ ': do notcom pute the rightSchurvectors;
$=\mathrm{V}^{\prime}$ : com pute the rightSchurvectors.

SORT (input)
Specifies w hether or not to order the eigenvalues
on the diagonal of the generalized Schur form . =
N ': Eigenvahues are notordered;
$=S$ ': Eigenvalues are ordered (see SELCTG).

SELCTG mustbe declaredEXTERNAL in the calling subroutine. If SORT $=N^{\prime}$ ',SELCTG is notreferenced. IfSORT = S', SELCTG is used to select eigenvalues to sort to the top leftof the Schur form . N ote that a selected com plex eigenvaluem ay no longer satisfy SELCTG (ALPHA ( ) ,BETA ( $\mathcal{j}$ ) $=$ .TRUE.afterordering, since ordering $m$ ay change the value of com plex eigenvalues (especially if the eigenvalue is ill-conditioned), in this case $\mathbb{N} F O$ is set to $\mathrm{N}+3 \mathrm{sec} \mathbb{I N}$ FO below ).

SEN SE (input)
D eterm ines which reciprocalcondition num bers are com puted. = N ': N one are com puted;
$=\mathrm{E}$ ':Com puted for average of selected eigenvalues only;
$=\mathrm{V}^{\prime}: \mathrm{C}$ om puted for selected deflating subspaces only;
$=\mathrm{B}^{\prime}:$ Com puted forboth. IfSENSE = $\mathrm{E}^{\prime}, \mathrm{V}$ ', or $B^{\prime}$ 'SORT m ustequal $S^{\prime}$.

N (input) The order of the m atrioes A, B, V SL, and V SR. N $>=0$.

A (input/output)
O $n$ entry, the first of the pair of $m$ atrices. On
exit, A has been overw ritten by its generalized Schur form $S$.

LD A (input)
The leading dim ension ofA. LD A $>=\max (1, N)$.
B (input/output)
On entry, the second of the pair ofm atrices. On
exit, B has been overw rilten by its generalized Schur form T.

LD B (input)
The leading dim ension ofB. LD B $>=\max (1, N)$.
SD $\mathbb{I M}$ (output)
If $S O R T=N ', S D \mathbb{M}=0$. IfSORT $=S^{\prime}, S D \mathbb{M}=$
num ber of eigenvalues (aftersorting) forw hich
SELCTG is true.
ALPHA (output)
On exit, ALPHA ( ) BETA ( $)$, $\ddagger 1, \ldots, N$, w illbe the generalized eigenvalues. ALPHA ( 1 ) and BETA ( $\mathcal{j}, \dot{j}=1, \ldots, N$ are the diagonals of the complex Schur form ( $\mathrm{S}, \mathrm{T}$ ). BETA ( $\mathcal{J}$ ) w ill be non-
negative real.

N ote: the quotients A LPHA ( 7 ) BETA ( ) may easily over- orunderflow, and BETA ( $)$ m ay even be zero. Thus, the user should avoid naively com puting the ratio alpha/beta. H ow ever, A LPH A w illbe alw ays less than and usually com parable w ith norm (A) in m agnitude, and BETA alw ays less than and usually com parable $w$ th norm (B).
BETA (output)
See description of A LPH A.

VSL (input)
If 30 BVSL $=V$ ',VSL willcontain the left Schur vectors. N ot referenced if $\mathrm{JOBVSL}=\mathrm{N}$ '.

LD V SL (input)
The leading dim ension of the $m$ atrix VSL. LDVSL
$>=1$, and if $\mathrm{JOBVSL}=\mathrm{V}$ ', LDVSL $>=\mathrm{N}$.
V SR (input)
If Jo BV SR = V', V SR w illcontain the right Schur vectors. N ot referenced if $\mathrm{JO} \mathrm{BV} \mathrm{SR}=\mathrm{N}$ '.

LDV SR (input)
The leading dim ension of the $m$ atrix $V$ SR .LD V SR >= 1 , and if $J 0 B V S R=V$ ', LDVSR $>=N$.

RCONDE (output)
IfSENSE = E'or B', RCONDE (1) and RCONDE (2)
contain the reciprocalcondition num bers for the average of the selected eigenvalues. N ot referenced if SENSE = N'or V'.

RCONDV (output)
If SENSE = V 'or B', RCONDV (1) and RCONDV (2)
contain the reciprocal condition num ber for the selected deflating subspaces. N ot referenced if SENSE = N 'or E'.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the aray $W$ ORK. LW ORK >= 2*N. If SENSE = E', V', or B',LW ORK $>=$ MAX $(2 \star N$, $2 * S D \mathbb{M} *(\mathbb{N}-S D \mathbb{I})$ ).

RW ORK (w orkspace)
dim ension ( $8 * N$ ) Realw orkspace.
IV ORK (w orkspace/output)
N otreferenced if $\mathrm{SEN} \mathrm{SE}=\mathrm{N}^{\prime}$. On exit, if $\mathbb{N} F O=$ $0, \mathbb{I V}$ ORK (1) retums the optim alL $\mathbb{I N}$ ORK.

LIW ORK (input)
The dim ension of the amay $W$ ORK.LIV ORK $>=N+2$.

BW ORK (w orkspace)
dim ension $(\mathbb{N}) N$ ot referenced if $S O R T=N^{\prime}$.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the i-th argum enthad an illegalvalue.
$=1, \ldots, N$ : The Q Z iteration failed. ( $\mathrm{A}, \mathrm{B}$ ) are not in Schur form , butA LPHA ( 1 ) and BETA ( 1 ) should
be comect for $\mathcal{F} \mathbb{N F O}+1, \ldots, N .>N:=N+1$ : other than Q Z iteration failed in C H G EQ Z
$=\mathrm{N}+2$ : after reordering, roundoff changed values of som e complex eigenvahues so that leading eigenvalues in the Generalized Schur form no longer satisfy $\mathrm{SELCTG}=$ TRUE . This could also be caused due to scaling. $=\mathrm{N}+3$ : reordering failed in CTGSEN .

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cggev - com pute for a pair of N -by -N com plex nonsym $m$ etric $m$ atrices $(A, B)$, the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors

## SYNOPSIS

```
SUBROUT\mathbb{NECGGEV (JOBVL,JOBVR,N,A,LDA,B,LDB,ALPHA,BETA,VL,}
    LDVL,VR,LDVR,W ORK,LW ORK,RW ORK,\mathbb{NFO)}
CHARACTER * 1 JOBVL,JOBVR
COM PLEX A (LDA,*),B (LDB,*),ALPHA (*), BETA (*), VL (LDVL,*),
VR (LDVR,*),W ORK (*)
INTEGERN,LDA,LDB,LDVL,LDVR,LW ORK,INFO
REAL RW ORK (*)
SU BROUT\mathbb{NECGGEV_64 (JO BVL,JOBVR,N,A,LDA , B ,LD B,ALPHA,BETA,VL,}
    LDVL,VR,LDVR,W ORK,LW ORK,RW ORK,INFO )
CHARACTER * 1 JOBVL,JOBVR
COM PLEX A (LDA ,*),B (LD B,*),ALPHA (*), BETA (*), VL (LDVL,*),
VR (LDVR,*),W ORK (*)
\mathbb{NTEGER*8N,LDA,LDB,LDVL,LDVR,LW ORK,INFO}
REAL RW ORK (*)
F95 INTERFACE
    SU BROUT\mathbb{NE GGEV (JOBVL,JOBVR, N ],A,[LDA ],B,[LDB ],ALPHA,BETA,}
        VL,[LDVL],VR,[LDVR], [W ORK ], [LW ORK], RW ORK ], [NFO ])
    CHARACTER (LEN=1)::JOBVL,JOBVR
    COMPLEX,D IM ENSION (:) ::ALPHA,BETA,W ORK
    COM PLEX,D IM ENSION (:,:)::A,B,VL,VR
    \mathbb{NTEGER ::N,LDA,LDB,LDVL,LDVR,LW ORK,INFO}
```

SU BROUT INE G GEV_64 (OBVL, OBVR, $\mathbb{N}], A,[L D A], B,[L D B], A L P H A$, $B E T A, V L,[L D V L], V R,[L D V R],[W O R K],[L W O R K],[R W O R K],[\mathbb{N F O}])$

CHARACTER ( $L E N=1$ ) :: JOBVL, JOBVR
COM PLEX,D $\mathbb{M}$ ENSION (:) ::ALPHA,BETA,W ORK
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : A , B , VL, VR
$\mathbb{N} \operatorname{TEGER}(8):: N$, LDA $, L D B, L D V L, L D V R, L W O R K, \mathbb{N} F O$
REAL,D $\mathbb{I}$ ENSION (:) ::RW ORK

## C INTERFACE

\#include <sunperfh>
void oggev (char jंbvl, char jobvr, intn, com plex *a, int lda, com plex *b, int ldb, com plex *alpha, com plex *beta, com plex *vl, int ldvl, com plex *Vr, int ldvr, int*info);
void cggev_64 (char j̣bvl, char jobvr, long n, com plex *a, long lda, com plex *b, long ldb, com plex *alpha, com plex *beta, com plex *vl, long ldvl, complex *vr, long ldvr, long *info);

## PURPOSE

cggev com putes for a pair of N łoy -N com plex nonsymm etric $m$ atrices $(A, B)$, the generalized eigenvalues, and optionally, the leftand/or rightgeneralized eigenvectors.

A generalized eigenvalue for a pair ofm atrices $(A, B)$ is a scalar lam bda or a ratio alpha/beta = lam bda, such thatA lam boda*B is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation for beta $=0$, and even forboth being zero.

The rightgeneralized eigenvectorv ( $\mathcal{I}$ ) comesponding to the generalized eigenvalue lam boda ( $\mathcal{I}$ ) of $(A, B)$ satisfies

$$
A * v(\mathcal{J})=\operatorname{lam} \operatorname{bda}(\mathcal{I}) * B * v(\mathcal{I})
$$

The leftgeneralized eigenvector u ( 7 ) comesponding to the generalized eigenvalues lam bda ( $\mathcal{I}$ ) of ( $\mathrm{A}, \mathrm{B}$ ) satisfies

$$
u(\mathfrak{j}) \star \star H * A=\operatorname{lam} \operatorname{bda}(\mathfrak{j}) * u(j) \star * H * B
$$

where $u(\mathcal{j}) * * H$ is the conjugate-transpose ofu (1) .

## ARGUMENTS

JO BVL (input)
$=\mathrm{N}$ ': do notcom pute the leftgeneralized eigenvectors;
$=\mathrm{V}$ ': com pute the leftgeneralized eigenvectors.
$J O B V R$ (input)
$=\mathrm{N}$ : : do not com pute the right generalized eigenvectors;
$=\mathrm{V}$ ': com pute the right generalized eigenvectors.

N (input) The order of the m atriges $\mathrm{A}, \mathrm{B}, \mathrm{VL}$, and $\mathrm{VR} . \mathrm{N}>=$ 0.

A (input/output)
On entry, them atrix $A$ in the pair $(A, B)$. On exit, A has been overw rilten.

LD A (input)
The leading dim ension ofA. LD A $>=\max (1, N)$.

B (input/output)
On entry, them atrix $B$ in the pair ( $A, B$ ). On exit, B has been overw ritten.

LD B (input)
The leading dim ension ofB. LD B >=max $(1, N)$.

ALPHA (output)
On exit, ALPHA ( ) BETA ( $)$, $1, \ldots, N$, w illbe the generalized eigenvalues.

N ote: the quotients A LPHA ( 7 ) $B E T A(7) \mathrm{m}$ ay easily over- orunderflow, and BETA ( 1 ) m ay even be zero. Thus, the user should avoid naively com puting the ratio alphaßbeta. H ow ever, A LPH A w illbe alw ays less than and usually com parable w ith norm (A) in $m$ agnitude, and BETA alw ays less than and usually com parable w ith norm (B).

BETA (output)
See description of A LPH A.

VL (input)
If $\mathrm{OBV} \mathrm{B}=\mathrm{V}$ ', the leftgeneralized eigenvectors
$u(7)$ are stored one after another in the colum ns ofVL, in the same order as their eigenvalues. Each eigenvector will be scaled so the largest com ponentw illhave abs(real part) + abs(im ag.
part $=1$. N ot referenced if $J 0 B V L=N^{\prime}$.

LDVL (input)
The leading dim ension of the $m$ atrix $V L$. LD V L $>=1$, and if $\mathrm{JOBVL}=\mathrm{V}$ ', LDVL $>=\mathrm{N}$.

VR (input)
If JO BVR = V ', the right generalized eigenvectors $v(j)$ are stored one after another in the colum ns ofVR, in the sam e order as their eigenvalues. Each eigenvector will be scaled so the largest com ponentw illhave abs(real part) + abs(m ag. part) $=1 . \operatorname{N}$ ot referenced if $J 0 B V R=N '$.

LDVR (input)
The leading dim ension of the $m$ atrix $V R$.LD $V R>=1$, and if $\mathrm{JOBVR}=\mathrm{V}$ ', LDVR $>=\mathrm{N}$.

W ORK (w orkspace)
On exit, if $\mathbb{N F F O}=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the anay $W$ ORK. LW ORK >= max ( $1,2 * \mathrm{~N}$ ). For good perform ance, LW ORK m ustgenerally be larger.

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of theW ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

RW ORK (w orkspace)
dim ension ( $8 * \mathrm{~N}$ )
$\mathbb{N F O}$ (output)
= 0: successfulexit
< 0: if $\mathbb{N}$ FO = -i, the i-th argum enthad an illegalvalue.
$=1, \ldots, N$ : The Q Z iteration failed. No eigenvectors have been calculated, but A LPHA ( $\mathcal{j}$ ) and BETA ( $\mathcal{j}$ ) should be comectfor $\mathcal{j} \mathbb{N} F O+1, \ldots, N . \quad>$ $N$ : $=N+1$ : other then $Q Z$ iteration failed in SHGEQZ,
$=N+2$ : error retum from STGEVC.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cggevx - com pute fora pairof $N$-by- N com plex nonsym m etric $m$ atrices ( $A, B$ ) the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors

## SYNOPSIS

SUBROUTINECGGEVX (BALANC, JOBVL, JOBVR,SENSE,N,A,LDA,B,LDB,
A LPHA, BETA, VL, LDVL, VR,LDVR, $\mathbb{L O}, \mathbb{H} I, L S C A L E, R S C A L E, A B N R M$, BBNRM,RCONDE,RCONDV,WORK,LWORK,RWORK, IN ORK,BWORK, $\mathbb{N} F O$ )

CHARACTER * 1 BALANC, JOBVL, JOBVR, SENSE
COM PLEX A (LDA,*), B (LD B ,*), ALPHA (*), BETA (*), VL (LDVL,*),
VR (LDVR,*), W ORK (*)
$\mathbb{N} T E G E R N, L D A, L D B, L D V L, L D V R, \mathbb{I} O, \mathbb{H} I, L W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R \mathbb{I N}$ ORK (*)
LOG ICALBW ORK (*)
REAL ABNRM,BBNRM
REAL LSCALE (*), RSCALE (*), RCONDE (*),RCONDV (*),RWORK (*)
SU BROUTINE CG GEVX_64 (BALANC, JOBVL, JOBVR, SENSE, N, A, LDA , B , LD B, A LPHA, BETA,VL,LDVL,VR,LDVR, $\mathbb{L} O, \mathbb{H} I, L S C A L E, R S C A L E, A B N R M$, BBNRM,RCONDE,RCONDV,WORK,LWORK,RWORK, IN ORK,BWORK, $\mathbb{N} F O$ )

CHARACTER * 1 BALANC, JOBVL, JOBVR,SENSE
COM PLEX A (LDA , *), B (LDB,*), ALPHA (*), BETA (*), VL (LDVL,*),
VR (LDVR,*), W ORK (*)
$\mathbb{N} T E G E R * 8 N, L D A, L D B, L D V L, L D V R, \mathbb{L} O, \mathbb{H} I, L W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I N}$ ORK (*)
LOG ICAL*8BW ORK (*)
REAL ABNRM,BBNRM
REAL LSCALE (*), RSCALE (*), RCONDE (*), RCONDV (*),RW ORK (*)

## F95 INTERFACE

SU BROUTINE G GEVX (BALANC, JOBVL, JOBVR, SENSE, $\mathbb{N}$ ],A, [LDA ], B, [LD B], A LPHA,BETA,VL, [LDVL],VR, [LDVR], $H O, \mathbb{H} I, L S C A L E, R S C A L E$, ABNRM, BBNRM,RCONDE,RCONDV, [W ORK], [LW ORK], RW ORK], [IW ORK], [BW ORK], [ $\mathbb{N F O}]$ )

CHARACTER (LEN =1) ::BALANC, JOBVL, JOBVR, SEN SE
COMPLEX,D $\mathbb{M}$ ENSION (:) ::ALPHA,BETA,W ORK
COM PLEX,D $\mathbb{I M}$ ENSION (: : : : : A, B, VL,VR
$\mathbb{N} T E G E R:: N, L D A, L D B, L D V L, L D V R, \mathbb{H}, \mathbb{H} I, L W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{I N}$ ORK
LOG ICAL,D IM ENSION (:) ::BW ORK
REAL ::ABNRM,BBNRM
REAL,D $\mathbb{I}$ ENSION (:) ::LSCALE,RSCALE,RCONDE,RCONDV,RW ORK SU BROUTINE GGEVX_64 (BALANC, JOBVL, JOBVR,SEN SE, $\mathbb{N}$ ], A, [LDA ],B, [LDB],A LPHA,BETA, VL, [LDVL],VR, [LDVR], $\amalg O, \mathbb{H} I, L S C A L E$, RSCALE, ABNRM, BBNRM, RCONDE,RCONDV, [W ORK ], [LW ORK], [RW ORK ], [IW ORK], [BW ORK], [ $\mathbb{N F O}$ ])

CHARACTER (LEN =1) ::BALANC, JOBVL, JOBVR, SEN SE
COMPLEX,D $\mathbb{M}$ ENSION (:) ::ALPHA,BETA,W ORK
COM PLEX,D $\mathbb{I M}$ ENSION (: : : : : A, B, VL,VR
$\mathbb{N} T E G E R(8):: N, L D A, L D B, L D V L, L D V R, \mathbb{L} O, \mathbb{H} I, L W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{I N}$ ORK
LOG ICAL (8),D IM ENSION (:) ::BW ORK
REAL ::ABNRM,BBNRM
REAL,D $\mathbb{M}$ ENSION (:) ::LSCALE,RSCALE,RCONDE,RCONDV,RW ORK

## C INTERFACE

\#include <sunperfh>
void oggevx (charbalanc, char jंbvl, char jobvr, char sense, int $n$, com plex *a, int lda, com plex *b, int ldb, com plex *alpha, com plex *beta, com plex *vl, int ldvl, com plex *vr, int ldvr, int *ilo, int *ihi, float*lscale, float*rscale, float*abnm, float *bonm, , float * rconde, float *rcondv, int *info);
void oggevx_64 (charbalanc, char jobvl, char jobvr, char sense, long n, com plex *a, long lda, com plex *b, long ldb, com plex *alpha, com plex *beta, com plex *vl, long ldvl, com plex *vr, long ldvr, long *ilo, long *ihi, float *lscale, float *rscale, float *abnım, float *bbnm, float *rconde, float *rcondv, long *info);

## PURPOSE

cggevx com putes for a pair of $N$-by -N com plex nonsym $m$ etric $m$ atrices $(A, B)$ the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors.

O ptionally, italso com putes a balancing transform ation to im prove the conditioning of the eigenvalues and eigenvectors ( $\mathbb{L} O, \mathbb{H} I, L S C A L E, R S C A L E, A B N R M$, and BBNRM), reciprocal condition num bers for the eigenvalues (RCONDE), and reciprocalcondition num bers for the righteigenvectors (RCONDV).

A generalized eigenvalue fora pairofm atrices $(A, B)$ is a scalar lam bda or a ratio alpha/beta = lam bda, such thatA lam bda*B is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation for beta= 0 , and even forboth being zero.
The righteigenvectorv (i) comesponding to the eigenvalue lam bda ( $)$ ) of $(A, B)$ satisfies

$$
A * v(\mathcal{j})=\operatorname{lam} \operatorname{bda}(\mathcal{j}) * B * v(\mathcal{j}) .
$$

The lefteigenvectoru ( $\mathcal{j}$ ) corresponding to the eigenvalue lam boda ( $)$ ) of $(A, B)$ satisfies

$$
u(j) \star * H * A=\operatorname{lam} \cdot \operatorname{bda}(j) * u(j) * * H * B .
$$

where $u(\mathcal{J}) * * H$ is the conjugate transpose of $u(\mathcal{)}$.

## ARGUMENTS

BALANC (input)
Specifies the balance option to be perform ed:
= N ': do notdiagonally scale orperm ute;
$=\mathrm{P}^{\prime}$ : perm ute only;
= S': scale only;
= B ': both perm ute and scale. C om puted reciprocal condition num bers $\mathrm{w} i l l$ be for the $m$ atrices afterperm uting and/orbalancing. Perm uting does not change condition numbers (in exactarithm etic), butbalancing does.

JO BVL (input)
= N ': do not com pute the left generalized eigenvectors;
$=\mathrm{V}$ ': com pute the left generalized eigenvectors.
JO BVR (input)
$=N^{\prime}$ : do not com pute the right generalized
eigenvectors;
$=\mathrm{V}$ : com pute the right generalized eigenvec-
tors.

D eterm ines which reciprocal condition num bers are com puted. = N ': none are com puted;
$=\mathrm{E}$ ': com puted foreigenvalues only;
= V ': com puted foreigenvectors only;
$=B$ ': com puted foreigenvalues and eigenvectors.
N (input) The order of the $m$ atrices $\mathrm{A}, \mathrm{B}, \mathrm{VL}$, and $V \mathrm{R} . \mathrm{N}>=$ 0.

A (input/output)
On entry, them atrix $A$ in the pair $(A, B)$. On exit, $A$ has been overw rilten. If $J O B V L=V$ 'or Jo BVR=V 'orboth, then A contains the first part of the com plex Schur form of the "balanced" versions of the input $A$ and $B$.

LD A (input)
The leading dim ension ofA. LD A $>=\max (1, N)$.
B (input/output)
On entry, them atrix $B$ in the pair $(A, B)$. On exit, $B$ has been overw ritten. If $\mathrm{JO} \mathrm{BV} \mathrm{L}=\mathrm{V}$ 'or JO BVR=V 'orboth, then B contains the second part of the com plex Schur form of the "balanced" versions of the inputA and $B$.

LD B (input)
The leading dim ension ofB. LD B $>=m$ ax $(1, N)$.
ALPHA (output)
On exit, A LPHA ( $)$ BETA ( $\mathcal{\nu}, \dot{于} 1, \ldots, N$, w illbe the generalized eigenvalues.

N ote: the quotient A LPHA ( ) BETA (j) ) may easily over- orunderflow, and BETA ( $)$ ) m ay even be zero. Thus, the user should avoid naively com puting the ratio A LPHA BETA. H ow ever, A LPHA w illbe alw ays less than and usually com parable w ith norm (A) in m agnitude, and BETA aliv ays less than and usually com parable w ith norm (B).

## BETA (output)

See description of A LPHA.
VL (output)
If JOBVL = V', the leftgeneralized eigenvectors
$u(\mathcal{)}$ are stored one after another in the colum ns of VL, in the sam e order as their eigenvalues. Each eigenvector will be scaled so the largest com ponentw illhave abs(real part) + abs(m ag.
part $=1$. N ot referenced if $J 0 B V L=N^{\prime}$.

LD V L (input)
The leading dim ension of the $m$ atrix $V L$. LD V L $>=1$, and if $\mathrm{JOBVL}=\mathrm{V}$ ', LDVL $>=\mathrm{N}$.
VR (input)
If $\mathrm{JOBVR}=\mathrm{V}$ ', the right generalized eigenvectors
$v(i)$ are stored one after another in the colum ns ofVR, in the sam e order as their eigenvalues. Each eigenvector will be scaled so the largest com ponentw illhave abs(real part) + abs(im ag. part) $=1 . \operatorname{N}$ ot referenced if $J 0 B V R=N{ }^{\prime}$.

LDVR (input)
The leading dim ension of the $m$ atrix $V R$.LD V $\gg=1$, and if $J O B V R=V$ ', LDVR $>=N$.

HO (output)
ШO is an integervalue such that on exitA $(i, 7)=$ 0 and $B(i, \gamma)=0$ ifi> jand $j=1, \ldots$, ILO -1 ori
$=\mathbb{H} \mathrm{I}+1, \ldots, \mathrm{~N}$. IfBALANC=N'or $\mathrm{S}^{\prime}, \mathbb{L} \mathrm{O}=1$ and $\mathbb{H} I=N$.

IH I (output)
IH $I$ is an integervalue such that on exitA $(i, 7)=$
0 and $B(i, j)=0$ if $i>$ jand $j=1, \ldots$, ILO -1 ori
$=\mathbb{H} \mathrm{I}+1, \ldots, \mathrm{~N}$. IfBALANC=$=\mathrm{N}^{\prime}$ or $\mathrm{S}^{\prime}$, $\mathbb{L} \mathrm{O}=1$
and $H I=N$.

LSCALE (output)
D etails of the perm utations and scaling factors
applied to the left side of $A$ and $B$. IfPL $(\mathcal{)}$ ) is the index of the row interchanged $w$ ith row $j$ and D L ( ) is the scaling factor applied to row $j$ then LSCALE ( $\mathcal{j}$ ) = PL ( $\mathfrak{j}$ ) for $j=1, \ldots$, ILO-1 = DL ( $)$ for $j=\mathbb{H O}, \ldots, \mathbb{H} I=P L(j)$ for $j=\mathbb{H} I+1, \ldots, N$. The order in which the interchanges are m ade is N to $\mathbb{H} \mathrm{I}+1$, then 1 to $\mathbb{L O}-1$.

## RSCALE (output)

D etails of the perm utations and scaling factors applied to the right side of A and B. IfPR ( $)$ ) is the index of the colum $n$ interchanged with colum $n$ $j$ and $\operatorname{DR}(\mathcal{j})$ is the scaling factor applied to column $j$ then RSCALE $(j)=\operatorname{PR}(\mathcal{j})$ for $j=$ $1, \ldots, \Pi \circ-1=\operatorname{DR}(j)$ for $j=\mathbb{H}, \ldots, I H I=\operatorname{PR}(\mathcal{j})$ for $j=\mathrm{HH} \mathrm{I}+1, \ldots, \mathrm{~N}$ The order in which the interchanges are m ade is N to $\mathbb{H} \mathrm{I}+1$, then 1 to $\mathbb{I} \mathrm{O}-1$.

The one-norm of the balanced $m$ atrix A.

## BBNRM (output)

The one-norm of the balanced $m$ atrix B .

## RCONDE (output)

IfSENSE = E'or B', the reciprocal condition
num bers of the selected eigenvalues, stored in consecutive elem ents of the array. If SENSE = V ',RCONDE is not referenced.

## RCONDV (output)

If $\mathrm{JOB}=\mathrm{V}$ 'or B ', the estim ated reciprocal condition numbers of the selected eigenvectors, stored in consecutive elem ents of the array. If the eigenvalues cannot be reordered to com pute RCONDV ( 7 ), RCONDV ( $j$ ) is setto 0 ; this can only occurw hen the true value w ould be very sm allanyway. If SENSE = E',RCONDV is not referenced. N ot referenced if $\mathrm{JOB}=\mathrm{E}$ '.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the aray W ORK. LW ORK >= $\mathrm{max}(1,2 \star \mathrm{~N})$. If SEN $S E=\mathrm{N}$ 'or E ', LW ORK $>=2 \star^{*} \mathrm{~N}$ 。 IfSENSE $=V$ 'or $B^{\prime}$ 'LW ORK $>=2 * N * N+2 \star N$.

If LW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of theW ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

RW ORK (w orkspace)
dim ension ( $6 \star$ N ) Realw orkspace.
IV ORK (w orkspace)
dim ension $(\mathbb{N}+2)$ If SEN $S E=E$ ', IV ORK is not referenced.

BW ORK (w orkspace)
dim ension $(\mathbb{N})$ If SEN $S E=N$ ', BW ORK is not referenced.
$\mathbb{I N F O}$ (output)
= 0 : successfulexit
< 0: if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an illegalvahue.
$=1, \ldots, \mathrm{~N}:$ The Q Z iteration failed. N o eigenvec-
tors have been calculated, but A LPHA ( ) ) and BETA ( $\mathcal{j}$ ) should be comect for $\mathcal{j} \mathbb{N} F O+1, \ldots, N .>$ $\mathrm{N}:=\mathrm{N}+1$ : other than Q Z teration failed in CHGEQZ.
$=\mathrm{N}+2$ : error retum from CTGEVC.

## FURTHER DETAILS

Balancing am atrix pair ( $A, B$ ) includes, first, perm uting row s and colum ns to isolate eigenvalues, second, applying diagonal sim ilarity transform ation to the row s and colum ns to $m$ ake the row $s$ and colum ns as close in norm as possible. The com puted reciprocal condition num bers comespond to the balanced m atrix. Perm uting row s and colum ns will not change the condition num bers (in exact arithm etic) but diagonal scaling will. For further explanation ofbalancing, see section 4.11 .12 of LA PA CK U sers'G uide.

A $n$ approxim ate errorbound on the chordal distance betw een the i-th computed generalized eigenvalue $w$ and the comesponding exacteigenvalue lam bda is hord (w , lam bda) <= EPS * norm (A BN RM, BBNRM) /RCONDE (I)

A $n$ approxim ate errorbound forthe angle betw een the $i$-th com puted eigenvectorV L (i) orVR (i) is given by PS * norm (ABNRM,BBNRM)/D $\mathbb{F}$ (i).

For further explanation of the reciprocal condition num bers RCONDE and RCONDV, see section 4.11 ofLAPACK U sers G uide.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cggglm -solve a general G auss M arkov linear model (G LM )
problem

## SYNOPSIS

```
SUBROUT\mathbb{NE CGGGLM N,M,P,A,LDA,B,LDB,D,X,Y,W ORK,LDW ORK,}
    INFO)
COM PLEX A (LDA,*),B (LDB,*),D (*),X (*),Y (*),W ORK (*)
INTEGERN,M,P,LDA,LDB,LDW ORK,\mathbb{NFO}
SUBROUT\mathbb{NECGGGLM_64 N,M,P,A,LDA,B,LDB,D,X,Y,W ORK,LDW ORK,}
    INFO)
```


$\mathbb{N} T E G E R * 8 N, M, P, L D A, L D B, L D W O R K, \mathbb{N} F O$

## F95 INTERFACE

SUBROUTINE GGGLM ( $\mathbb{N}], \mathbb{M}], \mathbb{P}], A,[\operatorname{LDA}], B,[L D B], D, X, Y,[\mathbb{O}$ ORK], [LDW ORK], [ $\mathbb{N F O}])$

COM PLEX,D $\mathbb{M} E N S I O N(:):: D, X, Y, W$ ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A, B
$\mathbb{N}$ TEGER ::N,M,P,LDA,LDB,LDW ORK, $\mathbb{N}$ FO
SU BROUTINE GGGLM_64 ( $\mathbb{N}], \mathbb{M}],[\mathbb{P}], A,[L D A], B,[L D B], D, X, Y,[\mathbb{W}$ ORK ], [LDW ORK], [ $\mathbb{N} F O]$ )

COM PLEX,D $\mathbb{M}$ ENSION (:) ::D,X,Y,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A,B
$\mathbb{N} T E G E R(8):: N, M, P, L D A, L D B, L D W O R K, \mathbb{N} F O$

## C INTERFACE

\#include <sunperfh>
void cggglm (intn, intm, intp, com plex *a, int lda, complex *b, int ldb, com plex *d, com plex *x, com plex *y, int*info);
void aggglm _64 (long n, long m, long p, com plex *a, long lda, com plex *b, long ldb, com plex *d, com plex *x, com plex *y, long *info);

## PURPOSE

cggglm solves a general Gauss M arkov linear model (G LM ) problem :
$m$ inim ize $\|y\| 2$ subject to $d=A *_{x}+B^{*} y$
X
$w$ here $A$ is an $N$ boy $-M m$ atrix, $B$ is an $N$ boy $P m$ atrix, and $d$ is a given N -vector. It is assum ed that $\mathrm{M}<=\mathrm{N}<=\mathrm{M}+\mathrm{P}$, and

$$
\operatorname{rank}(A)=M \quad \text { and } \quad \operatorname{rank}(A B)=N
$$

U nder these assum ptions, the constrained equation is alw ays consistent, and there is a unique solution $x$ and a minim al 2-nom solution $y$, which is obtained using a generalized $Q R$ factorization of $A$ and $B$.

In particular, ifm atrix $B$ is square nonsingular, then the problem G LM is equivalent to the follow ing w eighted linear least squares problem
$m$ inim ize $\|\operatorname{inv}(B) *(d-A * x)\| 2$
X
w here inv ( $B$ ) denotes the inverse of $B$.

## ARGUMENTS

N (input) The num ber of row s of the m atrices A and $\mathrm{B} . \mathrm{N}>=$ 0 .

M (input) The num ber of colum ns of them atrix A. $0<=\mathrm{M}<=$ N .

P (input) The num ber of colum ns of the m atrix B. P $>=\mathrm{N}-\mathrm{M}$.

A (input/output)
On entry, the $N$-by $-M$ m atrix A. On exit, A is destroyed.

LD A (input)
The leading dim ension of the aray A. LD A >= $\max (1, N)$.

B (input/output)
On entry, the N -by P m atrix B. On exit, B is destroyed.

LD B (input)
The leading dim ension of the array $\mathrm{B} . \mathrm{LD} \mathrm{B}>=$ $\max (1, N)$.

D (input/output)
On entry, $D$ is the lefthand side of the G LM equation. On exit, D is destroyed.

X (output)
On exit, X and Y are the solutions of the G LM problem.

Y (output)
On exit, X and Y are the solutions of the G LM
problem.

W ORK (w orkspace)
On exit, if $\mathbb{N F F O}=0, \mathrm{~W}$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the amay W ORK. LDW ORK >= $m a x(1, N+M+P)$. Foroptim um perform ance, LD W ORK >= $M+m$ in $(\mathbb{N}, P)+m$ ax $(\mathbb{N}, P) * N B$, where $N B$ is an upperbound for the optim al blocksizes forCGEQRF,CGERQF, CUNMQR and CUNMRQ.

IfLDW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
$\mathbb{I N} F O$ (output)
= 0: successfulexit.
< 0 : if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cgghrd - reduce a pair of com plex $m$ atrices ( $A, B$ ) to generalized upper H essenberg form using unitary transform ations, $w$ here $A$ is a generalm atrix and $B$ is upper triangular

## SYNOPSIS



```
    Z,LDZ,INFO)
```

CHARACTER * 1 COMPQ,COMPZ
COM PLEX A (LDA,*), B (LD B,*), Q (LDQ , *), Z (LD Z,*)
$\mathbb{N}$ TEGER $N, \mathbb{L O}, \mathbb{H} \mathrm{I}, \mathrm{LD} A, L D B, L D Q, L D Z, \mathbb{N}$ FO
SU BROUT $\mathbb{N} E C G G H R D \_64(C O M P Q, C O M P Z, N, \mathbb{L} O, \mathbb{H} I, A, L D A, B, L D B, Q$,
LD Q , Z, LD Z, $\mathbb{N} F O$ )
CHARACTER * 1 COMPQ,COMPZ
COM PLEX A (LDA, *), B (LDB,*), Q (LD Q , $\left.{ }^{\star}\right), \mathrm{Z}(\mathrm{LD} Z, \star)$
$\mathbb{N}$ TEGER* $8 \mathrm{~N}, \mathbb{L O}, \mathbb{H} \mathrm{I}, \mathrm{LD} A, L D B, L D Q, L D Z, \mathbb{N} F O$

## F95 INTERFACE

SU BROUTINE GGHRD (COMPQ,COMPZ, $\mathbb{N}], \mathbb{H}, \mathbb{H} I, A,[L D A], B,[L D B], Q$, [ LD Q$], \mathrm{Z},[\mathrm{LD} \mathrm{Z}],[\mathbb{N} F \mathrm{O}])$

CHARACTER (LEN=1): : COMPQ,COM PZ
COMPLEX,D $\mathbb{I}$ ENSION (:,:) :: A, $B, Q, Z$
$\mathbb{N}$ TEGER :: $\mathrm{N}, \mathbb{H} \mathrm{O}, \mathbb{H} \mathrm{I}, \mathrm{LD} A, L D B, L D Q, L D Z, \mathbb{N} F O$
SU BROUTINE GGHRD_64 (COMPQ,COMPZ, $\mathbb{N}], \mathbb{H O}, \mathbb{H} I, A,[L D A], B,[L D B]$, $Q,[\mathrm{LD} Q], \mathrm{Z},[\mathrm{LD} \mathrm{Z}],[\mathbb{N F O}])$

CHARACTER (LEN=1): $:$ COMPQ,COMPZ
COM PLEX, D $\mathbb{I}$ ENSION (:,:) ::A,B,Q,Z
$\mathbb{N}$ TEGER (8) ::N $, \mathbb{H O}, \mathbb{H} I, L D A, L D B, L D Q, L D Z, \mathbb{N} F O$

## C INTERFACE

\#include <sunperfh>
void cgghrd (char com pq, charcom pz, intn, int ilo, int ini, com plex *a, int lda, com plex *b, int ldb, com plex * $q$, int ldq, com plex *z, int ldz, int*info);
void cgghrd_64 (char com pq, charcom pz, long n, long ilo, long ihi, com plex *a, long lda, com plex *b, long ldb, com plex *q, long ldq, com plex *z, long ldz, long *info);

## PURPOSE

cgghrd reduces a pair of com plex $m$ atrices $(A, B)$ to generalized upper H essenberg form using unitary transform ations, $w$ here $A$ is a generalm atrix and $B$ is upper triangular: $Q$ '* $\mathrm{A} * \mathrm{Z}=\mathrm{H}$ and $\mathrm{Q}{ }^{\prime} * \mathrm{~B} * \mathrm{Z}=\mathrm{T}$, where H is upper H essenberg, T is uppertriangular, and Q and Z are unitary, and ' m eans conjugate transpose.

The unitary $m$ atrices $Q$ and $Z$ are determ ined as products of G ivens rotations. They $m$ ay eitherbe form ed explicitly, or they $m$ ay be postm ultiplied into inputm atrioes Q 1 and Z 1 , so that
$1 * A * Z 1{ }^{\prime}=\left(Q 1^{*} Q\right) * H *(Z 1 * Z) '$

## ARGUMENTS

COM PQ (input)
= N ': do not com pute Q ;
$=I^{\prime}: Q$ is initialized to the unit $m$ atrix, and the unitary m atrix Q is retumed; $=\mathrm{V}: \mathrm{Q} \mathrm{m}$ ust contain a unitary $m$ atrix $Q 1$ on entry, and the product $\mathrm{Q} 1 * \mathrm{Q}$ is retumed.

COMPZ (input)
$=\mathrm{N}$ ': do notcom pute Q ;
$=\mathrm{I}^{\prime}: \mathrm{Q}$ is in inialized to the unit m atrix, and the unitary m atrix Q is retumed; $=\mathrm{V}$ : $: \mathrm{Q}$ m ust contain a unitary $m$ atrix $Q 1$ on entry, and the product $\mathrm{Q} 1 * \mathrm{Q}$ is retumed.

N (input) The order of the m atrices A and $\mathrm{B} . \mathrm{N}>=0$.

ㅍO (input)
It is assum ed thatA is already upper triangular
in row sand colum ns $1: \mathbb{H O - 1}$ and $\mathbb{H} \mathrm{I}+1 \mathbb{N} . \mathbb{H O}$ and
HH I are norm ally setby a previous call to C G G BA L;
otherw ise they should be setto 1 and $N$ respec-
tively. $1<=\mathbb{L O}<=\mathbb{H} \mathrm{I}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{H}=1$ and $\mathbb{H} \mathrm{I}=0$, $\mathrm{if} \mathrm{N}=0$.

IH I (input)
See description of IIO .
A (input/output)
On entry, the N -by -N generalm atrix to be reduced.
On exit, the upper triangle and the first subdiagonalofA are overw ritten w th the upper $H$ essenberg $m$ atrix $H$, and the rest is set to zero.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, N)$.

B (input/output)
On entry, the $N-b y-N$ upper triangular $m$ atrix $B$.
On exit, the upper triangularm atrix $T=Q$ ' Z . The elem ents below the diagonal are set to zero.

LD B (input)
The leading dim ension of the aray $B$. LD B >= $\max (1, N)$.

Q (input/output)
If $C O M P Q=N$ : $Q$ is not referenced.
If COM PQ = 'I': on entry, $Q$ need notbe set, and on exit it contains the unitary $m$ atrix $Q$, where $Q$ 'is the product of the $G$ ivens transform ations which are applied to $A$ and $B$ on the left. If $C O M P Q=V$ ': on entry, Q m ustcontain a unitary m atrix Q 1 , and on exit this is overw ritten by $Q 1 * Q$.

LD Q (input)
The leading dim ension of the array $Q . L D Q>=N$ if $\mathrm{COMPQ}=\mathrm{V}$ 'or I '; LD Q >= 1 otherw ise.

Z (input/output)
If $C O M P Z=N^{\prime}: Z$ is not referenced.
If $\mathrm{COMPZ}=\mathrm{I}^{\prime}$ : on entry, Z need notbe set, and on exit it contains the unitary $m$ atrix $Z$, which is the product of the G ivens transform ations which
are applied to $A$ and $B$ on the right. If
$\mathrm{COMPZ=V}$ ': on entry, Z must contain a unitary $m$ atrix Z1, and on exit this is overw ritten by Z1*Z。

LD Z (input)
The leading dim ension of the array $Z$. LD $Z>=N$ if $C O M P Z=V$ 'or $I ; L D Z>=1$ otherw ise.
$\mathbb{N F O}$ (output)
= 0: successfulexit.
<0: if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an illegalvahue.

## FURTHER DETAILS

This routine reduces $A$ to $H$ essenberg and $B$ to triangular form by an unblocked reduction, as described in _M atrix_C om putations_, by G olub and van Loan (Johns H opkins Press).

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgglse - solve the linearequality-constrained least squares (LSE) problem

## SYNOPSIS

```
SUBROUT\mathbb{NE CGGLSE M ,N,P,A,LDA,B,LDB,C,D,X,W ORK,LDW ORK,}
    INFO)
COM PLEX A (LDA ,*),B (LD B ,*),C (*),D (*),X (*),W ORK (*)
```



```
SUBROUT\mathbb{NE CGGLSE_64M,N,P,A,LDA,B,LDB,C,D,X,W ORK,LDW ORK,}
    \mathbb{NFO )}
```

COM PLEXA (LDA, $\left.{ }^{\star}\right), \mathrm{B}\left(\mathrm{LDB},{ }^{\star}\right), \mathrm{C}\left(^{\star}\right), \mathrm{D}\left(^{\star}\right), \mathrm{X}\left({ }^{\star}\right), \mathrm{W} O R K\left({ }^{*}\right)$
$\mathbb{N} T E G E R * 8 M, N, P, L D A, L D B, L D W O R K, \mathbb{N} F O$

## F95 INTERFACE

SU BROUT $\mathbb{N} E \operatorname{GGLSE}(\mathbb{M}], \mathbb{N}],[\mathbb{P}], A,[L D A], B,[L D B], C, D, X,[\mathbb{N}$ ORK], [LDW ORK], [ $\mathbb{N} F O]$ )

COM PLEX,D $\mathbb{M}$ ENSION (:) ::C,D , X,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A,B
$\mathbb{N}$ TEGER :: M , N, P,LDA,LDB,LDW ORK, $\mathbb{N} F O$
SU BROUTINE GGLSE_64 ( $\mathbb{M}], \mathbb{N}], \mathbb{P}], A,[\operatorname{LDA}], B,[L D B], C, D, X,[\mathbb{W}$ ORK $]$, [LDW ORK], [ $\mathbb{N} F \mathrm{~F}]$ )

COM PLEX,D $\mathbb{M}$ ENSION (:) :: C,D,X,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A,B
$\mathbb{N} T E G E R(8):: M, N, P, L D A, L D B, L D W O R K, \mathbb{N} F O$

## C INTERFACE

\#include <sunperfh>
void agglse (intm, intn, intp, com plex *a, int lda, com plex *b, int ldb, com plex ${ }^{*} \mathrm{C}$, com plex *d, com plex * $x$, int*info);
void agglse_64 (long m , long $n$, long p, com plex *a, long lda, com plex *b, long ldb, com plex ${ }^{*}$ C, com plex ${ }^{\star}$ d, com plex *x, long *info);

## PURPOSE

cgglse solves the linearequality-constrained least squares (LSE ) problem :
$m$ inim ize $\left\|C-A *_{x}\right\| 2$ subject to $B *_{x}=d$
where $A$ is an $M$ boy $N \mathrm{~N}$ atrix, $B$ is a $P$ boy $N \mathrm{~m}$ atrix, $c$ is a given $M$-vector, and $d$ is a given $P$-vector. It is assum ed that $P<=N<=M+P$, and
$\operatorname{rank}(B)=P$ and $\operatorname{rank}((A))=N$. ( (B) )

These conditions ensure that the LSE problem has a unique solution, which is obtained using a G RQ factorization of the $m$ atrices $B$ and $A$.

## ARGUMENTS

M (input) The num ber of row s of the matrix $A . M>=0$.

N (input) The num ber of colum ns of the m atrioes A and B. N $>=0$.
$P$ (input) The num ber of row s of the m atrix B. $0<=P<=N<=$ $M+P$.

A (input/output)
O n entry, the $M-b y-N m$ atrix A. On exit, A is destroyed.

LD A (input)
The leading dim ension of the array A.LDA >= $\max (1, M)$.

B (input/output)
On entry, the $P-b y-N$ m atrix B. On exit, B is destroyed.

LD B (input)
The leading dim ension of the aray B . LD B >= $\max (1, \mathrm{P})$.

C (input/output)
On entry, $C$ contains the right hand side vector for the least squares part of the LSE problem. On exit, the residual sum of squares for the solution is given by the sum of squares ofelem ents $\mathrm{N}-\mathrm{P}+1$ to $M$ ofvector $C$.

D (input/output)
O n entry, $D$ contains the right hand side vector for the constrained equation. On exit, $D$ is destroyed.

X (output)
On exit, X is the solution of the LSE problem.
W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dim ension of the anay W ORK. LDW ORK >= $\max (1, M+N+P)$. For optim um perform ance LD $W$ ORK $>=$ $P+m$ in $M, N)+m$ ax $M, N) * N B$, where $N B$ is an upperbound for the optim al blocksizes forCGEQRF,CGERQF, CUNM QR and CUNMRQ.

If LD W ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LD W ORK is issued by XERBLA.
$\mathbb{I N F O}$ (output)
= 0: successfulexit.
<0: if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvahue.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cggqrf-com pute a generalized $Q R$ factorization of an $N$-by $-M$ $m$ atrix $A$ and an $N$ by $P m$ atrix $B$.

## SYNOPSIS

```
SUBROUT\mathbb{NE CGGQRF N,M,P,A,LDA,TAUA,B,LDB,TAUB,W ORK,LW ORK,}
    \mathbb{NFO )}
```

COM PLEX A (LDA, $\left.)^{*}\right)$,TAUA (*), B (LDB,*),TAUB (*), W ORK (*)
$\mathbb{N} T E G E R N, M, P, L D A, L D B, L W O R K, \mathbb{N} F O$
SU BROUTINECGGQRF_64 $\mathbb{N}, \mathrm{M}, \mathrm{P}, \mathrm{A}, \mathrm{LDA}, \mathrm{TAUA}, \mathrm{B}, \mathrm{LD} \mathrm{B}, \mathrm{TAUB}, \mathrm{W} O R K$,
LW ORK, $\mathbb{N} F O)$
COM PLEX A (LDA, *),TAUA (*), B (LDB,*),TAUB (*), W ORK (*)
$\mathbb{N}$ TEGER*8N,M,P,LDA,LDB,LWORK, $\mathbb{N} F O$

## F95 INTERFACE

SU BROUTINE GGQRF ( $\mathbb{N}], \mathbb{M}], \mathbb{P}], A,[L D A], T A U A, B,[L D B], T A U B,[\mathbb{O} O R]$, [LW ORK], [ $\mathbb{N F O}$ ])

COM PLEX,D $\mathbb{I M}$ ENSION (:) ::TAUA,TAUB,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : : A, B
$\mathbb{N} T E G E R:: N, M, P, L D A, L D B, L W O R K, \mathbb{N} F O$
SU BROUTINEGGQRF_64 ( $\mathbb{N}], \mathbb{M}],[\mathbb{P}], A,[L D A], T A U A, B,[L D B], T A U B$, [W ORK], [LW ORK], [ $\mathbb{N} F O$ ])

COM PLEX,D IM ENSION (:) ::TAUA,TAUB,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : A, B
$\mathbb{N}$ TEGER (8) ::N,M,P,LDA,LDB,LW ORK, $\mathbb{N} F O$

## C INTERFACE

\#include < sunperfh>
void cggqrf(intn, intm, intp, com plex *a, int lda, complex *taua, com plex *b, int ldb, com plex *taub, int*info);
void oggqrf_64 (long $n$, long $m$, long $p$, com plex *a, long lda, complex *taua, complex *b, long ldb, com plex *taub, long *info);

## PURPOSE

cggqrf com putes a generalized $Q$ R factorization of an $N$-by -M $m$ atrix $A$ and an $N$ by $P m$ atrix $B$ :

$$
A=Q * R, \quad B=Q * T * Z,
$$

where $Q$ is an $N$ by $N$ unitary $m$ atrix, $Z$ is a $P$-by $P$ unitary $m$ atrix, and $R$ and $T$ assum e one of the form $s$ :
if $N>=M, R=(R 11) M$, orif $N<M, R=(R 11 R 12$
) N,
( 0 ) N M
N $\mathrm{M}-\mathrm{N}$
M
where R11 is upper triangular, and
if $\mathrm{N}<=\mathrm{P}, \mathrm{T}=(0 \mathrm{~T} 12) \mathrm{N}$, orifN $>\mathrm{P}, \mathrm{T}=(\mathrm{T} 11$ )
$\mathrm{N}-\mathrm{P}$,

$$
\begin{equation*}
\mathrm{P}-\mathrm{N} \mathrm{~N} \tag{T21}
\end{equation*}
$$

P
where T12 or T 21 is uppertriangular.
In particular, if $B$ is square and nonsingular, the $G Q R$ factorization of $A$ and $B$ im plicitly gives the $Q R$ factorization of inv (B)*A :

$$
\operatorname{inv}(B) \star A=Z \text { * }(\operatorname{inv}(T) * R)
$$

$w$ here inv ( $B$ ) denotes the inverse of the $m$ atrix $B$, and $Z^{\prime}$ denotes the conjugate transpose ofm atrix $Z$.

## ARGUMENTS

N (input) The num ber of row sof the m atrioes A and $\mathrm{B} . \mathrm{N}>=$
0.

M (input) The num ber of colum ns of the m atrix A. M >=0.
$P$ (input) The num ber of colum ns of the $m$ atrix $B . P>=0$.

A (input/output)
On entry, the $N$ boy $M \mathrm{~m}$ atrix A. On exit, the ele$m$ ents on and above the diagonal of the amay contain the $m$ in $\mathbb{N}, M$ )-by $-M$ uppertrapezoidalm atrix $R$ $(R$ is upper triangular if $N>=M)$; the elem ents below the diagonal, w th the array TA U A , represent the unitary $m$ atrix $Q$ as a productofm in $(\mathbb{N}, M)$ ele$m$ entary reflectors (see FurtherD etails).

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, N)$.

TAUA (output)
The scalar factors of the elem entary reflectors $w$ hich represent the unitary $m$ atrix $Q$ (see Further D etails).

B (input/output)
On entry, the N -by P m atrix B. On exit, if $\mathrm{N}<=$ P , the upper triangle of the subaray $B(\mathcal{N}, \mathrm{P}-$ $\mathrm{N}+1: \mathrm{P}$ ) contains the N -by N uppertriangularm atrix
$T$; ifN > P, the elem ents on and above the $(\mathbb{N} P)-$ th subdiagonal contain the N Hoy -P upper trapezoidal m atrix T ; the rem aining elem ents, w th the array TA UB, represent the unitary matrix Z as a product of elem entary reflectors (see Further D etails).

LD B (input)
The leading dim ension of the aray $\mathrm{B} . \mathrm{LD} \mathrm{B}>=$ $\max (1, N)$.

TAUB (output)
The scalar factors of the elem entary reflectors which represent the unitary $m$ atrix $Z$ (see Further D etails).

W ORK (w orkspace)
On exit, if $\mathbb{N F O}=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the amay W ORK. LW ORK >=
$\max (1, N, M, P)$. For optim um perform ance LW ORK $>=$ $\max (\mathbb{N}, \mathbb{M}, P){ }^{m} \max (\mathbb{N} 1, N B 2, N B 3)$, where $N B 1$ is the optim al blocksize forthe QR factorization of an N toy -M m atrix, N B 2 is the optim al blocksize for the RQ factorization of an $N$ boy $P m$ atrix, and NB3 is the optim alblocksize for a call of CUNM QR .

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the $i$ th argum enthad an illegalvalue.

## FURTHER DETAILS

Them atrix $Q$ is represented as a product of elem entary reflectors

$$
Q=H(1) H(2) \ldots H(k), w \text { here } k=m \text { in }(n, m) .
$$

Each H (i) has the form

$$
H(i)=I-\operatorname{tana}{ }^{*} v^{*} v^{\prime}
$$

$w$ here taua is a com plex scalar, and $v$ is a com plex vector $w$ ith $v(1: i-1)=0$ and $v(i)=1 ; v(i+1 m)$ is stored on exit in A $(i+1 m, i)$, and taua in TAUA (i).
To form Q explicitly, use LA PACK subroutine CUNGQR. To use Q to update another matrix, use LAPACK subroutine CUNM QR.

Them atrix $Z$ is represented as a product of elem entary reflectors
$Z=H(1) H(2) \ldots H(k), w$ here $k=m$ in $(n, p)$.
Each H (i) has the form

$$
H(i)=I-\operatorname{taub} * v^{*} v^{\prime}
$$

$w$ here taub is a com plex scalar, and $v$ is a com plex vector $w$ ith $v(p-k+i+1 \mathrm{p})=0$ and $v(p-k+i)=1 ; v(1 \mathrm{p}-k+i-1)$ is stored on exitin B ( $n-k+i, 1$ p $k+i-1$ ), and taub in TA U B (i). To form Z explicitly, use LAPACK subroutine CUNGRQ. To use Z to update another matrix, use LAPACK subroutine CUNMRQ.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cggrqf-com pute a generalized $R Q$ factorization of an $M$ by $-N$ $m$ atrix $A$ and $a-b y-N$ m atrix B

## SYNOPSIS

```
SUBROUT\mathbb{NE CGGRQFM,P,N,A,LDA,TAUA,B,LDB,TAUB,W ORK,LW ORK,}
    \mathbb{NFO )}
```

COM PLEX A (LDA,*),TAUA (*), B (LDB,*),TAUB (*),W ORK (*)
$\mathbb{N}$ TEGERM, $\mathrm{P}, \mathrm{N}$, LDA, LDB,LWORK, $\mathbb{N} F O$
SU BROUTINECGGRQF_64M,P,N,A,LDA,TAUA,B,LDB,TAUB,WORK,
LW ORK, $\mathbb{N} F O$ )
COM PLEX A (LDA, *),TAUA (*), B (LDB,*),TAUB (*), W ORK (*)
$\mathbb{N}$ TEGER*8 M , P, N, LDA, LD B, LW ORK, $\mathbb{N} F O$

## F95 INTERFACE

SU BROUTINE GGRQF ( $\mathbb{M}],[\mathbb{P}], \mathbb{N}], A,[L D A], T A U A, B,[L D B], T A U B,[\mathbb{O}$ OR ], [LW ORK], [ $\mathbb{N F O}$ ])

COM PLEX,D $\mathbb{I M}$ ENSION (:) ::TAUA,TAUB,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : : A, B
$\mathbb{N} T E G E R:: M, P, N, L D A, L D B, L W O R K, \mathbb{N} F O$
SU BROUTINE GGRQF_64 (M) $\mathbb{M}, \mathbb{P}], \mathbb{N}], A,[L D A], T A U A, B,[L D B], T A U B$, [W ORK ], [LW ORK], [ $\mathbb{N F O}$ ])

COMPLEX,D $\mathbb{M}$ ENSION (:) ::TAUA,TAUB,W ORK
COM PLEX,D $\mathbb{I M}$ ENSION (: : : : :: A, B
$\mathbb{N}$ TEGER (8) ::M, P, N,LDA,LDB,LW ORK, $\mathbb{N} F O$

## C INTERFACE

\#include < sunperfh>
void cggrqf(intm, intp, intn, com plex *a, int lda, com plex *taua, com plex *b, int ldb, com plex *taub, int*info);
void cggrqf_64 (long m, long p, long n, com plex *a, long lda, complex *taua, complex *b, long ldb, com plex *taub, long *info);

## PURPOSE

cggrqf com putes a generalized RQ factorization of an M by -N $m$ atrix $A$ and $a P-b y-N m$ atrix $B$ :

$$
A=R * Q, \quad B=Z * T * Q,
$$

where $Q$ is an $N$ boy $N$ unitary $m$ atrix, $Z$ is a $P$-by $P$ unitary $m$ atrix, and $R$ and $T$ assum e one of the form $s$ :

```
ifM <= N, R = (0 R12)M, orifM > N, R = ( R11 )
```

$\mathrm{M}-\mathrm{N}$,

$$
\mathrm{N}-\mathrm{M} \text { M (R21)N }
$$

N
where R12 orR21 is upper triangular, and
if $\mathrm{P}>=\mathrm{N}, \mathrm{T}=(\mathrm{T} 11) \mathrm{N}$, orifP $<\mathrm{N}, \mathrm{T}=(\mathrm{T} 11 \mathrm{~T} 12$
) P,
( 0 ) P-N
P NP
N
where T11 is upper triangular.
In particular, if $B$ is square and nonsingular, the $G R Q$ factorization ofA and $B$ im plicitly gives the $R Q$ factorization of A *inv (B):

$$
A * \operatorname{inv}(B)=(R * \operatorname{inv}(I)) * Z^{\prime}
$$

where inv ( $B$ ) denotes the inverse of the $m$ atrix $B$, and $Z^{\prime}$ denotes the conjugate transpose of the $m$ atrix $Z$.

## ARGUMENTS

M (input) The num ber of row s of the $m$ atrix $A . M>=0$.
$P$ (input) The num ber of row sof the $m$ atrix $B . P>=0$.
N (input) The num ber of colum ns of the $m$ atrioes $A$ and $B . N$ $>=0$.

A (input/output)
On entry, the M -by -N m atrix A. On exit, if M <=
N , the upper triangle of the subaray A ( $1 \mathrm{M}, \mathrm{N}$ -
$\mathrm{M}+1 \mathrm{~N}$ ) contains the M -by -M uppertriangularm atrix
$R$; if $M>N$, the elem ents on and above the $M-N$ )th subdiagonal contain the $\mathrm{M}-$ by -N upper trapezoidal $m$ atrix $R$; the rem aining elem ents, $w$ ith the amay TAUA, represent the unitary $m$ atrix $Q$ as a product of elem entary reflectors (see Further D etails).
LDA (input)
The leading dim ension of the array A. LDA >= max (1, M).

TAUA (output)
The scalar factors of the elem entary reflectors which represent the unitary $m$ atrix $Q$ (see Further D etails).

B (input/output)
On entry, the $P-b y-N m$ atrix B. On exit, the ele$m$ ents on and above the diagonal of the array contain the $m$ in $(\mathbb{P}, N)-b y-N$ uppertrapezoidalm atrix $T$ ( $T$ is upper triangular if $P>=N$ ); the elem ents below the diagonal, w ith the amray TA U B, represent the unitary $m$ atrix $Z$ as a productofelem entary reflectors (see FurtherD etails).

LD B (input)
The leading dim ension of the array $\mathrm{B} . \operatorname{LD} \mathrm{B}>=$ $\max (1, P)$.

TAUB (output)
The scalar factors of the elem entary reflectors which represent the unitary $m$ atrix $Z$ (see Further D etails).

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the array W ORK. LW ORK >= $m a x(1, N, M, \mathbb{P})$. For optim um perform ance LW $O R K>=$
$\max (\mathbb{N}, \mathbb{M}, \mathbb{P})_{m a x}(N B 1, N B 2, N B 3)$, where NB1 is the optim al blocksize forthe $R Q$ factorization of an M by $-\mathrm{N} m$ atrix, NB2 is the optim al blocksize for the $Q R$ factorization of $P$-by $-N m$ atrix, and NB3 is the optim alblocksize for a callof $C U N M R Q$.

If LW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
$\mathbb{N} F O$ (output)
= 0 : successfiulexit
< 0: if $\mathbb{N F O}=-i$, the $i$-th argum enthad an illegal value.

## FURTHER DETAILS

Them atrix $Q$ is represented as a product of elem entary reflectors
$Q=H(1) H(2) \ldots H(k)$, where $k=m$ in $(m, n)$.
Each H (i) has the form

$$
\mathrm{H}(\mathrm{i})=\mathrm{I}-\operatorname{tana} * \mathrm{~V}^{*} \mathrm{~V}^{\prime}
$$

where taua is a com plex scalar, and $v$ is a complex vector $w$ ith $v(n-k+i+1 n)=0$ and $v(n-k+i)=1 ; v(1 n-k+i-1)$ is stored on exitin A ( $m-k+i, 1 n-k+i-1$ ), and taua in TAUA (i). To form $Q$ explicitly, use LAPACK subroutine CUNGRQ. To use $Q$ to update another $m$ atrix, use LAPACK subroutine CUNMRQ.

Them atrix $Z$ is represented as a product of elem entary reflectors

$$
Z=H(1) H(2) \ldots H(k), w \text { here } k=m \text { in }(p, n) .
$$

Each H (i) has the form

$$
\mathrm{H}(\mathrm{i})=\mathrm{I}-\operatorname{tanb} * \mathrm{~V}^{*} \mathrm{v}^{\prime}
$$

where taub is a com plex scalar, and $v$ is a com plex vector w ith $\mathrm{v}(1: i-1)=0$ and $v(i)=1 ; \mathrm{v}(i+1: p)$ is stored on exit in B (i+1 $\mathrm{P}, \mathrm{i}$ ), and taub in TA UB (i).
To form $Z$ explicitly, use LAPACK subroutine CUNGQR. To use $Z$ to update another matrix, use LAPACK subroutine CUNMQR.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cggsvd -com pute the generalized singular value decom position (G SVD) of an M -by-N com plex m atrix A and P-by-N com plex $m$ atrix B

## SYNOPSIS

```
SU BROUT\mathbb{NE CGGSVD(JOBU,NOBV,NOBQ,M ,N,P,K,L,A,LDA,B,LD B,}
    A LPHA,BETA,U,LDU,V,LDV,Q,LDQ,W ORK,W ORK 2,IN ORK 3, \mathbb{NFO)}
CHARACTER * 1 JOBU,NOBV,NOBQ
COM PLEX A (LDA,*),B (LD B,*), U (LDU ,*), V (LDV ,*), Q (LDQ ,*),
W ORK (*)
\mathbb{NTEGER M,N,P,K,L,LDA,LDB,LDU,LDV,LDQ, INFO}
\mathbb{NTEGER IN ORK 3 (*)}
REAL ALPHA (*),BETA (*),W ORK 2 (*)
SU BROUT\mathbb{NE CGGSVD_64(JO BU,NO BV,NOBQ,M,N,P,K,L,A ,LDA,B,LD B,}
    ALPHA,BETA,U,LDU,V,LDV,Q,LDQ,W ORK,W ORK 2,IN ORK 3, \mathbb{NFO)}
CHARACTER * 1 JOBU,NOBV,NOBQ
COM PLEX A (LDA,*),B (LD B,*), U (LDU ,*), V (LDV,*), Q (LDQ ,*),
W ORK (*)
\mathbb{NTEGER*8M,N,P,K,L,LDA,LDB,LDU,LDV,LDQ, NNFO}
\mathbb{NTEGER*8 IN ORK 3 (*)}
REAL ALPHA (*),BETA (*),W ORK 2 (*)
```


## F95 INTERFACE

SU BROUTINE GGSVD (JOBU, $\mathcal{D} \operatorname{BV}, \mathcal{D} B Q, \mathbb{M}], \mathbb{N}],[\mathbb{P}], K, L, A,[L D A], B$, [LDB],ALPHA,BETA, U, [LDU],V, [LDV],Q, [LDQ], [W ORK], [W ORK2], $\mathbb{I N}$ ORK3, $[\mathbb{N F O}])$

COM PLEX,D $\mathbb{I M} E N S I O N(:):$ W ORK
COM PLEX ,D $\mathbb{M}$ ENSION (:,:) :: A, B, U, V , Q

$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{I W} O R K 3$
REAL,D $\mathbb{I M} E N S I O N$ (:) ::ALPHA,BETA,W ORK2

SU BROUTINE G G SVD_64 (JOBU, JOBV, JOBQ, M ], $\mathbb{N}],[P], K, L, A,[L D A]$,
 [WORK2], $\mathbb{I W}$ ORK 3, [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) :: JOBU, JOBV, JOBQ
COM PLEX ,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : A , B , U , V , Q
$\mathbb{N}$ TEGER (8) :: M , N , P, K , L, LD A , LD B , LD U , LD V , LD Q , $\mathbb{N}$ FO
$\mathbb{N}$ TEGER (8), D $\mathbb{M}$ ENSION (:) :: $\mathbb{I N}$ ORK 3
REAL,D $\mathbb{M}$ ENSION (:) ::ALPHA,BETA,W ORK2

## C INTERFACE

\#include <sunperfh>
void oggsvd (char jobu, char jobv, char jobq, intm, int n, int p , int *k, int *l, com plex *a, int lda, com plex *b, int ldb, float * alpha, float *beta, com plex *u, int ldu, com plex *v, int ldv, com plex *q, int ldq, int *íw ork3, int*info);
void cggsvd_64 (char jojbu, char jobv, char jobq, long m, long n, long p, long *k, long *l, com plex *a, long lda, com plex *b, long ldb, float *alpha, float *beta, com plex *u, long ldu, com plex *v, long ldv, com plex *q, long ldq, long *ìw ork3, long *info);

## PURPOSE

cggsvd com putes the generalized singularvalue decom position (G SV D ) of an $M$ łoy $-N$ com plex $m$ atrix $A$ and $P$ boy $-N$ com plex m atrix B :

$$
U^{*} A * Q=D 1^{*}(0 R), \quad V{ }^{*} B * Q=D 2^{*}(0 R)
$$

$w$ here $U, V$ and $Q$ are unitary $m$ atriges, and $Z$ ' $m$ eans the conjugate transpose of $\mathrm{Z} . \mathrm{Let} \mathrm{K}+\mathrm{L}=$ the effective num erical rank of them atrix ( $A$ ', $B$ )', then $R$ is a $(\mathbb{K}+L$ ) -by $(\mathbb{K}+L$ ) nonsingular upper triangularm atrix, $D 1$ and $D 2$ are $M$ foy $-(K+L)$ and $\mathrm{P}-\mathrm{by}-(\mathrm{K}+\mathrm{L})$ "diagonal" $m$ atrioes and of the follow ing structures, respectively:

IfM $\mathrm{K}-\mathrm{L}>=0$,

K L

$$
\begin{aligned}
& D 1=K(I 0) \\
& \text { L ( } 0 \mathrm{C} \text { ) } \\
& M \text { K 乙 ( } 0 \text { O) } \\
& \text { K L } \\
& \mathrm{D} 2=\mathrm{L}(0 \mathrm{~S}) \\
& \mathrm{P} \longleftarrow(0 \quad 0) \\
& \mathrm{NH} \mathrm{~K} \mathrm{~K} \quad \mathrm{~L} \\
& \text { (0R) }=\mathrm{K} \text { (0 R11 R12) } \\
& \text { L ( } 0 \quad 0 \quad \mathrm{R} 22 \text { ) } \\
& \text { where } \\
& C=\operatorname{diag}(A L P H A(K+1), \ldots, A L P H A(K+L)) \text {, } \\
& S=\operatorname{diag}(\operatorname{BETA}(K+1), \ldots, B E T A(K+L)) \text {, } \\
& C * * 2+S * * 2=I \text {. }
\end{aligned}
$$

$R$ is stored in $A(1: K+L, N-K-1 \mathbb{N})$ on exit． IfM $-\mathrm{K} \mathrm{f}<0$ ，

$$
\begin{aligned}
& \text { K M K K + L }-\mathrm{M} \\
& D 1=K\left(\begin{array}{ll}
I & 0
\end{array}\right) \\
& M-K(0 C 0) \\
& \text { K M K K + L } \mathrm{M} \\
& D 2=M K(0 S 0) \\
& \mathrm{K}+\mathrm{L} \mathrm{M} \text { ( } 0 \text { O I ) } \\
& \text { P孔 (0 0 0 ) } \\
& N \not K \dashv K \quad M-K+L-M \\
& (0 R)=K(0 \text { R11 R12 R13 ) } \\
& \mathrm{M}-\mathrm{K} \text { ( } 0 \quad 0 \text { R22 R23) } \\
& \mathrm{K}+\mathrm{L} \mathrm{M} \text { (0 } 0 \text { (0 R33) }
\end{aligned}
$$

w here

$$
\begin{aligned}
& C=\operatorname{diag}(A L P H A(K+1), \ldots, \text { A LPHA } M)), \\
& S=\operatorname{diag}(\operatorname{BETA}(K+1), \ldots, \text { BETA }(M)), \\
& C \star * 2+S \star * 2=I .
\end{aligned}
$$

$(\mathrm{R} 11 \mathrm{R} 12 \mathrm{R} 13)$ is stored in $\mathrm{A}(1 \mathbb{M}, \mathrm{~N}-\mathrm{K}-\mathrm{L}+\mathbb{N})$ ，and R 33 is stored
（0 R 22 R 23 ）
in $B(M-K+1: N+M-K-\perp+1 \mathbb{N})$ on exit．

The routine com putes $C, S, R$ ，and optionally the unitary transform ation $m$ atrices $U, V$ and $Q$ ．

In particular，if $B$ is an $N$ boy -N nonsingular $m$ atrix，then the G SVD ofA and B im plicitly gives the SVD ofA＊inv（B）：

$$
A * \operatorname{inv}(B)=U *(D 1 * \operatorname{inv}(D 2)) * V V^{\prime} .
$$

If ( $A$ ', B )' 'has orthnorm alcolum ns, then the GSVD of $A$ and $B$ is also equal to the CS decom position of A and B.Further$m$ ore, the GSVD can be used to derive the solution of the eigenvalue problem :

$$
\text { A *A } \mathrm{x}=\operatorname{lam} \text { bda* } \mathrm{B} \text { *B } \mathrm{x} .
$$

In som e literature, the G SVD of $A$ and $B$ is presented in the form
$\mathrm{U} * \mathrm{~A} * \mathrm{X}=(0 \mathrm{D} 1), \mathrm{V}$ *B*X=(0D2)
where $U$ and $V$ are orthogonal and $X$ is nonsingular, and $D 1$ and D 2 are "diagonal". The form erG SVD form can be converted to the latter form by taking the nonsingularm atrix X as

$$
\begin{aligned}
X= & Q *\left(\begin{array}{ll}
I & 0
\end{array}\right) \\
& (0 \operatorname{inv}(R))
\end{aligned}
$$

## ARGUMENTS

$J O B U$ (input)
$=\mathrm{U}$ : U nitary $m$ atrix U is com puted;
$=N^{\prime}: U$ is notcom puted.
$J O$ BV (input)
= V : U nitary m atrix V is com puted;
$=\mathrm{N}: \mathrm{V}$ is not com puted.
$J O B Q$ (input)
$=Q$ : U nitary matrix Q is com puted;
$=\mathrm{N}^{\prime}: \mathrm{Q}$ is not com puted.

M (input) The num ber of row s of the m atrix $\mathrm{A} . \mathrm{M}>=0$.

N (input) The num ber of collm ns of the m atrioes A and B. N $>=0$.
$P$ (input) The num ber of row sof the $m$ atrix $B . P>=0$.
K (output)
On exit, $K$ and $L$ specify the dim ension of the subblocks described in Puppose. $\mathrm{K}+\mathrm{L}=$ effective num erical rank of (A 'B )'.

L (output)
On exit, $K$ and L specify the dim ension of the subblocks described in Punpose. $\mathrm{K}+\mathrm{L}=$ effective num erical rank of (A 'B ')'.

A (input/output)

On entry, the M by -N m atrix A. On exit, A contains the triangularm atrix $R$, orpart of $R$. See Purpose fordetails.

LD A (input)
The leading dim ension of the array A. LDA >= $m a x(1, M)$.

B (input/output)
On entry, the $P$-by $-N$ m atrix B. On exit, B contains part of the triangularm atrix $R$ if $M K-4<$ 0 . See Punpose fordetails.

LD B (input)
The leading dim ension of the array $\mathrm{B} . \operatorname{LD} \mathrm{B}>=$ $\max (1, \mathrm{P})$.

## A LPHA (output)

On exit, ALPHA andBETA contain the generalized singular value pairs of $A$ and $;$; LPHA $(1: K)=1$, A LPHA ( $1: K$ ) = 1 ,
BETA $(1: K)=0$, and ifM $K-L>=0$, ALPHA $(K+1 \mathbb{K}+L)$
= C,
BETA $(K+1: K+L)=S$, orifM $K-L<0$, ALPHA $(K+1 M)=$
C, ALPHA $M+1: K+L)=0$
$\operatorname{BETA}(K+1 M)=S, \operatorname{BETA}(M+1: K+L)=1$ and
A LPHA $(\mathbb{K}+\mathrm{L}+1 \mathbb{N})=0$
BETA $(\mathbb{K}+\mathrm{L}+1 \mathbb{N})=0$

BETA (output)
See description of A LPH A .
U (output)
If $\mathrm{JOBU}=\mathrm{U}$ ', U contains the M -by M unitary
$m$ atrix $U$. If $J O B U=N$ ', $U$ is not referenced.
LD U (input)
The leading dim ension of the array $U$. LD U >= $m a x(1, M)$ if $J O B U=U ' ; L D U>=1$ otherw ise.

V (output)
If $\mathrm{JOBV}=\mathrm{V}$ ', V contains the P -by P unitary m atrix V . If $\mathrm{JO} \mathrm{BV}=\mathrm{N}, \mathrm{V}$ is not referenced.

LDV (input)
The leading dim ension of the array $V$. LDV >= $\mathrm{max}(1, \mathrm{P})$ if $\mathrm{JOBV}=\mathrm{V}$; LDV $>=1$ otherw ise.

Q (output)
If $J O B Q=Q ', Q$ contains the $N$ by $-N$ unitary
$m$ atrix $Q$. If $J O B Q=N$ ', $Q$ is not referenced.

LD Q (input)
The leading dim ension of the array $Q . \operatorname{LDQ}>=$ $\max (1, N)$ if $J O B Q=Q ; L D Q>=1$ otherw ise.

W ORK (w orkspace)
dim ension $M$ AX $(3 * N, M, P)+N)$
W ORK 2 (w orkspace)
dim ension ( $2 * \mathrm{~N}$ )
IV ORK 3 (output)
dim ension $(\mathbb{N})$ On exit, $\mathbb{I W}$ ORK 3 stores the sorting inform ation. M ore precisely, the follow ing loop willsortA LPHA for $I=K+1$, $m$ in $M, K+L$ ) sw ap ALPHA (I) and ALPHA ( $\mathbb{W}$ ORK 3 (I)) endfor such that A LPHA ( 1 ) >=ALPHA (2) >= ... $>=A L P H A(N)$.
$\mathbb{I N} F \mathrm{O}$ (output)
= 0: successfulexit.
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum ent had an illegalvahue.
>0: if $\mathbb{N F O}=1$, the Jacobi-type procedure failed to converge. For further details, see subroutine C TG SJA.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

```
cggsvp - com pute unitary m atrices U,V and Q such that N -
```

$K-\mathrm{K} L \mathrm{U} * \mathrm{~A} * \mathrm{Q}=\mathrm{K}$ (0A12A13) if $\mathrm{M} \mathrm{K}-\mathrm{L}>=0$

## SYNOPSIS

```
SU BROUTINE CGGSVP(JOBU,NOBV,NOBQ,M,P,N,A,LDA,B,LDB,TOLA,
    TOLB,K,L,U,LDU,V,LDV,Q,LDQ,IN ORK,RW ORK,TAU,W ORK,
    INFO)
CHARACTER * 1 JOBU,JOBV,JOBQ
COM PLEX A (LDA,*),B (LD B,*), U (LDU ,*), V (LDV ,*), Q (LDQ ,*),
TAU (*),W ORK (*)
\mathbb{NTEGER M,P,N,LDA,LDB,K,L,LDU,LDV,LDQ, INFO}
INTEGER IN ORK (*)
REAL TOLA,TOLB
REAL RW ORK (*)
```

SU BROUTINE CGGSVP_64 (JO BU , JO BV , JO BQ , M , P, N, A , LD A , B, LD B, TOLA,
TOLB,K,L,U,LDU,V,LDV,Q,LDQ,IWORK,RWORK,TAU,WORK,
$\mathbb{N} F O$ )
CHARACTER * 1 JOBU, JOBV , JOBQ
COM PLEX A (LDA,*), B (LDB,*), U (LDU ,*), V (LDV ,*), Q (LDQ ,*),
TAU ( ${ }^{*}$ ), WORK ( ${ }^{*}$ )
$\mathbb{N}$ TEGER*8 M , P, N ,LDA ,LDB, K , L, LD U ,LDV ,LD Q , $\mathbb{N}$ FO
$\mathbb{N}$ TEGER*8 $\mathbb{I N}$ ORK (*)
REAL TOLA,TOLB
REAL RW ORK (*)

## F95 INTERFACE

SU BROU T INE G GSVP (OOBU , JOBV , $\mathcal{O B} \operatorname{BQ}, \mathbb{M}], \mathbb{P}], \mathbb{N}], A,[L D A], B,[L D B]$,

TOLA,TOLB,K,L,U,[LDU],V,[LDV],Q,[LDQ],[IN ORK], [RW ORK], [TAU], [W ORK], [NFO])

CHARACTER (LEN=1):: JOBU, JOBV, JOBQ
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::TAU,W ORK
COMPLEX,D $\mathbb{I M} E N S I O N(:,:):: A, B, U, V, Q$
$\mathbb{N} T E G E R:: M, P, N, L D A, L D B, K, L, L D U, L D V, L D Q, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: $\mathbb{I N}$ ORK
REAL ::TOLA,TOLB
REAL,D $\mathbb{M}$ ENSION (:) ::RW ORK

SU BROUTINE G G SVP_64 (JOBU, $\mathcal{J O B V}, \mathcal{J O B Q}, \mathbb{M}],[\mathbb{P}], \mathbb{N}], A,[L D A], B$,
 [RW ORK], [TAU ], [W ORK ], [ $\mathbb{N} F O]$ )

COM PLEX,D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::A,B, U,V,Q
$\mathbb{N}$ TEGER (8) ::M ,P,N,LDA,LDB,K,L,LDU,LDV,LDQ, $\mathbb{N}$ FO
$\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N}$ ORK
REAL ::TOLA,TOLB
REAL,D $\mathbb{M}$ ENSION (:) ::RW ORK

## C INTERFACE

\#include <sunperfh>
void cggsvp (char j̀jur char j̀jbv, char jobq, intm, int p, int $n$, com plex *a, int lda, com plex *b, int ldb, float tola, float tolb, int *k, int *l, com plex *u, int ldu, com plex ${ }^{*}$ v, int ldv, com plex *q, int ldq, int*info);
void cggsvp_64 (char jobu, char jobv, char jobor, long m, long
p, long n, com plex *a, long lda, com plex *b, long ldb, float tola, floattolb, long *k, long *l, com plex *u, long ldu, com plex *v, long ldv, com plex *q, long ldq, long *info);

## PURPOSE

cggsvp com putes unitary $m$ atrices $\mathrm{U}, \mathrm{V}$ and Q such that
L (0 0 A23)
$M$ Kـ ( $0 \quad 0 \quad 0 \quad)$

NK— K L
$=K(0$ A12 A13) ifM $K-4<0$;
M K (0 0 A 23)

NK- K L
$\mathrm{V} * \mathrm{~B} * \mathrm{Q}=\mathrm{L}\left(\begin{array}{lll}0 & 0 & \mathrm{~B} 13\end{array}\right)$
P 乙 ( 0000 )
where the $K-b y-K m$ atrix A 12 and $L-b y-L m$ atrix B 13 are nonsingularuppertriangular; A 23 is L-by- L uppertriangular if $\mathrm{M} \mathrm{K}-\mathrm{L}>=0$, otherw ise A 23 is M K) -by C upper trapezoidal. $K+L=$ the effective num erical rank of the $M+P)$-by $-N$ m atrix
( $\mathrm{A}, \mathrm{B}$ )'. Z 'denotes the conjugate transpose of $Z$.
This decom position is the preprocessing step for com puting the Generalized Singular V alue D ecom position (G SV D ), se subroutine CG G SV D .

## ARGUMENTS

$J 0 \mathrm{BU}$ (input)
$=\mathrm{U}$ : U nitary matrix U is com puted;
$=N^{\prime}: \mathrm{U}$ is not com puted.
$J O B V$ (input)
$=\mathrm{V}$ : U nitary m atrix V is com puted;
$=\mathrm{N}^{\prime}: \mathrm{V}$ is not com puted.
$J O B Q$ (input)
$=Q$ ': Unitary matrix $Q$ is com puted;
$=\mathrm{N}^{\prime}: \mathrm{Q}$ is notcom puted.

M (input) The num ber of row s of the $m$ atrix $\mathrm{A} . \mathrm{M}>=0$.
$P$ (input) The num ber of row sof the $m$ atrix $B . P>=0$.

N (input) The num ber of colmm ns of the $m$ atrioes $A$ and $B$. $N$ $>=0$.

A (input/output)
On entry, the M by-N m atrix A. On exit, A contains the triangular (or trapezoidal) matrix described in the Punpose section.

LD A (input)
The leading dim ension of the array A. LDA >= max (1, M).

B (input/output)
On entry, the $P-b y-N$ m atrix B. On exit, B contains the triangularm atrix described in the Purpose section.

LD B (input)

The leading dim ension of the array $\mathrm{B} . \operatorname{LD} \mathrm{B}>=$ $\max (1, \mathrm{P})$.

TO LA (input)
TO LA and TO LB are the thresholds to determ ine the effective num erical rank ofm atrix $B$ and a subblock ofA. Generally, they are set to TO LA =
 MAX $(P, N) *$ norm ( $B$ )*M ACHEPS. The size of TOLA and TO LB $m$ ay affect the size of backw ard emors of the decom position.
TOLB (input)
See description of TO LA .
K (output)
O n exit, $K$ and $L$ specify the dim ension of the subblocks described in Punpose section. $\mathrm{K}+\mathrm{L}=$ effective num erical rank of $(A, B)^{\prime}$.

L (output)
See the description ofK .
U (input) If $\mathrm{JOBU}=\mathrm{U}$ ', U contains the unitary matrix U . If $\mathrm{JOBU}=\mathrm{N}$ ', U is not referenced.

LD U (input)
The leading dim ension of the array U. LD U >= $m a x(1, M)$ if $\mathcal{O B B U}=U$; LD $U>=1$ otherw ise.

V (input) If $\mathrm{JO} \mathrm{BV}=\mathrm{V}$ ', V contains the unitary matrix V .
If $\mathrm{JOBV}=\mathrm{N}$ ', V is not referenced.
LD V (input)
The leading dim ension of the array $V$. LDV >= $\max (1, \mathrm{P})$ if $\mathrm{JOBV}=\mathrm{V}$ '; LD V >= 1 otherw ise.
$Q$ (input) If $J B Q=Q$ ', $Q$ contains the unitary $m$ atrix $Q$.
If $\mathrm{JOBQ}=\mathrm{N}^{\prime}, \mathrm{Q}$ is not referenced.
LDQ (input)
The leading dim ension of the aray $Q . L D Q>=$ $\max (1, N)$ if $\operatorname{OOB}=Q ; L D Q>=1$ otherw ise.

IW ORK (w orkspace)
dim ension $\mathbb{N}$ )

RW ORK (w orkspace)
dim ension ( $2 \star \mathrm{~N}$ )
TAU (w orkspace)
dim ension $\mathbb{N}$ )

W ORK (w orkspace)
dim ension $M A X(3 * N, M, P))$
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0$ : if $\mathbb{I N F O}=-i$, the i-th argum enthad an illegalvalue.

## FURTHER DETAILS

The subroutine uses LA PA CK subroutine CGEQPF for the QR factorization w ith column pivoting to detect the effective num erical rank of the a m atrix. It $m$ ay be replaced by a better rank determ ination strategy

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS


## NAME

cgssco -G eneral sparse solver condition num ber estim ate.

## SYNOPSIS

```
SUBROUTINECGSSCO(COND,HANDLE,\mathbb{ER})
```

$\mathbb{N T E G E R \quad \mathbb { E R }}$

REAL COND
DOUBLE PRECISION HANDLE (150)

## PURPOSE

CGSSCO -C ondition num berestim ate.

## PARAMETERS

COND -REAL
On exit, an estim ate of the condition num berof the factored $m$ atrix. M ustbe called after the num erical factorization subroutine, CGSSFA ().

HANDLE (150) -D OUBLE PREC IS IO N anay
On entry, HANDLE ( $*$ ) is an array containing inform ation needed by the solver, and $m$ ust.be passed unchanged to each sparse solver subroutine.
M odified on exit.
$\mathbb{E R} \quad-\mathbb{N} T E G E R$
Enrornum ber. If no errorencountered, unchanged on
exit. If emorencountered, it is set to a non-zero
integer. Errornum bers setby this subroutine:
-700 : Invalid calling sequence - need to call C G SSFA first.
-710 : C ondition num ber estim ate not available (notim plem ented for this H A N D LE sm atix type).

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS


## NAME

cgssda -D eallocate w orking storage for the general.sparse solver.

## SYNOPSIS

SUBROUTINECGSSDA (HANDLE, $\mathbb{E R}$ )
$\mathbb{N}$ TEGER $\quad \mathbb{R}$
DOUBLE PRECISION HANDLE (150)

## PURPOSE

CGSSDA -D eallocate dynam ically allocated w orking storage.

## PARAMETERS

HANDLE (150) -D OUBLE PREC IS IO N array
On entry, HANDLE (*) is an array containing inform ation needed by the solver, and $m$ ustbe passed unchanged to each sparse solver subroutine. M odified on exit.
$\mathbb{E R} \quad-\mathbb{N} T E G E R$
Errornum ber. If no errorencountered, unchanged on exit. If errorencountered, it is set to a non-zero integer. Errornum bers set.by this subroutine:
none

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS


## NAME

cgssfa -G eneral sparse solvernum eric factorization.

## SYNOPSIS

SUBROUTINECGSSFA (NEQNS,COLSTR,ROW $\mathbb{N D}, V A L U E S, H A N D L E, \mathbb{E R}$ )
$\mathbb{N}$ TEGER NEQNS,COLSTR (*),ROW $\mathbb{N D}(*), \mathbb{E R}$
COM PLEX VALUES (*)
DOUBLE PRECISION HANDLE (150)

## PURPOSE

CGSSFA -N um eric factorization of a sparse m atrix.

## PARAMETERS

NEQNS - $\mathbb{N}$ TEGER
On entry, NEQNS specifies the num ber of equations in coefficientm atrix. U nchanged on exit.
$\operatorname{COLSTR}\left(^{*}\right)-\mathbb{N}$ TEG ER array
On entry, $\operatorname{COLSTR}\left(^{*}\right)$ is an array of size $(\mathbb{N E Q N S + 1 ) , ~}$ containing the pointers of them atrix structure.
U nchanged on exit.
ROWIND (*) - $\mathbb{N}$ TEG ER anay
On entry, ROWIND ( ${ }^{*}$ ) is an array of size
CO LSTR $\mathbb{N} E Q N S+1)-1$, containing the indices of the $m$ atrix structure. U nchanged on exit.
$\operatorname{VALUES}$ ( ${ }^{\star}$ ) -COM PLEX aray
On entry, VALUES ( ${ }^{\star}$ ) is an array of size
CO LSTR $\mathbb{N E Q N S}+1$ )-1, containing the num eric values of
the sparse $m$ atrix to be factored. U nchanged on exit.

HANDLE (150) -D OUBLE PRECISIO N array On entry, HANDLE (*) is an array containing inform ation needed by the solver, and $m$ ust.be passed unchanged to each sparse solver subroutine. M odified on exit.
$\mathbb{E R} \quad-\mathbb{N} T E G E R$
Errornum ber. If no emrorencountered, unchanged on exit. If errorencountered, it is set to a non-zero integer. Enrornum bers set.by this subroutine:
-300 : Invalid calling sequence - need to callC G SSO R first.
-301 : Failure to dynam ically allocate $m$ em ory.
-666 : Intemalerror.

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS


## NAME

cgssis -G eneral sparse solver one call interface.

## SYNOPSIS

```
SUBROUTINE CGSSFS(M TXTYP,PIVOT,NEQNS,COLSTR,ROW IND,
        VALUES,NRHS ,RHS ,LDRHS,ORDM THD,
        OUTUNT,MSGLVL,HANDLE,\mathbb{ER)}
```

CHARACTER*2 M TXTYP
CHARACTER*1 PIVOT
$\mathbb{N}$ TEGER NEQNS,COLSTR (*),ROW $\mathbb{N} D\left({ }^{*}\right)$,NRHS,LDRHS,
OUTUNT,MSGLVL, $\mathbb{E R}$
CHARACTER*3 ORDMTHD
COMPLEX VALUES (*),RHS (*)
DOUBLE PRECISION HANDLE (150)

## PURPOSE

CGSSFS -G eneral sparse solver one call interface.

## PARAMETERS

MTXTYP -CHARACTER*2
On entry, M TX TY P specifies the coefficientm atrix type. Specifically, the valid options are:

Sp 'or SP '-sym m etric structure, H erm itian positive definite
values
ss'or SS '-sym m etric structure, sym $m$ etric values
su 'or SU '-sym m etric structure, unsym $m$ etric values
uu 'or UU '-unsym $m$ etric structure, unsym $m$ etric values
U nchanged on exit.

## PIVOT -CHARACTER*1

O n entry, pivotspecifies w hetherornotpivoting is used in the course of the num eric factorization. The valid options are:
h'or $\mathrm{N}^{\prime}$-no pivoting is used
(Pivoting is notsupported forthis release).

U nchanged on exit.

## NEQNS - $\mathbb{N}$ TEGER

On entry, N EQ N S specifies the num ber ofequations in the coefficientm atrix. N EQ N S m ustbe at least one. U nchanged on exit.
$\operatorname{COLSTR}$ (*) - $\mathbb{N}$ TEG ER aray
On entry, COLSTR (*) is an array of size (NEQNS+1), containing the pointers of them atrix structure.
U nchanged on exit.
ROWIND (*) - $\mathbb{N}$ TEG ER anay
On entry, ROWIND (*) is an array of size COLSTR $\mathbb{N E Q N S}+1)-1$, containing the indices of the $m$ atrix structure. U nchanged on exit.

## VALUES (*) -COM PLEX aray

O $n$ entry, $V A L U E S$ (*) is an aray of size
COLSTR $\mathbb{N} E Q N S+1$ )-1, containing the non-zero num eric values of the sparse $m$ atrix to be factored. U nchanged on exit.

NRHS - INTEGER
On entry, N RH S specifies the num ber of right hand sides to solve for. U nchanged on exit.

RHS (*) -COM PLEX aray
On entry, RH S (LD RH S NRHS) contains the NRHS right hand sides. On exit, itcontains the solutions.

LDRHS - $\mathbb{N}$ TEGER
O n entry, LD RH S specifies the leading dim ension of the RH S array. U nchanged on exit.

ORDMTHD -CHARACTER*3
On entry, ORD M THD specifies the fill-reducing ordering to be used by the sparse solver.
Specifically, the valid options are:
hat'or NAT'-natural ordering (no ordering) mmd'or M M D '-m ultiplem inim um degree
gnd 'or GND '-general nested dissection
uso 'or U SO '-user specified ordering (see C G SSU O )
U nchanged on exit.

OUTUNT - $\mathbb{N}$ TEGER
O utputunit. U nchanged on exit.
M SG LVL - $\mathbb{N}$ TEGER
M essage level.

0 -no output from solver.
( $N$ o m essages supported for this release.)
U nchanged on exit.
HANDLE (150) -DOUBLE PRECISIO N aray
On entry, HANDLE (*) is an amay of containing
inform ation needed by the solver, and $m$ ustibe passed
unchanged to each sparse solver subroutine.
M odified on exit.
$\mathbb{E R} \quad-\mathbb{N} T E G E R$
Enrornum ber. If no errorencountered, unchanged on
exit. If emorencountered, it is set to a non-zero
integer. Errornum bers set.by this subroutine:
-101: Failure to dynam ically allocate $m$ em ory.
-102 : Invalid m atrix type.
-103 : Invalid pivotoption.
-104: N um berofnonzeros is less than N EQ N S .
-105: NEQNS < 1
-201 : Faihure to dynam ically allocate $m$ em ory.
-301 : Failure to dynam ically allocate $m$ em ory.
-401 : Failure to dynam ically allocate $m$ em ory.
-402 :NRHS < 1
-403 : NEQN S > LD RHS
-666 : Intemalerror.

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS


## NAME

cgssin -Initialize the generalsparse solver.

## SYNOPSIS

```
SUBROUT\mathbb{NE CGSSIN (M TXTYP,PIVOT,NEQNS,COLSTR,ROW IND,OUTUNT,}
                        M SGLVL,HANDLE,\mathbb{ER )}
CHARACTER*2 M TXTYP
CHARACTER*1 PIVOT
\mathbb{NTEGER NEQNS,COLSTR (*),ROW IND (*),OUTUNT,MSGLVL,\mathbb{ER}}\mathbf{N}\mathrm{ (*)}
DOUBLE PRECISION HANDLE (150)
```


## PURPOSE

CGSSIN -Initialize the sparse solver and input the $m$ atrix
structure.

## PARAMETERS

## MTXTYP -CHARACTER*2

On entry, M TX TY P specifies the coefficientm atrix type. Specifically, the valid options are:

Sp 'or SP '-sym m etric structure, H erm itian positive definite
ss'or SS '-sym m etric structure, sym m etric values
su 'or SU '-sym m etric structure, unsym $m$ etric values
uu 'or UU '-unsym $m$ etric structure, unsym $m$ etric values
U nchanged on exit.

On entry, PIV OT specifies whether ornotpivoting is
used in the course of the num eric factorization.
The valid options are:
h'or $\mathrm{N}^{\prime}$-no pivoting is used
(Pivoting is not supported forthis release).

U nchanged on exit.
NEQNS - $\mathbb{N} T E G E R$
On entry, NEQNS specifies the num berofequations in the coefficientm atrix. N EQ N S m ustbe at leastone. U nchanged on exit.
$\operatorname{COLSTR}$ ( $\left.^{*}\right)-\mathbb{N}$ TEG ER amay
On entry, $\operatorname{COLSTR}$ ( ${ }^{*}$ ) is an array of size $\mathbb{N E Q N S + 1 ) \text { , }}$ containing the pointers of them atrix structure.
U nchanged on exit.
ROWIND ( ${ }^{\star}$ ) - $\mathbb{N}$ TEG ER array
On entry, ROWIND ( ${ }^{\star}$ ) is an array of size
CO LSTR $\mathbb{N} E Q N S+1$ )-1, containing the indices of the $m$ atrix structure. U nchanged on exit.

HANDLE (150) -D OUBLE PRECIS IO N array
On entry, $H A N D L E$ (*) is an array containing inform ation needed by the solver, and $m$ ust.be passed unchanged to each sparse solver subroutine. M odified on exit.

OUTUNT - $\mathbb{N}$ TEGER
O utputunit. U nchanged on exit.

M SGLVL - $\mathbb{N}$ TEGER
M essage level.
0 -no output from solver.
(N o m essages supported for this release.)
U nchanged on exit.
$\mathbb{E R} \quad$ - $\mathbb{N}$ TEGER
Enrornum ber. If no errorencountered, unchanged on
exit. If emrorencountered, it is set to a non-zero
integer. Enrornum bers setby this subroutine:
-101: Failure to dynam ically allocate mem ory.
-102 : Invalid m atrix type.
-103 : Invalid pivotoption.
-104 : N um berofnonzeros less than N EQ N S .
-105:NEQNS<1

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS


## NAME

cgssor-G eneral sparse solver ordering and sym bolic factorization.

## SYNOPSIS

```
SUBROUTINECGSSOR(ORDMTHD,HANDLE,\mathbb{ER})
CHARACTER*3 ORDMTHD
\mathbb{NTEGER ER}
DOUBLE PRECISION HANDLE (150)
```


## PURPOSE

CGSSOR -O rders and sym bolically factors a sparse m atrix .

## PARAMETERS

ORDMTHD -CHARACTER*3
On entry, ORDM THD specifies the fill-reducing ordering to be used by the sparse solver. Specifically, the valid options are:
hat'or NAT '-naturalordering (no ordering)
mmd'or M M D '-m ultiplem inim um degree
gnd 'or GND '-generalnested dissection
uso 'or U SO '-user specified ordering (see C G SSU O )
U nchanged on exit.
HANDLE (150) -DOUBLE PRECISIO N aray
On entry, HANDLE (*) is an array containing
inform ation needed by the solver, and $m$ ust.be passed unchanged to each sparse solver subroutine.

M odified on exit.
$\mathbb{E R} \quad-\mathbb{N} T E G E R$
Errornum ber. If no errorencountered, unchanged on
exit. If emrorencountered, it is set to a non-zero
integer. Enrornum bers set.by this subroutine:
-200 : Invalid calling sequence - need to callC G SSIN first.
-201 : Failure to dynam ically allocate $m$ em ory.
-666 : Intemalerror.

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS


## NAME

cgssps -P rint general.sparse solver statics.

## SYNOPSIS

SUBROUTINECGSSPS (HANDLE, $\mathbb{E R}$ )
$\mathbb{N} T E G E R \quad \mathbb{E R}$
DOUBLE PRECISION HANDLE (150)

## PURPOSE

CGSSPS -Print solver statistics.

## PARAMETERS

HANDLE (150) -DOUBLE PRECISION aray
On entry, HANDLE ( $*$ ) is an amay containing
inform ation needed by the solver, and $m$ ust.be passed
unchanged to each sparse solver subroutine.
M odified on exit.

ER

- $\mathbb{N}$ TEGER

Errornum ber. If no errorencountered, unchanged on
exit. If errorencountered, it is set to a non-zero
integer. Errornum bers setby this subroutine:
-800: Invalid calling sequence - need to callC G SSSL first.
-899 : Printed solver statistics not supported this release.

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS


## NAME

cgssip -R etum perm utation used by the general sparse solver.

## SYNOPSIS

SUBROUTINECGSSRP (PERM, HANDLE, $\mathbb{E R}$ )
$\mathbb{N} T E G E R \quad$ PERM (*), $\mathbb{E R}$
DOUBLE PRECISION HANDLE (150)

## PURPOSE

CGSSRP -R etums the perm utation used by the solver for the fill-reducing ordering.

## PARAMETERS

PERM $\mathbb{N E Q N S}$ ) - $\mathbb{N}$ TEGER amay
U ndefined on entry. PERM $\mathbb{N E Q N S}$ ) is the perm utation array used by the sparse solver for the fillreducing ordering. M odified on exit.

HANDLE (150) -D OUBLE PRECISION array
On entry, HANDLE ( $*$ ) is an array containing
inform ation needed by the solver, and $m$ ust.be passed unchanged to each sparse solver subroutine.
M odified on exit.
$\mathbb{E R} \quad-\mathbb{N} T E G E R$
Errornum ber. If no errorencountered, unchanged on exit. If errorencountered, it is set to a non-zero
integer. Errornum bers set.by this subroutine:
-600 : Invalid calling sequence - need to callC G SSO R first.

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS


## NAME

cgsssl-Solve routine for the general sparse solver.

## SYNOPSIS

SUBROUTINECGSSSL (NRHS,RHS,LDRHS,HANDLE, ER )
$\mathbb{N}$ TEGER NRHS,LDRHS, $\mathbb{E R}$
COM PLEX RHS (LDRHS,NRHS)
DOUBLE PRECISION HANDLE (150)

## PURPOSE

CGSSSL -Triangular solve of a factored sparse m atrix.

## PARAMETERS

NRHS - $\mathbb{N}$ TEGER
On entry, N RH S specifies the num ber of righthand
sides to solve for. U nchanged on exit.

RHS (LDRHS,*) -COM PLEX array
On entry, RH S (LDRHS,NRHS) contains the NRHS right hand sides. On exit, itcontains the solutions.

LDRHS - $\mathbb{N}$ TEGER
On entry, LD RH S specifies the leading dim ension of the RH S amay. U nchanged on exit.

HANDLE (150) -D OUBLE PREC ISIO N aray
O n entry, HANDLE ( ${ }^{\star}$ ) is an array containing
inform ation needed by the solver, and $m$ ustbe passed
unchanged to each sparse solver subroutine.
M odified on exit.

ER

- $\mathbb{N}$ TEGER

Errornum ber. If no errorencountered, unchanged on
exit. If errorencountered, it is set to a non-zero
integer. Errornum bers setby this subroutine:
-400 : Invalid calling sequence - need to callC G SSFA first.
-401 : Failure to dynam ically allocate $m$ em ory.
-402 : NRHS < 1
-403 : NEQN S > LD RHS

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS


## NAME

cgssuo -U ser supplied perm utation for ordering used in the generalsparse solver.

## SYNOPSIS

SUBROUTINECGSSUO (PERM,HANDLE, $\mathbb{E R}$ )
$\mathbb{N}$ TEGER PERM (*), $\mathbb{E R}$
DOUBLE PRECISION HANDLE (150)

## PURPOSE

CGSSUO -U ser supplied perm utation for ordering. M ust.be called after CGSS IN 0 (sparse solver initialization) and before CGSSOR () (sparse solver ordering).

## PARAMETERS

PERM $\mathbb{N} E Q N S$ ) - $\mathbb{N}$ TEGER array
On entry, PERM (NEQNS) is a perm utation array supplied by the user for the fill-reducing ordering.
U nchanged on exit.

HANDLE (150) -D OUBLE PREC IS IO N array
On entry, HANDLE (*) is an anay containing
inform ation needed by the solver, and $m$ ustbe passed
unchanged to each sparse solver subroutine.
M odified on exit.
$\mathbb{E R} \quad-\mathbb{N} T E G E R$
Errornum ber. If no errorencountered, unchanged on
exit. If emorencountered, it is set to a non-zero
integer. Errornum bers setby this subroutine:
-500 : Invalid calling sequence - need to callC G SSIN first.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgtcon -estim ate the reciprocal of the condition num ber of a com plex tridiagonalm atrix A using the LU factorization as com puted by CG TTRF

## SYNOPSIS

```
SUBROUT\mathbb{NECGTCON NORM,N,LOW,D IAG,UP1,UP2,\mathbb{PIVOT,ANORM,RCOND,}}\mathbf{N},
    W ORK,INFO)
CHARACTER * 1 NORM
COM PLEX LOW (*),D IAG (*),UP1 (*),UP2 (*),W ORK (*)
INTEGERN,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}
REAL ANORM,RCOND
```



```
        RCOND,W ORK,\mathbb{NFO)}
CHARACTER * 1 NORM
COM PLEX LOW (*),D IAG (*),UP1 (*),UP2 (*),W ORK (*)
INTEGER*8N,\mathbb{NFO}
\mathbb{NTEGER*8 P\mathbb{IVOT (*)}}\mathbf{(})
REAL ANORM,RCOND
```


## F95 INTERFACE

```
SU BROUTINE GTCON \(\mathbb{N} O R M, \mathbb{N}], L O W, D I A G, U P 1, U P 2, \mathbb{P} \mathbb{I V O T}, A N O R M\), RCOND, [WORK], [ \(\mathbb{N F O}\) ])
CHARACTER (LEN=1) ::NORM
COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::LOW ,D IA G ,UP1,UP2,W ORK
\(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V O T}\)
```

REAL ::ANORM,RCOND

SU BROUTINE GTCON_64 $\mathbb{N} O R M, \mathbb{N}], L O W, D I A G, U P 1, U P 2, \mathbb{P} \mathbb{I} O T, A N O R M$, RCOND, $[\mathbb{W} O R K],[\mathbb{N F O}])$

CHARACTER (LEN=1) ::NORM
COM PLEX,D $\mathbb{I}$ ENSION (:) :: LOW ,D IAG , UP1, UP2,W ORK
$\mathbb{N} T E G E R(8):: N, \mathbb{N F O}$
$\mathbb{N}$ TEGER (8), D $\mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{V} O T$
REAL ::ANORM,RCOND

## C INTERFACE

\#include <sunperfh>
void ogtoon (charnorm , intn, com plex *low , com plex *diag, complex *up1, complex *up2, int*ipivot, float anorm, float *rcond, int*info);
void cgtcon_64 (charnorm, long n, com plex *low, com plex *diag, com plex *up1, com plex *up2, long *ípivot, floatanorm , float *rcond, long *info);

## PURPOSE

cgtcon estim ates the reciprocal of the condition num ber of a com plex tridiagonal $m$ atrix A using the LU factorization as com puted by C G TTRF .

A $n$ estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND $=1 /(A N O R M *$ norm (inv (A))).

## ARGUMENTS

NORM (input)
Specifies w hether the 1-norm condition num ber or the infinity-norm condition num ber is required:
= ' 'or $\mathrm{D}^{\prime}$ : 1-norm;
= I ': Infinity-norm .

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

LOW (input)
The $(n-1) \mathrm{m}$ ultipliers that define the $m$ atrix $L$ from the LU factorization of $A$ as com puted by CGTTRF.

D IA G (input)

Then diagonalelem ents of the upper triangular $m$ atrix $U$ from the $L U$ factorization ofA.

UP1 (input)
The ( $n-1$ ) elem ents of the first superdiagonal of U.

UP2 (input)
The ( $n-2$ ) elem ents of the second superdiagonal of U.
$\mathbb{P I V O T}$ (input)
The pivotindices; for $1<=i<=n$, row $i$ of the matrix was interchanged with row PIVOT (i). IPIVOT (i) will always be either $i$ or i+1; PIVOT (i) = iindicates a row interchange w as not required.

ANORM (input)
IfNORM = ' 1 'or $\mathrm{D}^{\prime}$ ', the 1-nom of the original $m$ atrix $A$. IfNORM = I', the infinity-norm of the originalm atrix A .

RCOND (output)
The reciprocal of the condition number of the $m$ atrix $A$, com puted as RCOND $=1 /(A N O R M * A \mathbb{N} V N M)$, where $A \mathbb{N} V N M$ is an estim ate of the 1 -nom of $\operatorname{inv}(A)$ com puted in this routine.

W ORK (w orkspace)
dim ension $(2 * N)$
$\mathbb{N}$ FO (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS


## NAME

cgthr-G athers specified elem ents from $y$ into $x$.

## SYNOPSIS

SUBROUTINECGTHR(NZ,Y,X, $\mathbb{N} D X)$
COM PLEX Y (*), X (*)
$\mathbb{N}$ TEGER NZ
$\mathbb{N} T E G E R \mathbb{N} D X(*)$
SUBROUTINECGTHR_64 $\mathbb{N} Z, Y, X, \mathbb{N} D X)$
COM PLEX Y (*), X (*)
$\mathbb{N}$ TEGER*8NZ
$\mathbb{I N} T E G E R * 8 \mathbb{N} D X(*)$
F95 $\mathbb{I N}$ TERFACE
SUBROUTINE GTHR ( $\mathbb{N} Z], Y, X, \mathbb{N} D X)$
COMPLEX,D $\mathbb{M}$ ENSION (:) :: Y,X
$\mathbb{N} T E G E R:: N Z$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{N D X}$
SUBROUTINE GTHR_64( $\mathbb{N} Z], Y, X, \mathbb{N} D X)$
COM PLEX,D $\mathbb{I M} E N S I O N(:):$ :,$X$
$\mathbb{N}$ TEGER (8) ::NZ
$\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{N D X}$

## PURPOSE

CGTHR-G athers the specified elem ents from a vectory in fullstorage form into a vectorx in com pressed form. Only
the elem ents of $y$ w hose indices are listed in indx are referenced.
do $i=1, n$ $x(i)=y($ indx $(i))$
enddo

## ARGUMENTS

N Z (input) - $\mathbb{N}$ TEGER
$N$ um ber of elem ents in the com pressed form .
U nchanged on exit.
$Y$ (input)
V ectorin fullstorage form . U nchanged on exit.
X (output)
V ector in com pressed form. C ontains elem ents ofy
whose indices are listed in indx on exit.
$\mathbb{I N D X}$ (input) - $\mathbb{N}$ TEGER
$V$ ector containing the indices of the com pressed
form. It is assum ed that the elem ents in $\mathbb{N} D \mathrm{X}$ are
distinct and greater than zero. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS


## NAME

cgthrz -G ather and zero.

## SYNOPSIS

SUBROUTINECGTHRZ $\mathbb{N} Z, Y, X, \mathbb{N} D X)$
COM PLEX Y (*), X (*)
$\mathbb{N}$ TEGER NZ
$\mathbb{N} T E G E R \mathbb{N} D X(*)$
SUBROUTINECGTHRZ_64 $\mathbb{N} Z, Y, X, \mathbb{N} D X)$
COM PLEX Y (*), X (*)
$\mathbb{N}$ TEGER*8NZ
$\mathbb{I N} T E G E R * 8 \mathbb{N} D X(*)$
F95 $\mathbb{I N}$ TERFACE
SUBROUTINE GTHRZ (NZ], Y, X, $\mathbb{N} D \mathrm{X}$ )
COM PLEX,D IM ENSION (:) :: Y, X
$\mathbb{N}$ TEGER ::NZ
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{N D X}$
SUBROUTINEGTHRZ_64(NZ],Y,X, $\mathbb{N} D X)$
COM PLEX,D $\mathbb{I M} E N S I O N(:):$ :,$X$
$\mathbb{N}$ TEGER (8) ::NZ
$\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{N D X}$

## PURPOSE

CGTHRZ-G athers the specified elem ents from a vectory in fullstorage form into a vectorx in com pressed form. The
gathered elem ents ofy are set to zero. O nly the elem ents ofy w hose indices are listed in indx are referenced.

```
do i=1,n
    x (i) = y (indx (i))
    y(indx (i)) = 0
enddo
```


## ARGUMENTS

N Z (input) - $\mathbb{N}$ TEGER
N um ber of elem ents in the com pressed form .
U nchanged on exit.

Y (input/output)
V ector in fullstorage form. G athered elem ents are setto zero.
X (output)
V ector in com pressed form. C ontains elem ents ofy w hose indices are listed in indx on exit.
$\mathbb{N} D X$ (input) - $\mathbb{N} T E G E R$
V ector containing the indiges of the com pressed form. It is assum ed that the elem ents in $\mathbb{N D} X$ are distinctand greater than zero. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgtrfs - im prove the com puted solution to a system of linear equations when the coefficientm atrix is tridiagonal, and provides enrorbounds and backw ard enrorestim ates for the solution

## SYNOPSIS

```
SU BROUT\mathbb{NE CGTRFS (TRANSA,N,NRHS,LOW ,D IAG,UP,LOW F,D IAGF,UPF1,}
    UPF2,\mathbb{PIVOT,B,LDB,X,LDX,FERR,BERR,W ORK,W ORK 2, INFO)}
CHARACTER * 1 TRANSA
COM PLEX LOW (*),D IAG (*),UP (*),LOW F (*),D IAGF (*), UPF1 (*),
UPF2 (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,INFO
INTEGER \mathbb{PIVOT (*)}
REAL FERR (*),BERR (*),WORK2 (*)
SUBROUT\mathbb{NE CGTRFS_64 (TRANSA,N,NRHS,LOW ,D IAG,UP,LOW F,D IAGF,}
        UPF1,UPF2,\mathbb{PIVOT,B,LDB,X,LDX,FERR,BERR,W ORK,W ORK2,}
        INFO)
CHARACTER * 1 TRANSA
COM PLEX LOW (*),D IAG (*),UP (*),LOW F (*),D IAGF (*), UPF1 (*),
UPF2 (*),B (LD B,*),X (LDX ,*),W ORK (*)
\mathbb{NTEGER*8N,NRHS,LDB,LDX,}\mathbb{N}FO
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
REAL FERR (*),BERR (*),W ORK ( (*)
```


## F95 INTERFACE

SU BROUTINE GTRFS ([TRANSA ], $\mathbb{N}], \mathbb{N} R H S], L O W, D I A G, U P, L O W ~ F, D ~ I A G F$, UPF1, UPF2, $\mathbb{P} \mathbb{I V}$ OT,B, [LDB],X, [LDX],FERR,BERR, [W ORK], [W ORK2], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1) ::TRANSA
COM PLEX , D $\mathbb{M}$ ENSION (:) ::LOW ,D IAG, UP,LOW F, DIAGF, UPF1, UPF2,W ORK
COM PLEX, D $\mathbb{I}$ ENSION (:,: : : B, X
$\mathbb{N}$ TEGER :: N, NRHS,LDB,LDX, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V O T}$
REAL,D $\mathbb{M} E N S I O N(:):: F E R R, B E R R$, W ORK 2

SU BROUTINE GTRFS_64 ([TRANSA], $\mathbb{N}], \mathbb{N} R H S], L O W, D \mathbb{I} G, U P, L O W F$, D $\mathbb{A} G F, U P F 1, U P F 2, \mathbb{P} \mathbb{I} O T, B,[L D B], X,[L D X], F E R R, B E R R,[W O R K]$, [ $W$ ORK2], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::TRANSA
COM PLEX , D $\mathbb{M}$ ENSION (:) ::LOW ,D $\mathbb{I A G}, \mathrm{UP}$, LOW F, D IA GF, UPF1, UPF2,W ORK
COM PLEX, D $\mathbb{I M}$ ENSION (:,:) ::B,X
$\mathbb{N}$ TEGER (8) :: N , NRHS,LDB,LDX, $\mathbb{N}$ FO
$\mathbb{N} \operatorname{TEGER}$ (8), D $\mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$
REAL,D $\mathbb{M}$ ENSION (:) ::FERR,BERR,W ORK2

## C INTERFACE

\#include <sunperfh>
void cgtrfs (char transa, intn, intnrhs, com plex *low , com plex *diag, com plex *up, com plex *low f, com plex *diagf, com plex *upfl, com plex *upf2, int*ipivot, com plex *b, int ldb, com plex *x, int ldx, float * ferr, float *berr, int*info);
void cgtrfs_64 (chartransa, long n, long nrhs, com plex *low , com plex *diag, com plex *up, com plex *low f, com plex *diagf, complex *upfl, complex *upf2, long *ipivot, com plex *b, long ldb, com plex *x, long ldx, float * ferr, float *berr, long *info);

## PURPOSE

cgtrfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is tridiagonal, and provides errorbounds and backw ard error estim ates for the solution.

## ARGUMENTS

TRANSA (input)
Specifies the form of the system of equations:
$=N^{\prime}: A * X=B \quad$ ( 0 transpose)
$=T$ ': $A * * T * X=B \quad$ ( ranspose)
$=C$ ': $A * * H * X=B \quad$ (C onjugate transpose)

TRANSA is defaulted to $N$ 'forF $95 \mathbb{I N}$ TERFACE.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRH S $>=0$.

LOW (input)
The ( $n-1$ ) subdiagonalelem ents ofA.
D IA G (input)
The diagonalelem ents ofA.

UP (input)
The ( $\mathrm{n}-1$ ) superdiagonalelem ents of A.

LOW F (input)
The $(n-1) \mathrm{m}$ ultipliers that define the $m$ atrix $L$ from the LU factorization of $A$ as com puted by CGTTRF.

D IA GF (input)
Then diagonalelem ents of the upper triangular $m$ atrix $U$ from the $L U$ factorization ofA .

UPF1 (input)
The ( $n-1$ ) elem ents of the first superdiagonal of U.

UPF2 (input)
The ( $n-2$ ) elem ents of the second superdiagonal of U.

IPIVOT (input)
The pivotindioes; for $1<=i<=n$, row $i$ of the $m$ atrix $w a s$ interchanged w th row $\mathbb{P I V O T}(i)$. PIVOT (i) will always be either $i$ or $i+1$; $\mathbb{P} \mathbb{I V}$ OT (i) = i indicates a row interchange w as not required.

B (input) The righthand side m atrix B .

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.

X (input/output)
O $n$ entry, the solution $m$ atrix $X$, as com puted by CG TTRS. On exit, the im proved solution $m$ atrix $X$.

LD X (input)
The leading dim ension of the anay X . LD X >= $\max (1, N)$.

FERR (output)
The estim ated forw ard emrorbound for each solution vector $X()$ ) the $j$ th column of the solution matrix X). If XTRUE is the true solution corresponding to $X(\mathcal{H}), \operatorname{FERR}(\mathcal{)})$ is an estim ated upperbound for the $m$ agnitude of the largest ele$m$ ent in ( $X(\mathcal{i})-X$ TRUE) divided by the $m$ agnitude of the largestelem ent in $X(j)$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

## BERR (output)

The com ponentw ise relative backw ard error of each solution vector $X$ ( $j$ ) (ie., the sm allest relative change in any elem entofA orB thatm akes $X(\mathcal{J})$ an exactsolution).

W ORK (w orkspace)
dim ension ( $2 * \mathrm{~N}$ )

W ORK 2 (w orkspace)
dim ension $(\mathbb{N})$
$\mathbb{I N F O}$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgtsv -solve the equation $A * X=B$,

## SYNOPSIS

```
SUBROUT\mathbb{NECGTSV N,NRHS,LOW ,D IAG,UP,B,LDB,\mathbb{NFO)}}\mathbf{N},\textrm{L}
COM PLEX LOW (*),DIAG (*),UP (*),B (LD B,*)
\mathbb{NTEGER N,NRHS,LDB,INFO}
SUBROUT\mathbb{NECGTSV_64 N,NRHS,LOW,DIAG,UP,B,LDB,INFO)}
COM PLEX LOW (*),D IAG (*),UP (*),B (LDB,*)
INTEGER*8N,NRHS,LDB,INFO
```

F95 INTERFACE
SUBROUTINEGTSV ( $\mathbb{N}], \mathbb{N} R H S], L O W, D \mathbb{I A G}, \mathrm{UP}, \mathrm{B},[\operatorname{LDB}],[\mathbb{N} F O])$
COM PLEX,D $\mathbb{I}$ ENSION (:) ::LOW ,D $\mathbb{A} G, U P$
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : B
$\mathbb{N}$ TEGER :: $\mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} B, \mathbb{N}$ FO
SUBROUTINEGTSV_64 (N ], $\mathbb{N} R H S], L O W, D \mathbb{I} G, U P, B,[L D B],[\mathbb{N F O}])$
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::LOW ,D IAG,UP
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : B
$\mathbb{N}$ TEGER (8) :: N,NRHS,LDB, $\mathbb{N}$ FO
C INTERFACE
\#include <sunperfh>
void cgtsv (intn, intnrhs, com plex *low, com plex *diag,
com plex *up, com plex *b, int ldb, int*info);
void cgtsv_64 (long n, long nrhs, com plex *low, com plex *diag, com plex *up, com plex *b, long ldb, long *info);

## PURPOSE

cgtsv solves the equation
where $A$ is an $N$-by N tridiagonalm atrix, by $G$ aussian elim ination $w$ ith partialpivoting.

N ote that the equation $\mathrm{A} * \mathrm{X}=\mathrm{B}$ may be solved by interchanging the order of the argum ents $D U$ and $D L$.

## ARGUMENTS

N (input) The order of the matrix A. $\mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS >=0.

LOW (input/output)
On entry, LOW m ust contain the ( $n-1$ ) subdiagonal elem ents ofA. On exit, LOW is overw ritten by the ( $n-2$ ) elem ents of the second superdiagonal of the upper triangular $m$ atrix $U$ from the $L U$ factorization ofA, in LOW (1), ..., LOW (n-2).

D IA G (input/output)
On entry, D IA G mustcontain the diagonal elem ents of A. On exit, D IA G is overw ritten by the $n$ diagonalelem ents of $U$.

UP (input/output)
O n entry, UP m ust contain the ( $n-1$ ) superdiagonal elem ents of A. On exit, UP is overw ritten by the $(n-1)$ elem ents of the first superdiagonal of $U$.

B (input/output)
On entry, the N -by-NRH S righthand side m atrix $B$. On exi, if $\mathbb{N} F O=0$, the $N$ by N RH S solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the array B. LD B >=
$\max (1, N)$.
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$ th argum enthad an illegalvalue
$>0:$ if $\mathbb{N F O}=i, U(i, i)$ is exactly zero, and the solution has notbeen com puted. The factorization has notbeen com pleted unless $i=N$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgtsvx -use the LU factorization to com pute the solution to a complex system of linearequations $\mathrm{A} * \mathrm{X}=\mathrm{B}, \mathrm{A} * * \mathrm{~T} * \mathrm{X}=$ $B$,orA ${ }^{* *} H * X=B$,

## SYNOPSIS

```
SUBROUTINECGTSVX (FACT,TRANSA,N,NRHS,LOW,DIAG,UP,LOW F,D IAGF,
    UPF1,UPF2,\mathbb{PIVOT,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,}
    W ORK2, \mathbb{NFO)}
CHARACTER * 1 FACT,TRANSA
COM PLEX LOW (*),DIAG (*),UP (*),LOW F (*),DIAGF (*), UPF1 (*),
UPF2 (*),B (LD B ,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}
REALRCOND
REAL FERR (*),BERR (*),W ORK2 (*)
SUBROUTINE CGTSVX_64(FACT,TRANSA,N,NRHS,LOW,DIAG,UP,LOW F,
    D IAGF,UPF1,UPF2,\mathbb{PIVOT,B,LDB,X,LDX,RCOND,FERR,BERR,}
    W ORK,W ORK2, INFO)
CHARACTER * 1 FACT,TRANSA
COM PLEX LOW (*),D IAG (*),UP (*),LOW F (*),D IAGF (*), UPF1 (*),
UPF2 (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGER*8N,NRHS,LDB,LDX, INFO
INTEGER*8 \mathbb{PIVOT (*)}
REALRCOND
REAL FERR (*),BERR (*),WORK2 (*)
```


## F95 INTERFACE

SU BROUTINE GTSVX (FACT, [TRANSA], $\mathbb{N}], \mathbb{N} R H S], L O W, D I A G, U P, L O W F$,

D $\mathbb{I A G F}, \mathrm{UPF} 1, \mathrm{UPF} 2, \mathbb{P} \mathbb{I} O T, B,[L D B], \mathrm{X},[\operatorname{LD}], R C O N D, F E R R, B E R R$, [W ORK], [W ORK 2], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1) ::FACT,TRANSA
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::LOW ,D IAG,UP,LOW F, D IA GF, UPF1, UPF2,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : B, X
$\mathbb{N}$ TEGER ::N,NRHS,LDB,LDX, $\mathbb{N}$ FO
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T$
REAL ::RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::FERR,BERR,W ORK2

SUBROUTINE GTSVX_64 (FACT, [TRANSA], $\mathbb{N}], \mathbb{N} R H S], L O W, D I A G, U P, L O W F$, D $\mathbb{A} G \mathrm{G}, \mathrm{UPF} 1, \mathrm{UPF} 2, \mathbb{P} \mathbb{I V O T}, \mathrm{~B},[\mathrm{LD} \mathrm{B}], \mathrm{X},[\mathrm{LD} \mathrm{X}], R C O N D, F E R R, B E R R$, [W ORK], [W ORK 2], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1) ::FACT,TRANSA
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::LOW ,D IA G,UP,LOW F, D IA GF, UPF1, UPF2,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : B, X
$\mathbb{N}$ TEGER (8) ::N ,NRHS,LDB,LDX, $\mathbb{N}$ FO
$\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T$
REAL ::RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::FERR,BERR,W ORK 2

## C INTERFACE

\#include < sunperfh>
void cgtsvx (char fact, chartransa, intn, intnrhs, com plex *low, com plex *diag, com plex *up, com plex *low f, com plex *diagf, com plex *upfl, com plex *upf2, int *ịíivot, com plex *b, int ldb, com plex *x, int ldx, float *rcond, float *ferr, float *berr, int *info);
void cgtsvx_64 (char fact, char transa, long n, long nihs, com plex *low, com plex *diag, com plex *up, com plex *low f, com plex *diagf, com plex *upfl, com plex *upf2, long *ípívot, com plex *b, long ldb, com plex
*x, long ldx, float *rcond, float *ferr, float
*bers, long *info);

## PURPOSE

cgtsvx uses the LU factorization to com pute the solution to a complex system of linearequations $\mathrm{A} * \mathrm{X}=\mathrm{B}, \mathrm{A} * * \mathrm{~T}$ * $\mathrm{X}=$ $B$, orA $* * H * X=B, w h e r e A$ is a tridiagonalm atrix of order N and X and B are N -by N RH S m atrices.

E rrorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:

1. IfFACT $=N$ ', the LU decom position is used to factor the $m$ atrix A
as $A=L * U, w h e r e L$ is a product of perm utation and unit low er
bidiagonal m atrices and U is upper triangular w th
nonzeros in
only the $m$ ain diagonal and first tw o superdiagonals.
2. If som e $U(i, i)=0$, so that $U$ is exactly singular, then the routine
retums w ith $\mathbb{N}$ FO $=$ i. $O$ therw ise, the factored form of $A$ is used
to estim ate the condition num ber of the $m$ atrix $A$. If the reciprocal of the condition num ber is less than $m$ achine precision,
$\mathbb{N} F O=N+1$ is retumed as a waming, but the routine stillgoes on
to solve for X and com pute error bounds as described below .
3.The system ofequations is solyed for X using the factored form of A.
3. Iterative refinem ent is applied to im prove the com puted solution
m atrix and calculate error bounds and backw ard error estim ates
for it.

## ARGUMENTS

FACT (input)
Specifies whether ornot the factored form of A has been supplied on entry. = F': LOW F, D IA GF, $\mathrm{UPF} 1, \mathrm{UPF} 2$, and $\mathbb{P}$ IV OT contain the factored form of $A$; LOW, DIAG, UP, LOW F,D $\mathbb{A} G F, U P F 1, U P F 2$ and $\mathbb{P} \mathbb{V} O T \mathrm{w}$ illnotbe m odified. $=\mathrm{N}$ ': The m atrix w ill be copied to LOW F, D IA GF, and UPF1 and factored.

TRANSA (input)
Specifies the form of the system of equations:
$=N: A * X=B \quad$ N $\circ$ transpose)
$=T$ ': A **T * X = B (Transpose)
= C': A ** H * X = B (C onjugate transpose)

TRANSA is defaulted to $N$ 'forF95 $\mathbb{I N}$ TERFACE.

N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of them atrix B. NRHS $>=0$.

LOW (input)
The $(n-1)$ subdiagonalelem ents of $A$.

D IA G (input)
The n diagonalelem ents of A .

UP (input)
The ( $n-1$ ) superdiagonalelem ents ofA.

LOW F (input/output)
IfFACT $=\mathrm{F}^{\prime}$, then LOW F is an inputargum ent and on entry contains the ( $n-1$ ) multipliers that define the $m$ atrix $L$ from the $L U$ factorization of $A$ as com puted by C G TTRF.

IfFACT $=\mathrm{N}$ ', then LOW F is an outputargum ent and on exit contains the ( $n-1$ ) m ultipliers that define them atrix $L$ from the LU factorization of A.

D IA GF (input/output)
If FACT $=\mathrm{F}^{\prime}$, then $\mathrm{D} \mathbb{I} G \mathrm{~F}$ is an input argum ent and on entry contains the n diagonalelem ents of the uppertriangularm atrix $U$ from the LU factorization ofA.

IfFACT = $N$ ', then DIAGF is an output argum ent and on exit contains the $n$ diagonalelem ents of the uppertriangularm atrix $U$ from the $L U$ factorization of $A$.

UPF1 (input/output)
IfFACT = $\mathrm{F}^{\prime}$, then UPF 1 is an inputargum ent and on entry contains the ( $n-1$ ) elem ents of the first superdiagonalofU .

IfFACT = N ', then UPF1 is an outputargum ent and on exit contains the ( $n-1$ ) elem ents of the first superdiagonalofU.

UPF2 (input/output)
If FACT = F ', then UPF 2 is an input argum ent and on entry contains the ( $n-2$ ) elem ents of the second superdiagonalofU .

IfFACT $=\mathrm{N}$ ', then UPF2 is an output argum entand on exitcontains the ( $n-2$ ) elem ents of the second superdiagonalofU .

IPIVOT (input/output)
If $F A C T=F '$, then $\mathbb{P I V O T}$ is an input argum ent and on entry contains the pivotindioes from the LU factorization of A as com puted by CG TTRF.

IfFACT = $N$ ', then $\mathbb{P} \mathbb{I V O T}$ is an output argum ent and on exit contains the pivot indices from the LU factorization of ; row iof the $m$ atrix $w$ as interchanged w ith row $\mathbb{P} \mathbb{I V O T}$ (i). $\mathbb{P} \mathbb{I V O T}$ (i) w illalw ays be ether ior i+ 1 ; $\mathbb{P} \mathbb{I V O T}(i)=$ indicates a row interchange $w$ as not required.
$B$ (input) The $N$-by-N RH S righthand side $m$ atrix $B$.

LD B (input)
The leading din ension of the array $B$. LD B >= $\max (1, N)$.

X (output)
If $\mathbb{N} F O=0$ or $\mathbb{N} F O=\mathrm{N}+1$, the $\mathrm{N}-$ by -NRH S solution
$m$ atrix $X$.

LD $X$ (input)
The leading dim ension of the array X . LD X >= $\max (1, N)$.

RCOND (output)
The estim ate of the reciprocal condition num ber of the $m$ atrix $A$. IfRCOND is less than them achine precision (in particular, if RCOND $=0$ ), the $m$ atrix is singular to working precision. This condition is indicated by a retum code of $\mathbb{N}$ FO > 0.

FERR (output)
The estim ated forw ard enorbound for each solution vector $X(\mathcal{)}$ ) the $j$ th colum $n$ of the solution $m$ atrix X). If XTRUE is the true solution comesponding to $X(\mathcal{O}), \operatorname{FERR}(\mathcal{7})$ is an estim ated upperbound for the $m$ agnitude of the largest ele-
$m$ ent in $(X(\mathcal{O})$ X TRUE $)$ divided by the $m$ agnitude of the largestelem ent in $X(\mathcal{})$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

## BERR (output)

The com ponentw ise relative backw ard error of each
solution vector $X(\mathcal{)}$ ) (i.e., the sm allest relative change in any elem entof $A$ orB thatm akes $X(j)$ an exactsolution).

W ORK (w orkspace)
dim ension ( $2 * \mathrm{~N}$ )

W ORK 2 (w orkspace)
dim ension (N)
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum enthad an ille-
galvalue
>0: if $\mathbb{N} F O=i$, and $i$ is
$<=\mathrm{N}: \mathrm{U}(i, i)$ is exactly zero. The factorization has notbeen com pleted unless $i=N$, but the factor $U$ is exactly singular, so the solution and error bounds could notbe com puted. RCOND $=0$ is retumed. $=\mathrm{N}+1$ : U is nonsingular, but RCOND is less than $m$ achine precision, m eaning that the $m$ atrix is singular to working precision. $N$ evertheless, the solution and errorbounds are com puted because there are a num ber of situations where the com puted solution can bem ore accurate than the value of RCOND w ould suggest.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgturf-com pute an LU factorization of a com plex tridiagonalm atrix A using elim ination $w$ ith partialpivoting and row interchanges

## SYNOPSIS

```
SUBROUT\mathbb{NE CGTTRF N,LOW ,D IA G,UP1,UP2, PIVOT,INFO)}
COM PLEX LOW (*),DIAG (*),UP1 (*),UP2 (*)
\mathbb{NTEGER N,}\mathbb{N}FO
INTEGER \mathbb{PIVOT (*)}
```



```
COM PLEX LOW (*),DIAG (*),UP1 (*),UP2 (*)
\mathbb{NTEGER*8 N,NNFO}
\mathbb{NTEGER *8 \mathbb{PIVOT (*)}}\mathbf{(})
```


## F95 INTERFACE

```
SU BROUTINE GTTRF ( \(\mathbb{N}\) ],LOW ,D \(\mathbb{I A}, \mathrm{UP} 1, \mathrm{UP} 2, \mathbb{P} \mathbb{I V O T},[\mathbb{N} F O]\) )
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::LOW ,D \(\mathbb{A} G, U P 1, U P 2\)
\(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V} O T\)
SU BROUTINE GTTRF_64 (N ],LOW ,D \(\mathbb{I A G}, \mathrm{UP} 1, \mathrm{UP} 2, \mathbb{P} \mathbb{I} \operatorname{OT},[\mathbb{N} F O\) ])
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::LOW ,D \(\mathbb{A} G, U P 1, U P 2\)
\(\mathbb{N}\) TEGER ( 8 ): : \(\mathrm{N}, \mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T\)
```

void cgttrf(intn, com plex *low, complex *diag, com plex *up1, com plex *up2, int *ipivot, int *info);
void ogttrf_64 (long n, com plex *low , com plex *diag, com plex
*up1, com plex *up2, long *ípívot, long *info);

## PURPOSE

cgttrf com putes an LU factorization of a com plex tridiagonal $m$ atrix A using elim ination w ith partialpivoting and row interchanges.

The factorization has the form

$$
A=L * U
$$

where $L$ is a productofperm utation and unit low er bidiagonalm atrioes and $U$ is upper triangularw ith nonzeros in only the $m$ ain diagonal and firsttw o superdiagonals.

## ARGUMENTS

$N$ (input) The order of the $m$ atrix $A$.

LOW (input/output)
On entry, LOW m ustcontain the ( $\mathrm{n}-1$ ) sub-diagonal elem ents ofA.

On exit, LOW is overw rilten by the ( $n-1$ ) multipliers that define the $m$ atrix $L$ from the $L U$ factorization of A.

D IA G (input/output)
O n entry, D IA G m ust contain the diagonal elem ents of A.

On exit, D IA G is overw rilten by the $n$ diagonal elem ents of the upper triangularm atrix $U$ from the LU factorization of A.

UP1 (input/output)
On entry, UP1 must contain the ( $n-1$ ) superdiagonalelem ents ofA.

O n exit, UP1 is overw rilten by the ( $\mathrm{n}-1$ ) elem ents of the first super-diagonal of .

UP2 (output)
On exit, UP2 is overw rilten by the ( $n-2$ ) elem ents of the second super-diagonalofU.

IPIVOT (output)
The pivotindioes; for $1<=i<=n$, row $i$ of the matrix was interchanged w th row $\mathbb{P I V O T}$ (i). IPIVOT (i) will always be either $i$ or i+1; PIVOT (i) = iindicates a row interchange was not required.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0$ : if $\mathbb{N} F O=-k$, the $k$-th argum enthad an illegalvalue
> 0 : if $\mathbb{N} F O=k, U(k, k)$ is exactly zero. The factorization has been com pleted, but the factor $U$ is exactly singular, and division by zero will occur if it is used to solve a system ofequations.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cgttrs - solve one of the system sof equations $A * X=B$, $\mathrm{A} * * \mathrm{~T} * \mathrm{X}=\mathrm{B}$,orA ${ }^{*}{ }^{*} \mathrm{H} * \mathrm{X}=\mathrm{B}$,

## SYNOPSIS

```
SU BROUT\mathbb{NE CGTTRS (TRANSA,N,NRHS,LOW,DIAG,UP1,UP2,\mathbb{PIVOT,B,}}\mathbf{N},\textrm{N},\textrm{N}
    LDB,\mathbb{NFO)}
CHARACTER * 1 TRANSA
COM PLEX LOW (*),D IAG (*),UP1 (*),UP2 (*),B (LDB,*)
\mathbb{NTEGERN,NRHS,LDB,INFO}
\mathbb{NTEGER \mathbb{PIVOT(*)}}\mathbf{(})
```



```
    LDB,\mathbb{NFO)}
CHARACTER * 1 TRANSA
COM PLEX LOW (*),D IAG (*),UP1 (*),UP2 (*),B (LDB ,*)
\mathbb{NTEGER*8N,NRHS,LDB,INFO}
INTEGER*8 \mathbb{PIVOT (*)}
```


## F95 INTERFACE

SU BROUTINE GTTRS ([TRANSA ], $\mathbb{N}], \mathbb{N} R H S], L O W, D I A G, U P 1, U P 2, \mathbb{P} \mathbb{I V O T}$, B, [LDB], [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1) ::TRANSA
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::LOW ,D $\mathbb{A} G, U P 1, U P 2$
COM PLEX,D $\mathbb{I M}$ ENSION (: : : : : B
$\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V} O T$
SU BROUTINE GTTRS_64 ([TRANSA], $\mathbb{N}], \mathbb{N} R H S], L O W, D \mathbb{A} G, U P 1, U P 2$,
$\mathbb{P} \mathbb{V} O T, B,[\operatorname{LDB}],[\mathbb{N F O}])$

CHARACTER (LEN=1) ::TRANSA
COM PLEX , D $\mathbb{M}$ ENSION (:) ::LOW ,D $\mathbb{I} G, U P 1, U P 2$
COM PLEX, D IM ENSION (:,:) ::B
$\mathbb{N}$ TEGER (8) :: N , NRHS,LD B, $\mathbb{N}$ FO
$\mathbb{N} \operatorname{TEGER}$ (8), D $\mathbb{I M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT}$

## C INTERFACE

\#include <sunperfh>
void ogttrs (chartransa, intn, intnrhs, com plex *low , com plex *diag, complex *up1, complex *up2, int *ípivot, com plex *b, int ldb, int *info);
void agttrs_64 (chartransa, long $n$, long nihs, com plex *low , com plex *diag, com plex *up1, com plex *up2, long *ípívot, com plex *b, long ldb, long *info);

## PURPOSE

cgttrs solves one of the system s of equations $A * X=B, A * * T * X=B$, or $A * * H * X=B$, with a tridiagonal m atrix A using the LU factorization com puted by CGTTRF.

## ARGUMENTS

TRANSA (input)
Specifies the form of the system of equations. $=$
$N^{\prime}: A * X=B \quad$ N o transpose)
$=T ': A * * T * X=B \quad$ ( ranspose)
$=C$ ': $A * * H * X=B \quad$ (C onjugate transpose)

TRANSA is defaulted to $N$ 'forF95 $\mathbb{I N}$ TERFACE.
$N$ (input) The order of the m atrix A.

NRHS (input)
The num ber of righthand sides, i.e., the num ber
of colum ns of the m atrix B. NRHS $>=0$.

LOW (input)
The $(n-1) \mathrm{m}$ ultipliers that define the $m$ atrix $L$ from the LU factorization ofA.

D IA G (input)
The $n$ diagonalelem ents of the upper triangular
$m$ atrix $U$ from the $L U$ factorization ofA .

## UP1 (input)

The $(n-1)$ elem ents of the first super-diagonal of U .

UP2 (input)
The ( $n-2$ ) elem ents of the second super-diagonal of U.

IPIVOT (input)
The pivotindioes; for $1<=i<=n$, row $i$ of the $m$ atrix $w a s$ interchanged $w$ th row $\mathbb{P} \mathbb{I V O T}(i)$. IPIVOT (i) w ill always be either $i$ or i+1; IPIVOT (i) = iindicates a row interchange was not required.

B (input/output)
O n entry, the m atrix of righthand side vectors B . On exit, B is overw rilten by the solution vectors X .

LD B (input)
The leading dim ension of the aray $\mathrm{B} . \operatorname{LDB}>=$ $\max (1, N)$.

INFO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N ~ F O ~}=-\mathrm{k}$, the k -th argum enthad an illegalvałue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chbev - com pute all the eigenvalues and, optionally, eigenvectors of a com plex Herm itian band m atrix A

## SYNOPSIS

```
SUBROUT\mathbb{NE CHBEV (JOBZ,UPLO,N,KD,A,LDA,W ,Z,LD Z,W ORK,}
        W ORK2,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
COM PLEX A (LDA,*),Z (LD Z,*),W ORK (*)
\mathbb{NTEGERN,KD,LDA,LD Z,INFO}
REALW (*),W ORK2 (*)
SUBROUT\mathbb{NECHBEV_64(JOBZ,UPLO,N,KD,A,LDA,W ,Z,LD Z,W ORK,}
    WORK2, \mathbb{NFO)}
```

CHARACTER * 1 JOBZ, UPLO
COM PLEX A (LDA, ${ }^{*}$ ), Z (LD Z,*), W ORK (*)
$\mathbb{N} T E G E R * 8 N, K D, L D A, L D Z, \mathbb{N} F O$
REALW (*),WORK2 (*)

## F95 INTERFACE

SU BROUTINE HBEV (OBB , UPLO, $\mathbb{N}], K D, A,[L D A], W, Z,[L D Z],[W O R K]$, [ W ORK 2], [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1)::JOBZ, UPLO
COMPLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::A, Z
$\mathbb{N} T E G E R:: N, K D, L D A, L D Z, \mathbb{N F O}$
REAL,D $\mathbb{M}$ ENSION (:) ::W ,W ORK2
SUBROUTINE HBEV_64 (JOBZ, UPLO, $\mathbb{N}], K D, A,[L D A], W, Z,[L D Z]$,
[W ORK], [W ORK2], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1)::JOBZ,UPLO
COMPLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:): : A, Z
$\mathbb{N} T E G E R(8):: N, K D, L D A, L D Z, \mathbb{N} F O$
REAL,D $\mathbb{I}$ ENSION (:) ::W ,W ORK2

## C INTERFACE

\#include <sunperfh>
void chbev (char jobz, charuple, intn, int kd, com plex *a, int lda, float *w, com plex *z, int ldz, int *info);
void chbev_64 (char jojbz, char uplo, long n, long kd, com plex *a, long lda, float *w , com plex *z, long ldz, long *info);

## PURPOSE

chbev com putes all the eigenvalues and, optionally, eigenvectors of a com plex H erm itian band m atrix A .

## ARGUMENTS

JOBZ (input)
$=N^{\prime}:$ C om pute eigenvalues only;
= V ': C om pute eigenvalues and eigenvectors.
UPLO (input)
= U : U Upertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrix A. $\mathrm{N}>=0$.
KD (input)
The num ber of superdiagonals of the $m$ atrix A if UPLO = U',orthe num berof subdiagonals ifU PLO $=L^{\prime} . \mathrm{KD}>=0$.

A (input/output)
On entry, the upper or low ertriangle of the Her $m$ tian band $m$ atrix A, stored in the firstKD +1 row sof the array. The $j$ th colum $n$ of $A$ is stored in the $j$ th colum $n$ of the amay $A$ as follow $s$ : if $\mathrm{UPLO}=\mathrm{U}, \mathrm{A}(\mathrm{kd}+1+i-j, j)=A(i, j)$ for $\max (1, j$ $\mathrm{kd})<=i<=j$ if UPLO $=\mathrm{L}, \mathrm{A}(1+i-j, j)=A(i, j)$
for $\dot{j}=i<=m$ in $(n, j+k d)$.
On exit, A is overw rilten by values generated during the reduction to tridiagonal form. If P PLO = U ', the first superdiagonal and the diagonal of the tridiagonal $m$ atrix $T$ are retumed in row sK D and $K D+1$ ofA, and if $U P L O=L$ ', the diagonaland first subdiagonal of T are retumed in the first tw o row sofA.

LD A (input)
The leading dim ension of the array A. LD A >= KD + 1.

W (output) If $\mathbb{N} F O=0$, the eigenvalues in ascending order.

Z (input) If $\mathrm{JOBZ}=\mathrm{V}^{\prime}$, then if $\mathbb{N F O}=0, \mathrm{Z}$ contains the orthonom aleigenvectors of the $m$ atrix $A, w$ ith the $i$-th colum $n$ of $Z$ holding the eigenvector associated w th $W$ (i). If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced.

LD Z (input)
The leading dim ension of the amray Z . LD $\mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD Z $>=\mathrm{max}(1, N)$.

W ORK (w orkspace)
dim ension (N)
W ORK 2 (w orkspace)
dim ension (max (1,3*N-2))
$\mathbb{N} F O$ (output)
= 0 : successfulexit.
<0: if $\mathbb{N}$ FO = -i, the i-th argum ent had an illegalvahue.
> 0 : if $\mathbb{N} F O=$ i, the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chbevd - com pute all the eigenvalues and, optionally, eigenvectors of a com plex $H$ erm bitian band $m$ atrix $A$

## SYNOPSIS

```
SUBROUT\mathbb{NE CHBEVD (JOBZ,UPLO,N,KD,AB,LDAB,W ,Z,LDZ,W ORK,}
    LW ORK,RW ORK,LRW ORK,IN ORK,LIN ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
COM PLEX AB (LDAB,*),Z (LD Z,*),W ORK (*)
INTEGERN,KD,LDAB,LDZ,LW ORK,LRW ORK,LIN ORK,INFO
INTEGER IN ORK (*)
REALW (*),RW ORK (*)
SU BROUTINE CHBEVD_64(JOBZ,UPLO ,N,KD,AB,LDAB,W,Z,LDZ,W ORK,
    LW ORK,RW ORK,LRW ORK,\mathbb{IW ORK,L\mathbb{N ORK,NNFO)}}\mathbf{N},\mp@code{IN}
```

CHARACTER * 1 JOBZ, UPLO
COM PLEX AB (LDAB, $)$, Z (LD Z , *), W ORK (*)
$\mathbb{N}$ TEGER*8N,KD,LDAB,LDZ,LW ORK,LRW ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I N}$ ORK (*)
REALW (*),RWORK (*)

## F95 INTERFACE

SU BROUTINE HBEVD (OBZ,UPLO, $\mathbb{N}], K D, A B,[L D A B], W, Z,[L D Z],[W O R K]$, $[$ [LW ORK ], $\mathbb{R W}$ ORK ], [LRW ORK ], [ $\mathbb{W}$ ORK ], [LIN ORK ], [ $\mathbb{N} F \mathrm{FO}])$

CHARACTER (LEN=1): : JOBZ, UPLO
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::AB,Z
$\mathbb{N} T E G E R:: N, K D, L D A B, L D Z, L W O R K, L R W$ ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N}$ ORK

REAL,D IM ENSION (:) ::W ,RW ORK

SU BROUTINE HBEVD_64 (JOBZ,UPLO, $\mathbb{N}], K D, A B,[L D A B], W, Z,[L D Z]$,


CHARACTER (LEN=1)::JOBZ,UPLO
COMPLEX,DIM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::AB,Z
$\mathbb{N} T E G E R(8):: N, K D, L D A B, L D Z, L W$ ORK,LRW ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{I W}$ ORK
REAL,D IM ENSION (:) ::W ,RW ORK

## C INTERFACE

\#include < sunperfh>
void chbevd (char jobz, charuplo, int n, int kd, com plex
*ab, int ldab, float *w, com plex *z, int ldz, int
*info);
void chbevd_64 (char jobz, charuplo, long n, long kd, com plex *ab, long ldab, float *w, com plex *z, long ldz, long *info);

## PURPOSE

chbevd com putes all the eigenvalues and, optionally, eigenvectors of a com plex $H$ erm titian band $m$ atrix $A$. If eigenvectors are desired, it uses a divide and conquer algorithm .

The divide and conquer algorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on $m$ achines w th a guard digit in add/subtract, or on those binary $m$ achines $w$ thout guard digits w hich subtract like the C ray X - M P , C ray Y M P , C ray C-90, orC ray-2. It could conceivably fail on hexadecim al or decim al machines w thout guard digits, butw e know of none.

## ARGUMENTS

JOBZ (input)
$=N^{\prime}:$ C om pute eigenvalues only;
$=\mathrm{V}$ : C om pute eigenvalues and eigenvectors.
UPLO (input)
= U ': U ppertriangle ofA is stored;
= L' : Low ertriangle ofA is stored.

N (input) The order of the matrix A. $\mathrm{N}>=0$.
$K D$ (input)
The num ber of superdiagonals of the $m$ atrix A if UPLO $=\mathrm{U}$ ', orthe num berof subdiagonals if UPLO $=L^{\prime} . K D>=0$.

AB (input/output)
O $n$ entry, the upper or low er triangle of the Her$m$ itian band $m$ atrix $A$, stored in the firstKD +1 row sof the array. The $j$ th colum n of $A$ is stored in the $j$ th colum $n$ of the array $A B$ as follow $s$ : if $\mathrm{UPLO}=\mathrm{U}, \mathrm{AB}(\mathrm{kd}+1+\mathrm{i}-j)=\mathrm{j}(i, 1)$ for $\mathrm{max}(1, j$ $\mathrm{kd})<=i<=\dot{j}$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{AB}(1+i-j, 7)=A(i, j)$ for $j=i<=m$ in $(n, j+k d)$.
On exit, AB is overw rilten by values generated during the reduction to tridiagonal form. IfU PLO $=\mathrm{U}$ ', the first superdiagonal and the diagonal of the tridiagonal m atrix T are retumed in row sKD and $K D+1$ of $A B$, and if $U P L O=L$ ', the diagonal and first subdiagonal of $T$ are retumed in the first tw o row sofA B .

LDAB (input)
The leading dim ension of the array AB. LD A B > $>=K D$ +1 .

W (output) If $\mathbb{N} F O=0$, the eigenvalues in ascending order.
$Z$ (input) If $\mathcal{O B Z}=\mathrm{V}$ ', then if $\mathbb{N} F O=0, \mathrm{Z}$ contains the orthonorm aleigenvectors of the $m$ atrix $A, w$ ith the i-th colum $n$ of $Z$ holding the eigenvector associated $w$ th $W$ (i). If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced.

LD $Z$ (input)
The leading din ension of the array $\mathrm{Z} . \operatorname{LD} \mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD Z $>=\mathrm{max}(1, N)$.

W ORK (w orkspace)
On exit, if $\mathbb{N F} F=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array W ORK. If $\mathrm{N}<=1$, LW ORK must be at least1. If $J 0 B Z=N$ 'and $N>$ 1, LW ORK m ust.be at leastN. If JO BZ $=\mathrm{V}$ 'and N > 1, LW ORK m ustbe at least $2 * N$ **2.

IfLW ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

RW ORK (w orkspace)
dim ension (LRW ORK) On exit, if $\mathbb{I N} F O=0, R W O R K(1)$
retums the optim allRW ORK.
LRW ORK (input)
The dimension of array RW ORK. If $\mathrm{N}<=1$, LRW ORK mustibe at least 1. If $\mathrm{JOBZ}=\mathrm{N}$ 'and $\mathrm{N}>$ 1, LRW ORK m ustbe at leastN. If JO B Z $=\mathrm{V}$ 'and N > 1,LRW ORK m ust.be at least1 + 5*N + 2*N **2.

If LRW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the RW ORK amray, retums this value as the first entry of the RW ORK amay, and no enorm essage related to LRW ORK is issued by X ERBLA.

IV ORK (w orkspace/output)
On exit, if $\mathbb{N}$ FO $=0, \mathbb{I N}$ ORK (1) retums the optim al LIN ORK.

LIN ORK (input)
The dim ension of array $\mathbb{I N}$ ORK. If JOBZ = N 'or N $<=1, \mathrm{LIV} O R K$ mustbe at least1. If $\mathrm{JOBZ}=\mathrm{V}^{\prime}$ and $N>1, L \mathbb{I}$ ORK m ustbe at least $3+5{ }^{*} N$.

If $L \mathbb{I V} O R K=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the $\mathbb{I W}$ ORK array, retums this value as the first entry of the $\mathbb{I W} O R K$ array, and no errorm essage related to LIIN ORK is issued by X ERBLA.
$\mathbb{I N} F O$ (output)
= 0 : successfulexit.
< 0: if $\mathbb{N}$ FO = -i, the i-th argum ent had an illegalvahue. > 0 : if $\mathbb{N} F O=$ i, the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chbevx - com pute selected eigenvalues and, optionally, eigenvectors of a com plex H erm itian band m atrix A

## SYNOPSIS

```
SUBROUT\mathbb{NE CHBEVX (OOBZ,RANGE,UPLO,N,KD,A,LDA,Q,LDQ,VL,}
    VU,\mathbb{L},\mathbb{U},ABTOL,NFOUND,W,Z,LD Z,W ORK,W ORK 2, IN ORK 3, \mathbb{FA}\mathbb{I},
    \mathbb{NFO)}
CHARACTER * 1 JOBZ,RANGE,UPLO
COM PLEX A (LDA,*),Q (LDQ,*),Z (LD Z,*),W ORK (*)
\mathbb{NTEGERN,KD,LDA,LDQ,\mathbb{L},\mathbb{U},NFOUND,LDZ, NNFO}
\mathbb{NTEGER IN ORK3(*),\mathbb{FA [H (*)}}\mathbf{(})
REALVL,VU,ABTOL
REALW (*),W ORK2 (*)
SU BROUTINE CHBEVX_64(JOBZ,RANGE,UPLO,N,KD,A,LDA,Q,LDQ,VL,
```



```
    \mathbb{NFO)}
```

CHARACTER * 1 JOBZ,RANGE,UPLO
COM PLEX A (LDA, *), Q (LDQ , *), Z (LD Z, *), W ORK (*)
$\mathbb{N} T E G E R * 8 N, K D, L D A, L D Q, \mathbb{L}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I N}$ ORK 3 (*), $\mathbb{F A} \mathbb{L}\left({ }^{*}\right)$
REALVL,VU,ABTOL
REALW (*),WORK2 (*)

## F95 INTERFACE

SU BROUTINE HBEVX (JOBZ,RANGE,UPLO, $\mathbb{N}], K D, A,[L D A], Q,[L D Q]$, VL, VU, $\mathbb{I}, \mathbb{I}, A B T O L, ~ \mathbb{N F O U N D}], W, Z,[L D Z],[W O R K],[W O R K 2]$, [ $\mathbb{I W}$ ORK3], $\mathbb{F A} \mathbb{I},[\mathbb{N} F O]$ )

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COMPLEX,D $\mathbb{I M} E N S I O N(:):: W$ ORK
COM PLEX,D $\operatorname{IM}$ ENSION (: : : : : A , Q , Z
$\mathbb{N} T E G E R:: N, K D, L D A, L D Q, \mathbb{L}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{I N} O R K 3, \mathbb{F} A \mathbb{L}$
REAL ::VL,VU,ABTOL
REAL,D $\mathbb{M}$ ENSION (:) ::W ,W ORK2

SU BROUTINE HBEVX_64 (JOBZ,RANGE, UPLO, $\mathbb{N}], K D, A,[L D A], Q,[L D Q]$, $\mathrm{VL}, \mathrm{VU}, \mathbb{I}, \mathbb{Z}, \mathrm{ABTOL}, \mathbb{N} F O U N D], W, Z,[\mathrm{ZD} \mathrm{Z}],[\mathrm{W} O R K],[W O R K 2]$, [ $\mathbb{I N} O R K 3], \mathbb{F A} \mathbb{I},[\mathbb{N F O}])$

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COMPLEX,D $\mathbb{M} E N S I O N(:):$ ORK
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : A , Q , Z
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{KD}, \mathrm{LDA}, \mathrm{LD} Q, \mathbb{I}, \mathbb{U}, \mathrm{NFOUND}, \mathrm{LD} Z, \mathbb{N} F O$
$\mathbb{N}$ TEGER (8), D $\mathbb{I M}$ ENSION (:) :: IN ORK 3, $\mathbb{F} A \mathbb{I}$
REAL ::VL,VU,ABTOL
REAL,D $\mathbb{I M} E N S I O N(:):: W$,W ORK 2

## C INTERFACE

\#include <sunperfh>
void chbevx (char jobz, char range, charuplo, intn, int kd, com plex *a, int lda, com plex *q, int ldq, float vl, floatvu, int il, int in, float abtol, int ${ }^{*}$ nfound, float ${ }^{*} \mathrm{w}$, com plex ${ }^{*} \mathrm{z}$, int ldz , int *ifail, int*info);
void chbevx_64 (char j.bzz, char range, char uplo, long n, long kd, complex *a, long lda, com plex *q, long ldq, float vl, floatvu, long il, long iu, float abtol, long *nfound, float*w, com plex *z, long ldz, long *ifail, long *info);

## PURPOSE

chbevx com putes selected eigenvahues and, optionally, eigenvectors of a com plex H erm itian band m atrix A. E igenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}$ : C om pute eigenvahues only;
$=\mathrm{V}$ : C om pute eigenvalues and eigenvectors.

RANGE (input)
= A : alleigenvalues w illibe found;
$=\mathrm{V}$ ::alleigenvalues in the half-open interval
(NL,VU] will be found; = 'I': the $\mathbb{L}$-th through
$\mathbb{I J}$-th eigenvaluesw illlbe found.

UPLO (input)
= U ': U ppertriangle ofA is stored;
$=\mathrm{L}$ ': Low er triangle of A is stored.
N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.
KD (input)
The num ber of superdiagonals of the $m$ atrix A if UPLO = U', orthe num berof subdiagonals if UPLO $=\mathbb{L}^{\prime} . \mathrm{KD}>=0$.

A (input/output)
On entry, the upper or low er triangle of the Her$m$ itian band $m$ atrix $A$, stored in the firstKD +1 row sof the array. The $j$ th colum n of A is stored in the $j$ th colum $n$ of the amay A as follows: if $\mathrm{UPLO}=\mathrm{U}, \mathrm{A}(\mathrm{kd}+1+\mathrm{i}-j)=\mathrm{A}(i, j)$ for max $(1, j$ $\mathrm{kd})<=i<=j$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{A}(1+i-j, j)=A(i, j)$ for $j=i<=m$ in $(n, j+k d)$.

On exit, A is overw rilten by values generated during the reduction to tridiagonal form .

LD A (input)
The leading dim ension of the amray A. LD A >=KD + 1.

Q (output)
If $\mathrm{JOBZ}=\mathrm{V}$ ', the N -by-N unitary m atrix used in the reduction to tridiagonal form. If $J 0 B Z=N$ ', the array Q is not referenced.

LD Q (input)
The leading dim ension of the array $Q$. If $\operatorname{JOBZ}=$ V ', then LD $Q>=\max (1, N)$.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU.
N ot referenced ifRANGE = A 'or I'.
VU (input)

IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRA N G E = A' 'or I'.

II (input)
IfRA N G E= I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{U}<=\mathbb{U}<=N$, if $N>0$; $\mathbb{L}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N otreferenced ifRANGE= A 'or V'.
UU (input)
IfRA N G E= I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}$, if $\mathrm{N}>0 ; \mathbb{L}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N otreferenced ifRANGE= A 'or V'.

ABTOL (input)
The absolute error tolerance for the eigenvalues. A $n$ approxim ate eigenvalue is accepted as converged when it is determ ined to $l i e$ in an interval [a,b]
of w idth less than orequal to
ABTOL + EPS * $\max (k|,||$,$) ,$
where EPS is them achine precision. If ABTOL is less than or equal to zero, then EPS* $\mid$ |w illbe used in its place, where $T$ | is the 1 -norm of the tridiagonal m atrix obtained by reducing $A$ to tridiagonalform .

E igenvalues w illbe com puted m ost accurately when
ABTOL is set to tw ice the underflow threshold $2 * S L A M C H$ ( ${ }^{\eta}$ ), not zero. If this routine retums w ith $\mathbb{N}$ FO $>0$, indicating that som e eigenvectors did not converge, try setting ABTOL to 2 *SLAM CH (S ).

See "C om puting Sm allSingularV alues ofB idiagonal M atrices w ith G uaranteed H igh Relative A ccuracy," by D em m eland K ahan, LA PA CK W orking N ote \#3.

NFOUND (output)
The total num ber of eigenvalues found. $0<=$
NFOUND <= N. IfRANGE = A',NFOUND = N, and if RANGE = 'I',NFOUND = $\mathbb{U}-\mathbb{L}+1$.

W (output)
The firstNFOUND elem ents contain the selected eigenvalues in ascending order.
$Z$ (input) If $J O B Z=V '$, then if $\mathbb{N} F O=0$, the first $N F O U N D$ colum ns of $Z$ contain the orthonorm aleigenvectors of them atrix A corresponding to the selected eigenvalues, $w$ ith the $i$-th colum $n$ of $Z$ holding the eigenvector associated w ith $W$ (i). If an eigenvector fails to converge, then that colum n of $Z$ contains the latest approxim ation to the eigenvector, and the index of the eigenvector is retumed in FAII. If $J O B Z=N$ ', then $Z$ is not referenced.
$N$ ote: the user must ensure that at least
$\max (1, N F O U N D)$ colum ns are supplied in the anray $Z$; if RANGE = V', the exact value ofNFOUND is not know $n$ in advance and an upperbound $m$ ustbe used.

LD Z (input)
The leading $d i m$ ension of the array $Z$. LD $Z>=1$, and if $\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{max}(1, \mathrm{~N})$.

W ORK (w orkspace)
dim ension (N)

W ORK 2 (w orkspace)
dim ension ( $7 * \mathrm{~N}$ )

IV ORK 3 (w orkspace)
dim ension ( $5 * \mathrm{~N}$ )
FFA II (output)
If $\mathrm{JOBZ}=\mathrm{V}$ ', then if $\mathbb{N F O}=0$, the first NFOUND
elem ents of $\mathbb{F A} I I$ are zero. If $\mathbb{N} F O>0$, then
FAII contains the indices of the eigenvectors
that failed to converge. If $\mathrm{JOBZ}=\mathrm{N}$ ', then
$\mathbb{F A} \mathbb{I L}$ is not referenced.
$\mathbb{N} F O$ (output)
= 0: successfulexit
<0: if $\mathbb{I N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue
> 0 : if $\mathbb{N F O}=$ i, then ieigenvectors failed to converge. Their indioes are stored in array ㅍFAI.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chbgst-reduce a com plex H erm itian-definite banded generalized eigenproblem $\mathrm{A} * \mathrm{x}=\operatorname{lam}$ bda* $\mathrm{B} * \mathrm{x}$ to standard form C * $\mathrm{Y}=$ lam bda*y,

## SYNOPSIS

```
SUBROUT\mathbb{NECHBGST NECT,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,X,LDX,}
    W ORK,RW ORK,INFO)
CHARACTER * 1 VECT,UPLO
COM PLEX AB (LDAB,*),BB (LDBB,*),X (LDX,*),W ORK (*)
INTEGERN,KA,KB,LDAB,LDBB,LDX,INFO
REAL RW ORK (*)
SUBROUTINE CHBGST_64NECT,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,X,
    LDX,W ORK,RW ORK,\mathbb{NFO)}
CHARACTER * 1 VECT,UPLO
COM PLEX AB (LDAB,*),BB (LDBB,*),X (LDX,*),W ORK (*)
\mathbb{NTEGER*8N,KA,KB,LDAB,LDBB,LDX,}\mathbb{N}FO
REAL RW ORK (*)
```


## F95 INTERFACE

SU BROUTINE HBGST $N E C T, U P L O, \mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B], X$, [LDX], [W ORK], $\mathbb{R W}$ ORK], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1)::VECT,UPLO
COMPLEX,D IM ENSION (:) ::W ORK
COMPLEX,D $\mathbb{M}$ ENSION (:,:) ::AB, BB,X
$\mathbb{N} T E G E R:: N, K A, K B, L D A B, L D B B, L D X, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::RW ORK

SU BROUTINE HBGST_64 NECT,UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B]$, $\mathrm{X},[\mathrm{LD} \mathrm{X}],[\mathrm{W}$ ORK ], $\mathbb{R W}$ ORK ], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::VECT,UPLO
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK COMPLEX,D IM ENSION (:,:) ::AB,BB,X
$\mathbb{N} T E G E R(8):: N, K A, K B, L D A B, L D B B, L D X, \mathbb{N} F O$
REAL,D $\mathbb{I M} E N S I O N(:):: R W$ ORK

C INTERFACE
\#include <sunperfh>
void chbgst(charvect, charuplo, intn, int ka, int kb, com plex *ab, int ldab, com plex *bb, int ldbb, com plex *x, int ldx, int*info);
void chbgst_64 (charvect, charuplo, long n, long ka, long kb, com plex *ab, long ldab, com plex *bb, long ldbb, com plex *x, long ldx, long *info);

## PURPOSE

chbgst reduces a com plex Herm tian-definite banded generalized eigenproblem $\mathrm{A} * \mathrm{X}=\operatorname{lam}$ bda* $\mathrm{B} * \mathrm{x}$ to standard form $\mathrm{C} * \mathrm{Y}=$ lam bda* $y$, such that $C$ has the sam e bandw idth as A.

B musthave been previously factorized as $S * * H * S$ by CPBSTF, using a split Cholesky factorization. A is overw rilten by C $=\mathrm{X} * * \mathrm{H} * \mathrm{~A} * \mathrm{X}$, where $\mathrm{X}=\mathrm{S} * *(-1) * \mathrm{Q}$ and Q is a unitary matrix chosen to preserve the bandw idth of A.

## ARGUMENTS

VECT (input)
$=N^{\prime}$ : do not form the transform ation $m$ atrix $X$;
= V ': form X .

UPLO (input)
= U ': U pper triangle ofA is stored;
= L': Low er triangle of A is stored.

N (input) The order of the m atrioes A and $\mathrm{B} . \mathrm{N}>=0$.

KA (input)
The num ber of superdiagonals of the $m$ atrix $A$ if UPLO = U',orthe num berof subdiagonals ifU PLO
$=\mathbb{L}^{\prime} . \mathrm{KA}>=0$.

KB (input)
The num ber of superdiagonals of the $m$ atrix $B$ if UPLO = U',or the num berof subdiagonals ifU PLO $=L^{\prime} . K A>=K B>=0$.

AB (input/output)
On entry, the upper or low ertriangle of the Her$m$ tian band $m$ atrix $A$, stored in the firstka+1 row s of the array. The $j$ th colum n of A is stored in the $j$ th colum $n$ of the array $A B$ as follow $s$ : if $\mathrm{UPLO}=\mathrm{U}, \mathrm{AB}(\mathrm{ka}+1+i-j)=\mathrm{A}(i, 7)$ for $\max (1, j$ $\mathrm{ka})<=i<=j ;$ ifUPLO $=\mathrm{L}, \mathrm{AB}(1+i-j, j)=A(i, j)$ for $j=i<=m$ in $(n, j+k a)$.
On exit, the transform ed $m$ atrix $X * * H * A * X$, stored in the sam e form atasA.

LD AB (input)
The leading dim ension of the array AB. LDAB >= KA+1.

BB (input)
The banded factorS from the split Cholesky factorization ofB, as retumed by CPBSTF, stored in the first kb+1 row sof the anay.

LD BB (input)
The leading dim ension of the array $\mathrm{BB} . \operatorname{LDBB}>=$ K B +1 .

X (output)
If VECT = V', the $n-b y-n m$ atrix $X$. If $V E C T=$ N ', the array X is not referenced.

LD X (input)
The leading dim ension of the array X . LD X >= $\max (1, N)$ ifVECT $=V$ '; LD $X>=1$ otherw ise.

W ORK (w orkspace)
dim ension (N)
RW ORK (w orkspace)
dim ension $(\mathbb{N})$
$\mathbb{N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{I N}$ FO $=-i$, the $i$-th argum ent had an illegalvalue.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chbgv -com pute allthe eigenvalues, and optionally, the eigenvectors of a com plex generalized $H$ erm itian-definite banded eigenproblem, of the form $A * x=(\operatorname{lam} . b d a) * B * x$

## SYNOPSIS

```
SUBROUTINE CHBGV (JOBZ,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,W,Z,
    LD Z,W ORK,RW ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
COM PLEX AB (LDAB,*),BB (LDBB,*),Z (LD Z ,*),W ORK (*)
INTEGERN,KA,KB,LDAB,LDBB,LD Z,INFO
REALW (*),RW ORK (*)
SUBROUTINE CHBGV_64(OBZ,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,W,Z,
    LD Z,W ORK,RW ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
COM PLEX AB (LDAB,*),BB (LDBB,*),Z (LD Z,*),W ORK (*)
NNTEGER*8N,KA,KB,LDAB,LDBB,LD Z, INFO
REALW (*),RW ORK (*)
```


## F95 INTERFACE

SU BROUTINE HBGV (JBZ, UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B], W$, Z, [LD Z], [W ORK], RW ORK], [NFO])

CHARACTER (LEN=1):: JOBZ,UPLO
COMPLEX,D IM ENSION (:) ::W ORK
COMPLEX,D IM ENSION (:,:) ::AB,BB,Z
$\mathbb{N} T E G E R:: N, K A, K B, L D A B, L D B B, L D Z, \mathbb{N F O}$
REAL,D $\mathbb{I M}$ ENSION (:) ::W ,RW ORK

SU BROUTINE HBGV_64 (JOBZ, UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B]$,
W, Z, [LD Z], [W ORK], RW ORK], [ $\mathbb{N} F O])$
CHARACTER (LEN=1):: JOBZ,UPLO
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::AB,BB,Z
$\mathbb{N} T E G E R(8):: N, K A, K B, L D A B, L D B B, L D Z, \mathbb{N} F O$
REAL,D IM ENSION (:) ::W ,RW ORK

## C INTERFACE

\#include < sunperfh>
void chbgv (char j̀jbz, charuplo, intn, int ka, int kb, com plex *ab, int ldab, com plex *bb, int ldbb, float ${ }^{*}$ w , com plex *z, int ldz, int *info);
void chbgv_64 (char jobz, charuplo, long n, long ka, long kb, com plex *ab, long ldab, com plex *bb, long ldbb, float *w , com plex *z, long ldz, long *info);

## PURPOSE

chbogv com putes all the eigenvalues, and optionally, the eigenvectors of a com plex generalized $H$ erm itian-definite banded eigenproblem, of the form $A * x=(l a m ~ b d a) * B * x$. Here A and $B$ are assum ed to be $H$ erm itian and banded, and $B$ is also positive definite.

## ARGUMENTS

$J 0 \mathrm{BZ}$ (input)
= N ': C om pute eigenvalues only;
$=\mathrm{V}$ : C om pute eigenvalues and eigenvectors.

UPLO (input)
= U ': U pper triangles ofA and B are stored;
$=\mathrm{L}$ ': Low ertriangles of $A$ and $B$ are stored.

N (input) The order of the m atrioes A and $\mathrm{B} . \mathrm{N}>=0$.
KA (input)
The num ber of superdiagonals of the $m$ atrix $A$ if UPLO $=U$ ', ort the num berof subdiagonals if UPLO $=\mathrm{L} \cdot \mathrm{KA}>=0$.

KB (input)
The num ber of superdiagonals of the $m$ atrix $B$ if
UPLO = U',orthe num berof subdiagonals ifU PLO
$=L^{\prime} \cdot \mathrm{KB}>=0$.

AB (input/output)
On entry, the upper or low er triangle of the Her$m$ itian band $m$ atrix $A$, stored in the firstka+1 row sof the anray. The $j$ th colum $n$ of $A$ is stored in the jth colum $n$ of the amay AB as follows: if $\mathrm{UPLO}=\mathrm{U}, \mathrm{AB}(\mathrm{ka}+1+\mathrm{i}-j)=\mathrm{A}(i, j)$ for $\mathrm{max}(1, j$ $\mathrm{ka})<=i<=j$ ifUPLO $=\mathbb{L}, \mathrm{AB}(1+i-j, j)=A(i, j)$ for $j<=i<=m$ in $(n, j+k a)$.

On exit, the contents of $A$ are destroyed.
LDAB (input)
The leading dim ension of the array AB. LD AB >= $K A+1$.

BB (input/output)
O n entry, the upper or low ertriangle of the Her$m$ tian band $m$ atrix $B$, stored in the first kb+1 row sof the array. The $j$ th colum n of $B$ is stored in the jth colum $n$ of the array $B B$ as follow s: if $\mathrm{UPLO}=\mathrm{U}, \mathrm{BB}(\mathrm{kb}+1+\mathrm{i}-j, j)=\mathrm{B}(i, j)$ for $\mathrm{max}(1, j$ $\mathrm{kb})<=\mathrm{i}<=\dot{j}$ if P PLO $=\mathrm{L}, \mathrm{BB}(1+i-j, j)=B(i, 7)$ for $j<=i<=m$ in $(n, \dot{j}+k b)$.

On exit, the factors from the splitCholesky factorization $\mathrm{B}=\mathrm{S} * * \mathrm{H} * \mathrm{~S}$, as retumed by CPBSTF .

LD BB (input)
The leading dim ension of the array BB. LD BB >= $\mathrm{KB}+1$.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.
$Z$ (input) If $\mathrm{JOBZ}=\mathrm{V}^{\prime}$, then if $\mathbb{N} F O=0, \mathrm{Z}$ contains the $m$ atrix $Z$ ofeigenvectors, $w$ th the $i$-th colum n of Z holding the eigenvector associated with W (i). The eigenvectors are norm alized so that Z **H *B *Z = I. If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced.

## LD $Z$ (input)

The leading dim ension of the array $Z$. LD $Z>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD $\mathrm{Z}>=\mathrm{N}$.

W ORK (w orkspace)
dim ension (N)
RW ORK (w orkspace)
dim ension $(3 * N)$
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue
> 0: if $\mathbb{N F O}=\mathrm{i}$, and i is:
<= N : the algorithm failed to converge: i off-
diagonal elem ents of an interm ediate tridiagonal form did notconverge to zero; > N : if $\mathbb{N} F O=N$ +i , for $1<=\mathrm{i}<=\mathrm{N}$, then CPBSTF
retumed $\mathbb{N} F O=i: B$ is not positive definite. The factorization ofB could notbe com pleted and no eigenvalues oreigenvectors w ere com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

chbgvd - com pute allthe eigenvalues, and optionally, the eigenvectors of a com plex generalized H erm itian-definite banded eigenproblem, of the form $A * x=(l a m ~ b d a) * B * x$

## SYNOPSIS

```
SUBROUT\mathbb{NE CHBGVD (JOBZ,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,W,Z,}
    LD Z,W ORK,LW ORK,RW ORK,LRW ORK,IN ORK,LIN ORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
COM PLEX AB (LDAB,*),BB (LDBB,*),Z (LD Z,*),W ORK (*)
\mathbb{NTEGER N,KA,KB,LDAB,LDBB,LD Z,LW ORK, LRW ORK,LIN ORK,}
\mathbb{NFO}
INTEGER IN ORK (*)
REALW (*),RW ORK (*)
SUBROUTINE CHBGVD_64 (JOBZ,UPLO,N,KA,KB,AB,LDAB,BB,LD BB,W,Z,
    LD Z,W ORK,LW ORK,RW ORK,LRW ORK,IN ORK,LIN ORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
COM PLEX AB (LDAB,*),BB (LDBB,*),Z (LD Z ,*),W ORK (*)
\mathbb{NTEGER*8N,KA,KB,LDAB,LDBB,LDZ,LW ORK,LRW ORK,LIN ORK,}
\mathbb{NFO}
INTEGER*8 IN ORK (*)
REALW (*),RW ORK (*)
```


## F95 INTERFACE

SU BROUTINE HBGVD (JOBZ,UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B], W$, Z, [LD Z], [W ORK ], [LW ORK], RW ORK ], [LRW ORK ], [WW ORK ], [LIW ORK ], [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1):: OBZ, UPLO
COMPLEX,D $\mathbb{I M} E N S I O N(:):: W O R K$
COM PLEX, D $\mathbb{I M} E N S I O N$ (: : : : : AB, BB, Z
$\mathbb{N}$ TEGER ::N, KA, KB, LDAB, LDBB, LD Z, LW ORK, LRW ORK,
LIW ORK, $\mathbb{N} F O$
$\mathbb{N}$ TEGER,D $\mathbb{I M} E N S I O N(:):: \mathbb{I W}$ ORK
REAL,D $\mathbb{I M}$ ENSION (:) ::W ,RW ORK

SU BROUTINE HBGVD_64 (OBZ, UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B]$, W , Z, [LDZ], [W ORK], [LW ORK], [RW ORK], [LRW ORK ], [ $\mathbb{W}$ ORK ], [ $\mathbb{Z} \mathbb{W} O R K]$, [ $\mathbb{N} \mathrm{FO}]$ )

CHARACTER (LEN=1): : OBBZ, UPLO
COM PLEX,D $\mathbb{I M} E N S I O N(:):$ ORK
COM PLEX , D $\mathbb{I M} E N S I O N$ (:,:) ::AB,BB,Z
$\mathbb{N} \operatorname{TEGER}(8):: N, K A, K B, L D A B, L D B B, L D Z, L W O R K, L R W O R K$,
L $\mathbb{V}$ ORK, $\mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}$ (:) :: $\mathbb{I V}$ ORK
REAL,D $\mathbb{I}$ ENSION (:) ::W ,RW ORK

## C INTERFACE

\#include <sunperfh>
void chbgvd (char jobz, charuplo, intn, int ka, int kb, complex *ab, int ldab, com plex *bbb, int ldbb, float * ${ }_{\text {w }}$, com plex * $z$, int ldz, int *info);
void chbogvd_64 (char jobz, char uplo, long n, long ka, long kb, com plex *ab, long ldab, com plex *bb, long ldbb, float *w , com plex * z, long ldz, long *info);

## PURPOSE

chbgvd com putes all the eigenvalues, and optionally, the eigenvectors of a com plex generalized $H$ erm itian-definite banded eigenproblem, of the form $A * x=\left(l a m\right.$ bda) ${ }^{*} B{ }^{*} x$. H ere $A$ and $B$ are assum ed to be $H$ erm titian and banded, and $B$ is also positive definite. If eigenvectors are desired, it uses a divide and conquer algorithm .

The divide and conqueralgorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on $m$ achines w ith a guard digit in add/subtract, or on those binary $m$ achines $w$ thout guard digits $w$ hich subtract like the $C$ ray $\mathrm{X}-\mathrm{M} P, \mathrm{C}$ ray $\mathrm{Y}-\mathrm{M} \mathrm{P}$, C ray $\mathrm{C}-90$, or C ray -2 . It could conceivably fail on hexadecim al or decim al machines $w$ ithout guard digits, butw e know of none.

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}$ ': C om pute eigenvalues only;
$=\mathrm{V}:$ : C om pute eigenvalues and eigenvectors.
UPLO (input)
$=\mathrm{U}$ ': U pper triangles of $A$ and $B$ are stored;
$=\mathrm{L}$ ': Low ertriangles of $A$ and $B$ are stored.

N (input) The order of the m atriges $A$ and $B . N>=0$.

KA (input)
The num ber of superdiagonals of the $m$ atrix $A$ if $\mathrm{UPLO}=\mathrm{U}$ ', orthe num berof subdiagonals if UPLO
$=\mathbb{L}$ '.KA >=0.
K B (input)
The num ber of superdiagonals of the $m$ atrix $B$ if $\mathrm{UPLO}=\mathrm{U}$ ', orthe num ber of subdiagonals if UPLO $=\mathbb{L} . \mathrm{KB}>=0$.

A B (input/output)
O n entry, the upper or low ertriangle of the H er$m$ titian band $m$ atrix $A$, stored in the firstka+1 row s of the array. The $j$ th colum n of A is stored in the jth colum n of the amay AB as follow s: if $\mathrm{UPLO}=\mathrm{U}, \mathrm{AB}(\mathrm{ka}+1+i-j)=\mathrm{A}(i, 1)$ for $\mathrm{max}(1, j$ $\mathrm{ka})<=\dot{i}=j$ ifUPLO $=\mathrm{L}^{\prime}, \mathrm{AB}(1+i-j, j)=A(i, j)$ for $\dot{j}=i<=m$ in $(n, j+k a)$.

On exit, the contents of A B are destroyed.

LDAB (input)
The leading dim ension of the array AB. LDAB >= $K A+1$.

BB (input/output)
O n entry, the upper or low er triangle of the H er$m$ titian band $m$ atrix $B$, stored in the first $k b+1$ row s of the aray. The $j$ th colum n of $B$ is stored in the $j$ th colum $n$ of the array $B B$ as follow $s$ : if $\mathrm{UPLO}=\mathrm{U}, \mathrm{BB}(\mathrm{kb}+1+i-j, j)=\mathrm{B}(i, j)$ for $\mathrm{max}(1, j$ $\mathrm{kb})<=\dot{i}<=\dot{j}$ ifUPLO $=\mathrm{L}, \mathrm{BB}(1+i-j, j)=\mathrm{B}(i, 7)$ for $\dot{j}=i<=m$ in $(n, j+k b)$.

O n exit, the factors from the splitCholesky factorization $B=S * * H * S$, as retumed by CPBSTF .

LD BB (input)

The leading dim ension of the array BB. LD BB >= K B+1.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.
Z (input) If $\mathrm{OOBZ}=\mathrm{V}$ ', then if $\mathbb{N} F O=0, \mathrm{Z}$ contains the $m$ atrix $Z$ of eigenvectors, $w$ ith the $i$-th colum $n$ of $Z$ holding the eigenvector associated $w$ ith $W$ (i). The eigenvectors are norm alized so that $\mathrm{Z} * * \mathrm{H} * \mathrm{~B} * \mathrm{Z}=$ I. If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced.

LD $Z$ (input)
The leading din ension of the array $\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{N}$.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array $\mathrm{W} O R \mathrm{~K}$. If $\mathrm{N}<=1$, LW ORK >=1. If $\mathrm{JOBZ}=\mathrm{N}$ 'and $\mathrm{N}>1$, LW ORK $>=\mathrm{N}$. If $\mathrm{JOBZ}=\mathrm{V}$ 'and $\mathrm{N}>1$, LW $O R K>=2 * \mathrm{~N} * * 2$.

If LW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

RW ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, R W$ ORK (1) retums the optim al LRW ORK.

LRW ORK (input)
The dimension of array RW ORK. If $\mathrm{N}<=1$, LRW ORK >= 1. If $\mathrm{JOBZ}=\mathrm{N}$ 'and $N>1$, LRW $O R K>=$ N . If JOBZ $=\mathrm{V}$ 'and $\mathrm{N}>1$,LRW $O R K>=1+5^{\star} \mathrm{N}+$ $2{ }^{*} N * * 2$ 。

If LRW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the RW ORK array, retums this value as the first entry of the RW ORK amay, and no emorm essage related to LRW ORK is issued by X ERBLA.

IV ORK (w orkspace/output)
On exit, if $\mathbb{N} F O=0, \mathbb{I N}$ ORK (1) retums the optim al LIV ORK.

LIV ORK (input)
The dim ension of array $\mathbb{I N}$ ORK. If JOBZ $=\mathrm{N}$ 'or N
$<=1$, LIV ORK >= 1. If $\mathrm{OOBZ}=\mathrm{V}$ 'and $\mathrm{N}>1$, LIV ORK $>=3+5{ }^{*} \mathrm{~N}$.

If $\mathrm{L} \mathbb{I V}$ ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim al size of the $\mathbb{I W}$ ORK array, retums this value as the first entry of the $\mathbb{I V}$ ORK array, and no errorm essage related to $L \mathbb{I W} O R K$ is issued by XERBLA.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the i-th argum enthad an illegalvalue
$>0$ : if $\mathbb{N F O}=i$, and $i$ is:
$<=\mathrm{N}$ : the algorithm failed to converge: i offdiagonal elem ents of an interm ediate tridiagonal form did not converge to zero; $>\mathrm{N}$ : if $\mathbb{N} F \mathrm{FO}=\mathrm{N}$ +i , for $1<=\mathrm{i}<=\mathrm{N}$, then CPBSTF
retumed $\mathbb{N} F O=i: B$ is not positive definite. The factorization ofB could notbe com pleted and no eigenvalues or eigenvectors w ere com puted.

## FURTHER DETAILS

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U nìv.ofK entucky, USA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

chbgvx - com pute allthe eigenvalues, and optionally, the eigenvectors of a com plex generalized H erm itian-definite banded eigenproblem, of the form $A * x=(l a m ~ b d a) * B * x$

## SYNOPSIS

```
SUBROUTINECHBGVX (JOBZ,RANGE,UPLO,N,KA,KB,AB,LDAB,BB,LDBB, \(Q, L D Q, V L, V U, \mathbb{L}, \mathbb{U}, A B S T O L, M, W, Z, L D Z, W\) ORK,RW ORK, \(\mathbb{I W} O R K\), \(\mathbb{F A} \mathbb{L}, \mathbb{N} F O)\)
```

CHARACTER * 1 JobZ,RANGE,UPLO
COM PLEX AB (LDAB,*), BB (LDBB,*), Q (LDQ , *), Z (LDZ,*), W ORK (*)
$\mathbb{N} T E G E R N, K A, K B, L D A B, L D B B, L D Q, \mathbb{L}, \mathbb{U}, M, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R \mathbb{I N} O R K(*), \mathbb{F A} \mathbb{L}(*)$
REALVL,VU,ABSTOL
REALW (*) ,RWORK (*)
SUBROUTINE CHBGVX_64 (OBZ,RANGE, UPLO,N,KA,KB,AB,LDAB,BB, LD BB, Q,LDQ,VL,VU, $\mathbb{L}, \mathbb{I}, A B S T O L, M, W, Z, L D Z, W$ ORK,RW ORK, $\mathbb{I N} O R K, \mathbb{F} A \mathbb{I}, \mathbb{N} F O)$

CHARACTER * 1 JOBZ,RANGE, UPLO

$\mathbb{N} T E G E R * 8 N, K A, K B, L D A B, L D B B, L D Q, \mathbb{L}, \mathbb{U}, M, L D Z, \mathbb{N} F O$
$\mathbb{N}$ TEGER*8 $\mathbb{I V}$ ORK (*), $\mathbb{F A} \mathbb{I}(*)$
REALVL,VU,ABSTOL
REALW (*),RWORK (*)

## F95 INTERFACE

SU BROUTINE HBGVX (OBZ,RANGE, UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B$, $\left[\begin{array}{ll}\text { D BB }], Q,[L D Q], V L, V U, \mathbb{L}, \mathbb{U}, A B S T O L, M, W, Z,[L D Z],[W O R K], ~\end{array}\right.$
$[R W$ ORK], [IN ORK], $\mathbb{F A} \mathbb{I},[\mathbb{N} F O])$

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{I M}$ ENSION (: : : : : AB, BB, $\mathrm{Q}, \mathrm{Z}$
$\mathbb{N} T E G E R:: N, K A, K B, L D A B, L D B B, L D Q, \mathbb{L}, \mathbb{U}, M, L D Z, \mathbb{N F O}$
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{I N} O R K, \mathbb{F A} \mathbb{L}$
REAL ::VL,VU,ABSTOL
REAL,D $\mathbb{I M} E N S I O N(:):: W, R W O R K$
SUBROUTINE HBGVX_64 (JOBZ,RANGE,UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B$, [LDBB],Q, [LDQ],VL,VU, $\mathbb{L}, \mathbb{U}, A B S T O L, M, W, Z,[L D Z],[W O R K]$, [RW ORK], [IN ORK], $\mathbb{F} A \mathbb{I},[\mathbb{N} F O]$ )

CHARACTER (LEN=1):: JOBZ,RANGE,UPLO
COMPLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{I M}$ ENSION (: : : : : AB, $B B, Q, Z$
$\mathbb{N} \operatorname{TEGER}(8):: N, K A, K B, L D A B, L D B B, L D Q, \mathbb{L}, \mathbb{U}, M, L D Z$,
$\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N} O R K, \mathbb{F A} \mathbb{I}$
REAL ::VL,VU,ABSTOL
REAL,D IM ENSION (:) ::W ,RW ORK

## C INTERFACE

\#include < sunperfh>
void chbgvx (char jobz, char range, charuplo, intn, int ka, int kb, com plex *ab, int ldab, com plex *bb, int ldbb, com plex *q, int ldq, float vl, float vu, int il, int iu, floatabstol, int *m, float* w, com plex *z, int ldz, int*ifail, int*info);
void chbovvx_64 (char jobz, char range, char uplo, long n, long ka, long kb, com plex *ab, long ldab, com plex
*bb, long ldbb, com plex *q, long ldq, float vl, float vu, long il, long iu, float abstol, long $\star_{m}$, float *w , com plex *z, long ldz, long *ifail, long *info);

## PURPOSE

chbgvx com putes all the eigenvalues, and optionally, the eigenvectors of a com plex generalized $H$ erm itian-definite banded eigenproblem, of the form $A * x=(l a m . b d a) * B * x$. H ere A and $B$ are assum ed to be $H$ erm itian and banded, and $B$ is also positive definite. Eigenvalues and eigenvectors can be selected by specifying either alleigenvalues, a range of values ora range of indices for the desired eigenvalues.

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}^{\prime}:$ C om pute eigenvalues only;
$=\mathrm{V}$ ': C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvahues w illbe found;
$=\mathrm{V}$ : alleigenvahues in the half-open interval (VL, VU ] w ill be found; = I': the II-th through $\mathbb{I U}$-th eigenvaluesw illbe found.

UPLO (input)
$=\mathrm{U}$ ': U pper triangles of $A$ and $B$ are stored;
= ${ }^{L}$ ': Low er triangles ofA and B are stored.
N (input) The order of the m atriges A and $\mathrm{B} . \mathrm{N}>=0$.

KA (input)
The num ber of superdiagonals of the $m$ atrix $A$ if $\mathrm{UPLO}=\mathrm{U}$ ', or the num ber of subdiagonals if $\mathrm{U} P L O$
$=\mathbb{L} \cdot \mathrm{KA}>=0$.

K B (input)
The num ber of superdiagonals of the $m$ atrix $B$ if $\mathrm{UPLO}=\mathrm{U}$ ', orthe num ber of subdiagonals if U PLO $=\mathrm{L} \cdot \mathrm{KB}>=0$ 。

A B (input/output)
O n entry, the upper or low ertriangle of the H er$m$ titian band $m$ atrix $A$, stored in the firstka+1 row s of the array. The $j$ th colum n of A is stored in the fth colum $n$ of the array A B as follow s: if $\mathrm{UPLO}=\mathrm{U}$ ', AB $(k a+1+i-j)=A(i, j)$ for $m$ ax $(1, j$ $\mathrm{ka})<=\dot{i}=\dot{j}$ ifUPLO $=\mathrm{L} ', A B(1+i-j, j)=A(i, j)$ for $\dot{j}=i<=m$ in $(n, j+k a)$.

On exit, the contents ofA B are destroyed.

LDAB (input)
The leading dim ension of the aray AB. LDAB >= K A +1 .

B B (input/output)
O n entry, the upper or low ertriangle of the H er$m$ Itian band $m$ atrix $B$, stored in the firstkb+1
row s of the anray. The $j$ th colum n ofB is stored in the $j$ th colum n of the array B B as follow $s$ : if $\mathrm{UPLO}=\mathrm{U}, \mathrm{BB}(\mathrm{kb}+1+i-j, j)=\mathrm{B}(1, j)$ for m ax $(1, j$
$\mathrm{kb})<=\mathrm{i}<=\dot{j}$ if UPLO $=\mathrm{L}, \mathrm{BB}(1+i-j, j)=B(i, 7)$
for $j=i<=m$ in $(n, j+k b)$.
On exit, the factors from the splitCholesky factorization $B=S * * H * S$, as retumed by CPBSTF .

LD BB (input)
The leading dim ension of the array $B B$. LD BB $>=$ K B+1.

Q (output)
If $J O B Z=V$ ', the $n-b y-n$ matrix used in the reduction of ${ }^{*} \mathrm{x}=\left(\mathrm{lam}\right.$ bda) ${ }^{\mathrm{B}} \mathrm{B}^{\mathrm{x}} \mathrm{x}$ to standard form, i.e. $C * x=\left(l a m\right.$ bda) ${ }^{*} \mathrm{x}$, and consequently C to tridiagonal form. If $J O B Z=N$ ', the array $Q$ is not referenced.

LDQ (input)
The leading dim ension of the array Q . If $\mathrm{JOBZ}=$ $N^{\prime}, L D Q>=1$. If $\mathrm{OBBZ}=\mathrm{V}$ ', LD Q >= $\max (1, N)$.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE=A 'or I'.

VU (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE=A 'or I '.

II (input)
If RA N GE= 'I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{\Pi}<=\mathbb{U}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{L}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N ot referenced ifRANGE $=$ A'or V'.
$\mathbb{I U}$ (input)
IfRA NGE= I', the indices (in ascending order) of the sm allest and largest eigenvalues to be retumed. $1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{L}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N otreferenced ifRANGE $=$ A'or V'.

ABSTOL (input)
The absolute error tolerance for the eigenvalues.
A $n$ approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b]
of w idth less than orequalto

ABSTOL + EPS * $\max (|k|, \mid)$,
where EPS is the machine precision. IfABSTOL is less than or equalto zero, then EPS* $\mid$ |w illibe used in its place, where $T$ is the 1 -norm of the tridiagonalm atrix obtained by reducing AP to tridiagonal form .
E igenvalues w illbe com puted m ostaccurately when
ABSTOL is set to tw ice the underflow threshold $2 \star$ SLAM CH (S ), notzero. If this routine retums w ith $\mathbb{N}$ FO $>0$, indicating that som e eigenvectors did not converge, try setting ABSTO L to $2 *$ SLAM CH (S ).

M (output)
The total num ber ofeigenvalues found. $0<=\mathrm{M}$ <= N . IfRANGE $=\mathrm{A}^{\prime}, \mathrm{M}=\mathrm{N}$, and ifRANGE $=\mathrm{I}^{\prime}, \mathrm{M}=$ $\mathbb{I U}-\mathbb{L}+1$.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.
$Z$ (input) If $J O B Z=V '$, then if $\mathbb{N} F O=0, Z$ contains the $m$ atrix $Z$ ofeigenvectors, $w$ th the $i$-th colum n of Z holding the eigenvector associated with W (i). The eigenvectors are norm alized so that $\mathrm{Z} * * \mathrm{H} * \mathrm{~B} * \mathrm{Z}=$ I. If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced.

LD Z (input)
The leading dim ension of the array $Z$. LD $Z>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD Z >=N.

W ORK (w orkspace)
dim ension $(\mathbb{N})$
RW ORK (w orkspace)
dim ension ( $7 * \mathrm{~N}$ )
IN ORK (w orkspace)
dim ension ( $5 * \mathrm{~N}$ )
FAII (output)
If $\mathrm{OBZ}=\mathrm{V}$ ', then if $\mathbb{N F O}=0$, the firstM ele$m$ ents of $\mathbb{F} A \mathbb{L}$ are zero. If $\mathbb{N} F O>0$, then $\mathbb{F} A \mathbb{L}$ contains the indices of the eigenvectors that failed to converge. If $J O B Z=N$ ', then $\mathbb{F A} I$ is not referenced.
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N F O}=-$ i, the $i$-th argum enthad an illegalvahue
> 0: if $\mathbb{N F O}=\mathrm{i}$, and i is:
$<=\mathrm{N}$ : then i eigenvectors failed to converge. Their indices are stored in array $\mathbb{F A} \mathbb{I} .>N$ : if $\mathbb{N} F O=N+i$, for $1<=i<=N$, then CPBSTF retumed $\mathbb{N} F O=i: B$ is not positive definite. The factorization ofB could notbe com pleted and no eigenvalues oreigenvectors w ere com puted.

## FURTHER DETAILS

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv. of K entucky, USA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chbm v -perform the m atrix-vectoroperation $\mathrm{y}:=\operatorname{alpha} \mathrm{A} \mathrm{A} * \mathrm{x}$

+ beta*y


## SYNOPSIS

```
SUBROUTINE CHBMV (UPLO,N,K,ALPHA,A,LDA,X,\mathbb{NCX,BETA,Y,} \(\mathbb{N} C Y\) )
CHARACTER * 1 UPLO
COM PLEX ALPHA,BETA
COM PLEX A (LDA,*),X (*), Y (*)
\(\mathbb{N} T E G E R N, K, L D A, \mathbb{N} C X, \mathbb{N} C Y\)
SU BROUTINE CHBMV_64 (UPLO ,N,K,ALPHA,A,LDA,X, \(\mathbb{N} C X, B E T A, Y\), \(\mathbb{N} C Y\) )
```

CHARACTER * 1 UPLO
COM PLEX ALPHA,BETA
COM PLEX A (LDA,$\star), X(*), Y(*)$
$\mathbb{N}$ TEGER*8 $\mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathbb{N} C X, \mathbb{N} \mathrm{CY}$

## F95 INTERFACE

SU BROUT $\mathbb{N} E$ HBMV $\mathbb{U}$ PLO , $\mathbb{N}], K, A L P H A, A,[L D A], X,[\mathbb{N C X}], B E T A$, Y, [ $\mathbb{N} C Y]$ )

CHARACTER (LEN=1) ::UPLO
COMPLEX ::ALPHA,BETA
COM PLEX,D IM ENSION (:) :: X,Y
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : A
$\mathbb{N} T E G E R:: N, K, L D A, \mathbb{N} C X, \mathbb{N} C Y$

SU BROUTINE HBM V_64 (UPLO, $\mathbb{N}], K, A L P H A, A,[L D A], X,[\mathbb{N} C X]$,

BETA, Y, [ $\mathbb{N} C Y])$

CHARACTER (LEN=1) ::UPLO
COMPLEX ::ALPHA,BETA
COM PLEX,D IM ENSION (:) :: X,Y
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathbb{N} \mathrm{CX}, \mathbb{N} \mathrm{CY}$

## C INTERFACE

\#include < sunperfh>
void chbm $v$ (charuplo, intn, intk, com plex *alpha, com plex
*a, int lda, com plex *x, int incx, com plex *beta, com plex *y, int incy);
void chbm v_64 (char uplo, long n, long k, com plex *alpha, com plex *a, long lda, com plex *x, long incx, com plex *beta, com plex *y, long incy);

## PURPOSE

chbm $v$ perform $s$ the $m$ atrix-vector operation $y:=a l p h a * A * x+$ beta* $y$ where alpha and beta are scalars, $x$ and $y$ are $n$ ele$m$ ent vectors and $A$ is an $n$ by $n$ herm tian band $m$ atrix, $w$ ith k super-diagonals.

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the bandmatrix $A$ is being supplied as follow s:

UPLO $=\mathrm{U}$ 'or $\mathrm{L}^{\prime}$ ' The uppertriangularpartofA is being supplied.

UPLO = L'or $\mathrm{I}^{\prime}$ ' The low ertriangularpart of A is being supplied.

U nchanged on exit.

N (input)
O $n$ entry, $N$ specifies the order of the $m$ atrix A. $\mathrm{N}>=0$. U nchanged on exit.

K (input)
On entry, $K$ specifies the number of superdiagonals of them atrix A. K m ust satisfy 0 .le.

K . U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
Before entry w ith UPLO = U 'or G ', the leading ( $k+1$ ) by $n$ part of the array A $m$ ust contain the upper triangular band part of the herm tian $m$ atrix, supplied colum $n$ by colum $n$, $w$ th the leading diagonal of the $m$ atrix in row ( $k+1$ ) of the array, the first super-diagonalstarting atposition 2 in row $k$, and so on. The top left $k$ by $k$ triangle of the anray $A$ is not referenced. The follow ing program segm entw illtransfer the upper triangular part of a herm itian band $m$ atrix from conventional fullm atrix storage to band storage:

$$
\begin{aligned}
& \text { DO } 20, \mathrm{~J}=1, \mathrm{~N} \\
& \mathrm{M}=\mathrm{K}+1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{MAX}(1, \mathrm{~J}-\mathrm{K}), \mathrm{J} \\
& \text { A }(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \text { CONTINUE } \\
& 20 \text { CONTINUE }
\end{aligned}
$$

Before entry w ith UPLO = L 'or 1', the leading ( $k+1$ ) by $n$ part of the array A m ust contain the low er triangular band part of the herm itian $m$ atrix, supplied colum $n$ by colum $n$, $w$ th the leading diagonalof them atrix in row 1 of the amay, the first sub-diagonalstarting atposition 1 in row 2 , and so on. The bottom right k by $k$ triangle of the amay $A$ is not referenced. The follow ing program segm entw ill transfer the low er triangular part of a herm itian band $m$ atrix from conventional fullm atrix storage to band storage:

$$
\begin{aligned}
& \text { DO } 20, J=1, N \\
& \mathrm{M}=1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{J}, \mathrm{M} \mathbb{N}(\mathrm{~N}, \mathrm{~J}+\mathrm{K}) \\
& \mathrm{A}(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \text { CONTINUE } \\
& 20 \text { CONTINUE }
\end{aligned}
$$

N ote that the im aginary parts of the diagonalele$m$ ents need notbe set and are assum ed to be zero. U nchanged on exit.

O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD A >= ( $\mathrm{k}+1$ ). U nchanged on exit.

X (input)
$(1+(n-1) * a b s(\mathbb{N} C X))$. Before entry, the
increm ented array $X$ must contain the vectorx. U nchanged on exit.
$\mathbb{N C X}$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of $\mathrm{X} . \mathbb{N} C X$ m ustnotbe zero. U nchanged on exit.
BETA (input)
On entry, BETA specifies the scalar beta. U nchanged on exit.

Y (input/output)
$(1+(n-1) * a b s(\mathbb{N} C Y))$. Before entry, the increm ented array $Y$ m ustcontain the vectory. On exit, $Y$ is overw rilten by the updated vectory.
$\mathbb{N C Y}$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of Y. $\mathbb{N} C Y$ m ustnotbe zero. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

chbtrd -reduce a com plex H erm tian band $m$ atrix $A$ to real sym metric tridiagonal form $T$ by a unitary sim ilarity transform ation

## SYNOPSIS

```
SUBROUT\mathbb{NECHBTRD NECT,UPLO,N,KD,AB,LDAB,D,E,Q,LDQ,W ORK,}
    \mathbb{NFO)}
CHARACTER * 1VECT,UPLO
COM PLEX AB (LDAB,*),Q (LDQ,*),W ORK (*)
\mathbb{NTEGER N,KD,LDAB,LDQ,INFO}
REALD (*),E (*)
SUBROUT\mathbb{NECHBTRD_64NECT,UPLO,N,KD,AB,LDAB,D,E,Q,LDQ,W ORK,}
    \mathbb{NFO)}
CHARACTER * 1 VECT,UPLO
COM PLEX AB (LDAB,*),Q (LDQ ,*),W ORK (*)
\mathbb{NTEGER*8N,KD,LDAB,LDQ,INFO}
REALD (*),E (*)
```


## F95 INTERFACE

```
SU BROUTINE HBTRD (NECT, UPLO, \(\mathbb{N}], K D, A B,[L D A B], D, E, Q,[L D Q]\), [ W ORK ], \([\mathbb{N} F O\) ])
CHARACTER (LEN=1) ::VECT,UPLO
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{I M}\) ENSION (: : : : : AB, Q
\(\mathbb{N} T E G E R:: N, K D, L D A B, L D Q, \mathbb{N} F O\)
REAL,D \(\mathbb{I M}\) ENSION (:) ::D ,E
```

SU BROUTINE HBTRD_64 NECT, UPLO, $\mathbb{N}], K D, A B,[L D A B], D, E, Q,[L D Q]$, [ W ORK], [ $\mathbb{N} F \mathrm{O}$ ])

CHARACTER (LEN=1) ::VECT,UPLO
COM PLEX,D IM ENSION (:) ::W ORK
COMPLEX,D $\mathbb{M}$ ENSION (: : : : : AB, Q
$\mathbb{N} \operatorname{TEGER}(8):: N, K D, L D A B, L D Q, \mathbb{N} F O$
REAL,D IM ENSION (:) ::D, E

## C INTERFACE

\#include < sunperfh>
void chbtrd (charvect, charuplo, int n, int kd, com plex *ab, int ldab, float *d, float *e, com plex *q, int ldq, int *info);
void chbtrd_64 (charvect, charuplo, long n, long kd, com plex *ab, long ldab, float*d, float*e, com plex
*q, long ldq, long *info);

## PURPOSE

chbtrod reduces a com plex $H$ erm itian band $m$ atrix $A$ to real symmetric tridiagonal form $T$ by a unitary similarity transform ation: Q ** $\mathrm{H} * \mathrm{~A} * \mathrm{Q}=\mathrm{T}$.

## ARGUMENTS

VECT (input)
$=N^{\prime}$ : do not form $Q$;
$=\mathrm{V}$ : form Q ;
$=\mathrm{U}$ : update am atrix X , by form ing $\mathrm{X} * \mathrm{Q}$.

UPLO (input)
= U ': U pper triangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the matrix A. $\mathrm{N}>=0$.
KD (input)
The num ber of superdiagonals of the $m$ atrix $A$ if UPLO $=\mathrm{U}$ ', ort the num berof subdiagonals if UPLO $=\mathbb{L}^{\prime} . \mathrm{KD}>=0$.

AB (input/output)
On entry, the upper or low ertriangle of the Her $m$ itian band $m$ atrix A, stored in the firstK $D+1$
row sof the anray. The $j$ th colum n of $A$ is stored in the $j$ th colum $n$ of the array AB as follow s: if $\mathrm{UPLO}=\mathrm{U}, \mathrm{AB}(\mathrm{kd}+1+\mathrm{i}-j)=\mathrm{j}(i, 1)$ for $\mathrm{max}(1, j$ $\mathrm{kd})<=i<=j$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{AB}(1+i-j)=A(i, 7)$ for $\dot{j}=i<=m$ in $(n, \dot{j}+\mathrm{kd})$. O $n$ exit, the diagonalele$m$ ents of AB are overw ritten by the diagonalele$m$ ents of the tridiagonalm atrix T ; if K D > 0, the elem ents on the first superdiagonal (if UPLO $=$ U ) orthe first subdiagonal (ifU PLO = L ) are overw rilten by the off-diagonalelem ents of $T$; the rest of A B is overw ritten by values generated during the reduction.

LDAB (input)
The leading dim ension of the array AB. LDAB >= K D +1 .

D (output)
The diagonalelem ents of the tridiagonalm atrix T .
E (output)
The off-diagonal elem ents of the tridiagonal m atrix $\mathrm{T}: \mathrm{E}(\mathrm{i})=\mathrm{T}(\mathrm{i}, \mathrm{i}+1)$ if $\mathrm{PLO}=\mathrm{U} ; \mathrm{E}(\mathrm{i})=$ $T(i+1, i)$ if $\mathrm{PLLO}=\mathrm{L}^{\prime}$ 。

Q (input/output)
Onentry, ifVECT = U', then $Q \mathrm{must}$ contain an N by -N m atrix X ; if VECT $=\mathrm{N}$ 'or V ', then Q need notbe set.

On exit: if $\mathrm{VECT}=\mathrm{V}$ ', Q contains the N -by -N unitary matrix $Q$; ifVECT = U', $Q$ contains the product $X$ * $Q$; ifVECT $=N$ ', the array $Q$ is not referenced.

LD Q (input)
The leading $d i m$ ension of the array $Q . L D Q>=1$, and LDQ >= N ifVECT = V'or U'.

W ORK (w orkspace)
dim ension (N)
$\mathbb{N} F O$ (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an illegalvalue

## FURTHER DETAILS

M odified by Linda K aufn an, Bell Labs.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

checon -estim ate the reciprocal of the condition num ber of a complex Herm tian $m$ atrix $A$ using the factorization $A=$ $\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}$ orA $=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}$ com puted by CHETRF

## SYNOPSIS

```
SUBROUT\mathbb{NE CHECON(UPLO,N,A,LDA, \mathbb{PIVOT,ANORM,RCOND,WORK,INFO)}}\mathbf{N}\mathrm{ (N,}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGERN,LDA,}\mathbb{NFO}
\mathbb{NTEGER \mathbb{PIVOT (*)}}\mathbf{(*)}
REAL ANORM,RCOND
SUBROUTINECHECON_64(UPLO,N,A,LDA, \mathbb{PIVOT,ANORM,RCOND,WORK,}
        \mathbb{NFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGER*8N,LDA,}\mathbb{N}FO
INTEGER*8 \mathbb{PIVOT (*)}
REALANORM,RCOND
F95 INTERFACE
```



```
        [\mathbb{NFO ])}
    CHARACTER (LEN=1) ::UPLO
    COMPLEX,DIM ENSION (:) ::W ORK
    COM PLEX,D IM ENSION (:,:) ::A
    \mathbb{NTEGER::N,LDA,}\mathbb{NFO}
    \mathbb{NTEGER,D IM ENSION (:)::\mathbb{PIVOT}}\mathbf{T}\mathrm{ (: }
```

REAL ::ANORM,RCOND

SU BROUTINE HECON_64 (UPLO, $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I} O T, A N O R M, R C O N D,[W O R K]$, [ $\mathbb{N} \mathrm{FO}]$ )

CHARACTER (LEN=1) :: UPLO
COM PLEX,D $\mathbb{I M} E N S I O N(:):$ W ORK
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : A
$\mathbb{I N}$ TEGER (8) :: N, LD A , $\mathbb{N F}$ F
$\mathbb{N} \operatorname{TEGER}$ (8), D $\mathbb{M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT}$
REAL ::ANORM,RCOND

## C INTERFACE

\#include <sunperfh>
void checon (charuplo, int $n$, com plex *a, int lda, int *ípívot, floatanorm , float *roond, int *info);
void checon_64 (charuplo, long n, com plex *a, long lda, long *ipivot, floatanorm , float *roond, long *info);

## PURPOSE

checon estim ates the reciprocal of the condition num ber of a complex Herm itian matrix A using the factorization A = $\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}$ orA $=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}$ com puted by CHETRF.

A $n$ estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND $=1$ / (ANORM * norm (inv (A )) ).

## ARGUMENTS

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
$=\mathrm{U}$ : U ppertriangular, form is $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}$ ** H ;
$={ }^{\prime} \mathrm{L}$ ': Low er triangular, form is $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}$.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input) The block diagonalm atrix D and the m ultipliers used to obtain the factorU orL as com puted by CHETRF.

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, N)$.
$\mathbb{P} \mathbb{V} O T$ (input)
D etails of the interchanges and the block structure ofD as determ ined by CHETRF.

## ANORM (input)

The 1-norm of the originalm atrix A.

## RCOND (output)

The reciprocal of the condition number of the $m$ atrix $A$, com puted as RCOND $=1$ ( $A N O R M * A \mathbb{N} V N M$ ), where $A \mathbb{N} V N M$ is an estim ate of the 1 -nom of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension ( $2 * \mathrm{~N}$ )
$\mathbb{I N F O}$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cheev - com pute alleigenvalues and, optionally, eigenvectors of a com plex H erm itian m atrix A

## SYNOPSIS

```
SUBROUT\mathbb{NE CHEEV (JOBZ,UPLO,N,A,LDA,W ,W ORK,LDW ORK,WORK2,\mathbb{NFO )}}\mathbf{N}\mathrm{ , (T)}
CHARACTER * 1 OOBZ,UPLO
COM PLEX A (LDA,*),W ORK (*)
INTEGERN,LDA,LDW ORK, INFO
REALW (*),W ORK 2 (*)
SU BROUTINE CHEEV_64(JOBZ,UPLO,N,A,LDA,W ,W ORK,LDW ORK,W ORK2,
    INFO)
CHARACTER * 1 JOBZ,UPLO
COM PLEX A (LDA,*),W ORK (*)
INTEGER*8N,LDA,LDW ORK,INFO
REALW (*),W ORK2 (*)
```


## F95 INTERFACE

```
SU BROUTINE HEEV (OBZ,UPLO, \(\mathbb{N}], A,[L D A], W,[\mathbb{W}\) ORK ], [LDW ORK], [W ORK 2], [ \(\mathbb{N} F \mathrm{~F}\) ])
CHARACTER (LEN=1): : JOBZ, UPLO
COMPLEX,D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: N, L D A, L D W O R K, \mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::W ,W ORK2
SU BROUTINE HEEV_64 (JOBZ, UPLO, \(\mathbb{N}], A,[L D A], W,[\mathbb{W}\) ORK ], [LDW ORK ], [W ORK2], [ \(\mathbb{N} F \mathrm{O}\) ])
```

CHARACTER (LEN=1): : OBBZ, UPLO
COM PLEX,D $\mathbb{I M} E N S I O N(:):$ ORK
COMPLEX ,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R(8):: N$, LDA, LDW ORK, $\mathbb{N} F O$
REAL,D $\mathbb{I}$ ENSION (:) ::W ,W ORK2

## C INTERFACE

\#include <sunperfh>
void cheev (char jobz, char uplo, intn, com plex *a, int lda, float *w, int *info);
void cheev_64 (char joboz, charuplo, long n, com plex *a, long lda, float *w , long *info);

## PURPOSE

cheev com putes alleigenvalues and, optionally, eigenvectors of a com plex $H$ erm itian $m$ atrix A .

## ARGUMENTS

JO B Z (input)
$=\mathrm{N}$ : C om pute eigenvalues only;
$=\mathrm{V}$ : C om pute eigenvalues and eigenvectors.

UPLO (input)
$=U^{\prime}:$ U ppertriangle of $A$ is stored;
$=\mathbb{L}$ ': Low ertriangle ofA is stored.

N (input) The order of the matrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
On entry, the H erm itian $m$ atrix A . If $\mathrm{UPLO}=\mathrm{U}$ ', the leading N -by N uppertriangularpart of A contains the upper triangular part of the $m$ atrix $A$. If U PLO $=\mathrm{L}$ ', the leading N -by-N low er triangular part ofA contains the low er triangular part of the $m$ atrix $A$. On exit, if $J O B Z=V$ ', then if $\mathbb{N} F O=0, A$ contains the orthonorm al eigenvectors of them atrix A . If $\mathrm{OOBZ}=\mathrm{N}$ ', then on exit the low er triangle (if $\mathrm{U} P \mathrm{PO}=\mathrm{L}$ ) or the upper triangle (if $\mathrm{UPLO}=\mathrm{U}$ ) of $A$, including the diagonal, is destroyed.

The leading dim ension of the array A. LDA >= $\max (1, N)$.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.
W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al
LDW ORK.
LDW ORK (input)
The length of the array W ORK. LDW ORK >= max $(1,2 * \mathrm{~N}-1)$. For optim alefficiency, LD W ORK >= $(N B+1) * N$, where $N B$ is the blocksize for CHETRD retumed by ILAENV.

If LD W ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 2 (w orkspace)
dim ension (max (1,3*N-2))
$\mathbb{N} F O$ (output)
= 0 : successfiulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue
$>0$ : if $\mathbb{N} F O=$ i, the algonithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cheevd - com pute alleigenvalues and, optionally, eigenvectors of a com plex H erm itian m atrix A

## SYNOPSIS

```
SUBROUT\mathbb{NE CHEEVD (JOBZ,UPLO,N,A,LDA,W ,W ORK,LW ORK,RW ORK,}
    LRW ORK,IN ORK,LIN ORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
COM PLEX A (LDA,*),W ORK (*)
```



```
INTEGER IV ORK (*)
REALW (*),RW ORK (*)
```

SU BROUTINE CHEEVD_64 (JOBZ, UPLO ,N,A,LDA,W,W ORK,LW ORK,RW ORK,
LRW ORK, $\mathbb{I W}$ ORK,LIW ORK, $\mathbb{N} F O$ )
CHARACTER * 1 JOBZ, UPLO
COM PLEX A (LDA, ${ }^{\star}$ ), W ORK ( $\left.{ }^{( }\right)$
$\mathbb{N} T E G E R * 8 N, L D A, L W$ ORK,LRW ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I N}$ ORK (*)
REALW (*),RWORK (*)

## F95 INTERFACE

SUBROUTINE HEEVD ( $\mathbb{O B} \mathrm{B}, \mathrm{UPLO}, \mathbb{N}], A,[L D A], W,[\mathbb{W}$ ORK ], [LW ORK], [RW ORK ], [LRW ORK], [ $\mathbb{W}$ ORK ], [LIN ORK], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1): : JOBZ,UPLO
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER :: N,LDA,LW ORK,LRW ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{I W}$ ORK
REAL,D $\mathbb{M}$ ENSION (:) ::W ,RW ORK

SU BROUTINE HEEVD _64 ( $\operatorname{DOBZ}, \mathrm{UPLO}, \mathbb{N}], A,[L D A], W,[\mathbb{W}$ ORK $],[L W$ ORK $]$, $[R W$ ORK $],[L R W$ ORK $],[\mathbb{W}$ ORK $],[L \mathbb{N}$ ORK $],[\mathbb{N F O}])$

CHARACTER (LEN=1): : JOBZ, UPLO
COMPLEX,D $\mathbb{I M} E N S I O N(:):$ W ORK
COM PLEX , D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R(8):: N, L D A, L W O R K, L R W O R K, L \mathbb{W} O R K, \mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}$ (8), D $\mathbb{M}$ ENSION (:) :: $\mathbb{I W}$ ORK
REAL,D $\mathbb{I}$ ENSION (:) ::W ,RW ORK

## C INTERFACE

\#include <sunperfh>
void cheevd (char j̀bz, charuplo, int n, com plex *a, int lda, float *w , int *info);
void cheevd_64 (char jobz, charuplo, long n, com plex *a, long lda, float *w , long *info);

## PURPOSE

cheevd com putes alleigenvalues and, optionally, eigenvectors of a com plex H erm itian m atrix A. If eigenvectors are desired, ituses a divide and conquer algorithm .

The divide and conquer algorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on m achines w ith a guard digit in add/subtract, or on those binary $m$ achines $w$ thout guard digits $w$ hich subtract like the $C$ ray $\mathrm{X}-\mathrm{M} P, \mathrm{C}$ ray $\mathrm{Y}-\mathrm{M} \mathrm{P}, \mathrm{C}$ ray $\mathrm{C}-90$, or C ray -2 . It could conceivably fail on hexadecim al or decim al $m$ achines $w$ ithout guard digits, butw e know of none.

## ARGUMENTS

```
JO B Z (input)
    = N':}\mathrm{ C om pute eigenvahues only;
    = V ': C om pute eigenvalues and eigenvectors.
UPLO (input)
    = U ': U pper triangle ofA is stored;
    = L': Low ertriangle ofA is stored.
N (input) The order of them atrix A. N >= 0.
```

A (input/output)
On entry, the $H$ em itian m atrix A. If P PLO = U', the leading N -by -N uppertriangularpartofA contains the uppertriangularpart of the $m$ atrix $A$. If U PLO = L', the leading $N$-by N low er triangular part ofA contains the low er triangular part of the $m$ atrix $A$. On exit, if $J O B Z=V$ ', then if $\mathbb{N} F O=0, A$ contains the orthonorm al eigenvectors of the $m$ atrix $A$. If $J O B Z=N$ ', then on exit the low er triangle (if $\mathrm{PLLO}=\mathrm{L}$ ) or the upper triangle (if $U P L O=U$ ) of $A$, including the diagonal, is destroyed.

LDA (input)
The leading dim ension of the array A. LD A >= $\max (1, \mathbb{N})$.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.

W ORK (w orkspace)
On exit, if $\mathbb{N F O}=0, \mathrm{~W}$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The length of the array $\mathrm{W} O R K$. If $\mathrm{N}<=1$, LW ORK mustibe at least 1. If $\mathrm{JOBZ}=\mathrm{N}$ 'and $\mathrm{N}>$ 1, LW ORK m ustbe at least $\mathrm{N}+1$. If $\mathrm{JOBZ}=\mathrm{V}^{\prime}$ and $\mathrm{N}>1$, LW ORK m ustbe at least2*N + N **2.

If LW O RK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of theW ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

RW ORK (w orkspace)
dim ension (LRW ORK) On ex止, if $\mathbb{N F F O}=0$, RW ORK (1) retums the optim alLRW ORK .

LRW ORK (input)
The dim ension of the aray RW ORK. If $\mathrm{N}<=1$, LRW ORK m ustbe at least1. If $\mathrm{JOBZ}=\mathrm{N}$ 'and $\mathrm{N}>$ 1,LRW ORK m ust.be at leastN. If JOBZ $=V^{\prime}$ and $\mathrm{N}>1$,LRW ORK m ust.be at least1 + 5*N + 2*N **2.

If LRW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim al size of the RW ORK array, retums this value as the first entry of the RW ORK aray, and no enrorm essage
related to LRW ORK is issued by XERBLA.
IN ORK (w orkspace/output)
On exit, if $\mathbb{N} F O=0, \mathbb{I}$ O RK (1) retums the optim al LIN ORK.

LIV ORK (input)
The dim ension of the array $\mathbb{I N}$ ORK. If $\mathrm{N}<=1$, LIN ORK must.be at least1. If $J 0 \mathrm{BZ}=\mathrm{N}$ 'and $\mathrm{N}>$ $1, \mathrm{LIV}$ ORK m ustbe at least 1. If $\mathrm{JOBZ}=\mathrm{V}^{\prime}$ and $\mathrm{N}>1, \mathrm{~L} \mathbb{I} \mathrm{O}$ ORK mustbe at least $3+5 * \mathrm{~N}$.

If LIV ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the $\mathbb{I V}$ ORK array, retums this value as the first entry of the $\mathbb{I W}$ ORK array, and no errorm essage related to $L \mathbb{I N} O R K$ is issued by XERBLA.
$\mathbb{N}$ FO (output)
= 0 : successfulexit
$<0$ : if $\mathbb{N} F O=-i$, the $i$-th argum enthad an illegalvalue
$>0$ : if $\mathbb{N F O}=$ i, the algonithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

## FURTHER DETAILS

B ased on contributions by JeffR utter, C om puter Science D ívision, U niversity of C alifomia at B erkeley, U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cheevr - com pute selected eigenvalues and, optionally, eigenvectors of a com plex H erm itian tridiagonalm atrix T

## SYNOPSIS

```
SU BROUT\mathbb{NE CHEEVR(JOBZ,RANGE,UPLO,N,A,LDA,VL,VU,I|,\mathbb{U,}}\mathbf{N},
    ABSTOL,M,W ,Z,LD Z,ISUPPZ,W ORK,LW ORK,RW ORK,LRW ORK,IW ORK,
    LINORK,\mathbb{NFO)}
CHARACTER * 1 OBZ,RANGE,UPLO
COM PLEX A (LDA,*),Z (LD Z,*),W ORK (*)
```



```
INTEGER ISUPPZ (*), \mathbb{N ORK (*)}
REALVL,VU,ABSTOL
REALW (*),RW ORK (*)
SUBROUTINE CHEEVR_64 (JOBZ,RANGE,UPLO,N,A,LDA,VL,VU,IL,IU,
    ABSTOL,M,W ,Z,LD Z,ISUPPZ,W ORK,LW ORK,RW ORK,LRW ORK,IN ORK,
    L\mathbb{IN ORK, INFO)}
```

CHARACTER * 1 JOBZ,RANGE, UPLO
COM PLEX A (LDA,*), Z (LD Z, *),W ORK (*)
$\mathbb{N} T E G E R * 8 N, L D A, \mathbb{H}, \mathbb{U}, M, L D Z, L W O R K, L R W O R K, L \mathbb{I N} O R K$,
$\mathbb{N} F O$
$\mathbb{N}$ TEGER *8 $\operatorname{ISUPPZ}$ (*), $\mathbb{I N}$ ORK (*)
REALVL,VU,ABSTOL
REALW (*) ,RWORK (*)

## F95 INTERFACE

SU BROUTINE HEEVR (JOBZ,RANGE, UPLO, $\mathbb{N}], A,[L D A], V L, V U, \mathbb{L}, \mathbb{I}$, ABSTOL,M,W,Z,[LD Z], ISU PPZ, [W ORK], [LW ORK ], RW ORK ], [LRW ORK ],
[ $\mathbb{I N}$ ORK], [LIN ORK ], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COMPLEX,D $\mathbb{M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : : A, Z
$\mathbb{N} T E G E R:: N, L D A, \mathbb{L}, \mathbb{U}, M, L D Z, L W$ ORK, LRW ORK, LIW ORK,
$\mathbb{N F O}$
$\mathbb{N}$ TEGER,D $\mathbb{M}$ ENSION (:) :: ISU PPZ, $\mathbb{I N}$ ORK
REAL ::VL,VU,ABSTOL
REAL,D IM ENSION (:) ::W ,RW ORK

SU BROUTINE HEEVR_64 (JOBZ,RANGE,UPLO, $\mathbb{N}], A,[L D A], V L, V U, \mathbb{I}, \mathbb{U}$, ABSTOL,M,W,Z,[LDZ], ISUPPZ, [W ORK], [LW ORK], [RW ORK], [LRW ORK], [ $\mathbb{I N}$ ORK], [LIN ORK], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : : A, Z
$\mathbb{N}$ TEGER (8) ::N,LDA, $\mathbb{I}, \mathbb{I}, \mathrm{M}, \mathrm{LD} Z, L W$ ORK,LRW ORK,LIW ORK,
$\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{M}$ ENSION (:) :: $\mathbb{I S U P P Z , I N O R K}$
REAL ::VL,VU,ABSTOL
REAL,D IM ENSION (:) ::W ,RW ORK

## C INTERFACE

\#include <sunperfh>
void cheevr(char jobz, char range, charuple, intn, com plex
*a, int lda, float vl, float vu, int il, intiu,
floatabstol, int ${ }^{\mathrm{m}} \mathrm{m}$, float ${ }_{\mathrm{w}}$, com plex ${ }^{\text {z }}$, int ldz, int *isuppz, int*info);
void cheevr_64 (char jobz, char range, char uplo, long n, com plex *a, long lda, floatvl, float vu, long il, long iu, float abstol, long *m, float *w, com plex *z, long ldz, long *isuppz, long *info);

## PURPOSE

cheevr com putes selected eigenvalues and, optionally , eigenvectors of a com plex H erm itian tridiagonalm atrix T. Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

W heneverpossible, CHEEVR calls C STEGR to com pute the eigenspectrum using Relatively Robust Representations. CSTEGR com putes eigenvalues by the dqds algorithm, while orthogonaleigenvectors are com puted from various "good" L D
$L^{\wedge} \mathrm{T}$ representations (also known as Relatively Robust Representations). G ram -Schm idtorthogonalization is avoided as far as possible. M ore specifically, the various steps of the algorithm are as follow s.For the i-th unreduced block of T ,
(a) C om pute $T$-sigm a_i= L_iD _iL_i^T, such that L_i D_iL_i^T is a relatively robust representation,
(b) C om pute the eigenvalues, lam bda_j, of L_i D _i L_i^T to high relative accuracy by the dqds algorithm,
(c) If there is a cluster of close eigenvalues, "choose" sigm a_i close to the cluster, and go to step (a),
(d) G iven the approxim ate eigenvalue lam bda_jofL_i D _i L_i^T, com pute the comesponding eigenvectorby form ing a rank-revealing tw isted factorization.
The desired accuracy of the output can be specified by the inputparam eterA BSTOL.

Form ore details, see "A new O ( $n^{\wedge} 2$ ) algorithm for the sym m etric tridiagonal eigenvahue/eigenvector problem ", by Inder光D hillon, C om puter Science D ivision TechnicalR eport N o.U CB //C SD -97-971, U C B erkeley, M ay 1997.

N ote 1 : CHEEVR calls CSTEGR when the fill spectrum is requested on $m$ achines $w$ hich conform to the ieee-754 floating pointstandard. CHEEVR calls SSTEBZ and CSTEIN on non-ieee $m$ achines and
w hen partialspectrum requests are $m$ ade.

N orm alexecution of CSTEGR m ay create NaNs and infinities and hence $m$ ay abort due to a floating pointexception in environm ents w hich do nothandle N aN s and infinities in the ieee standard defaultm anner.

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}$ ': C om pute eigenvalues only;
$=\mathrm{V}$ : C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvalues w illbe found.
$=\mathrm{V}$ ': alleigenvalues in the half-open interval
(VL, V U ] w ill be found. = I': the II th through
IU th eigenvalues w illbe found.

UPLO (input)
= U : U ppertriangle ofA is stored;
= L': Low er triangle of A is stored.

N (input) The orderof the m atrix A. $\mathrm{N}>=0$.

A (input/output)
On entry, the $H$ erm itian $m$ atrix $A$. If $\mathrm{PLO}=\mathrm{U}$ ', the leading N -by -N uppertriangularpartofA contains the upper triangularpart of the $m$ atrix $A$. If UPLO = L', the leading N -by-N low er triangular part ofA contains the low er triangular part of the $m$ atrix $A$. On exit, the low ertriangle (if $\mathrm{UPLO}=\mathrm{L}$ ) or the upper triangle (if UPLO = U ) of A , including the diagonal, is destroyed.
LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, \mathbb{N})$.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE=A'or I'.

VU (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRA N G E = A 'or I''.

II (input)
IfRA NGE= I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{L}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N otreferenced ifRANGE= A 'or V'.
$\mathbb{I U}$ (input)
IfRA N G E= I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{L}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N otreferenced ifRANGE= A 'or V'.

ABSTOL (input)
The absolute error tolerance forthe eigenvalues. A $n$ approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b] of w idth less than orequal to

ABSTOL + EPS * $\max (|k|, \mid)$,
where EPS is the m achine precision. IfA BSTOL is less than or equal to zero, then EPS* $\mid$ | w illbe used in its place, w here $F \mid$ is the 1 -norm of the tridiagonal $m$ atrix obtained by reducing $A$ to tridiagonal form .

See "C om puting Sm allSingularV ahues of B idiagonal $M$ atrices $w$ ith $G$ uaranteed $H$ igh Relative A ccuracy," by D em meland K ahan, LA PA CK W orking N ote \#3.

If high relative accuracy is im portant, setA BSTO L to SLAMCH (Safe minimum '). Doing so will guarantee thateigenvalues are com puted to high relative accuracy when possible in future releases. The current code does not $m$ ake any guarantees abouthigh relative accuracy, but funutre releasesw ill. See J.Barlow and J. Demmel, "C om puting A ccurate E igensystem sofScaled D iagonally D om inantM atrioes", LA PA CK W orking N ote \#7, for a discussion of $w$ hich $m$ atrices define their eigenvalues to high relative accuracy.

M (output)
The totalnum berofeigenvalues found. $0<=\mathrm{M}$ <= N . IfRANGE $=A^{\prime}, \mathrm{M}=\mathrm{N}$, and ifRANGE $=\mathrm{I}^{\prime}, \mathrm{M}=$ $\mathbb{U}-\mathbb{L}+1$.

W (output)
The firstM elem ents contain the selected eigenvalues in ascending order.

Z (output)
If $\mathrm{OOBZ}=\mathrm{V}$ ', then if $\mathbb{N F O}=0$, the first M colum ns of $Z$ contain the orthonorm aleigenvectors of them atrix A comesponding to the selected eigenvalues, $w$ ith the $i$-th colum $n$ of $Z$ holding the eigenvector associated w ith W (i). If $\mathrm{JOBZ}=\mathrm{N}$ ', then $Z$ is not referenced. N ote: the userm ust ensure that at leastm ax $(1, M)$ colum ns are supplied in the array $Z$; ifRANGE = V', the exact value of $M$ is not know $n$ in advance and an upperbound $m$ ust be used.

LD $Z$ (input)
The leading dim ension of the array $Z . L D Z>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD $Z>=\max (1, N)$.

ISUPPZ (output)

The support of the eigenvectors in $Z$, i.e., the indices indicating the nonzero elem ents in $Z$. The i-th eigenvector is nonzero only in elem ents $\operatorname{ISUPPZ}(2 \star i-1)$ through $\operatorname{ISU} \operatorname{PPZ}$ (2*i).

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.
LW ORK (input)
The length of the aray $W$ ORK. LW ORK >= $\max (1,2 \star \mathrm{~N})$. For optim al efficiency, LW ORK $>=$ $(N B+1) * N$, where $N B$ is the $m$ ax of the blocksize for CHETRD and forCUNM TR as retumed by HAENV.

IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

RW ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, R W$ ORK (1) retums the optim al (andm inim al) LRW ORK.

LRW ORK (input)
The length of the array RW ORK. LRW ORK >= $\max (1,24 \star \mathrm{~N})$.

If LRW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the RW ORK array, retums this value as the first entry of the RW ORK anay, and no enorm essage related to LRW ORK is issued by XERBLA.

IW ORK (w orkspace/output)
On exit, if $\mathbb{N} F O=0, \mathbb{I N}$ ORK (1) retums the optim al (and $m$ inim al) $L \mathbb{I N} O R K$.

LIN ORK (input)
The dimension of the array $\mathbb{I W}$ ORK. LIW ORK >= $\max (1,10 \star \mathrm{~N})$.

If LIV ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the $\mathbb{I W}$ ORK array, retums this value as the first entry of the $\mathbb{I W}$ ORK anay, and no enrorm essage related to $L \mathbb{I N} O R K$ is issued by X ERBLA.
$\mathbb{I N} F O$ (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N N F O}=-i$, the $i$-th argum ent had an illegalvahue
> 0: Intemalemor

## FURTHER DETAILS

$B$ ased on contributions by
Inder所D hillon, $\mathbb{B M}$ A m aden, U SA
O sniM arques, LBN L NERSC, U SA
K en Stanley, C om puterScience D ivision, U niversity of C alifomia atB erkeley, U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cheevx - com pute selected eigenvalues and, optionally, eigenvectors of a com plex Herm itian m atrix A

## SYNOPSIS

```
SUBROUT\mathbb{NE CHEEVX (JOBZ,RANGE,UPLO,N,A,LDA,VL,VU,\mathbb{I},\mathbb{U},};=,
    ABTOL,NFOUND,W ,Z,LDZ,W ORK,LDW ORK,W ORK2,IN ORK 3, \mathbb{FA IL,}
    \mathbb{NFO )}
CHARACTER * 1 JOBZ,RANGE,UPLO
COM PLEX A (LDA,*),Z (LD Z,*),W ORK (*)
\mathbb{N}TEGERN,LDA,\mathbb{I},\mathbb{U},NFOUND,LDZ,LDWORK,\mathbb{NFO}
\mathbb{NTEGER IN ORK3(*),\mathbb{FA LI (*)}}\mathbf{(})
REALVL,VU,ABTOL
REALW (*),W ORK2 (*)
```



```
    ABTOL,NFOUND,W ,Z,LDZ,W ORK,LDW ORK,W ORK2, IN ORK 3, \mathbb{FAIL,}
    \mathbb{NFO)}
```

CHARACTER * 1 JOBZ,RANGE,UPLO
COM PLEX A (LDA,*), Z (LD Z, *),W ORK (*)
$\mathbb{I N} T E G E R * 8 N, L D A, \mathbb{I}, \mathbb{U}, N F O U N D, L D Z, L D W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I N}$ ORK 3 (*), $\mathbb{F A} \mathbb{H}$ (*)
REALVL,VU,ABTOL
REALW (*),WORK2 (*)

## F95 INTERFACE

SU BROUTINE HEEVX (JOBZ,RANGE,UPLO, $\mathbb{N}], A,[L D A], V L, V U, \mathbb{I}, \mathbb{I}$, ABTOL, $\mathbb{N} F O U N D], W, Z,[L D Z],[W O R K],[L D W$ ORK ], [W ORK 2], [W ORK 3], $\mathbb{F A} \mathbb{L},[\mathbb{N F O}])$

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COM PLEX ,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\operatorname{IM}$ ENSION (:,:) ::A,Z
$\mathbb{N} T E G E R:: N, L D A, \mathbb{L}, \mathbb{U}, N F O U N D, L D Z, L D W O R K, \mathbb{N F O}$
$\mathbb{N}$ TEGER,D $\mathbb{I M} E N S I O N(:):: \mathbb{I N} O R K 3, \mathbb{F} A \mathbb{L}$
REAL ::VL,VU,ABTOL
REAL,D $\mathbb{I M}$ ENSION (:) ::W ,W ORK2

SU BROUTINE HEEVX_64 (OBZ,RANGE,UPLO, $\mathbb{N}], A,[L D A], V L, V U, \mathbb{I}, \mathbb{U}$, ABTOL, $\mathbb{N} F O U N D], W, Z,[L D Z],[W O R K],[L D W O R K],[W O R K 2],[\mathbb{W} O R K 3]$, $\mathbb{F A} \mathbb{I},[\mathbb{N F O}])$

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COMPLEX,D $\mathbb{M} E N S I O N(:):$ W ORK
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : A , Z
$\mathbb{N}$ TEGER (8) :: N , LDA $, \mathbb{I}, \mathbb{Z}, N F O U N D, L D Z, L D W O R K, \mathbb{N F O}$
$\mathbb{N}$ TEGER (8), D $\mathbb{I M}$ ENSION (:) :: IN ORK 3, $\mathbb{F} A \mathbb{I}$
REAL ::VL,VU,ABTOL
REAL,D $\mathbb{I M} E N S I O N(:):: W$,W ORK 2

## C INTERFACE

\#include <sunperfh>
void cheevx (char j́bib, char range, charuplo, intn, com plex
*a, int lda, floatvl, floatvu, intil, intiu, floatabtol, int *nfound, float * ${ }_{\mathrm{w}}$, com plex *z, int $1 d z$, int *ifail, int*info);
void cheevx_64 (char jobz, charrange, char uplo, long n, com plex *a, long lda, floatvl, floatvu, long il, long iu, floatabtol, long *nfound, float *w , com plex *z, long ldz, long *ifail, long *info);

## PURPOSE

cheevx com putes selected eigenvalues and, optionally, eigenvectors of a com plex $H$ erm itian $m$ atrix A. E igenvalues and eigenvectors can be selected by specifying either a range of values or range of indices for the desired eigenvalues.

## ARGUMENTS

JO B Z (input)
$=\mathrm{N}:$ : Com pute eigenvahes only;
$=\mathrm{V}$ ': C om pute eigenvalues and eigenvectors.
= A ': alleigenvalues $w$ illibe found.
= V : alleigenvalues in the half-open interval
(NL, VU ] will be found. = I': the $\mathbb{I}$-th through
$\mathbb{I U}$-th eigenvaluesw illbe found.

UPLO (input)
= U ': U ppertriangle ofA is stored;
= LL': Low ertriangle ofA is stored.
N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
On entry, the $H$ erm itian $m$ atrix A. If $\mathrm{PLO}=\mathrm{U}$ ', the leading N -by N upper triangularpart of A contains the upper triangularpart of the $m$ atrix A. If U PLO = L', the leading N -by-N low er triangular part of A contains the low er triangular part of the $m$ atrix $A$. On exit, the low ertriangle (if $\mathrm{UPLO}=\mathrm{L}$ ) or the uppertriangle (if $\mathrm{UPLO}=\mathrm{U}$ ) of A , including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, N)$.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRA N G E = A 'or I''.

VU (input)
IfRANGE=V ', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE = A 'or I''.

II (input)
If RA NGE= 'I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{L}<=\mathbb{U}<=N$, if $N>0$; $\mathbb{L}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N ot referenced ifRANGE= A 'or V'.

IU (input)
IfRA N G E= 'I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{H}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N otreferenced ifRANGE= A 'or V'.

ABTOL (input)

The absolute error tolerance for the eigenvalues. A $n$ approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b] of w idth less than orequal to

ABTOL + EPS * max (k|,b|),
where EPS is them achine precision. If ABTOL is less than or equalto zero, then EPS* $\mid$ | w illibe used in its place, w here $F \mid$ is the 1 -norm of the tridiagonal $m$ atrix obtained by reducing $A$ to tridiagonal form .

E igenvalues w illlbe com puted m ostaccurately when ABTOL is set to tw ice the underflow threshold 2*SLAM CH (S ), not zero. If this routine retums w ith $\mathbb{N}$ FO $>0$, indicating that som e eigenvectors did not converge, try setting ABTOL to $2 *$ SLAM CH (S ).

See "C om puting Sm allSingularV alues of B idiagonal M atrices w ith G uaranteed H igh Relative A ccuracy," by Dem m eland K ahan, LA PA CK W orking N ote \#3.

NFOUND (output)
The total num ber of eigenvalues found. $0<=$ NFOUND <= N. IfRANGE = A',NFOUND = N, and if RANGE $=I^{\prime}$ ', NFOUND $=\mathbb{U}-\mathbb{H}+1$.

W (output)
On norm alexit, the firstN FOUND elem ents contain the selected eigenvalues in ascending order.
$Z$ (input) If $J O B Z=V '$, then if $\mathbb{N} F O=0$, the first $N F O U N D$ colum ns of $Z$ contain the orthonorm aleigenvectors of thematrix A comesponding to the selected eigenvalues, $w$ ith the i-th colum $n$ of $Z$ holding the eigenvector associated w ith $W$ (i). If an eigenvector fails to converge, then that colum n of $Z$ contains the latest approxim ation to the eigenvector, and the index of the eigenvector is retumed in $\mathbb{F A} \mathbb{I}$. If $J O B Z=N$ ', then $Z$ is not referenced.
$N$ ote: the user must ensure that at least
$m$ ax ( $1, N$ FOUND ) colum ns are supplied in the array $Z$; if RANGE = V', the exact value ofNFOUND is not know $n$ in advance and an upperbound $m$ ustbe used.

LD Z (input)
The leading dim ension of the array $Z$. LD $Z>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD $Z>=\mathrm{max}(1, N)$.

W ORK (w orkspace)
On exit, if $\mathbb{N F O}=0, W$ ORK (1) retums the optim al LDW ORK.

LD W ORK (input)
The length of the array W ORK. LDW ORK $>=$ $\max (1,2 * N)$. For optim al efficiency, LDW ORK >= $(\mathbb{N B}+1) \star \mathrm{N}$, w here $N B$ is the $m$ ax of the blocksize for CHETRD and forCUNM TR as retumed by $\amalg A E N V$.

IfLDW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by X ERBLA.

W ORK 2 (w orkspace)
dim ension ( $7 \star \mathrm{~N}$ )

IV ORK 3 (w orkspace)
dim ension ( $5 * \mathrm{~N}$ )
$\mathbb{F A} I I$ (output)
If $J O B=V^{\prime}$ ', then if $\mathbb{N F O}=0$, the first $N F O U N D$ elem ents of $\mathbb{F A} I I$ are zero. If $\mathbb{N F O}>0$, then $\mathbb{F} A$ II contains the indices of the eigenvectors that failed to converge. If $\mathrm{OBZ}=\mathrm{N}$ ', then $\mathbb{F A} \mathbb{I}$ is notreferenced.
$\mathbb{N F O}$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum ent had an illegalvalue
> 0: if $\mathbb{N N F O}=i$, then ieigenvectors failed to converge. Their indioes are stored in array正AI.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chegs2 - reduce a com plex Herm itian-definite generalized eigenproblem to standard form

## SYNOPSIS

```
SU BROUT\mathbb{NE CHEGS2 (TTYPE,UPLO,N,A,LDA,B,LDB, NNFO )}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LDB,*)
INTEGER ITYPE,N,LDA,LDB,INFO
SUBROUT\mathbb{NE CHEGS2_64(ITYPE,UPLO,N,A,LDA,B,LDB,INFO )}
CHARACTER * 1 UPLO
COM PLEXA (LDA,*),B (LDB,*)
INTEGER*8 TTYPE,N,LDA,LDB,INFO
F95 INTERFACE
```



```
    CHARACTER (LEN=1)::UPLO
    COM PLEX,D IM ENSION (:,:)::A,B
    \mathbb{NTEGER ::TTYPE,N,LDA,LDB,INFO}
    SUBROUT\mathbb{NE HEGS2_64 (TTYPE,UPLO,N ,A ,[LDA ],B, [LD B ], [NFO ])}
    CHARACTER (LEN=1) ::UPLO
    COM PLEX,D IM ENSION (:,:)::A,B
    INTEGER (8):: ITYPE,N,LDA,LDB,\mathbb{NFO}
C INTERFACE
    #include <sunperfh>
```

void chegs2 (int itype, charuplo, int n, com plex *a, int lda, com plex *b, int ldb, int *info);
void chegs2_64 (long itype, charuple, long n, com plex *a, long lda, com plex *b, long ldb, long *info);

## PURPOSE

chegs2 reduces a complex Herm tian-definite generalized eigenproblem to standard form .

If ITYPE $=1$, the problem is $A * x=\operatorname{lam}$ bda ${ }^{*}{ }^{*} \mathrm{x}$, and $A$ is overw ritten by inv ( $U 1$ ) *A *inv (U) orinv (L) *A *inv (L) If ITYPE $=2$ or 3 , the problem is $A * B * x=\operatorname{lam}$ bda* $x$ or $B * A * X=\operatorname{lam}$ bda ${ }^{\mathrm{X}} \mathrm{X}$, and A is overw ritten by $\mathrm{U} * \mathrm{~A} * \mathrm{U}$ `orL ${ }^{*} \mathrm{~A} * \mathrm{~L}$. B m usthave been previously factorized as U *U or L *L' by CPOTRF.

## ARGUMENTS

ITYPE (input)

$=2$ or 3: com pute $U * A$ *U 'orL *A *L.

UPLO (input)
Specifies w hether the upper or low er triangular
part of the $H$ erm itian $m$ atrix A is stored, and how
B has been factorized. = U ': U pper triangular
= L': Low ertriangular
N (input) The order of the m atrices A and $\mathrm{B} . \mathrm{N}>=0$.

A (input/output)
O n entry, the $H$ erm tian matrix A. If U PLO $=\mathrm{U}$ ', the leading $n$ by $n$ upper triangularpart of $A$ contains the upper triangular part of the $m$ atrix $A$, and the strictly low ertriangularpartofA is not referenced. If UPLO $=\mathrm{L}$ ', the leading n by n low er triangularpart ofA contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of $A$ is not referenced.

Onexit, if $\mathbb{N} F O=0$, the transform ed $m$ atrix, stored in the sam e form atas A.

## LD A (input)

The leading dim ension of the array A. LDA >= $\max (1, N)$.
$B$ (input) The triangular factor from the C holesky factorization ofB, as retumed by CPO TRF .

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.
$\mathbb{I N} F O$ (output)
= 0: successfulexit.
< 0: if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an illegalvahue.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chegst-reduce a com plex Herm itian-definite generalized eigenproblem to standard form

## SYNOPSIS

```
SUBROUT\mathbb{NE CHEGST (TTYPE,UPLO,N,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LDB,*)
INTEGER TTYPE,N,LDA,LDB,INFO
SUBROUT\mathbb{NE CHEGST_64(ITYPE,UPLO,N,A ,LDA,B,LDB, INFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LDB,*)
INTEGER*8 ITYPE,N,LDA,LDB,INFO
F95 INTERFACE
    SUBROUT\mathbb{NE HEGST (TTYPE,UPLO,N,A , [LDA ],B, [LDB],[INFO])}
    CHARACTER (LEN=1)::UPLO
    COM PLEX,D IM ENSION (:,:)::A,B
```



```
    SUBROUT\mathbb{NE HEGST_64 (TTYPE,UPLO,N,A,[LDA ],B,[LDB],[NFO])}
    CHARACTER (LEN=1) ::UPLO
    COM PLEX,D IM ENSION (:,:) ::A,B
    INTEGER (8):: ITYPE,N,LDA,LDB,\mathbb{NFO}
```

C INTERFACE
\#include <sunperfh>
void chegst(int itype, charuplo, int $n$, com plex *a, int lda, com plex *b, int ldb, int *info);
void chegst 64 (long itype, charuplo, long n, com plex *a, long lda, com plex *b, long ldb, long *info);

## PURPOSE

chegst reduces a com plex H erm itian-definite generalized eigenproblem to standard form .

If ITY $\mathrm{PE}=1$, the problem is $\mathrm{A}{ }_{\mathrm{x}}=$ lam bda*B ${ }_{\mathrm{X}} \mathrm{X}$, and $A$ is overw ritten by $\operatorname{inv}(U * * H) * A * \operatorname{inv}(U)$ or $\operatorname{inv}(\amalg) * A * \operatorname{inv}(\amalg * * H)$
If ITYPE $=2$ or 3 , the problem is $A * B *_{x}=$ lam bda* x or $B * A * x=\operatorname{lam} . b d a * x$, and $A$ is overw rilten by $U * A * U * * H$ or L** $\mathrm{H} * \mathrm{~A}$ *L.

B m usthave been previously factorized as U ** $\mathrm{H} * \mathrm{U}$ or $\mathrm{L} * \mathrm{~L}$ ** H by CPOTRF .

## ARGUMENTS

ITYPE (input)
$=1:$ compute $\quad \operatorname{inv}(\mathrm{U} * * H) * A * \operatorname{inv}(\mathrm{U}) \quad$ or
$\operatorname{inv}(\amalg) \star A$ *inv $(\amalg * * H)$;
$=2$ or 3 :com pute $\mathrm{U} * \mathrm{~A} * \mathrm{U} * * \mathrm{H}$ orL** $\mathrm{H} * \mathrm{~A} * \mathrm{~L}$.

UPLO (input)
$=U^{\prime}$ : Uppertriangle of $A$ is stored and $B$ is factored as $\mathrm{U} * * \mathrm{H} * \mathrm{U} ;=\mathrm{L}:$ : Low er triangle of is stored and B is factored as $\mathrm{L} * \mathrm{~L}^{\star *} \mathrm{H}$.

N (input) The order of the $m$ atriges $A$ and $B . N>=0$.

A (input/output)
On entry, the $H$ erm itian $m$ atrix A. If $\mathrm{UPLO}=\mathrm{U}$ ', the leading N -by N uppertriangularpartofA contains the upper triangular part of the $m$ atrix $A$, and the strictly low er triangularpart of A is not referenced. IfUPLO = L ', the leading N -by -N low er triangularpart ofA contains the low ertriangularpart of them atrix A, and the strictly upper triangularpart of A is not referenced.

On exit, if $\mathbb{N F O}=0$, the transform ed matrix,
stored in the sam e form at as A .

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, N)$.

B (input) The triangular factor from the Cholesky factorization ofB , as retumed by CPO TRF .

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the $i-$ th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chegv - com pute all the eigenvalues, and optionally, the eigenvectors of a com plex generalized $H$ erm itian-definite eigenproblem, of the form $A * x=(\operatorname{lam} . b d a) * B * x, A * B x=(\operatorname{lam} . b d a) * x$, or $\mathrm{B}^{*} \mathrm{~A}$ * $\mathrm{x}=\left(\operatorname{lam}\right.$ bda) ${ }^{*} \mathrm{x}$

## SYNOPSIS

```
SU BROUT\mathbb{NE CHEGV (TTYPE,JOBZ,UPLO,N,A,LDA,B,LDB,W ,W ORK,}
    LDW ORK,W ORK2,INFO)
CHARACTER * 1 JOBZ,UPLO
COM PLEX A (LDA,*),B (LD B ,*),W ORK (*)
\mathbb{NTEGER ITYPE,N,LDA,LDB,LDW ORK,\mathbb{NFO}}\mathbf{N}=\mp@code{L}
REALW (*),W ORK 2 (*)
SUBROUT\mathbb{NE CHEGV_64 (TTYPE,JOBZ,UPLO ,N,A,LDA,B,LDB,W ,W ORK,}
    LDW ORK,W ORK2,INFO)
CHARACTER * 1 JOBZ,UPLO
COM PLEX A (LDA,*),B (LD B,*),W ORK (*)
\mathbb{NTEGER*8 ITYPE,N,LDA,LDB,LDW ORK,\mathbb{NFO}}\mathbf{N}=\mp@code{L}
REALW (*),W ORK2 (*)
```


## F95 INTERFACE

SU BROUTINE HEGV (TTYPE, $\operatorname{JoB} \mathrm{Z}, \mathrm{U} P L O, N, A,[L D A], B,[L D B], W, \mathbb{W} O R K]$, [LDW ORK], [WORK2], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1)::JOBZ, UPLO
COMPLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A,B
$\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D A, L D B, L D W O R K, \mathbb{N F O}$
REAL,D $\mathbb{I M}$ ENSION (:) ::W ,W ORK2

SU BROUTINE HEGV_64 (TTYPE, $\operatorname{OOBZ}, \mathrm{UPLO}, \mathrm{N}, \mathrm{A},[\mathrm{LDA}], \mathrm{B},[\mathrm{LDB}], \mathrm{W},[\mathbb{W} O R K]$, [LDW ORK], [W ORK2], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1):: JOBZ, UPLO
COMPLEX,D $\mathbb{M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : :: A, B
$\mathbb{N}$ TEGER (8) :: $\mathbb{T} Y$ PE, $N$,LDA,LDB,LDW ORK, $\mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::W ,W ORK2

## C INTERFACE

\#include < sunperfh>
void chegv (int itype, char jobz, charuplo, int $n$, com plex
*a, int lda, com plex *b, int ldb, float *w, int
*info);
void chegv_64 (long itype, char jobz, charuplo, long n, com plex *a, long lda, com plex *b, long ldb, float *w , long *info);

## PURPOSE

chegv com putes all the eigenvalues, and optionally, the eigenvectors of a com plex generalized H erm itian-definite eigenproblem, of the form $A * x=(l a m ~ b d a) * B * x$, $\mathrm{A} * \mathrm{~B} \mathrm{x}=(\operatorname{lam} . \mathrm{bda}){ }^{\star} \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{X}=\left(\mathrm{lam}\right.$ bda) ${ }^{*} \mathrm{x}$. H ere A and B are assum ed to be $H$ erm itian and $B$ is also positive definite.

## ARGUMENTS

ITYPE (input)
Specifies the problem type to be solved:
$=1: A{ }^{*} \mathrm{x}=\left(\mathrm{lam}\right.$ bda) ${ }^{\mathrm{B}}{ }^{*}{ }_{\mathrm{x}}$
$=2: \mathrm{A} * \mathrm{~B} * \mathrm{X}=\left(\mathrm{lam}\right.$ bda) ${ }^{*} \mathrm{X}$
$=3: B * A * X=(l a m ~ b d a){ }^{*} \mathrm{x}$

JOBZ (input)
$=N^{\prime}:$ C om pute eigenvalues only;
= V ': C om pute eigenvalues and eigenvectors.
UPLO (input)
$=\mathrm{U}$ ': U ppertriangles of $A$ and $B$ are stored;
$=\mathrm{L}$ ': Low ertriangles of $A$ and $B$ are stored.

N (input) The order of the m atrices A and $\mathrm{B} . \mathrm{N}>=0$.

A (input/output)
On entry, the $H$ erm itian $m$ atrix A. If UPLO = U', the leading N -by N upper triangularpartof A contains the upper triangularpart of the $m$ atrix $A$. If UPLO = L', the leading N -by-N low er triangular partofA contains the low er triangular part of the m atrix A.

On exit, if $J \mathrm{OBZ}=\mathrm{V}^{\prime}$, then if $\mathbb{N} F O=0$, A contains the $m$ atrix $Z$ ofeigenvectors. The eigenvectors are norm alized as follow s: if ITYPE = 1 or $2, Z * * H * B * Z=I ;$ if $I T Y P E=3, Z * * H * i n v(B) * Z=I$. If $\mathrm{JOBZ}=\mathrm{N}$ ', then on exit the upper triangle (if $\mathrm{U} P L O=\mathrm{U}$ ) or the low er triangle (if $\mathrm{U} P L O=\mathrm{L}$ ) of A , including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, N)$.

B (input/output)
On entry, the $H$ erm itian positive definite $m$ atrix $B$. If $U P L O=U$ ', the leading $N$ by $N$ uppertriangularpartofB contains the upper triangular part of them atrix B. IfU PLO = L', the leading N toy -N low er triangularpart of B contains the low er triangularpart of the $m$ atrix B .

On exit, if $\mathbb{N} F O<=N$, the part of $B$ containing the $m$ atrix is overw rilten by the triangular factor U orL from the Cholesky factorization $\mathrm{B}=\mathrm{U} * * \mathrm{H} * \mathrm{U}$ or $B=\mathrm{L} * \mathrm{~L} * * \mathrm{H}$.

LD B (input)
The leading dim ension of the aray B. LD B >= $\max (1, N)$.

W (output)
If $\mathbb{N}$ FO $=0$, the eigenvalues in ascending order.

## W ORK (w orkspace)

On exit, if $\mathbb{N} F O=0, W$ ORK ( 1 ) retums the optim al LDW ORK.

LDW ORK (input)
The length of the array $W$ ORK. LDW ORK >= max $\left(1,2^{\star} \mathrm{N}-1\right)$. For optim alefficiency, LD W ORK >= $(\mathrm{NB}+1) * \mathrm{~N}$, where NB is the blocksize for CHETRD retumed by LLAENV.

IfLD W ORK $=-1$, then aw orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 2 (w orkspace)
dim ension (max (1,3*N-2))
$\mathbb{I N F O}$ (output)
= 0 : successfulexit
< $0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue
> 0: CPOTRF orCH EEV retumed an errorcode:
$<=\mathrm{N}$ : if $\mathbb{N F O}=\mathrm{i}, \mathrm{CHEEV}$ failed to converge; i off-diagonalelem ents of an interm ediate tridiagonal form did notconverge to zero; $>N$ : if $\mathbb{N F O}$
$=N+i$, for $1<=i<=N$, then the leading $m$ inor oforderiofB is not positive definite. The factorization of B could notbe com pleted and no eigenvahues or eigenvectors w ere com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

chegvd - com pute all the eigenvalues, and optionally, the eigenvectors of a com plex generalized H erm itian-definite eigenproblem, of the form $A * x=(l a m . b d a) * B * x, A * B x=(l a m ~ b d a) * x$, or $B * A * X=\left(l a m\right.$ bda) ${ }^{*} X$

## SYNOPSIS

```
SU BROUTINE CHEGVD (ITYPE,NOBZ,UPLO,N,A,LDA,B,LDB,W,W ORK,
    LW ORK,RW ORK,LRW ORK,IN ORK,LIN ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
COM PLEX A (LDA,*),B (LD B,*),W ORK (*)
INTEGER ITYPE,N,LDA,LDB,LW ORK,LRW ORK,LIN ORK,\mathbb{NFO}
INTEGER IN ORK (*)
REALW (*),RW ORK (*)
SU BROUT\mathbb{NE CHEGVD_64 (TTYPE,JOBZ,UPLO,N,A,LDA,B,LDB,W,W ORK,}
        LW ORK,RW ORK,LRW ORK,IN ORK,LIN ORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
COM PLEX A (LDA,*),B (LD B,*),W ORK (*)
INTEGER*8 ITYPE,N,LDA,LDB,LW ORK,LRW ORK,LIN ORK,INFO
INTEGER*8 \mathbb{N ORK (*)}
REALW (*),RW ORK (*)
```


## F95 INTERFACE

SU BROUTINE HEGVD (TTYPE, JOBZ, UPLO, $\mathbb{N}], A,[L D A], B,[L D B], W,[W O R K]$, [LW ORK ], $\mathbb{R W}$ ORK ], [LRW ORK ], [ $\mathbb{W}$ ORK ], [LIN ORK ], [ $\mathbb{N} F O]$ )

COM PLEX, D $\mathbb{M}$ ENSION (: : : : : A , B
$\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D A, L D B, L W O R K, L R W O R K, L \mathbb{I N} O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{I W} O R K$
REAL,D IM ENSION (:) ::W ,RW ORK

SU BROU T $\mathbb{N} E$ HEGVD_64 (ITYPE, $\mathrm{OB} \mathrm{B}, \mathrm{UPLO}, \mathbb{N}], A,[L D A], B,[L D B], W$, $[\mathbb{W} O R K],[L W$ ORK $],[R W$ ORK $],[L R W O R K],[\mathbb{W} O R K],[L \mathbb{W} O R K],[\mathbb{N} F O])$

CHARACTER (LEN=1) :: JOBZ, UPLO
COM PLEX,D $\mathbb{I M} E N S I O N(:):$ W ORK
COM PLEX, D $\mathbb{M}$ ENSION (:,:) ::A,B
$\mathbb{N} \operatorname{TEGER}$ (8) :: $\mathbb{I T Y} \mathrm{PE}, \mathrm{N}, \mathrm{LDA}, \mathrm{LDB}, \mathrm{LW}$ ORK, LRWORK, LIV ORK,
$\mathbb{N}$ FO

REAL,D $\mathbb{I}$ ENSION (:) ::W ,RW ORK

## C INTERFACE

\#include <sunperfh>
void chegvd (int itype, char j̣bz, charuplo, intn, com plex *a, int lda, com plex *b, int ldb, float * w , int *info);
void chegvd_64 (long itype, char jobz, char uplo, long n, com plex *a, long lda, com plex *b, long ldb, float *W, long *info);

## PURPOSE

chegvd com putes all the eigenvalues, and optionally, the eigenvectors of a com plex generalized $H$ erm itian-definite eigenproblem, of the form $A{ }^{*} x=(l a m ~ b d a) * B{ }^{*} x$, $\mathrm{A} * \mathrm{~B} \mathrm{x}=(\operatorname{lam} \mathrm{bda})^{*} \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\operatorname{lam} \mathrm{bda}) * \mathrm{x}$. H ere A and B are assum ed to be $H$ erm itian and $B$ is also positive definite. If eigenvectors are desired, ituses a divide and conquer algorithm .

The divide and conquer algorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on m achines w ith a guard digit in add/subtract, or on those binary $m$ achines $w$ thout guard digits $w$ hich subtract like the $C$ ray $\mathrm{X}-\mathrm{M} P, \mathrm{C}$ ray $\mathrm{Y}-\mathrm{M} \mathrm{P}, \mathrm{C}$ ray $\mathrm{C}-90$, or C ray -2 . It could conceivably fail on hexadecim al or decim al machines $w$ ithout guard digits, butw e know of none.

## ARGUMENTS

ITYPE (input)

Specifies the problem type to be solved:
$=1: A{ }^{*}=\left(\operatorname{lam}\right.$ bda) ${ }^{\mathrm{B}} \mathrm{A}^{*} \mathrm{x}$
$=2: A * B * x=\left(l a m\right.$ bda) ${ }^{*} \mathrm{x}$
$=3: B{ }^{*}{ }^{*}{ }^{x}=\left(\operatorname{lam}\right.$ bda) ${ }^{*} \mathrm{X}$

JOBZ (input)
$=\mathrm{N}$ ': C om pute eigenvalues only;
$=\mathrm{V}$ :: C om pute eigenvalues and eigenvectors.
UPLO (input)
$=\mathrm{U}:$ : U pper triangles of $A$ and $B$ are stored;
= L': Low ertriangles of $A$ and $B$ are stored.
N (input) The order of the m atrices A and $\mathrm{B} . \mathrm{N}>=0$.
A (input/output)
On entry, the $H$ erm itian m atrix A. If UPLO = U', the leading N -by N uppertriangularpart of A contains the upper triangularpart of the $m$ atrix $A$. If UPLO = L', the leading N -by N low er triangular partofA contains the low er triangular part of them atrix $A$.

Onexit, if $J O B Z=V^{\prime}$, then if $\mathbb{N} F O=0, A$ contains the $m$ atrix $Z$ of eigenvectors. The eigenvectors are norm alized as follow s: if ITYPE = 1 or $2, Z * * H * B * Z=I ;$ if $I T Y P E=3, Z * * H * i n v(B) * Z=I$. If $J O B Z=N$ ', then on exit the upper triangle (if $\mathrm{UPLO}=\mathrm{U}$ ) or the low er triangle (if $\mathrm{UPLO}=\mathrm{L}$ ) of A , including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the array A. LD A >= $\max (1, N)$.

B (input/output)
O n entry, the $H$ erm itian m atrix B. If UPLO = U', the leading N -by N uppertriangularpart of B contains the upper triangularpart of the $m$ atrix $B$. If UPLO = L', the leading N by N low er triangular part ofB contains the low er triangular part of them atrix B .

Onexit, if $\mathbb{N} F O<=N$, the part of $B$ containing the $m$ atrix is overw rilten by the triangular factor U orL from the Cholesky factorization $\mathrm{B}=\mathrm{U} * * \mathrm{H} * \mathrm{U}$ or $B=L * L^{* *} \mathrm{H}$.

LD B (input)
The leading dim ension of the aray $B$. LD B >= $\max (1, N)$.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The length of the amay $W$ ORK. If $N<=1$, LW ORK $>=1$. If $O B Z=N$ 'and $N>1, L W O R K>=N$ +1 . If $O B Z=V$ 'andN $>1$, LW ORK $>=2 * N+$ $\mathrm{N} * * 2$ 。

If LW O RK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK aray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

RW ORK (w orkspace)
On exit, if $\mathbb{N} F O=0$, RW ORK (1) retums the optim al LRW ORK.

LRW ORK (input)
The dim ension of the aray RW ORK. If $\mathrm{N}<=1$, LRW ORK $>=1$. If $\mathrm{OBZ}=\mathrm{N}$ 'and $\mathrm{N}>1$,LRW ORK $>=$ N . If $\mathrm{OBB}=\mathrm{V}^{\prime}$ and $\mathrm{N}>1$,LRW ORK $>=1+5 \star \mathrm{~N}+$ $2 * N * * 2$ 。

If LRW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the RW ORK aray, retums this value as the first entry of the RW ORK array, and no errorm essage related to LRW ORK is issued by X ERBLA.

IW ORK (w orkspace/output)
On exit, if $\mathbb{N} F O=0$, $\mathbb{I N}$ O RK (1) retums the optim al
LIW ORK.

LIW ORK (input)
The dim ension of the anay $\mathbb{I N} O R K$. If $\mathrm{N}<=1$, LIW ORK $>=1$. If $O B Z=N$ 'andN $>1, L \mathbb{W} O R K>=$ 1. If $\mathrm{OBBZ}=\mathrm{V}^{\prime}$ and $\mathrm{N}>1, \mathrm{~L} \mathbb{I} \mathrm{ORK}>=3+5 * \mathrm{~N}$.

INFO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$ th argum enthad an illegalvalue
$>0$ : CPOTRF orCHEEVD retumed an error code:
<= N: if $\mathbb{N} F O=i, C H E E V D$ failed to converge; i off-diagonalelem ents of an interm ediate tridiagonal form did notconverge to zero; > N : if $\mathbb{N}$ FO $=N+i$, for $1<=i<=N$, then the leading $m$ inor oforderiofB is not positive definite. The factorization of B could notbe com pleted and no eigenvalues or eigenvectors w ere com puted.

## FURTHER DETAILS

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv. of K entucky, USA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

chegvx - com pute selected eigenvalues, and optionally, eigenvectors of a com plex generalized H erm itian-definite eigenproblem, of the form $\mathrm{A} * \mathrm{x}=\left(\mathrm{lam}\right.$ bda) ${ }^{\mathrm{B}} \mathrm{B}$ x, $\mathrm{A} * \mathrm{~B} x=\left(\operatorname{lam}\right.$ bda) ${ }^{\mathrm{x}}$, or $B{ }^{*} A * X=\left(l a m\right.$ bda) ${ }^{*} X$

## SYNOPSIS

```
SU BROUT\mathbb{NE CHEGVX (TTYPE,JOBZ,RANGE,UPLO,N,A,LDA,B,LDB,VL,}
    VU,\mathbb{L,IU,ABSTOL,M ,W ,Z,LD Z,W ORK,LW ORK,RW ORK,IN ORK,}
    \mathbb{FA}\mathbb{L},\mathbb{NNFO}
CHARACTER * 1 JOBZ,RANGE,UPLO
COM PLEX A (LDA,*),B (LD B,*), Z (LD Z ,*),W ORK (*)
\mathbb{NTEGER ITYPE,N,LDA,LDB,\mathbb{L},\mathbb{U},M,LD Z,LW ORK,\mathbb{NFO}}\mathbf{N},\mp@code{L}
\mathbb{NTEGER IN ORK (*), \mathbb{FA}\mathbb{L}(*)}
REALVL,VU,ABSTOL
REALW (*),RWORK (*)
SUBROUT\mathbb{NE CHEGVX_64 (TTYPE,JOBZ,RANGE,UPLO,N,A,LDA,B,LDB,VL,}
    VU,\mathbb{L},\mathbb{IU},ABSTOL,M,W,Z,LDZ,W ORK,LW ORK,RW ORK,IN ORK,
    FA|,\mathbb{NFO)}
CHARACTER * 1 JOBZ,RANGE,UPLO
COM PLEX A (LDA,*),B (LD B,*), Z (LD Z ,*),W ORK (*)
\mathbb{NTEGER*8 ITYPE,N,LDA,LDB,\mathbb{L},\mathbb{U},M,LD Z,LW ORK,\mathbb{NFO}}\mathbf{N}=\mp@code{L}
INTEGER*8 \mathbb{N ORK (*), \mathbb{FA IL (*)}}\mathbf{(})
REALVL,VU,ABSTOL
REALW (*),RW ORK (*)
```


## F95 INTERFACE

SU BROUTINE HEGVX (TTYPE, $\operatorname{OOBZ}, R A N G E, U P L O, \mathbb{N}], A,[L D A], B,[L D B]$,
$\mathrm{VL}, \mathrm{VU}, \mathbb{I}, \mathbb{I}, \mathrm{ABSTOL}, \mathrm{M}, \mathrm{W}, \mathrm{Z},[\mathrm{LD} \mathrm{Z}],[\mathrm{W}$ ORK ], [LW ORK], RW ORK], [ $\mathbb{I N}$ ORK], $\mathbb{F A} \mathbb{H},[\mathbb{N} F O]$ )

CHARACTER (LEN=1):: JOBZ,RANGE,UPLO
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A, B , Z
$\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D A, L D B, \mathbb{L}, \mathbb{U}, M, L D Z, L W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{I N}$ ORK, $\mathbb{F} A \mathbb{I}$
REAL ::VL,VU,ABSTOL
REAL,D IM ENSION (:) ::W ,RW ORK

SU BROUTINE HEGVX_64 (TTYPE, JOBZ,RANGE,UPLO, $\mathbb{N}]$ ], A, [LDA ], B, [LDB], VL,VU, $\mathbb{I}, \mathbb{I}, A B S T O L, M, W, Z,[L D Z],[W O R K],[L W O R K], \mathbb{R W}$ ORK ], [ $\mathbb{I N}$ ORK], $\mathbb{F A} \mathbb{H},[\mathbb{N} F O])$

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A, B , Z
$\mathbb{N}$ TEGER (8) :: ITYPE,N,LDA,LDB, $\mathbb{L}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W$ ORK,
$\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{I N}$ ORK, $\mathbb{F} A \mathbb{I}$
REAL ::VL,VU,ABSTOL
REAL,D IM ENSION (:) ::W ,RW ORK

## C INTERFACE

\#include <sunperfh>
void chegvx (int itype, char j̀bz, char range, charuplo, int n, com plex *a, int lda, com plex *b, int ldb, float vl, float vu, int il, int iu, float abstol, int *m , float *w , com plex *z, int ldz, int *ifail, int *info);
void chegvx_64 (long itype, char jobz, char range, char uplo, long n, com plex *a, long lda, com plex *b, long ldb, float vl, float vu, long il, long in, float abstol, long *m, float* ${ }_{\mathrm{w}}$, com plex *z, long ldz, long *ifail, long *info);

## PURPOSE

chegvx com putes selected eigenvalues, and optionally, eigenvectors of a com plex generalized H erm itian-definite eigenproblem, of the form $A * x=\left(\operatorname{lam}\right.$ bda) ${ }^{*}{ }^{*}{ }^{x}, A$ A $\mathrm{Bx}=\left(\operatorname{lam}\right.$ bda) ${ }^{*} \mathrm{x}$, or $B{ }^{*} A * x=(l a m ~ b d a){ }^{*} x$. H ere $A$ and $B$ are assum ed to be $H$ erm itian and $B$ is also positive definite. Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

## ARGUMENTS

ITYPE (input)
Specifies the problem type to be solved:
$=1: A{ }^{*} \mathrm{x}=\left(\operatorname{lam}\right.$ bda)${ }^{\mathrm{B}}{ }^{*} \mathrm{x}$
$=2: \mathrm{A} * \mathrm{~B} * \mathrm{x}=\left(\operatorname{lam}\right.$ bda) ${ }^{*} \mathrm{x}$
$=3: B * A * x=\left(l a m\right.$ bda) ${ }^{*} \mathrm{x}$
JOBZ (input)
$=\mathrm{N}$ ': C om pute eigenvalues only;
= V ': C om pute eigenvalues and eigenvectors.
RANGE (input)
= A : alleigenvalues w illbe found.
= V ': alleigenvalues in the half-open interval ( $\mathrm{L}, \mathrm{VU}]$ w ill be found. = ' I ': the I -th through $\mathbb{I U}$-th eigenvaluesw illlbe found.

UPLO (input)
$=\mathrm{U}$ ': U pper triangles of $A$ and $B$ are stored;
$=\mathrm{L}$ ': Low ertriangles of $A$ and $B$ are stored.
$N$ (input) The order of the $m$ atrices $A$ and $B . N>=0$.
A (input/output)
O n entry, the $H$ erm itian $m$ atrix $A$. If $U P L O=U '$, the leading N -by N uppertriangular partofA contains the upper triangular part of the $m$ atrix $A$. If UPLO = L', the leading N by -N low er triangular part ofA contains the low er triangular part of them atrix $A$.

On exit, the low er triangle (if $\mathrm{UPLO}=\mathrm{L}$ ) or the upper triangle (if $\mathrm{U} P \mathrm{O}=\mathrm{U}$ ) of , including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, \mathbb{N})$.

B (input/output)
O n entry, the $H$ erm tian $m$ atrix $B$. If $U P L O=U '$, the leading N -by N uppertriangular partofB contains the upper triangular part of the $m$ atrix $B$. If UPLO $=\mathrm{L}$ ', the leading N -by -N low er triangular partofB contains the low er triangular part of the $m$ atrix $B$.

On exit, if $\mathbb{N} F O<=N$, the part of $B$ containing the $m$ atrix is overw rilten by the triangular factor
U orL from the Cholesky factorization $\mathrm{B}=\mathrm{U} * * \mathrm{H} * \mathrm{U}$ or $B=L{ }^{*} \mathrm{~L}^{* *} \mathrm{H}$.

LD B (input)
The leading dim ension of the aray $B$. LD B >= $\max (1, N)$.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRA NGE = A 'or I'.
VU (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU. N ot referenced ifRANGE=A 'or I '.

II (input)
IfRA N G E= I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}$, ifN $>0$; $\mathbb{H}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N ot referenced ifRANGE $=$ A'or V'.

IU (input)
If RA NGE= 'I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{Z}<=\mathbb{U}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{H}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N otreferenced ifRANGE $=$ A'or V'.

ABSTOL (input)
The absolute error tolerance for the eigenvalues.
A $n$ approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b]
of w idth less than orequal to
ABSTOL + EPS * max ( $k|$,$| b|),$
where EPS is the m achine precision. IfA BSTOL is less than or equalto zero, then EPS* $\mid$ | w illbe used in its place, where $F \mid$ is the 1 -norm of the tridiagonal $m$ atrix obtained by reducing $A$ to tridiagonal form .

E igenvalues w illbe com puted m ost accurately when ABSTOL is set to tw ioe the underflow threshold $2 *$ SLAM CH ( $\mathrm{S}^{\prime}$ ), not zero. If this routine retums w ith $\mathbb{N}$ FO $>0$, indicating that som e eigenvectors did
not converge, try setting ABSTOL to $2 *$ SLAM CH (S ).
M (output)
The total num ber ofeigenvalues found. $0<=\mathrm{M}<=$
N . IfRANGE $=\mathrm{A}^{\prime}, \mathrm{M}=\mathrm{N}$, and ifRANGE $=\mathrm{I}^{\prime}, \mathrm{M}=$ $\mathbb{U}-\mathbb{L}+1$.

W (output)
The firstM elem ents contain the selected eigenvalues in ascending order.
Z (output)
If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced. If JOBZ $=\mathrm{V}$ ', then if $\mathbb{N} F O=0$, the firstM columns of $Z$ contain the orthonorm aleigenvectors of the $m$ atrix A comesponding to the selected eigenvalues, $w$ ith the i-th colum $n$ of $Z$ holding the eigenvectorassociated $w$ ith $W$ (i). The eigenvectors are norm alized as follow s: if $I T Y P E=1$ or $2, Z * * T * B * Z=I$; if TTYPE $=3, Z * * T * \operatorname{inv}(B) * Z=I$.

If an eigenvector fails to converge, then that colum $n$ of $Z$ contains the latestapproxim ation to the eigenvector, and the index of the eigenvector is retumed in $\mathbb{F A} \mathbb{I}$. N ote: the userm ustensure that at leastm ax $(1, M)$ colum ns are supplied in the aray $Z$; ifRANGE = V', the exactvalue of $M$ is notknow $n$ in advance and an upper bound $m$ ust be used.

LD Z (input)
The leading dim ension of the array $Z . L D Z>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD $Z>=\mathrm{max}(1, N)$.

W ORK (w orkspace/output)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The length of the array $W$ ORK. LW ORK >= $\max \left(1,2{ }^{*} \mathrm{~N}-1\right)$. For optim al efficiency, LW ORK $>=$ $(N B+1) * N$, where $N B$ is the blocksize for CHETRD retumed by ILAENV.

If LW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

RW ORK (w orkspace)
dim ension ( $7 * \mathrm{~N}$ )

IV ORK (w orkspace)
dim ension ( $5 * N$ )

FAII (output)
If $J O B Z=V^{\prime}$, then if $\mathbb{N F O}=0$, the firstM ele$m$ ents of $\mathbb{F} A \mathbb{I}$ are zero. If $\mathbb{N} F O>0$, then $\mathbb{F} A \mathbb{H}$ contains the indices of the eigenvectors that failed to converge. If $\mathrm{OBB}=\mathrm{N}$ ', then $\mathbb{F} A \mathbb{I}$ is notreferenced.
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i-$ th argum enthad an illegalvalue
> 0: CPO TRF orCHEEVX retumed an errorcode:
$<=\mathrm{N}:$ if $\mathbb{N F O}=\mathrm{i}, \mathrm{CHEEVX}$ failed to converge; i eigenvectors failed to converge. Their indices are stored in amay $\mathbb{F A} \mathbb{I} .>N:$ if $\mathbb{N F O}=\mathrm{N}+$ $i$, for $1<=i<=N$, then the leading $m$ inor of orderiofB is notpositive definite. The factorization of $B$ could not be com pleted and no eigenvalues or eigenvectors w ere com puted.

## FURTHER DETAILS

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv. of K entucky, U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chem $m$-perform one of the $m$ atrix-m atrix operations $C:=$ alpha*A *B + beta*C orC : alpha*B *A + beta*C

## SYNOPSIS

```
SUBROUT\mathbb{NE CHEMM (S\mathbb{DE,UPLO,M,N,ALPHA,A,LDA,B,LDB,BETA,C,}}\mathbf{N},\textrm{M},\textrm{M}
    LD C )
CHARACTER * 1SIDE,UPLO
COM PLEX ALPHA,BETA
COM PLEX A (LDA,*),B (LD B,*),C (LD C ,*)
INTEGERM,N,LDA,LDB,LDC
```



```
    LD C)
CHARACTER * 1 SIDE,UPLO
COM PLEX ALPHA,BETA
COM PLEX A (LDA,*),B (LD B,*),C (LDC,*)
INTEGER*8M,N,LDA,LDB,LDC
```


## F95 INTERFACE

SU BROUTINE HEMM (SDE, UPLO, $\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A], B,[L D B]$, BETA, C, [LDC])

CHARACTER (LEN=1) ::SDE,UPLO
COM PLEX ::ALPHA,BETA
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A,B,C
$\mathbb{N} T E G E R:: M, N, L D A, L D B, L D C$
SUBROUTINE HEMM _64 (SDE, UPLO, $\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A], B,[L D B]$, BETA, C, [LDC])

CHARACTER ( $\llcorner E N=1$ ) : : SDE E , UPLO
COM PLEX ::ALPHA,BETA
COMPLEX, D $\mathbb{M}$ ENSION (: : : : : A, B, C
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{LD} A, L D B, L D C$

## C INTERFACE

\#include <sunperfh>
void chemm (charside, char uplo, int m, int n, com plex *alpha, com plex *a, int lda, com plex *b, int ldb, com plex *beta, com plex *c, int ldc);
void chem m _64 (charside, charuplo, long m, long n, com plex *alpha, com plex *a, long lda, com plex *b, long ldlb, com plex *beta, com plex * C , long ldc);

## PURPOSE

chem $m$ perform sone of the $m$ atrix $m$ atrix operations $C:=$ alpha*A *B + beta*C orC $:=$ alpha*B *A + beta*C where alpha and beta are scalars, $A$ is an herm titian $m$ atrix and $B$ and $C$ arem by $n \mathrm{~m}$ atriges.

## ARGUMENTS

SIDE (input)
On entry, SIDE specifiesw hether the herm itian $m$ atrix A appears on the leftorright in the operation as follow s:
$S \mathbb{D E}=$ L'or I' $C:=$ a耳pha*A *B + beta* $C$,

SIDE = R 'or 'r' C : alpha*B *A + beta*C ,

U nchanged on exit.

## UPLO (input)

On entry, UPLO specifies whether the upper or lower triangular part of the herm itian $m$ atrix $A$ is to be referenced as follow s:
$\mathrm{UPLO}=\mathrm{U}$ 'or U ' Only the upper triangularpart of the herm itian $m$ atrix is to be referenced.
$\mathrm{UPLO}=\mathrm{L}$ 'or $\mathrm{I}^{\prime}$ ' O nly the low ertriangularpart of the herm itian m atrix is to be referenced.

U nchanged on exit.
M (input)
O $n$ entry, M specifies the num ber of row sof the $m$ atrix $C . M$ M $=0$. U nchanged on exit.

N (input)
O n entry, N specifies the num ber of colum ns of the $m$ atrix $C . N>=0$. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.
A (input)
COM PLEX array ofD $\mathbb{I M}$ ENSION (LDA,ka), where ka is $m$ when $S D E=\mathbb{L}$ 'or $I^{\prime}$ and is $n$ otherw ise.

Before entry with SDE = L'or 1 ', the $m$ by m partof the array A mustcontain the herm itian $m$ atrix, such thatw hen UPLO $=U$ ' or $L^{\prime}$ ', the leading $m$ by $m$ uppertriangular part of the array A mustcontain the upper triangular part of the herm itian $m$ atrix and the strictly low er triangularpart of $A$ is not referenced, and $w$ hen UPLO = L' or ${ }^{1}$ ', the leading $m$ by $m$ lowertriangularpart of the array A m ust contain the lowertriangularpart of the herm itian $m$ atrix and the strictly upper triangular part of A is not referenced.

Before entry w ith $S D E=R$ 'or $r$ ', the $n$ by $n$ partof the array A must contain the herm itian $m$ atrix, such thatw hen UPLO = U ' or L ', the leading n by n uppertriangularpart of the array A mustcontain the upper triangular part of the herm itian $m$ atrix and the strictly low er triangularpart of A is not referenced, and when UPLO = L' or 1 ', the leading $n$ by $n$ low er triangularpart of the array A must contain the lowertriangularpart of the herm itian $m$ atrix and the strictly upper triangular part of A is not referenced.
$N$ ote that the im aginary parts of the diagonal elem ents need notbe set, they are assum ed to be zero. U nchanged on exit.

O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen SDE = L'or I' then LD A >= max ( $1, \mathrm{~m}$ ), otherw ise LD $A>=\max (1, n)$. U nchanged on exit.

B (input)
COM PLEX array ofD $\mathbb{I M}$ ENSION (LD $B, n$ ). Before
entry, the leading $m$ by $n$ part of the array $B$
$m$ ust contain the $m$ atrix $B$. Unchanged on exit.
LD B (input)
O $n$ entry, LD B specifies the firstdim ension of $B$ as declared in the calling (sub) program.
LDB $m$ ust be at leastm ax ( $1, \mathrm{~m}$ ). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen
BETA is supplied as zero then $C$ need notbe set on input. U nchanged on exit.

C (input/output)
COMPLEX aray ofD $\mathbb{I M} E N S I O N(L D C, n)$.
Before entry, the leading $m$ by $n$ part of the array $C$ mustcontain the $m$ atrix $C$, exceptw hen beta is zero, in which case $C$ need notbe set on entry.

On exit, the array $C$ is overw ritten by the $m$ by $n$ updated $m$ atrix.

LD C (input)
On entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program. LD C must be at leastm ax ( $1, \mathrm{~m}$ ). U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chem v -perform the m atrix-vectoroperation $\mathrm{y}:=\operatorname{alpha} \mathrm{A} \mathrm{A} * \mathrm{x}$ + beta*y

## SYNOPSIS

```
SUBROUT\mathbb{NE CHEMV (UPLO,N,ALPHA,A,LDA,X, NNCX,BETA,Y, INCY)}
CHARACTER * 1 UPLO
COM PLEX ALPHA,BETA
COM PLEX A (LDA,*),X (*),Y (*)
\mathbb{NTEGER N,LDA,INCX,}\mathbb{NCY}
```



```
CHARACTER * 1 UPLO
COM PLEX ALPHA,BETA
COM PLEX A (LDA,*),X (*),Y (*)
\mathbb{NTEGER*8N,LDA, INCX,}\mathbb{N}CY
```


## F95 INTERFACE

SU BROUTINE HEMV (UPLO, $\mathbb{N}], A L P H A, A,[L D A], X,[\mathbb{N C X}], B E T A, Y,[\mathbb{N C Y}])$
CHARACTER (LEN=1) ::UPLO
COMPLEX ::ALPHA,BETA
COM PLEX,D $\mathbb{I M}$ ENSION (:) :: X,Y
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : : A
$\mathbb{N} T E G E R:: N, L D A, \mathbb{N C X}, \mathbb{N} C Y$
SU BROUTINE HEMV_64 (UPLO, $\mathbb{N}], A L P H A, A,[L D A], X,[\mathbb{N} C X], B E T A, Y$, [ $\mathbb{N} C Y$ ])

COM PLEX ::ALPHA,BETA
COM PLEX , D $\mathbb{M}$ ENSION (:) ::X,Y
COM PLEX , D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER (8) :: N , LDA $, \mathbb{N} C X, \mathbb{N} C Y$

## C INTERFACE

\#include <sunperfh>
void chem v (charuple, intn, com plex *alpha, com plex *a, int lda, com plex *x, int incx, com plex *beta, com plex *y, intincy);
void chem v_64 (charuplo, long n, com plex *alpha, com plex *a, long lda, com plex *x, long incx, com plex *beta, com plex *y, long incy);

## PURPOSE

chem $v$ perform $s$ the $m$ atrix-vector operation $y:=a l p h a \star A * x+$ beta*y w here alpha and beta are scalars, $x$ and $y$ are $n$ ele$m$ entvectors and $A$ is an $n$ by $n$ herm itian $m$ atrix.

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the array $A$ is to be referenced as follow s:
$\mathrm{UPLO}=\mathrm{U}$ 'or L ' Only the upper triangularpart of A is to be referenced.

UPLO = L 'or I' O nly the low ertriangularpart of A is to be referenced.

U nchanged on exit.

N (input)
O $n$ entry, $N$ specifies the order of the $m$ atrix $A$. $\mathrm{N}>=0$. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha.
U nchanged on exit.

A (input)
Before entry w ith UPLO = U 'or L ', the leading
n by n uppertriangularpart of the anay A m ust contain the upper triangular part of the herm itian $m$ atrix and the strictly low ertriangularpart of $A$ is not referenced. Before entry w ith UPLO = L' or I', the leading $n$ by $n$ low er triangularpart of the array A m ustcontain the low er triangular part of the herm itian $m$ atrix and the strictly uppertriangularpart of $A$ is not referenced. $N$ ote that the im aginary parts of the diagonalele$m$ ents need notbe set and are assum ed to be zero. U nchanged on exit.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A >= $\max (1, n)$. U nchanged on exit.
X (input)
$(1+(n-1) \star \operatorname{abs}(\mathbb{N} C X))$. Before entry, the increm ented array $X$ must contain the $n$ elem ent vectorx. U nchanged on exit.
$\mathbb{N} C X$ (input)
On entry, $\mathbb{I N C X}$ specifies the increm ent for the elem ents ofX. $\mathbb{N} C X<>0$. U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then Y need notbe set on input. U nchanged on exit.

Y (input/output)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))$. Before entry, the increm ented array $Y$ must contain the $n$ elem ent vectory. On exit, $Y$ is overw rilten by the updated vectory.
$\mathbb{N} C Y$ (input)
On entry, $\mathbb{N N C Y}$ specifies the increm ent for the elem ents of $. \mathbb{N} C Y<>0$. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cher-perform the herm itian rank 1 operation $A:=$ alpha*x*conjg ( $x^{\prime}$ ) + A

## SYNOPSIS

```
SUBROUTINE CHER (UPLO,N,ALPHA,X,INCX,A,LDA)
CHARACTER * 1 UPLO
COM PLEX X (*),A (LDA,*)
INTEGER N, INCX,LDA
REAL A LPHA
SUBROUT\mathbb{NE CHER_64 (UPLO,N,ALPHA,X,NNCX,A,LDA)}
CHARACTER * 1 UPLO
COM PLEXX (*),A (LDA,*)
INTEGER*8 N, INCX,LDA
REALALPHA
F95 INTERFACE
```



```
CHARACTER (LEN=1)::UPLO
COMPLEX,D IM ENSION (:) ::X
COM PLEX,D IM ENSION (:,:)::A
\mathbb{NTEGER::N,\mathbb{NCX,LDA}}\mathbf{N}=\mp@code{N}
REAL ::ALPHA
SUBROUTINE HER_64 (UPLO, N ],ALPHA ,X , [NCX ],A,[LDA ])
CHARACTER (LEN=1)::UPLO
COMPLEX,D IM ENSION (:) ::X
```


## C INTERFACE

\#include <sunperfh>
void cher(charuple, intn, float alpha, complex *x, int incx, com plex *a, int lda);
void cher_64 (charuplo, long n, float alpha, com plex *x, long incx, com plex *a, long lda);

## PURPOSE

cher performs the herm titian rank 1 operation $A:=$ alpha* $x^{\star}$ con $j g\left(x^{\prime}\right)+A$ where alpha is a realscalar, $x$ is an $n$ elem ent vector and $A$ is an $n$ by $n$ herm titian $m$ atrix.

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the array $A$ is to be referenced as follow s:
$\mathrm{UPLO}=\mathrm{U}$ 'or L ' Only the upper triangularpant of $A$ is to be referenced.
$\mathrm{UPLO}=\mathrm{L}$ 'or l' O nly the low er triangularpart of $A$ is to be referenced.

U nchanged on exit.

N (input)
O n entry, $N$ specifies the order of the $m$ atrix $A$. $N>=0$. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C X))$. Before entry, the increm ented array $X$ must contain the $n$ elem ent vectorx. U nchanged on exit.
$\mathbb{N C X}$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents ofX. $\mathbb{N} C X<>0$. U nchanged on exit.

A (input/output)
Before entry w ith UPLO = U 'or 4 ', the leading $n$ by $n$ upper triangular part of the array A m ust contain the upper triangular part of the herm itian $m$ atrix and the strictly low er triangularpart of A is not referenced. O n exit, the upper triangular part of the array A is overw ritten by the upper triangularpart of the updated $m$ atrix. Before entry w ith UPLO = L 'or I', the leading $n$ by $n$ low er triangularpart of the anray A m ust contain the low er triangularpart of the herm itian $m$ atrix and the strictly upper triangularpart of A is not referenced. On exit, the low er triangularpart of the array $A$ is overw rilten by the low er triangular part of the updated $m$ atrix. N ote that the im aginary parts of the diagonalelem ents need not be set, they are assum ed to be zero, and on exit they are setto zero.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD A >= $\max (1, n)$. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cher2 -perform the herm itian rank 2 operation A := alpha*x*conjg( $y^{\prime}$ ) + conjg (alpha ) ${ }^{\star} y^{\star}$ conjg $\left(x^{\prime}\right)+$ A

## SYNOPSIS

```
SUBROUT\mathbb{NE CHER2 (UPLO,N,ALPHA,X, NNCX,Y, INCY,A,LDA )}
CHARACTER * 1 UPLO
COM PLEX ALPHA
COM PLEX X (*),Y (*),A (LDA ,*)
\mathbb{N TEGER N, INCX, \mathbb{NCY,LDA}}\mathbf{N},\textrm{L}
SU BROUTINE CHER2_64(UPLO ,N,ALPHA,X, NNCX,Y, INCY,A,LDA )
CHARACTER * 1 UPLO
COM PLEX ALPHA
COM PLEX X (*),Y (*),A (LDA ,*)
\mathbb{N TEGER*8N, INCX, INCY,LDA}
```


## F95 INTERFACE

SU BROUTINE HER2 (UPLO, $\mathbb{N}], A L P H A, X,[\mathbb{N C X}], Y,[\mathbb{N C Y}], A,[L D A])$
CHARACTER (LEN=1)::UPLO
COMPLEX ::ALPHA
COM PLEX,D $\mathbb{I M}$ ENSION (:) :: X,Y
COM PLEX,D $\mathbb{M}$ ENSION (: : : ) ::A
$\mathbb{N} T E G E R:: N, \mathbb{N C X}, \mathbb{N} C Y, L D A$
SU BROUTINE HER2_64 (UPLO, $\mathbb{N}]$, A LPHA $, \mathrm{X},[\mathbb{N} C X], Y,[\mathbb{N} C Y], A,[L D A])$
CHARACTER (LEN=1) ::UPLO
COMPLEX ::ALPHA

COM PLEX, D $\mathbb{M} \operatorname{ENSION(:):~X,Y~}$
COM PLEX , D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER (8) :: $N, \mathbb{N} C X, \mathbb{N} C Y, L D A$

## C INTERFACE

\#include <sunperfh>
void cher2 (charuplo, intn, com plex *alpha, com plex *x, int incx, com plex *y, int incy, com plex *a, int lda);
void cher2_64 (charuplo, long n, com plex *alpha, com plex *x, long incx, com plex *y, long incy, com plex *a, long lda);

## PURPOSE

cher2 performs the herm titian rank 2 operation $A:=$ alpha* $x^{\star}$ con $\dot{\jmath}\left(y^{\prime}\right)+$ con $\dot{g}(\text { alpha })^{\star} y^{\star}$ con $\dot{g}\left(x^{\prime}\right)+A$ where alpha is a scalar, $x$ and $y$ are $n$ elem ent vectors and $A$ is an $n$ by $n$ herm itian m atrix.

## ARGUMENTS

UPLO (input)
O n entry, UPLO specifies whether the upper or low er triangular part of the array $A$ is to be referenced as follow s:
$\mathrm{UPLO}=\mathrm{U}$ 'or L ' Only the upper triangularpant of A is to be referenced.
$\mathrm{UPLO}=\mathrm{L}$ 'or I' O nly the low ertriangularpart of $A$ is to be referenced.

U nchanged on exit.

N (input)
O n entry, $N$ specifies the order of the $m$ atrix A. $\mathrm{N}>=0$. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
$(1+(n-1) * a b s(\mathbb{N} C X))$. Before entry, the increm ented array $X$ must contain the $n$ elem ent
vectorx. U nchanged on exit.
$\mathbb{N C X}$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents ofX. $\mathbb{N} C X<>0$. U nchanged on exit.

Y (input)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))$. Before entry, the increm ented array $Y$ must contain the $n$ elem ent vectory. U nchanged on exit.
$\mathbb{N C Y}$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $Y$. $\mathbb{N C} C$ <> 0 . U nchanged on exit.

A (input/output)
Before entry w ith UPLO = U 'or L ', the leading $n$ by $n$ upper triangular part of the array A $m$ ust contain the upper triangular part of the herm itian $m$ atrix and the strictly low ertriangularpartofA is not referenced. On exit, the upper triangular part of the array A is overw rilten by the upper triangularpart of the updated $m$ atrix. Before entry $w$ ith UPLO $=\mathrm{L}$ 'or I ', the leading $n$ by $n$ low er triangularpart of the array A m ust contain the low er triangularpart of the hem itian $m$ atrix and the strictly uppertriangularpart of A is not referenced. On exit, the low er triangularpart of the array $A$ is overw ritten by the low er triangular part of the updated $m$ atrix. N ote that the im aginary parts of the diagonalelem ents need not be set, they are assum ed to be zero, and on exit they are set to zero.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A $>=$ $\max (1, n)$. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cher2k -perform one of the H erm titian rank 2 k operations C $:=$ alpha*A*conjg ( $\mathrm{B}^{\prime}$ ) + con $\dot{g}($ alpha $){ }^{*} \mathrm{~B} * \operatorname{con} \dot{g}\left(\mathrm{~A}^{\prime}\right)+$ beta*C orC $:=$ alpha*con $\dot{g}\left(A^{\prime}\right) * B+$ con $\dot{g}($ alpha $) *$ con $\dot{g}($ $\left.B^{\prime}\right)^{\star} A+$ beta* $C$

## SYNOPSIS

```
SUBROUT\mathbb{NE CHER2K (UPLO,TRANSA,N,K,ALPHA,A,LDA,B,LDB,BETA,C,}
    LD C )
CHARACTER * 1 UPLO,TRANSA
COM PLEX ALPHA
COM PLEX A (LDA,*), B (LD B ,*), C (LD C ,*)
IN TEGER N,K,LDA,LDB,LDC
REALBETA
SU BROUTINE CHER2K_64 (UPLO,TRANSA,N,K,ALPHA,A,LDA,B,LDB,BETA,
        C,LDC )
CHARACTER * 1 UPLO,TRANSA
COM PLEX ALPHA
COM PLEX A (LDA,*), B (LD B ,*), C (LD C ,*)
INTEGER*8N,K,LDA,LD B ,LDC
REALBETA
```


## F95 INTERFACE

```
SU BROUTINE HER2K (UPLO, [TRANSA], \(\mathbb{N}],[K], A L P H A, A,[L D A], B,[L D B]\), BETA, C, [LDC])
CHARACTER (LEN=1) ::UPLO,TRANSA
COMPLEX ::ALPHA
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,B,C
```

$\mathbb{N}$ TEGER :: N, K , LDA , LD B , LD C
REAL ::BETA

SU BROUTINE HER2K_64 (UPLO, [TRANSA ], $\mathbb{N}],[K], A L P H A, A,[L D A], B$, [LDB],BETA, C , [LDC])

CHARACTER (LEN=1) :: UPLO, TRANSA
COMPLEX ::ALPHA
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : A , B , C
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{K}, \mathrm{LD} A, L D B, L D C$
REAL ::BETA

## C INTERFACE

\#include <sunperfh>
void cher2k (charuplo, chartransa, intn, int $k$, com plex* alpha, com plex *a, intlda, com plex *b, int ldb, floatbeta, com plex ${ }^{*}$ c, int ldc);
void cher2k_64 (charuplo, chartransa, long n, long k, com plex *alpha, com plex *a, long lda, com plex *b, long ldlo, floatbeta, com plex *c, long ldc);

## PURPOSE

cher 2 k perform s one of the H erm titian rank 2 k operations $\mathrm{C}:=$

 beta*C where alpha and beta are scalarsw ith beta real, $C$ is an $n$ by $n H$ erm itian $m$ atrix and $A$ and $B$ are $n$ by $k$ $m$ atrices in the first case and $k$ by $n m$ atrioes in the second case.

## ARGUMENTS

## UPLO (input)

On entry, UPLO specifies whether the upper
or lower triangular part of the array $C$ is
to be referenced as follow $s$ :
UPLO = U'or L' Only the upper triangular partof $C$ is to be referenced.

UPLO = L'or I' Only the lower triangular partof $C$ is to be referenced.

U nchanged on exit.

TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA $=\mathrm{N}^{\prime}$ 'or $\mathrm{h}^{\prime} \mathrm{C}=$ alpha*A*conjg ( $\mathrm{B}^{\prime}$ )

+ congg (alpha)*B*conjg (A') + beta*C .
TRANSA $=C^{\prime}$ ort' $\mathrm{C}:=$ alpha*conjg ( $\left.\mathrm{A}^{\prime}\right) \star$ B + cong (alpha )*conjg ( $\mathrm{B}^{\prime}$ )*A + beta*C .

U nchanged on exit.

TRANSA is defaulted to N 'forF $95 \mathbb{I N}$ TERFACE.

N (input)
O n entry, $N$ specifies the order of the $m$ atrix C. N m ustbe at least zero. U nchanged on exit.

K (input)
On entry w th TRANSA $=N$ 'or $h$ ', $K$ specifies the num berof colum ns of the $m$ atrices $A$ and $B$, and on entry $w$ ith TRANSA $=C^{\prime}$ or $\mathrm{K}^{\prime}, \mathrm{K}$ specifies the num ber of row sof the $m$ atrices $A$ and B. K m ustbe at least zero. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
COM PLEX anay ofD $\mathbb{M}$ ENSION (LDA, ka ), where ka isk when TRANSA = N 'or h ', and is n otherw ise. Before entry w th TRANSA $=\mathrm{N}$ ' or h ', the leading n by k partof the array $A$ $m$ ust contain the $m$ atrix $A$, otherw ise the leading k by n partof the array A mustcontain the $m$ atrix A. U nchanged on exit.

LDA (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program.
W hen TRANSA $=N$ 'orh'then LDA must be at least $\max (1, n)$, otherw ise LDA $m$ ust.be at least $\max (1, k)$. U nchanged on exit.

B (input)
COM PLEX anay ofD $\mathbb{M}$ ENSION (LDB, kb ), where kb isk when TRANSA = N 'or h', and is
n otherw ise. Before entry w ith TRANSA $=\mathrm{N}$ ' or
h ', the leading n by k part of the anay B $m$ ust contain the $m$ atrix $B$, otherw ise the leading k by n part of the array $\mathrm{B} m$ ustcontain the $m$ atrix $B$. Unchanged on exit.

LD B (input)
O n entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program. W hen TRANSA $=N$ 'or $h$ 'then LDB must be at least $\max (1, n)$, otherw ise LDB mustbe at least $\max (1, k)$. U nchanged on exit.
BETA (input)
On entry, BETA specifies the scalar beta. U nchanged on exit.

C (input/output)
COMPLEX amay ofD $\mathbb{M} E N S I O N(L D C, n)$.

Before entry with UPLO = U 'or $L$ ', the leading $n$ by $n$ upper triangularpart of the array $C$ $m$ ustcontain the upper triangular part of the H erm itian $m$ atrix and the strictly low er triangularpartof $C$ is not referenced. On exit, the uppertriangularpart of the amay $C$ is overw ritten by the upper triangularpart of the updated $m$ atrix.

Before entry w ith UPLO = L'or I', the leading $n$ by $n$ low er triangular part of the array $C$ $m$ ustcontain the low er triangular part of the H erm itian $m$ atrix and the strictly upper triangularpartof C is not referenced. On exit, the low er triangularpart of the amay $C$ is overw ritten by the low er triangularpart of the updated $m$ atrix.
$N$ ote that the im aginary parts of the diagonalele$m$ ents need not be set, they are assum ed to be zero, and on exit they are set to zero.

LD C (input)
O n entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program.
LD C m ust be at leastm $a x(1, n)$. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cherfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is H erm itian indefintite, and provides emorbounds and backw ard emror estim ates for the solution

## SYNOPSIS

```
SU BROUT\mathbb{NE CHERFS (UPLO,N,NRHS,A,LDA,AF,LDAF, \mathbb{P IV OT,B,LDB,X,}}\mathbf{N},\textrm{L}
    LD X,FERR,BERR,W ORK,W ORK 2, INFO)
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),AF (LDAF,*),B (LDB,*),X (LDX,*),W ORK (*)
\mathbb{NTEGERN,NRHS,LDA,LDAF,LDB,LDX,}\mathbb{N}FO
\mathbb{NTEGER \mathbb{PIVOT (*)}}\mathbf{(})
REAL FERR(*),BERR(*),WORK2 (*)
SU BROUT\mathbb{NE CHERFS_64 UPLO,N,NRHS,A,LDA,AF,LDAF, PPIVOT,B,LDB,}
    X,LDX,FERR,BERR,W ORK,W ORK 2,INFO)
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),AF (LDAF,*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGER*8N,NRHS,LDA,LDAF,LDB,LDX,}\mathbb{N}F
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
REAL FERR (*),BERR (*),W ORK2 (*)
```


## F95 INTERFACE

```
SU BROUTINE HERFS (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F], \mathbb{P} \mathbb{V} O T, B\), [LD B], X, [LDX ],FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N F O}]\) )
CHARACTER (LEN=1) ::UPLO
COM PLEX,D IM ENSION (:) ::W ORK
COMPLEX,D \(\mathbb{M}\) ENSION (:,:): :A,AF,B,X
```

$\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$
REAL,D $\mathbb{M} E N S I O N(:):: F E R R, B E R R, W$ ORK 2

SUBROUT $\mathbb{N} E$ HERFS_64 (UPLO, $\mathbb{N}], \mathbb{N R H S}], A,[L D A], A F,[L D A F], \mathbb{P} \mathbb{I} O T$, $B,[L D B], X,[L D X], F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N} F O])$

CHARACTER (LEN=1) ::UPLO
COM PLEX,D $\mathbb{I M} E N S I O N(:):: W O R K$
COM PLEX, D $\mathbb{M}$ ENSION (: : : : : A, AF, B, X
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} A, L D A F, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}$ (8), D $\mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{V} O T$
REAL,D $\mathbb{M}$ ENSION (:) :: FERR,BERR,W ORK2
C INTERFACE
\#include <sunperfh>
void cherfs (char uplo, intn, intnrhs, com plex *a, int lda, com plex *af, int ldaf, int*ipivot, com plex *b, int ldb, com plex *x, int ldx, float *ferr, float *berr, int *info);
void cherfs_64 (charuplo, long n, long nrhs, com plex *a, long lda, com plex *af, long ldaf, long *ipivot, com plex *b, long ldb, com plex *x, long ldx, float * ferrr, float *berr, long *info);

## PURPOSE

cherfs im proves the com puted solution to a system of linear equations w hen the coefficientm atrix is H erm itian indefinite, and provides errorbounds and backw ard error estim ates for the solution.

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': Upper triangle ofA is stored;
$=\mathbb{L}$ ': Low ertriangle of is stored.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber
of colum ns of the m atrices B and X. NRHS $>=0$.

A (input) The $H$ erm tian $m$ atrix $A$. If $U P L O=U$ ', the leading N -by- N uppertriangularpartofA contains the
upper triangularpart of the $m$ atrix $A$, and the strictly low ertriangularpartofA is not referenced. IfU PLO = L', the leading N -by-N lower triangularpart ofA contains the low er triangular part of the $m$ atrix A, and the strictly upper triangularpartofA is not referenced.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, N)$.

## AF (input)

The factored form of them atrix A. AF contains the block diagonal $m$ atrix $D$ and the $m$ ultipliers used to obtain the factor $U$ orL from the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}$ orA $=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}$ as com puted by CHETRF.

LDAF (input)
The leading dim ension of the array AF. LDAF >= $\max (1, N)$.
$\mathbb{P I V O T}$ (input)
D etails of the interchanges and the block structure ofD as determ ined by CHETRF.
$B$ (input) The righthand side $m$ atrix $B$.
LD B (input)
The leading dim ension of the amay $B$. LD B >= $\max (1, \mathbb{N})$.

X (input/output)
On entry, the solution $m$ atrix $X$, as com puted by CHETRS. On exit, the im proved solution $m$ atrix $X$.

LD $X$ (input)
The leading dim ension of the anay X . LD X >= $\max (1, \mathbb{N})$.

FERR (output)
The estim ated forw ard errorbound for each solution vector $X()$ ) the $j$ th column of the solution $m$ atrix X). If XTRUE is the true solution corresponding to $X(\mathcal{H}), \operatorname{FERR}(\mathcal{)})$ is an estim ated upperbound for the $m$ agnitude of the largest ele$m$ ent in ( $X(\mathcal{\nu})-X$ TRUE) divided by the $m$ agnitude of the largestelem ent in $X(j)$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

## BERR (output)

The com ponentw ise relative backw ard error of each solution vector $X$ ( $\mathcal{j}$ ) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( $\mathcal{J}$ ) an exactsolution).
W ORK (w orkspace)
dim ension $(2 * N)$
W ORK 2 (w orkspace)
dim ension (N)
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{I N}$ FO = -i, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
cherk -perform one of the H erm itian rank k operations C
:= alpha*A*conjg(A') + beta*C orC := alpha*conjg(A')*A
+ beta*C
```


## SYNOPSIS

```
SUBROUTINE CHERK (UPLO,TRANSA,N,K,ALPHA,A,LDA,BETA,C,LDC)
CHARACTER * 1 UPLO,TRANSA
COM PLEX A (LDA,*),C (LDC,*)
INTEGER N,K,LDA,LDC
REALALPHA,BETA
SUBROUTINE CHERK_64 (UPLO,TRANSA,N,K,ALPHA,A,LDA,BETA,C,LDC)
CHARACTER * 1 UPLO,TRANSA
COM PLEX A (LDA,*),C (LDC ,*)
INTEGER*8N,K,LDA,LDC
REALALPHA,BETA
```


## F95 INTERFACE

SU BROUTINE HERK (UPLO, [TRANSA], $\mathbb{N}],[K], A L P H A, A,[L D A], B E T A, C$, [LD C ])

CHARACTER (LEN=1) ::UPLO,TRANSA
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::A, C
$\mathbb{N} T E G E R:: N, K, L D A, L D C$
REAL ::ALPHA,BETA

SU BROUTINE HERK_64 (UPLO, [TRANSA ], $\mathbb{N}],[K], A L P H A, A,[L D A], B E T A$, C, [LD C ])

CHARACTER (LEN=1) ::UPLO,TRANSA
COM PLEX,D $\mathbb{I}$ ENSION (:,:) ::A, C
$\mathbb{N}$ TEGER (8) ::N,K,LDA,LDC
REAL ::ALPHA,BETA

## C INTERFACE

\#include <sunperfh>
void cherk (charuplo, char transa, int n, int k, float alpha, com plex *a, int lda, floatbeta, com plex ${ }^{*} \mathrm{C}$, int ldd );
void cherk_64 (charuplo, chartransa, long n, long k, float aloha, com plex *a, long lda, floatbeta, com plex *C, long ldc);

## PURPOSE

cherk perform s one of the $H$ erm itian rank $k$ operations $C:=$ alpha*A *conjg ( $A^{\prime}$ ) + beta*C orC := alpha*cong ( $\mathrm{A}^{\prime}$ ) ${ }^{*} \mathrm{~A}+$ beta*C where alpha and beta are realscalars, C is an n by $n$ Herm itian $m$ atrix and $A$ is an $n$ by $k m$ atrix in the first case and $a k$ by $n m$ atrix in the second case.

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whether the upper or lower triangular part of the array $C$ is to be referenced as follow s:

UPLO = U'or L' Only the upper triangular partof $C$ is to be referenced.

UPLO = L'or I' Only the lower triangular part of $C$ is to be referenced.

U nchanged on exit.

TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA $=$ N 'or $h^{\prime} \mathrm{C}:=$ alpha*A *conjg (A') + beta*C.

TRANSA = C'or $\boldsymbol{E}^{\prime} \mathrm{C}:=$ alpha*cong (A')*A +
beta*C .

U nchanged on exit.
TRANSA is defaulted to N 'forF95 $\mathbb{I N}$ TERFACE.

N (input)
On entry, $N$ specifies the order of the $m$ atrix $C$.
N m ustbe at least zero. U nchanged on exit.
K (input)
On entry with TRANSA = N 'or h', K specifies the number of columns of the matrix $A$, and on entry $w$ ith TRANSA $=C^{\prime}$ or $\mathrm{C}^{\prime}$, $K$ specifies the num ber of row sof the $m$ atrix A. K m ustbe at least zero. U nchanged on exit.
ALPHA (input)
On entry, ALPHA specifies the scalar alpha.
U nchanged on exit.

A (input)
COM PLEX aray ofD $\mathbb{M}$ ENSION (LDA, ka), where ka isk when TRANSA = N 'or h', and is
n otherw ise. Before entry w th TRANSA $=\mathrm{N}$ ' or h ', the leading n by k part of the aray $A$ $m$ ust contain the $m$ atrix $A$, otherw ise the leading k by n partofthe array A mustcontain the m atrix A. U nchanged on exit.

LDA (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen TRANSA $=\mathrm{N}$ 'or h 'then LDA must be at least $\max (1, n)$, otherw ise LDA m ust.be at least $\max (1, k)$. U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. U nchanged on exit.

C (input/output)
COM PLEX array ofD $\mathbb{M}$ ENSION (LDC,n).
Before entry with UPLO = U 'or U ', the leading $n$ by $n$ upper triangularpart of the array $C$ $m$ ust contain the upper triangular part of the H erm itian $m$ atrix and the strictly low ertriangularpartofC is not referenced. On exit, the upper triangularpart of the array $C$ is overw ritten by the upper triangularpart of the updated
m atrix.

Before entry w ith UPLO = L 'or I', the leading $n$ by $n$ low er triangular part of the array $C$ $m$ ust contain the low er triangular part of the H erm itian $m$ atrix and the strictly upper triangularpartof $C$ is not referenced. On exit, the low er triangularpart of the amay $C$ is overw ritten by the low er triangularpart of the updated $m$ atrix.

N ote that the im aginary parts of the diagonalele$m$ ents need not be set, they are assum ed to be zero, and on exit they are set to zero.
LD C (input)
O n entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program. LD C must be at leastm ax ( $1, \mathrm{n}$ ). U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chesv -com pute the solution to a com plex system of linear equations $A * X=B$,

## SYNOPSIS

```
SUBROUTINE CHESV (UPLO,N,NRHS,A,LDA, \mathbb{PIVOT,B,LDB,W ORK,LDW ORK,}
    \mathbb{NFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LD B,*),W ORK (*)
\mathbb{NTEGERN,NRHS,LDA,LDB,LDW ORK,INFO}
INTEGER \mathbb{PIVOT (*)}
SU BROUT\mathbb{NE CHESV_64 (UPLO,N,NRHS,A,LDA, \mathbb{PIVOT,B,LDB,W ORK,}}\mathbf{N}\mathrm{ , N}
    LDW ORK,\mathbb{NFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LD B,*),W ORK (*)
\mathbb{NTEGER*8N,NRHS,LDA,LDB,LDW ORK, INFO}
INTEGER*8 \mathbb{PIVOT (*)}
```


## F95 INTERFACE

SU BROUTINE HESV (UPLO, $\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{V} O T, B,[L D B],[W O R K]$, [LDW ORK], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::UPLO
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::A,B
$\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, L D W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {I}OT}$
SU BROUTINE HESV_64 (UPLO, $\mathbb{N}], \mathbb{N R H S}], A,[L D A], \mathbb{P} \mathbb{I V O T}, \mathrm{B},[\operatorname{LD} B]$,
[W ORK], [LDW ORK], [ $\mathbb{N F O}])$

CHARACTER (LEN=1) :: UPLO
COM PLEX ,D $\mathbb{M}$ ENSION (:) ::W ORK
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : A , B
$\mathbb{N}$ TEGER (8) :: N , NRHS,LDA,LDB,LDWORK, $\mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \mathrm{ENSION}(:):: \mathbb{P} \mathbb{I} O T$

## C INTERFACE

\#include <sunperfh>
void chesv (char uplo, intn, intnrhs, com plex *a, int lda, int *ipivot, com plex *b, int ldb, int *info);
void chesv_64 (charuplo, long n, long nms, com plex *a, long lda, long *ipivot, com plex *b, long ldb, long *info);

## PURPOSE

chesv com putes the solution to a com plex system of linear equations
$A * X=B, w h e r e A$ is an $N$ boy $N$ H erm titian $m$ atrix and $X$ and $B$ are N boy-N RH S m atrices.

The diagonalpivoting $m$ ethod is used to factorA as
$A=U * D * U * * H$, if $U P L O=U$ ', or
$A=L * D * L * * H$, if $U P L O=L^{\prime}$,
where U (orL) is a productofperm utation and unit upper (low er) triangular matrices, and D is H erm itian and block diagonalw th 1 boy-1 and 2 -by-2 diagonalblocks. The factored form of $A$ is then used to solve the system of equations A * X $=$ B .

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ : U Ppertriangle of $A$ is stored;
$=\mathbb{L}$ ': Low er triangle of A is stored.

N (input) The num ber of linearequations, i.e., the order of them atrix $A . N>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRH S $>=0$.

A (input/output)

On entry, the $H$ erm itian $m$ atrix $A$. If $U P L O=U$ ', the leading N -by N uppertriangularpart of A contains the upper triangular part of the $m$ atrix $A$, and the strictly low er triangularpart of A is not referenced. If $\mathrm{UPLO}=\mathrm{L}$ ', the leading N -by -N low er triangularpart ofA contains the low ertriangularpart of the matrix A, and the strictly upper triangular partofA is not referenced.

On exit, if $\mathbb{N F} F=0$, the block diagonalm atrix $D$ and the multipliers used to obtain the factor $U$ or L from the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}$ or $\mathrm{A}=$ L*D *L**H as com puted by CHETRF.
LDA (input)
The leading dim ension of the array A. LDA >= $\max (1, \mathbb{N})$.

IPIVOT (output)
D etails of the interchanges and the block structure ofD, as determ ined by CHETRF. If $\mathbb{P} \mathbb{V} O T(k)$ $>0$, then row $s$ and colum nsk and $\mathbb{P} \mathbb{V O T}(\mathrm{k})$ were interchanged, and $D(k, k)$ is a 1 -by -1 diagonal block. If UPLO $=\mathrm{U}$ 'and $\mathbb{P} \mathbb{I V O T}(\mathrm{k})=\mathbb{P} \mathbb{V} O T(k-1)$ $<0$, then row s and colum ns k-1 and -IP IV O T (k) w ere interchanged and $D(k-1 *, k-1 k)$ is a 2 -by-2 diagonal block. If UPLO = L' and $\mathbb{P} \mathbb{I V O T}(k)=$ $\mathbb{P I V O T}(k+1)<0$, then row $s$ and colum ns $k+1$ and $-\mathbb{P} \mathbb{I V O T}(k)$ w ere interchanged and $D(k \cdot k+1, k \cdot k+1)$ is a 2-by-2 diagonalblock.

B (input/output)
On entry, the $\mathrm{N}-\mathrm{by}-\mathrm{N}$ RH S righthand side m atrix B . On exit, if $\mathbb{N} F O=0$, the $N$ by-NRHS solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the array $B$. LD B $>=$ $\max (1, \mathbb{N})$.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length of O ORK. LDW ORK $>=1$, and for best perform ance LDW ORK $>=N * N B$, where $N B$ is the optim alblocksize forCHETRF.

If LD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of
the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LD W ORK is issued by XERBLA.
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N F O}=-$ i, the $i$-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N} F O=i, D(i, i)$ is exactly zero. The factorization has been com pleted, but the block diagonalm atrix $D$ is exactly singular, so the solution could notbe com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chesvx - use the diagonalpivoting factorization to com pute the solution to a com plex system of linearequations A * $\mathrm{X}=$ B,

## SYNOPSIS

```
SUBROUT\mathbb{NECHESVX (FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,B,}}\mathbf{N},\textrm{N},\textrm{N}
    LDB,X,LDX,RCOND,FERR,BERR,W ORK,LDW ORK,W ORK2,INFO)
```

CHARACTER * 1 FACT, UPLO
COM PLEX A (LDA,*),AF (LDAF,*), B (LDB,*),X (LDX,*),W ORK (*)
$\mathbb{N} T E G E R N, N R H S, L D A, L D A F, L D B, L D X, L D W O R K, \mathbb{N} F O$
$\mathbb{N}$ TEGER $\mathbb{P} \mathbb{I V O T}$ ( )
REAL RCOND

SU BROUTINE CHESVX_64 (FACT, UPLO,N,NRHS,A,LDA,AF,LDAF, $\mathbb{P} \mathbb{I V O T}$,
$B, L D B, X, L D X, R C O N D, F E R R, B E R R, W$ ORK,LDW ORK,W ORK 2, $\mathbb{N} F O$ )

CHARACTER * 1 FACT,UPLO
COM PLEX A (LDA,*),AF (LDAF,*), B (LDB,*), X (LDX,*), W ORK (*)
$\mathbb{N} T E G E R * 8 N, N R H S, L D A, L D A F, L D B, L D X, L D W O R K, \mathbb{N} F O$
$\mathbb{N}$ TEGER * $8 \mathbb{P} \mathbb{I V O T}$ ( ${ }^{*}$ )
REAL RCOND
REAL FERR ( ${ }^{*}$ ), BERR (*), $\mathrm{W} O \operatorname{OR} 2\left({ }^{*}\right)$

## F95 INTERFACE

SU BROUTINE HESVX (FACT,UPLO, $\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F]$, $\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LD} B], \mathrm{X},[\operatorname{LD} \mathrm{X}], R C O N D, F E R R, B E R R,[W O R K],[L D W O R K]$, [W ORK2], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1)::FACT,UPLO

COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A,AF,B,X
$\mathbb{N}$ TEGER ::N,NRHS,LDA,LDAF,LDB,LDX,LDW ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{V} O T$
REAL ::RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::FERR,BERR,W ORK 2

SU BROUTINE HESVX_64 (FACT,UPLO, $\mathbb{N}], \mathbb{N R H S ] , A , [ L D A ] , A F , [ L D A F ] , ~}$ IPIVOT,B,[LDB],X, [LDX],RCOND,FERR,BERR, [W ORK], [LDW ORK], [W ORK2], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::FACT,UPLO
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COMPLEX,D $\mathbb{M}$ ENSION (:,:) ::A,AF,B,X
$\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D A F, L D B, L D X, L D W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V} O T$
REAL ::RCOND
REAL,D $\mathbb{I M}$ ENSION (:) ::FERR,BERR,W ORK 2

## C INTERFACE

\#include <sunperfh>
void chesvx (char fact, charuplo, intn, int nrhs, com plex
*a, int lda, com plex *af, int ldaf, int *ipivot, com plex *b, int ldb, com plex *x, int ldx, float *rcond, float * ferr, float *berr, int *info);
void chesvx_64 (char fact, charuplo, long n, long nrhs, com plex *a, long lda, com plex *af, long ldaf, long *ịívot, com plex *b, long ldb, com plex *x, long ldx, float * rcond, float * ferr, float *berr, long *info);

## PURPOSE

chesvx uses the diagonal pivoting factorization to com pute the solution to a com plex system of linearequations A * $\mathrm{X}=$ B , where A is an N -by N H erm Hian m atrix and X and B are N -by-N RH S m atrices.

Errorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:

1. IfFACT $=\mathrm{N}$ ', the diagonal pivoting m ethod is used to factorA.
The form of the factorization is

$$
A=U * D * U * * H \text {, if } U P L O=U ' \text { or }
$$

$$
A=L * D * L * * H, \text { if } U P L O=L \prime,
$$

where $U$ (orL) is a product of perm utation and unit upper (low er)
triangularm atrices, and $D$ is $H$ erm itian and block diagonalw th
1-by-1 and 2-by-2 diagonalblocks.
2. If som eD $(i, i)=0$, so thatD is exactly singular, then the routine
retums w ith $\mathbb{N} F O=$ i. O therw ise, the factored form of $A$ is used
to estim ate the condition num ber of the $m$ atrix $A$. If the reciprocal of the condition num ber is less than $m$ achine precision,
$\mathbb{N} F O=N+1$ is retumed as a waming, but the routine stillgoes on
to solve for $X$ and com pute error bounds as described below.
3.The system of equations is solved for $X$ using the factored form of A.
4. Iterative refinem ent is applied to im prove the com puted solution
$m$ atrix and calculate error bounds and backw ard enror estim ates
for it.

## ARGUMENTS

FACT (input)
Specifies w hether ornot the factored form of A has been supplied on entry. = F ': On entry, A F and $\mathbb{P}$ IV OT contain the factored form of A. A, AF and $\mathbb{P} \mathbb{V O T}$ will not be modified. = N : The $m$ atrix A w ill be copied to A F and factored.

UPLO (input)
$=U$ : U ppertriangle ofA is stored;
= L': Low er triangle ofA is stored.
N (input) The num ber of linearequations, i.e., the order of them atrix $A . N>=0$.

NRHS (input)
The num ber of right hand sides, i.e., the num ber of collm ns of the m atrices B and X. NRHS >=0.

A (input) The $H$ erm itian $m$ atrix $A$. If $U P L O=U$ ', the leading $N$-by- N upper triangularpart of $A$ contains the upper triangularpart of the $m$ atrix $A$, and the strictly low ertriangularpartofA is not referenced. If $\mathrm{PLO}=\mathrm{L}$ ', the leading N by- N lower triangularpartofA contains the low er triangular partof the $m$ atrix $A$, and the strictly upper triangularpart of A is not referenced.

LD A (input)
The leading dim ension of the anay A. LDA >= max ( $1, \mathbb{N}$ ) .

AF (input/output)
If $F A C T=F$ ', then $A F$ is an inputargum entand on entry contains the block diagonalm atrix $D$ and the m ultipliers used to obtain the factorU orl from the factorization $A=U * D * U * * H$ orA $=L * D * L * * H$ as com puted by CHETRF .

If FA C T = N ', then AF is an output argum ent and on exit retums the block diagonalm atrix $D$ and the multipliers used to obtain the factorU or L from the factorization $A=U * D * U * * H$ or $A=$ L*D *L**H.

LDAF (input)
The leading dim ension of the array AF. LDAF >= $\max (1, N)$.

PIVOT (inputoroutput)
If $F A C T=F '$, then $\mathbb{P I V O T}$ is an input argum ent and on entry contains details of the interchanges and the block structure of D, as determ ined by CHETRF. If $\mathbb{P}$ IV OT $(k)>0$, then row sand colum ns k and $\mathbb{P} \mathbb{I V O T}(k)$ w ere interchanged and $D(k, k)$ is a 1 -by-1 diagonal block. If UPLO $=U^{\prime}$ and $\mathbb{P} \mathbb{I V} O T(k)=\mathbb{P} \mathbb{I V} O T(k-1)<0$, then row sand colum ns $\mathrm{k}-1$ and $-\mathbb{P} \mathbb{I V O T}(\mathrm{k})$ were interchanged and $\mathrm{D}(\mathrm{k}-$ $1 \mathrm{k}, \mathrm{k}-1 \mathrm{k})$ is a $2-$ by-2 diagonalblock. IfU PLO $=$ L'and $\mathbb{P} \mathbb{I V}$ OT $(k)=\mathbb{P} \mathbb{I V}$ OT $(k+1)<0$, then row $s$ and colum nsk+1 and - $\mathbb{P}$ IV OT (k) were interchanged and D ( $k: k+1, k: k+1$ ) is a 2 -by-2 diagonalblock.

IfFACT = $N$ ', then $\mathbb{P} \mathbb{I V O T}$ is an output argum ent and on exit contains details of the interchanges and the block structure of D, as determ ined by CHETRF.
$B$ (input) The $N$-by-N RH S righthand side $m$ atrix $B$.

LD B (input)
The leading dim ension of the aray B. LD B >= $\max (1, N)$.

X (output)
If $\mathbb{N} F O=0$ or $\mathbb{N} F O=N+1$, the $N$ boy $-N$ RH S solution $m$ atrix $X$.

LD X (input)
The leading dim ension of the array $\mathrm{X} . \mathrm{LDX}>=$ $\max (1, N)$.

## RCOND (output)

The estim ate of the reciprocal condition num ber of
the matrix A. IfRCOND is less than the m achine precision (in particular, if RCOND $=0$ ), the $m$ atrix is singular to $w$ orking precision. This condition is indicated by a retum code of $\mathbb{N}$ FO > 0 .

FERR (output)
The estim ated forw ard errorbound for each solution vectorX ( 1 ) the $j$ th colum $n$ of the solution $m$ atrix $X)$. If $X T R U E$ is the true solution comesponding to $\mathrm{X}(\mathcal{1}), \mathrm{FERR}(\mathcal{)}$ is an estim ated upperbound forthe $m$ agnitude of the largest ele$m$ entin ( $X(\mathcal{)})-X$ TRU $E$ ) divided by the $m$ agnitude of the largestelem entin X ( 7 ) . The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slightoverestim ate of the true error.

## BERR (output)

The com ponentw ise relative backw ard error of each solution vectorX (i) (ie., the sm allest relative change in any elem entof $A$ orB thatm akes $X$ ( 7 ) an exactsolution).

W ORK (w orkspace)
On exit, if $\mathbb{N F O}=0, W$ ORK (1) retums the optim al LDW ORK.

LD W ORK (input)
The length of W ORK. LDW ORK $>=2 \star N$, and for best perform ance LDW ORK $>=N * N B$, where $N B$ is the optim alblocksize forCHETRF.

If LD W ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK aray, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA .

W ORK 2 (w orkspace)
dim ension (N)
$\mathbb{N} F O$ (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N F O}=-$ i, the $i$-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N F O}=i$, and $i$ is
<= N : D (i,i) is exactly zero. The factorization has been completed but the factorD is exactly singular, so the solution and error bounds could not be com puted. RCOND $=0$ is retumed. $=N+1$ : D is nonsingular, butRCOND is less than machine precision, $m$ eaning that the $m$ atrix is singular to w orking precision. N evertheless, the solution and enror bounds are com puted because there are a num ber of situationsw here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

chetf2 -com pute the factorization of a com plex Herm itian $m$ atrix A using the Bunch K aufn an diagonalpivoting $m$ ethod

## SYNOPSIS



CHARACTER * 1 UPLO
COM PLEX A (LDA,*)
$\mathbb{N}$ TEGER N,LDA, $\mathbb{N} F O$
$\mathbb{N} T E G E R \mathbb{P} \mathbb{I}\left({ }^{*}\right)$
SU BROUTINE CHETF2_64 (UPLO, N, A, LDA, $\mathbb{P} \mathbb{I V}, \mathbb{N} F O$ )

CHARACTER * 1 UPLO
COM PLEX A (LDA,*)
$\mathbb{N} T E G E R * 8 N, L D A, \mathbb{N F O}$
$\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{I V}\left({ }^{*}\right)$

## F95 INTERFACE

SU BROUTINE HETF2 (UPLO, $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I}$, $[\mathbb{N} F O])$
CHARACTER (LEN=1) ::UPLO
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R:: N, L D A, \mathbb{N F O}$
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V}$
SU BROUTINE HETF2_64 (UPLO, $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I},[\mathbb{N} F O])$
CHARACTER (LEN=1)::UPLO
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER (8) :: $\mathrm{N}, \mathrm{LD} A, \mathbb{N}$ FO
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I}$

## C INTERFACE

\#include <sunperfh>
void chetf2 (charuplo, int n, com plex *a, int lda, int *ị̆̇v, int *info);
void chetf2_64 (charuplo, long n, com plex *a, long lda, long *ipiv, long *info);

## PURPOSE

chetf2 com putes the factorization of a com plex H erm itian $m$ atrix A using the Bunch-K aufm an diagonalpivoting $m$ ethod:

$$
A=U * D * U^{\prime} \text { or } A=L * D * L^{\prime}
$$

where $U$ (orL) is a product of perm utation and unit upper (low er) triangular m atrices, $U$ 'is the conjugate transpose of $U$, and $D$ is $H$ erm itian and block diagonalw ith 1-by-1 and 2-by-2 diagonalblocks.

This is the unblocked version of the algorithm , calling Level2 BLAS.

## ARGUMENTS

## UPLO (input)

Specifies w hether the upper or low er triangular part of the $H$ erm itian $m$ atrix $A$ is stored:
$=\mathrm{U}$ ': Upper triangular
$=\mathrm{L}$ ': Low ertriangular

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
On entry, the $H$ erm itian $m$ atrix A. If $\mathrm{U} P \mathrm{O}=\mathrm{U}$ ', the leading $n-b y-n$ uppertriangularpartofA contains the upper triangular part of the $m$ atrix $A$, and the strictly low er triangularpart of A is not referenced. If UPLO $=\mathbb{L}$ ', the leading $n-b y-n$ low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of $A$ is notreferenced.

On exit, the block diagonalm atrix D and the mul
tipliers used to obtain the factorU orl (see below for further details).

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, \mathbb{N})$.

IPIV (output)
D etails of the interchanges and the block structure ofD. If $\mathbb{P} \mathbb{I V}(k)>0$, then row $s$ and colum ns $k$ and $\mathbb{P} \mathbb{I V}(k)$ were interchanged and $D(k, k)$ is a 1 -by-1 diagonalblock. IfUPLO $=U$ 'and $\mathbb{P} \mathbb{I V}(k)$ $=\mathbb{P} \mathbb{I V}(k-1)<0$, then row $s$ and colum ns $k-1$ and $-\mathbb{P} \mathbb{I V}(k)$ w ere interchanged and $D(k-1 *, k-1 k)$ is a 2 -by-2 diagonalblock. IfUPLO $=\mathbb{L}$ 'and $\mathbb{P} \mathbb{I V}(k)$
$=\mathbb{P} \mathbb{I}(k+1)<0$, then row sand colum ns $k+1$ and $-\mathbb{P} \mathbb{V}(k)$ w ere interchanged and $D(k, k+1, k \mathrm{k}+1)$ is a 2-by-2 diagonalblock.
$\mathbb{N F O}$ (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N} F O=-\mathrm{k}$, the k -th argum enthad an illegalvalue $>0:$ if $\mathbb{N} F O=k, D(k, k)$ is exactly zero. The factorization has been com pleted, but the block diagonalm atrix $D$ is exactly singular, and division by zero w ill occur if it is used to solve a system of equations.

## FURTHER DETAILS

1-96-B ased on m odifications by
J. Lew is, B oeing C om puter Servioes C om pany
A. Petitet, C om puterScience D ept, U niv . of Tenn., K noxville, U SA

If $U P L O=U$ ', then $A=U * D * U '$, where
$U=P(n) \star U(n) * \ldots$... $(k) U(k) * \ldots$,
i.e., $U$ is a product of term $s P(k) * U(k)$, where $k$ decreases from $n$ to 1 in steps of 1 or 2 , and $D$ is a block diagonal $m$ atrix $w$ th 1 -by -1 and 2 -by-2 diagonalblocks $D(k) . P(k)$ is a perm utation $m$ atrix as defined by $\mathbb{P} \mathbb{I}(k)$, and $U(k)$ is a unituppertriangularm atrix, such that if the diagonal block D (k) is of orders ( $s=1$ or 2 ), then

```
    ( I v 0 ) k-s
U(k)=(0 I 0 ) s
    ( 0 0 I ) n-k
        k-s s n-k
```

If $s=1, D(k)$ overw rites $A(k, k)$, and $v$ overw rites $A(1 k-$ $1, k)$. If $s=2$, the uppertriangle of $(k)$ overw rites $A(k-$ $1, k-1), A(k-1, k)$, and $A(k, k)$, and $v$ overw rites $A(1 k-2, k-$ 1 k).

IfUPLO $=\mathrm{L}$ ', then $A=\mathrm{L} * \mathrm{D} * \mathrm{~L}$ ', where $L=P(1) \star L(1)^{\star} \ldots * P(k) \star L(k) \star \ldots$
ie., $L$ is a productofterm $s P(k) * L(k)$, where $k$ increases from 1 to n in steps of 1 or 2 , and D is a block diagonal $m$ atrix w ith 1 -by -1 and 2 -by -2 diagonalblocks $D(k) . P(k)$ is a perm utation $m$ atrix as defined by $\mathbb{P} \mathbb{V}(k)$, and $L(k)$ is a unitlow ertriangularm atrix, such that if the diagonal
block D (k) is of orders ( $s=1$ or 2 ), then
$\left(\begin{array}{llll}I & 0 & 0\end{array}\right) k-1$
$L(k)=\left(\begin{array}{lll}0 & I & 0\end{array}\right) s$
( 0 v I ) n-k-s+1
$\mathrm{k}-1$ s n-k-s+1

If $s=1, D(k)$ overw rites $A(k, k)$, and $v$ overw rites $A(k+1 m, k)$. If $s=2$, the low ertriangle ofD $(k)$ overw rites $A(k, k), A(k+1, k)$, and $A(k+1, k+1)$, and $V$ overw rites A $(k+2 m, k: k+1)$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

chetrd - reduce a com plex H erm itian m atrix A to real sym $m$ etric tridiagonal form $T$ by a unitary sim ilarity transfor$m$ ation

## SYNOPSIS

```
SUBROUT\mathbb{NECHETRD (UPLO,N,A,LDA,D,E,TAU,W ORK,LW ORK,INFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),TAU (*),W ORK (*)
\mathbb{NTEGER N,LDA,LW ORK, INFO}
REALD (*),E (*)
SU BROUT\mathbb{NE CHETRD_64(UPLO,N,A,LDA,D,E,TAU,W ORK,LW ORK,INFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),TAU (*),W ORK (*)
\mathbb{NTEGER*8N,LDA,LW ORK,INFO}
REALD (*),E (*)
```


## F95 INTERFACE

SU BROUTINE HETRD (UPLO, $\mathbb{N}], A,[L D A], D, E, T A U,[W O R K],[L W ~ O R K]$, [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1) ::UPLO
COMPLEX,D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : : A
$\mathbb{N} T E G E R:: N, L D A, L W$ ORK, $\mathbb{N} F O$
REAL,D IM ENSION (:) ::D,E

SU BROUTINE HETRD_64 (UPLO, $\mathbb{N}], A,[L D A], D, E, T A U,[W O R K],[L W ~ O R K]$,
[ $\mathbb{N}$ FO ])

CHARACTER (LEN=1) ::UPLO
COM PLEX ,D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
COM PLEX , D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER (8) :: N, LD A , LW ORK, $\mathbb{N} F O$
REAL,D $\mathbb{I}$ ENSION (:) :: D , E

## C INTERFACE

\#include <sunperfh>
void chetrd (charuplo, intn, com plex *a, int lda, float *d, float *e, com plex *tau, int *info);
void chetrd_64 (charuplo, long n, complex *a, long lda, float *d, float *e, com plex *tau, long *info);

## PURPOSE

chetrd reduces a com plex $H$ erm itian $m$ atrix $A$ to real sym m etric tridiagonal form T by a unitary sim ilarity transformation: Q ** H * $\mathrm{A} * \mathrm{Q}=\mathrm{T}$.

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': Upper triangle ofA is stored;
= L': Low er triangle ofA is stored.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input) On entry, the $H$ erm itian $m$ atrix $A$. If UPLO $=U \prime$, the leading N boy N uppertriangularpartofA contains the upper triangular part of the $m$ atrix $A$, and the strictly low er triangularpart of A is not referenced. IfUPLO $=\mathrm{L}$ ', the leading N -by -N low er triangularpart of A contains the low ertriangularpart of the $m$ atrix $A$, and the strictly upper triangularpart of A is notreferenced. On exit, if $\mathrm{UPLO}=\mathrm{U}$ ', the diagonal and first superdiagonalofA are overw ritten by the comesponding elem ents of the tridiagonalm atrix T , and the ele$m$ ents above the first superdiagonal, $w$ ith the array TAU, represent the unitary $m$ atrix $Q$ as a product of elem entary reflectors; ifU PLO = 'L', the diagonal and firstsubdiagonal of A are overw ritten by the comesponding elem ents of the tri-
diagonalm atrix T , and the elem ents below the first subdiagonal, $w$ ith the array TA $U$, represent the unitary $m$ atrix $Q$ as a product of elem entary reflectors. See FurtherD etails.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, N)$.

D (output)
The diagonal elem ents of the tridiagonalm atrix T : D (i) $=A(i, i)$.

E (output)
The off-diagonal elem ents of the tridiagonal $m$ atrix $T: E(i)=A(i, i+1)$ if $U P L O=U^{\prime}, E(i)=$ A (i+1,i) if UPLO = L'.
TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if $\mathbb{N F O}=0, W$ ORK ( 1 ) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array W ORK. LW ORK >=1. For optim um penform ance LW ORK $>=N * N B$, where $N B$ is the optim alblocksize.

If LW O RK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue

## FURTHER DETAILS

If U PLO $=\mathrm{U}$ ', them atrix Q is represented as a product of elem entary reflectors

$$
Q=H(n-1) \ldots H(2) H(1)
$$

Each H (i) has the form
$H(i)=I-\tan * V^{*} V^{\prime}$
where tau is a com plex scalar, and $v$ is a com plex vector w ith $\mathrm{v}(\mathrm{i}+1 \mathrm{n})=0$ and $\mathrm{v}(\mathrm{i})=1 ; \mathrm{v}(1: i-1)$ is stored on exit in

A ( $1: i-1, i+1$ ), and tau in TAU (i).

If $\mathrm{ULO}=\mathrm{L}$ ', them atrix Q is represented as a product of elem entary reflectors

$$
Q=H(1) H(2) \ldots H(n-1)
$$

Each H (i) has the form

$$
H(i)=I-\tan { }^{\star} v^{*} v^{\prime}
$$

where tau is a com plex scalar, and $v$ is a com plex vector w ith $\mathrm{v}(1: i)=0$ and $v(i+1)=1 ; v(i+2 \mathrm{n})$ is stored on exit in $A(i+2 m, i)$, and tau in TAU (i).

The contents ofA on exit are illustrated by the follow ing exam ples $w$ th $n=5$ :

```
ifUPLO = U ': ifUPLO = L ':
```

    ( d e v2 v3 v4 ) ( d
    )
( d e v3 v4 ) ( e d
)
( d e v4 ) (v1 e d
)
( d e ) ( v1 v2 e d
)
( d ) (v1 v2 v3 e d
)
where d and e denote diagonal and off-diagonal elem ents of $T$, and videnotes an elem ent of the vectordefining $H$ (i).

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

chetrf-com pute the factorization of a com plex Herm tian $m$ atrix A using the Bunch $-K$ aufn an diagonalpivoting $m$ ethod

## SYNOPSIS



CHARACTER * 1 UPLO
COM PLEX A (LDA, $\left.{ }^{\star}\right)$, W ORK ( $\left.{ }^{( }\right)$
$\mathbb{N}$ TEGER $N, L D A, L D W$ ORK, $\mathbb{N} F O$
$\mathbb{N T E G E R} \mathbb{P} \mathbb{I V O T}{ }^{( }$)
SU BROUTINE CHETRF_64 (UPLO ,N,A,LDA, $\mathbb{P} \mathbb{I V O T}, \mathrm{W}$ ORK, LDW ORK, $\mathbb{N} F O$ )

CHARACTER * 1 UPLO
COM PLEX A (LDA, $\left.{ }^{*}\right)$, W ORK ( $\left.{ }^{( }\right)$
$\mathbb{N}$ TEGER*8N,LDA,LDW ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{V} O T\left({ }^{*}\right)$

## F95 INTERFACE

SU BROUT $\mathbb{N} E$ HETRF (UPLO, $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V O T},[\mathbb{W}$ ORK ], [LDW ORK ], [ $\mathbb{N F O}])$
CHARACTER (LEN=1) ::UPLO
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R:: N, L D A, L D W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$
SU BROUTINE HETRF_64 (UPLO, $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I} O T,[\mathbb{W}$ ORK ], [LDW ORK ], [ $\mathbb{N} F \mathrm{FO}$ ])

CHARACTER (LEN=1) ::UPLO
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R(8):: N, L D A, L D W O R K, \mathbb{N} F O$
$\mathbb{N}$ TEGER (8),D $\mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V}$ OT

## C INTERFACE

\#include <sunperfh>
void chetrf(charuplo, int $n$, com plex *a, int lda, int *ipivot, int*info);
void chetrf_64 (charuplo, long n, com plex *a, long lda, long *ịívot, long *info);

## PURPOSE

chetrf com putes the factorization of a com plex Herm itian $m$ atrix $A$ using the $B$ unch- $K$ aufn an diagonalpivoting $m$ ethod. The form of the factorization is

$$
\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H} \text { or } \mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}
$$

where $U$ (orL) is a productof perm utation and unit upper (low er) triangular matrioes, and $D$ is H em itian and block diagonalw ith 1 -by-1 and 2 -by-2 diagonalblocks.

This is the blocked version of the algorithm, calling Level 3 BLAS.

## ARGUMENTS

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.
A (input/output)
O n entry, the $H$ erm tian m atrix A. If $\mathrm{PLO}=\mathrm{U}$ ', the leading N -by -N uppertriangularpartofA contains the upper triangular part of the $m$ atrix $A$, and the strictly low er triangularpartofA is not referenced. If $\mathrm{UPLO}=\mathrm{L}$ ', the leading N -by -N low er triangularpart ofA contains the low ertriangularpart of the m atrix A, and the strictly
upper triangularpartofA is not referenced.

On exit, the block diagonalm atrix D and the multipliers used to obtain the factorU orL (see below for further details).

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, N)$.

## IPIVOT (output)

D etails of the interchanges and the block structure of D. If $\mathbb{P I V O T}(k)>0$, then row sand columnsk and $\mathbb{P I V O T}(k)$ were interchanged and $D(k, k)$ is a $1-b y-1$ diagonalblock. If $U P L O=U^{\prime}$ and $\mathbb{P} \mathbb{V} O T(k)=\mathbb{P} \mathbb{V} O T(k-1)<0$, then row $s$ and colum ns $k-1$ and - $\mathbb{P I V O T}(k)$ were interchanged and D ( $k-1 * k, k-1 k)$ is a $2-b y-2$ diagonal block. If UPLO $=\mathrm{L}$ 'and $\mathbb{P I V O T}(k)=\mathbb{P} \mathbb{I V O T}(k+1)<0$, then row sand colum ns $k+1$ and $-\mathbb{P} \mathbb{V} O T(k)$ were interchanged and $D(k: k+1, k k+1)$ is a $2-b y-2$ diagonal block.

W ORK (w orkspace)
On exit, if $\mathbb{N F F O}=0, \mathrm{~W}$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length ofW ORK. LDW ORK >=1. Forbestperfor$m$ ance LDW ORK >=N *NB, where NB is the block size retumed by ㅍAENV .
$\mathbb{I N}$ FO (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the i-th argum ent had an illegalvalue
$>0:$ if $\mathbb{N} F O=\mathrm{i}, \mathrm{D}(\mathrm{i}, \mathrm{i})$ is exactly zero. The factorization has been com pleted, but the block diagonalm atrix $D$ is exactly singular, and division by zero w illoccur if it is used to solve a system ofequations.

## FURTHER DETAILS

If U PLO $=\mathrm{U}$ ', then $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}$ ', where
$U=P(n) \star U(n)^{\star} \ldots * P(k) U(k)^{\star} . . .$,
i.e., $U$ is a product of term $s P(k) * U(k)$, where $k$ decreases
from $n$ to 1 in steps of 1 or 2 , and $D$ is a block diagonal $m$ atrix $w$ ith 1 -by- 1 and 2 -by-2 diagonalblocks $D(k) . P(k)$ is a perm utation $m$ atrix as defined by $\mathbb{P} \mathbb{I V O T}(k)$, and $U(k)$ is a unit uppertriangularm atrix, such that if the diagonal
block $D(k)$ is of orders ( $s=1$ or 2 ), then

$$
\begin{aligned}
& \text { ( I v 0 ) k-s } \\
& U(k)=\left(\begin{array}{lll}
0 & I
\end{array}\right) s \\
& \text { ( } 0 \text { O I ) } n-k \\
& \mathrm{k}-\mathrm{s} \text { s } \mathrm{n}-\mathrm{k}
\end{aligned}
$$

If $s=1, D(k)$ overw rites $A(k, k)$, and $v$ overw rites $A(1 k-$ $1, k)$. If $s=2$, the uppertriangle of $(k)$ overw rites $A(k-$ $1, k-1), A(k-1, k)$, and $A(k, k)$, and $v$ overw rites $A(1 k-2, k-$ 1 k).

If $U P L O=L$ ', then $A=L * D * L '$, where
$L=P(1) \star L(1)^{\star} \ldots * P(k) \star L(k)^{\star} \ldots$,
ie., $L$ is a product ofterm $s P(k) * L(k)$, where $k$ increases from 1 to $n$ in steps of 1 or 2 , and $D$ is a block diagonal $m$ atrix $w$ ith 1 -by -1 and 2 -by-2 diagonalblocks $D(k) . P(k)$ is a perm utation $m$ atrix as defined by $\mathbb{P} \mathbb{I V} O T(k)$, and $L(k)$ is a unitlow ertriangularm atrix, such that if the diagonal
block D (k) is of orders ( $s=1$ or 2 ), then

$$
\begin{aligned}
& \text { ( I 0 0 ) k-1 } \\
& L(k)=\left(\begin{array}{lll}
0 & I & 0
\end{array}\right) s \\
& \text { ( } 0 \text { v I ) n-k-s+1 } \\
& \mathrm{k}-1 \text { s } \mathrm{n}-\mathrm{k}-\mathrm{s}+1
\end{aligned}
$$

If $s=1, D(k)$ overw rites $A(k, k)$, and $v$ overw rites $A(k+1 m, k)$. If $s=2$, the low ertriangle ofD $(k)$ overw rites $A(k, k), A(k+1, k)$, and $A(k+1, k+1)$, and $v$ overw rites $A(k+2 m, k: k+1)$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chetri-com pute the inverse of a com plex Herm itian indefinthe $m$ atrix $A$ using the factorization $A=U * D * U * * H$ orA $=$ L*D*L**H com puted by CHETRF

## SYNOPSIS


CHARACTER * 1 UPLO
COM PLEX A (LDA, $\left.{ }^{*}\right), \mathrm{W}$ ORK (*)
$\mathbb{N}$ TEGER $N, L D A, \mathbb{N} F O$
$\mathbb{N} T E G E R \mathbb{P} \mathbb{I} O T\left({ }^{( }\right)$
SU BROUTINE CHETRI_64 (UPLO,N,A,LDA, $\mathbb{P} \mathbb{I V O T}, \mathrm{W} O R K, \mathbb{N} F O$ )
CHARACTER * 1 UPLO
COM PLEX A (LDA, ${ }^{\star}$ ), W ORK ( $\left.{ }^{( }\right)$
$\mathbb{N} T E G E R * 8 N, L D A, \mathbb{N F O}$
$\left.\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{I V O T}{ }^{( }\right)$

## F95 INTERFACE

SUBROUTINE HETRI(UPLO, $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{N O T},[\mathbb{W} O R K],[\mathbb{N} F O])$

CHARACTER (LEN=1) ::UPLO
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D IM ENSION (:,:) ::A
$\mathbb{N} T E G E R:: N, L D A, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S T O N(:):: \mathbb{P} \mathbb{I V O T}$

SU BROUTINE HETRI_64 (UPLO, $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V O T},[\mathbb{W} O R K],[\mathbb{N} F O])$
CHARACTER (LEN=1)::UPLO

COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{I}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER (8) :: N, LDA, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{V} O T$

## C INTERFACE

\#include <sunperfh>
void chetri(char uplo, int n, com plex *a, int lda, int *ipivot, int*info);
void chetri_ 64 (charuplo, long n, com plex *a, long lda, long *ịívot, long *info);

## PURPOSE

chetricom putes the inverse of a com plex Herm itian indefinthe $m$ atrix $A$ using the factorization $A=U * D * U * * H$ orA $=$ L*D *L**H com puted by CHETRF .

## ARGUMENTS

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
$=U$ ': Uppertriangular, form is $A=U * D * U * * H$;
= L ': Low ertriangular, form is A = L *D *L** .

N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
O n entry, the block diagonalm atrix D and the m ultipliers used to obtain the factorU orL as com puted by CHETRF.

On exit, if $\mathbb{N} F O=0$, the ( H erm titian) inverse of the original matrix. If $\mathrm{UPLO}=\mathrm{U}$ ', the upper triangularpart of the inverse is form ed and the partofA below the diagonal is not referenced; if UPLO = L' the low er triangular part of the inverse is formed and the partofA above the diagonal is not referenced.

LD A (input)
The leading din ension of the aray A. LDA >= $\max (1, N)$.
$\mathbb{P I V O T}$ (input)
D etails of the interchanges and the block structure ofD as determ ined by CHETRF.

W ORK (w orkspace)
dim ension $\mathbb{N}$ )
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N} F O=-$ i, the $i$-th argum enthad an illegalvahue
$>0:$ if $\mathbb{N} F O=i, D(i, i)=0$; the $m$ atrix is singular and its inverse could not.be com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
chetrs-solve a system of linearequationsA*X = B w th a
complex Herm itian m atrix A using the factorization A =
U *D *U**H orA = L*D *L**H com puted by CHETRF
```


## SYNOPSIS



```
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LD B,*)
INTEGERN,NRHS,LDA,LDB,INFO
INTEGER \mathbb{PIVOT(*)}
SUBROUT\mathbb{NE CHETRS_64(UPLO,N,NRHS,A,LDA,\mathbb{PIVOT,B,LDB,INFO)}}\mathbf{(N,N}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LD B,*)
INTEGER*8N,NRHS,LDA,LDB,INFO
INTEGER*8 \mathbb{PIVOT (*)}
```

F95 INTERFACE
SUBROUTINE HETRS (UPLO, $\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I} O T, B,[L D B],[\mathbb{N} F O])$
CHARACTER (LEN=1) ::UPLO
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : A, B
$\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION(:)::\mathbb {P}\mathbb {O}OT}$
SU BROUTINE HETRS_64 (UPLO, $\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I V O T}, B,[L D B]$,
[ $\mathbb{N} F O$ ])
CHARACTER (LEN=1)::UPLO

COM PLEX , D $\mathbb{M}$ ENSION (: : : : : A , B
$\mathbb{N}$ TEGER (8) :: N , NRHS,LDA,LDB, $\mathbb{N}$ FO
$\mathbb{N} \operatorname{TEGER}$ (8), D $\mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$

## C INTERFACE

\#include <sunperfh>
void chetrs (charuple, intn, intnrhs, com plex *a, int lda, int *ipivot, com plex *b, int ldb, int *info);
void chetrs_64 (charuplo, long n, long nrhs, complex *a, long lda, long *ípívot, com plex *b, long lolo, long *info);

## PURPOSE

chetrs solves a system of linearequations $A * X=B$ with $a$ complex $H$ erm itian $m$ atrix $A$ using the factorization $A=$ $\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}$ orA $=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}$ com puted by CHETRF.

## ARGUMENTS

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
$=\mathrm{U}$ : U ppertriangular, form is $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}$;
$=\mathrm{L}:$ : Low ertriangular, form is $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}$.

N (input) The order of the matrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS $>=0$.

A (input) The block diagonalm atrix D and the m ultipliers used to obtain the factorU orL as com puted by CHETRF.

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, N)$.

IPIVOT (input)
D etails of the interchanges and the block structure ofD as determ ined by CHETRF.

B (input/output)

O $n$ entry, the right hand side $m$ atrix $B$. On exit, the solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the aray B. LD B >= $\max (1, N)$.
$\mathbb{I N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the $i$ th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

chgeqz -im plem enta single-shift version of the $Q Z$ m ethod for finding the generalized eigenvalues w (i)=ALPHA (i)BETA (i) of the equation $\operatorname{det}(\mathrm{A}-\mathrm{w}$ (i) B$)=0$ If $J 0 B=S$ ', then the pair $(A, B)$ is sim ultaneously reduced to Schurform (i.e., $A$ and $B$ are both upper triangular) by applying one unitary tranform ation (usually called Q) on the left and another (usually called $Z$ ) on the right

## SYNOPSIS

```
SUBROUTINE CHGEQZ (JOB,COM PQ,COMPZ,N, IOO, \mathbb{HI,A,LDA,B,LDB,}
    ALPHA,BETA,Q,LDQ,Z,LD Z,W ORK,LW ORK,RW ORK,INFO)
```

CHARACTER * $1 \mathrm{JOB}, \mathrm{COMPQ}, \mathrm{COMPZ}$
COM PLEX A (LDA,*), B (LDB,*), ALPHA (*), BETA (*), Q (LDQ ,*),
Z (LD Z, ${ }^{*}$ ), W ORK (*)
$\mathbb{N}$ TEGER N, $\mathbb{I L}, \mathbb{H} I, L D A, L D B, L D Q, L D Z, L W O R K, \mathbb{N} F O$
REAL RW ORK (*)
SU BROUTINE CHGEQZ_64 (JOB , COMPQ, COMPZ,N, $\mathbb{L} O, \mathbb{H} I, A, L D A, B, L D B$,
A LPHA, BETA, $Q, L D Q, Z, L D Z, W$ ORK, LW ORK,RWORK, $\mathbb{N} F O$ )
CHARACTER * $1 \mathrm{JOB}, \mathrm{COMPQ}, \mathrm{COMPZ}$
COMPLEX A (LDA,*), B (LDB,*), ALPHA (*), BETA (*), Q (LDQ , $\left.{ }^{*}\right)$,
Z (LD Z, ${ }^{\star}$ ), W ORK ( ${ }^{*}$ )
$\mathbb{N} T E G E R * 8 N, \mathbb{L} O, \mathbb{H} I, L D A, L D B, L D Q, L D Z, L W O R K, \mathbb{N} F O$
REAL RW ORK ( ${ }^{*}$ )

## F95 INTERFACE

SU BROUTINE HGEQZ (JOB,COMPQ,COMPZ, $\mathbb{N}], \mathbb{L} O, \mathbb{H} I, A,[L D A], B,[L D B]$, ALPHA,BETA, $Q$, [LDQ], $Z$, [LD Z], [W ORK], [LW ORK], [RW ORK], [ $\mathbb{N F O}])$

CHARACTER ( $L E N=1$ ) :: OB , COMPQ, COMPZ
COM PLEX ,D $\mathbb{I}$ ENSION (:) ::ALPHA,BETA, W ORK
COMPLEX , D $\mathbb{M}$ ENSION (: : : : : A, B, Q, Z
$\mathbb{N} T E G E R:: N, \Pi O, \mathbb{H} I, L D A, L D B, L D Q, L D Z, L W O R K, \mathbb{N} F O$
REAL,D $\mathbb{M} E N S I O N(:):: R W O R K$

SU BROUT INE H GEQ Z_64 (OB , COMPQ, COMPZ, $\mathbb{N}], \mathbb{L} O, \mathbb{H} I, A,[L D A], B$, $[L D B], A L P H A, B E T A, Q,[L D Q], Z,[L D Z],[W O R K],[L W O R K],[R W O R K]$, [ $\mathbb{N}$ FO ])

CHARACTER ( $L E N=1$ ) :: OB , COMPQ, COMPZ
COM PLEX , D $\mathbb{M} E N S I O N(:):: A L P H A, B E T A, W O R K$
COM PLEX, D $\mathbb{I M} E N S I O N(:,:): A, B, Q, Z$
$\mathbb{N}$ TEGER (8) :: N , $\mathbb{L} O, \mathbb{H} I, L D A, L D B, L D Q, L D Z, L W O R K, \mathbb{N} F O$
REAL,D $M$ ENSION (:) ::RW ORK

## C INTERFACE

\#include <sunperfh>
void chgeqz (char job, char com pq, char com pz, int n, int ilo, int ihi, com plex *a, int lda, com plex *b, int ldb, com plex *alpha, complex *beta, complex *q, int ldq, com plex * $z$, int ldz, int *info);
void chgeqz_64 (char job, char com pq, char com pz, long n,
long ilo, long ihi, com plex *a, long lda, com plex
*b, long ldb, com plex *alpha, com plex *beta, com -
plex *q, long ldq, complex *z, long ldz, long
*info);

## PURPOSE

chgeqz im plem ents a single-shiftversion of the $\mathrm{Q} Z \mathrm{~m}$ ethod
for finding the generalized eigenvalues
$w(i)=A L P H A(i) B E T A$ (i) of the equation $A$ are then A LPHA (1), $\ldots, A \operatorname{LPHA}(\mathbb{N})$, and ofB areBETA $(1), \ldots, B E T A \mathbb{N})$.

If $J O B=S$ 'and $C O M P Q$ and $C O M P Z$ are $V$ 'or 'I', then the unitary transform ations used to reduce ( $A, B$ ) are accum ulated into the arrays $Q$ and $Z$ s.t.:
(in) A (in) Z (in) ${ }^{\star}=\mathrm{Q}$ (out) A (out) Z (out)*

Ref: C B.M oler \& G N . Stew art, "A n A lgorithm for Generalized M atrixigenvalue Problem s", SIAM J. Num er. A nal, 10 (1973) „. 241-256.

## ARGUMENTS

JOB (input)
= E': com pute only A LPHA and BETA. A and B will
not necessarily be put into generalized Schur
form . $=5$ ': putA and B into generalized Schur
form, asw ellas com puting A LPHA and BETA.
COMPQ (input)
$=N^{\prime}$ : do notm odify Q .
$=\mathrm{V}$ ':m ultiply the array Q on the right by the conjugate transpose of the unitary tranform ation that is applied to the left side of A and B to reduce them to Schur form . = 'I': like COM PQ = V', except that Q w illbe initialized to the identity first.

COMPZ (input)
$=\mathrm{N}$ ': do notm odify Z .
= V ':m ultiply the array Z on the right by the unitary tranform ation that is applied to the right side of $A$ and $B$ to reduce them to Schur form . = I': like COM PZ=V', exœet thatZ w illbe initialized to the identity first.

N (input) The order of the m atrices $\mathrm{A}, \mathrm{B}, \mathrm{Q}$, and $\mathrm{Z} . \mathrm{N}>=0$.

## ㅍO (input)

It is assum ed that A is already upper triangular in row s and colum ns $1: \mathbb{H O}-1$ and $\mathbb{H} \mathrm{I}+1 \mathbb{N} .1<=\mathbb{L} \mathrm{O}$ $<=\mathbb{H} \mathrm{I}<=\mathrm{N}$, if $\mathrm{N}>0 ; \mathbb{H}=1$ and $\mathbb{H} \mathrm{I}=0$, if $\mathrm{N}=0$.

IH I (input)
It is assum ed that A is already upper triangular
in row sand colum ns 1: $\mathbb{H} O-1$ and $\mathbb{H} \mathrm{I}+1 \mathrm{~N} .1<=\mathbb{L O}$
$<=\mathbb{H} \mathrm{I}<=\mathrm{N}$, if $\mathrm{N}>0 ; \mathbb{H} \mathrm{O}=1$ and $\mathbb{H} \mathrm{I}=0$, if $\mathrm{N}=0$.
A (input) On entry, the N -by-N upper H essenberg m atrix A .
E lem ents below the subdiagonalm ustbe zero. If
JO B $=S$ ', then on exit A and B w ill have been sim ultaneously reduced to upper triangular form . If $\mathcal{O C B}=\mathrm{E}$ ', then on exitA w ill have been destroyed.

LD A (input)
The leading dim ension of the array A. LD A $>=m a x$ ( 1, N ).
$B$ (input) $O n$ entry, the $N$ boy $-N$ upper triangular $m$ atrix $B$. Elem ents below the diagonal m ust be zero. If JO $B=S$ ', then on exit $A$ and $B$ will have been
sim ultaneously reduced to upper triangular form . If $J O B=E$ ', then on exitB will have been destroyed.

LD B (input)
The leading dim ension of the array $B$. LD B $>=m a x$ ( $1, \mathrm{~N})$.

## A LPHA (output)

The diagonalelem ents of $A$ when the pair $(A, B)$ has been reduced to Schur form. ALPHA (i)ßETA (i) $i=1, \ldots, N$ are the generalized eigenvalues.
BETA (output)
The diagonalelem ents of $B$ w hen the pair $(A, B)$ has been reduced to Schur form. ALPHA (i) BETA (i) $\mathrm{i}=1, \ldots, \mathrm{~N}$ are the generalized eigenvalues. A and B are norm alized so thatBETA (1),...,BETA $\mathbb{N}$ ) are non-negative real num bers.

Q (input/output)
If $C O M P Q=N$ ', then $Q \mathrm{w}$ ill notbe referenced. If $\mathrm{COM} P Q=\mathrm{V}$ ' or ' I ', then the conjugate transpose of the unitary transform ationsw hich are applied to $A$ and $B$ on the leftw illbe applied to the array $Q$ on the right.

LD Q (input)
The leading dim ension of the array $Q . L D Q>=1$. If $C O M P Q=V$ 'or $I$ ', then $L D Q>=N$.

Z (input/output)
If COM PZ=N', then Z w ill notbe referenced. If $\mathrm{COMPZ}=\mathrm{V}$ 'or I', then the unitary transform ations which are applied to $A$ and $B$ on the rightw ill be applied to the amay $Z$ on the right.

LD $Z$ (input)
The leading dim ension of the array Z . LD $\mathrm{Z}>=1$. If COM PZ=V 'or 'I', then LD Z >= N .

W ORK (w orkspace)
On exit, if $\mathbb{N}$ FO >=0, W ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the array W ORK. LW ORK >= $\max (1, \mathbb{N})$.

If LW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim al size of
the W ORK amray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA .

RW ORK (w orkspace)
dim ension (N)
$\mathbb{N F O}$ (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N F O}=-$ i, the $i$-th argum enthad an illegalvalue
$=1, \ldots, N$ : the QZ iteration did not converge.
( $A, B$ ) is not in Schur form, butA LPHA (i) and BETA (i), $i=\mathbb{N} F O+1, \ldots, N$ should be correct. $=$ $\mathrm{N}+1, \ldots, 2 \star \mathrm{~N}$ : the shift calculation failed. ( $A, B$ ) is not in Schur form, but A LPH A (i) and BETA (i), $\mathrm{i}=\mathbb{N} F O-\mathrm{N}+1, \ldots, \mathrm{~N}$ should be comect. > 2*N : various "im possible" enrors.

## FURTHER DETAILS

W e assum e that com plex A BS w orks as long as its value is less than overflow .

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chpcon -estim ate the reciprocal of the condition num ber of a com plex H em itian packed $m$ atrix A using the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}$ orA $=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}$ com puted by CHPTRF

## SYNOPSIS



```
CHARACTER * 1 UPLO
COM PLEX A (*),W ORK (*)
INTEGER N,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}
REAL ANORM,RCOND
SUBROUT\mathbb{NECHPCON_64(UPLO,N,A,\mathbb{PIVOT,ANORM,RCOND,W ORK,INFO)}}\mathbf{N},\textrm{N},\textrm{N}
CHARACTER * 1 UPLO
COM PLEX A (*),W ORK (*)
INTEGER*8 N, INFO
INTEGER*8 \mathbb{PIVOT (*)}
REAL ANORM,RCOND
```


## F95 INTERFACE

```
SU BROUTINE HPCON (UPLO,N,A, \(\mathbb{P} \mathbb{I V O T}, A N O R M, R C O N D,[W O R K],[\mathbb{N} F O])\)
CHARACTER (LEN=1) ::UPLO
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::A,W ORK
\(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{V} O T\)
REAL ::ANORM,RCOND
SU BROUTINE HPCON_64 (UPLO,N,A, \(\mathbb{P} \mathbb{I} O T, A N O R M, R C O N D,[\mathbb{N} O R K],[\mathbb{N} F O])\)
```

CHARACTER (LEN=1) ::UPLO
COMPLEX,D $\mathbb{M} E N S I O N$ (:) ::A,W ORK
$\mathbb{N} T E G E R(8):: N, \mathbb{N F O}$
$\mathbb{N} T E G E R(8), D \mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$
REAL ::ANORM,RCOND

## C INTERFACE

\#include <sunperfh>
void chpcon (charuple, intn, com plex *a, int *ịìiot, float anorm, float *rcond, int*info);
void chpcon_64 (char uplo, long n, com plex *a, long *íivot, floatanorm, float *rcond, long *info);

## PURPOSE

chpcon estim ates the reciprocal of the condition num ber of a com plex $H$ erm itian packed $m$ atrix A using the factorization A $=U * D * U * * H$ orA $=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}$ com puted by CHPTRF.

A $n$ estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND $=1$ /(ANORM * norm (inv (A )) ).

## ARGUMENTS

UPLO (input)
Specifies w hether the details of the factorization are stored as an upper or low er triangularm atrix. $=\mathrm{U}$ ': U ppertriangular, form is $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}$;
$=L^{\prime}:$ Low ertriangular, form is $A=L * D * L * * H$.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input) The block diagonalm atrix D and the m ultipliers used to obtain the factorU orL as com puted by CH PTRF, stored as a packed triangularm atrix.

IPIVOT (input)
D etails of the interchanges and the block structure ofD as determ ined by CHPTRF.

ANORM (input)
The 1 -norm of the originalm atrix A.

## RCOND (output)

The reciprocal of the condition num ber of the
$m$ atrix $A$, com puted as RCOND = 1/(ANORM *A $\mathbb{N} V N M)$,
where $A \mathbb{N} V N M$ is an estim ate of the 1 -nom of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension ( $2 * \mathrm{~N}$ )
$\mathbb{N} F O$ (output)
= 0 : successfulexit
< 0: if $\mathbb{N}$ FO = -i, the i-th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chpev - com pute all the eigenvalues and, optionally, eigenvectors of a com plex H erm Hian $m$ atrix in packed storage

## SYNOPSIS

```
SUBROUT\mathbb{NE CHPEV (JOBZ,UPLO,N,A,W ,Z,LD Z,W ORK,W ORK 2, INFO)}
CHARACTER * 1 JOBZ,UPLO
COM PLEX A (*),Z (LD Z,*),W ORK (*)
\mathbb{NTEGER N,LD Z, INFO}
REALW (*),W ORK 2 (*)
SU BROUT\mathbb{NE CHPEV_64(OOBZ,UPLO,N,A,W ,Z,LD Z,W ORK,W ORK 2, INFO)}
CHARACTER * 1 OOBZ,UPLO
COM PLEX A (*),Z (LD Z,*),W ORK (*)
\mathbb{NTEGER*8N,LD Z,INFO}
REALW (*),W ORK2 (*)
```


## F95 INTERFACE

SU BROUTINE HPEV ( $\mathrm{O} \mathrm{BZ}, \mathrm{UPLO}, \mathrm{N}, \mathrm{A}, \mathrm{W}, \mathrm{Z},[\operatorname{LD} Z],[\mathbb{W}$ ORK ], [W ORK 2], [ $\mathbb{N} F \mathrm{~F}$ ])

CHARACTER (LEN=1):: JOBZ, UPLO
COMPLEX,D $\mathbb{M}$ ENSION (:) ::A,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::Z
$\mathbb{N}$ TEGER :: $\mathrm{N}, \mathrm{LD} \mathrm{Z}, \mathbb{N} F \mathrm{O}$
REAL,D $\mathbb{I}$ ENSION (:) ::W ,W ORK2

SU BROUTINE HPEV_64 (JOBZ, UPLO,N,A,W,Z,[LD Z], [W ORK],[WORK2], [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1)::JOBZ,UPLO
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::A,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::Z
$\mathbb{N} T E G E R(8):: N, L D Z, \mathbb{N} F O$
REAL,D $\mathbb{I}$ ENSION (:) ::W ,W ORK2

## C INTERFACE

\#include <sunperfh>
void chpev (char jobz, charuplo, intn, com plex *a, float ${ }^{*}$ w , com plex *z, int ldz, int *info);
void chpev_64 (char jobz, char uplo, long n, com plex *a, float *w , com plex *z, long ldz, long *info);

## PURPOSE

chpev com putes all the eigenvalues and, optionally, eigenvectors of a com plex H erm itian m atrix in packed storage.

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}$ ': C om pute eigenvalues only;
= V ': C om pute eigenvalues and eigenvectors.

UPLO (input)
= U ': U ppertriangle ofA is stored;
= 'L': Low er triangle ofA is stored.

N (input) The order of the m atrix A. $\mathrm{N}>=0$.
A (input/output)
O $n$ entry, the upper or low er triangle of the Her $m$ tian $m$ atrix A, packed colum nw ise in a linear array. The jth column of A is stored in the aray A as follows: if UPLO = U',A (i+ (j
 $(j-1) \star(2 * n-j / 2)=A(i, 7)$ for $j=i<=n$.

On exit, A is overw rilten by values generated during the reduction to tridiagonal form. If $\mathrm{PLO}=$ U ', the diagonal and first superdiagonal of the tridiagonal m atrix T overv rite the comesponding elem ents ofA, and if UPLO = L', the diagonal and first subdiagonal of $T$ overw rite the comesponding elem ents ofA.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.

Z (input) If $\mathcal{O} \mathrm{BZ}=\mathrm{V}$ ', then if $\mathbb{N} F O=0, \mathrm{Z}$ contains the orthonorm aleigenvectors of the $m$ atrix $A, w$ ith the i-th column of $Z$ holding the eigenvector associated with $W$ (i). If $J O B Z=N$ ', then $Z$ is not referenced.

LD $Z$ (input)
The leading dim ension of the array Z . LD $\mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD Z >= $\mathrm{max}(1, \mathrm{~N})$.
W ORK (w orkspace)
dim ension M AX ( $1,2 * \mathrm{~N}-1)$ )
W ORK 2 (w orkspace)
dim ension (max (1,3*N-2))
$\mathbb{I N} F O$ (output)
= 0 : successfulexit.
< 0: if $\mathbb{N}$ FO $=-$-i, the $i$-th argum ent had an illegalvalue.
>0: if $\mathbb{N} F O=$ i, the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chpevd - com pute all the eigenvalues and, optionally, eigenvectors of a com plex H erm itian m atrix A in packed storage

## SYNOPSIS

```
SU BROUT\mathbb{NE CHPEVD (OBZ,UPLO,N,AP,W,Z,LD Z,W ORK,LW ORK,RW ORK,}
    LRW ORK,INORK,LIN ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
COM PLEX AP (*),Z (LD Z ,*),W ORK (*)
INTEGER N,LDZ,LW ORK,LRW ORK,LIN ORK,NNFO
INTEGER IN ORK (*)
REALW (*),RW ORK (*)
SUBROUTINE CHPEVD_64(JOBZ,UPLO,N,AP,W ,Z,LD Z,W ORK,LW ORK,
    RW ORK,LRW ORK,INORK,LINORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
COM PLEX AP (*),Z (LD Z ,*),W ORK (*)
INTEGER*8N,LD Z,LW ORK,LRW ORK,LIN ORK,INFO
INTEGER*8 \mathbb{N ORK (*)}
REALW (*),RW ORK (*)
```


## F95 INTERFACE

SU BROUTINE HPEVD (JOBZ,UPLO,N,AP,W,Z, [LD Z], [W ORK ], [LW ORK], $[R W$ ORK ], [LRW ORK], [ $\mathbb{W}$ ORK ], [ $\mathbb{L} \mathbb{W}$ ORK ], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1)::JOBZ,UPLO
COM PLEX,D $\mathbb{M}$ ENSION (:) ::AP,W ORK
COM PLEX,D $\mathbb{I}$ ENSION (:,:) ::Z
$\mathbb{N} T E G E R:: N, L D Z, L W$ ORK,LRW ORK,LIN ORK, $\mathbb{N} F O$


SU BROUTINE HPEVD_64 (OBZ, UPLO ,N,AP, W , Z, [LD Z], [W ORK ], [LW ORK ], [RW ORK], [LRW ORK], [ $\mathbb{W}$ ORK], [LIW ORK], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) :: JOBZ, UPLO
COM PLEX,D $\mathbb{M}$ ENSION (:) ::AP,W ORK
COM PLEX, D $\mathbb{M}$ ENSION (:,:) :: Z
$\mathbb{N}$ TEGER (8) :: N, LD Z,LW ORK,LRW ORK,L $\mathbb{I W}$ ORK, $\mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}$ (8), D $\mathbb{M} \operatorname{ENSION(:):~:~\mathbb {N~ORK~}}$
REAL,D $\mathbb{I M}$ ENSION (:) ::W ,RW ORK

## C INTERFACE

\#include <sunperfh>
void chpevd (char jobz, charuple, intn, com plex *ap, float
${ }^{*} \mathrm{~W}$, com plex ${ }^{2} z$, int ldz, int *info);
void chpevd_64 (char jobz, charuplo, long n, complex *ap, float *w , com plex *z, long ldz, long *info);

## PURPOSE

chpevd com putes all the eigenvalues and, optionally, eigenvectors of a com plex H erm itian m atrix A in packed storage. Ifeigenvectors are desired, ituses a divide and conquer algorithm .

The divide and conquer algorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on m achines w ith a guard digit in add/subtract, or on those binary $m$ achines $w$ thout guard digits $w$ hich subtract like the $C$ ray $X$-M P , C ray Y M P , C ray C-90, orC ray- 2 . Itcould conceivably fail on hexadecim al or decim al $m$ achines $w$ ithout guard digits, butw e know of none.

## ARGUMENTS

```
JO B Z (input)
    = N ': C om pute eigenvalues only;
    = V ': C om pute eigenvalues and eigenvectors.
UPLO (input)
    = U ': U pper triangle ofA is stored;
    = L': Low ertriangle ofA is stored.
N (input) The order of them atrix A. N >=0.
```

AP (input/output)
O $n$ entry, the upper or low er triangle of the H er$m$ tian $m$ atrix A, packed colum nw ise in a linear array. The jth column of A is stored in the array AP as follows: ifUPLO = U',AP (i+ (j
 $+(j-1)^{\star}\left(2{ }^{*} n-j / 2\right)=A(i, 7)$ for $j=i<=n$.

On exit, AP is overw rilten by values generated during the reduction to tridiagonal form. If PLO $=U$ ', the diagonal and first superdiagonal of the tridiagonal $m$ atrix $T$ overw rite the corresponding elem ents ofA, and if UPLO = L', the diagonal and first subdiagonalof $T$ overw rite the corresponding elem ents ofA.
W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.

Z (input) If $\mathrm{JOBZ}=\mathrm{V}^{\prime}$, then if $\mathbb{N F O}=0, \mathrm{Z}$ contains the orthonorm aleigenvectors of the $m$ atrix $A, w$ ith the $i-t h$ colum $n$ of $Z$ holding the eigenvector associated w ith $W$ (i). If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced.

LD Z (input)
The leading dim ension of the array $\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD $Z>=\max (1, N)$.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of array $\mathrm{W} O R K$. If $\mathrm{N}<=1$, LW ORK must be at least1. If $\mathrm{JOBZ}=\mathrm{N}$ 'and $\mathrm{N}>$ 1, LW ORK m ust.be at least N . If $\mathrm{JOBZ}=\mathrm{V}$ 'and N $>1$, LW ORK m ustbe at least $2 \star_{\mathrm{N}}$.

If LW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

## RW ORK (w orkspace)

dim ension (LRW ORK) On exit, if $\mathbb{N} F O=0$, RW ORK (1)
retums the optim alLRW ORK .
LRW ORK (input)
The dim ension of aray RW ORK. If $\mathrm{N}<=1$,

LRW ORK m ustbe atleast1. If $\operatorname{OBZ}=N^{\prime}$ 'and $N>$ 1,LRW ORK m ustbe atleastN. If OBZ $=\mathrm{V}$ 'and $N$ $>1$, LRW ORK m ustbe at least $1+5 * N+2 * N * * 2$.

IfLRW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the RW ORK array, retums this value as the first entry of the RW ORK array, and no errorm essage related to LRW ORK is issued by X ERBLA.
$\mathbb{I W}$ ORK (w orkspace/output)
O n exit, if $\mathbb{N} F O=0, \mathbb{I N}$ ORK (1) retums the optim al LIW ORK.

LIW ORK (input)
The dim ension of array $\mathbb{I W}$ ORK. If JOBZ $=\mathrm{N}$ 'orN $<=1$, LIN ORK m ustbe at least1. If $\mathrm{OBZ}=\mathrm{V}^{\prime}$ and $N>1, L \mathbb{I N} O R K m$ ust.be at least $3+5 * N$.

If LIV ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the $\mathbb{I V}$ ORK array, retums this value as the first entry of the $\mathbb{I W}$ ORK anay, and no errorm essage related to $L \mathbb{I W}$ ORK is issued by XERBLA.
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the i-th argum enthad an illegalvalue.
$>0:$ if $\mathbb{N F O}=$ i, the algorithm failed to con-
verge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chpevx - com pute selected eigenvalues and, optionally, eigenvectors of a complex Herm inian matrix A in packed storage

## SYNOPSIS

```
SUBROUT\mathbb{NE CHPEVX (JOBZ,RANGE,UPLO,N,A,VL,VU,\mathbb{I},\mathbb{IU},ABTOL,}
    NFOUND,W,Z,LD Z,W ORK,W ORK2,IN ORK 3, \mathbb{FA}\mathbb{I},\mathbb{NNFO)}
CHARACTER * 1 JOBZ,RANGE,UPLO
COM PLEX A (*),Z (LD Z,*),W ORK (*)
```



```
\mathbb{NTEGER IN ORK3(*),\mathbb{FA LI (*)}}\mathbf{(})
REAL VL,VU,ABTOL
REALW (*),W ORK2 (*)
SUBROUT\mathbb{NE CHPEVX_64(JOBZ,RANGE,UPLO,N,A,VL,VU, IL,\mathbb{U},ABTOL,}
        NFOUND,W,Z,LD Z,W ORK,W ORK2,IN ORK 3, \mathbb{FA}\mathbb{I},\mathbb{NNFO)}
```

CHARACTER * 1 JOBZ,RANGE, UPLO
COM PLEX A (*), Z (LD Z , *), W ORK (*)
$\mathbb{N}$ TEGER* $8 \mathrm{~N}, \mathbb{I}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I V}$ ORK 3 (*), $\mathbb{F A} \mathbb{L}(*)$
REALVL,VU,ABTOL
REALW (*), W ORK 2 (*)

## F95 INTERFACE

SU BROUTINE HPEVX (JOBZ,RANGE,UPLO,N,A,VL,VU, $\mathbb{I}, \mathbb{I}, A B T O L$, $\mathbb{N} F O U N D], W, Z,[L D Z],[\mathbb{W}$ ORK $], \mathbb{W}$ ORK2], [IN ORK 3], $\mathbb{F} A \mathbb{I},[\mathbb{N} F O])$

CHARACTER (LEN=1):: OOBZ,RANGE,UPLO
COMPLEX,D $\mathbb{M}$ ENSION (:) ::A,W ORK

COM PLEX, D $\mathbb{M}$ ENSION (: : : : : Z
$\mathbb{N}$ TEGER :: N, $\mathbb{I}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{I N} O R K 3, \mathbb{F A} \mathbb{I}$
REAL ::VL,VU,ABTOL
REAL,D $\mathbb{M}$ ENSION (:) ::W ,W ORK2

SU BROUTINE HPEVX_64 (OBZ,RANGE, UPLO,N,A,VL,VU, $\mathbb{I}, \mathbb{U}, A B T O L$, $[\mathbb{N} F O U N D], W, Z,[L D Z],[W O R K],[W O R K 2],[\mathbb{W}$ ORK 3], $\mathbb{F} A \mathbb{L},[\mathbb{N} F O])$

CHARACTER ( $几 E N=1$ ) : : OBZ,RANGE, UPLO
COM PLEX,D $\mathbb{I M} E N S I O N(:):: A, W$ ORK
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : Z
$\mathbb{N}$ TEGER (8) :: N , $\mathbb{I}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N F O}$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} E N S I O N(:):: \mathbb{I N} O R K 3, \mathbb{F} A \mathbb{I}$
REAL ::VL,VU,ABTOL
REAL,D $\mathbb{M}$ ENSION (:) ::W ,W ORK2

## C INTERFACE

\#include <sunperfh>
void chpevx (char jobz, char range, charuplo, intn, com plex *a, float vl, float vu, intil, intiu, float abtol, int *nfound, float *w, com plex *z, int ldz, int*ifail, int*info);
void chpevx_64 (char j̀jbz, char range, char uplo, long n, complex *a, floatvl, floatvu, long il, long iu, floatabtol, long *nfound, float *W, complex *z, long ldz, long *ifail, long *info);

## PURPOSE

chpevx com putes selected eigenvalues and, optionally, eigenvectors of a com plex $H$ erm itian m atrix $A$ in packed storage. E igenvalues/vectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

## ARGUMENTS

JOBZ (input)
= N ': C om pute eigenvahues only;
$=\mathrm{V}:$ : C om pute eigenvalues and eigenvectors.

RANGE (input)
= A : : alleigenvalues willbe found;
= V $:$ :alleigenvalues in the half-open interval (NL,VU] will be found; = I': the $\mathbb{I}$-th through

IU -th eigenvaluesw illube found.

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L ': Low ertriangle ofA is stored.
N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
On entry, the upperor low ertriangle of the Her $m$ tian matrix A, packed collm nw ise in a linear array. The jth column of A is stored in the aray A as follows: if UPLO = U',A (i+ (j)
 $(j-1) *(2 * n-j / 2)=A(i, 7)$ for $\dot{j}=i<=n$.
On exit, A is overw ritten by values generated during the reduction to tridiagonal form. If UPLO $=$ U', the diagonaland firstsuperdiagonal of the tridiagonal m atrix T overw rite the comesponding elem ents ofA, and if U PLO = L', the diagonal and first subdiagonal of $T$ overw rite the corresponding elem ents ofA .

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE=A'or I'.

VU (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRA N G E = A' 'or I'.

II (input)
If RA N G E= 'I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{L}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N otreferenced ifRANGE= A 'or V'.

IU (input)
IfRA N G E= I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{L}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N otreferenced ifRANGE= A 'or V'.

ABTOL (input)
The absolute error tolerance for the eigenvalues. A $n$ approxim ate eigenvalue is accepted as converged
when it is determ ined to lie in an interval [a,b]
of w idth less than orequal to
ABTOL + EPS * max (k|, b|),
where EPS is them achine precision. If ABTOL is less than or equal to zero, then EPS* $\mid$ | w illbe used in its place, where $F \mid$ is the 1 -norm of the tridiagonal m atrix obtained by reducing A to tridiagonal form.

E igenvalues w illbe com puted m ostaccurately when ABTOL is set to tw ice the underflow threshold $2 *$ SLAM CH ( ${ }^{\prime}$ ), notzero. If this routine retums w ith $\mathbb{N}$ FO $>0$, indicating that som e eigenvectors did not converge, try setting ABTOL to $2 *$ SLAM CH (S ).

See "C om puting Sm allSingularV alues ofB idiagonal M atrices w ith G uaranteed H igh Relative A ccuracy," by $D$ em m eland $K$ ahan, LA PA CK $W$ orking $N$ ote \#3.

## NFOUND (output)

The total num ber of eigenvalues found. $0<=$ NFOUND <= N. IfRANGE = A',NFOUND = N, and if RANGE $=I^{\prime}$, NFOUND $=\mathbb{U}-\mathbb{L}+1$.

W (output)
If $\mathbb{N}$ FO $=0$, the selected eigenvalues in ascending order.
$Z$ (input) If $J O B Z=V^{\prime}$, then if $\mathbb{N F O}=0$, the first $N F O U N D$ colum ns of $Z$ contain the orthonorm aleigenvectors of them atrix A comesponding to the selected eigenvalues, $w$ ith the i-th colum $n$ of $Z$ holding the eigenvector associated w ith W (i). If an eigenvector fails to converge, then that colum n of $Z$ contains the latest approxim ation to the eigenvector, and the index of the eigenvector is retumed in FAII. If $J 0 B Z=N$ ', then $Z$ is not referenced. $N$ ote: the user must ensure that at least m ax (1,NFOUND) colum ns are supplied in the array $Z$; if RANGE = V', the exactvalue ofNFOUND is not know $n$ in advance and an upperbound $m$ ustbe used.

## LD Z (input)

The leading dim ension of the array $\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\max (1, N)$.

W ORK (w orkspace)
dim ension $(2 * N)$

W ORK 2 (w orkspace)
dim ension ( $7 * \mathrm{~N}$ )

IV ORK 3 (w orkspace)
dim ension ( $5 * \mathrm{~N}$ )
IFAII (output)
If $\mathrm{OBZ}=\mathrm{V}^{\prime}$, then if $\mathbb{N F O}=0$, the first NFOUND
elem ents of $\mathbb{F} A \mathbb{I}$ are zero. If $\mathbb{N F O}>0$, then
IFA II contains the indices of the eigenvectors that failed to converge. If $\mathrm{OBZ}=\mathrm{N}$ ', then $\mathbb{F} A \mathbb{I}$ is notreferenced.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N ~ F O ~}=-$ i, the i-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N} F O=i$, then ieigenvectors failed to converge. Their indices are stored in aray $\mathbb{F} A \mathbb{I}$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chpgst-reduce a com plex Herm itian-definite generalized eigenproblem to standard form, using packed storage

## SYNOPSIS

SU BROUTINE CHPGST (TYPE, UPLO, N, AP, BP, $\mathbb{N} F O$ )
CHARACTER * 1 UPLO
COM PLEX AP $\left(^{*}\right)$, BP ( ${ }^{*}$ )
$\mathbb{N} T E G E R \mathbb{T T} \mathrm{PE}, \mathrm{N}, \mathbb{N} F O$
SUBROUTINECHPGST_64 (TTYPE, UPLO, N, AP, BP, $\mathbb{N} F O$ )

CHARACTER * 1 UPLO
COM PLEX AP (*), BP (*)
$\mathbb{N} T E G E R * 8 \mathbb{I T Y P E}, N, \mathbb{N} F O$

## F95 INTERFACE

SUBROUTINE HPGST (TTYPE, UPLO, N,AP, BP, [ $\mathbb{N} F O$ ])
CHARACTER (LEN=1) ::UPLO
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::AP,BP
$\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, \mathbb{N} F O$
SUBROUTINE HPGST_64 (TTYPE, UPLO ,N,AP,BP, [ $\mathbb{N F F O}$ ])

CHARACTER (LEN=1) ::UPLO
COMPLEX,D $\mathbb{I}$ ENSION (:) ::AP,BP
$\mathbb{N} T E G E R(8):: \mathbb{T} Y \mathrm{PE}, \mathrm{N}, \mathbb{N} F O$

## C INTERFACE

\#include <sunperfh>
void chpgst(int itype, charuplo, intn, com plex *ap, com plex *bp, int*info);
void chpgst_64 (long itype, char uplo, long n, com plex *ap, com plex *bp, long *info);

## PURPOSE

chpgst reduces a complex H erm tian-definite generalized eigenproblem to standard form, using packed storage.

If ITY $\mathrm{PE}=1$, the problem is $\mathrm{A} * \mathrm{X}=\operatorname{lam} \mathrm{bda}^{*} \mathrm{~B}{ }^{*} \mathrm{x}$, and $A$ is overw ritten by inv $\left(U{ }^{* *} H\right) * A * \operatorname{inv}(U)$ or $\operatorname{inv}(\mathbb{L}) * A * \operatorname{inv}(\amalg * * H)$
If ITYPE $=2$ or 3 , the problem is $A * B * x=l a m$ bda* $x$ or $B * A * x=\operatorname{lam} b d a{ }^{*} x$, and $A$ is overw rilten by $U * A * U * * H$ or $\mathrm{L} * \star_{\mathrm{H}}{ }^{*} \mathrm{~A} * \mathrm{~L}$.

B m usthave been previously factorized as $\mathrm{U} * * \mathrm{H} * \mathrm{U}$ or $\mathrm{L} * \mathrm{~L} * * \mathrm{H}$ by CPPTRF.

## ARGUMENTS

ITYPE (input)
$=1$ : compute $\quad \operatorname{inv}(U * * H) * A * \operatorname{inv}(U)$ or
inv (L) ${ }^{\text {A } A * i n v ~(L * * H) ; ~}$
$=2$ or 3: com pute U *A $* \mathrm{U} * * \mathrm{H}$ or $\mathrm{L} * * \mathrm{H} * \mathrm{~A} * \mathrm{~L}$.
UPLO (input)
= U ': Uppertriangle ofA is stored and B is factored as $\mathrm{U}{ }^{* *} \mathrm{H} * \mathrm{U} ;=\mathrm{L}$ ': Low ertriangle ofA is stored and $B$ is factored as $L * L * * H$.

N (input) The order of the m atrices A and $\mathrm{B} . \mathrm{N}>=0$.
AP (input/output)
O $n$ entry, the upper or low er triangle of the H er$m$ itian matrix A, packed colum nw ise in a linear aray. The $j$ th colum n of A is stored in the array AP as follows: ifUPLO = U', AP (i+ (j 1) ${ }^{2} 2$ ) $=A(i, 7)$ for $1<=i<=j$ if $U P L O=L ', A P(i$ $+(j-1)^{*}(2 n-j / 2)=A(i, j)$ for $j=i<=n$.

Onexit, if $\mathbb{N F F O}=0$, the transform ed matrix, stored in the sam e form at as A.
$B P$ (input)
The triangular factor from the $C$ holesky factoriza-
tion of B, stored in the sam e form at as A, as retumed by CPPTRF.
$\mathbb{I N F O}$ (output)
= 0 : successfulexit
<0: if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
chpgv -com pute allthe eigenvalues and, optionally, the
eigenvectors of a com plex generalized H erm itian-definite
eigenproblem, of the form \(A * x=(l a m . b d a) * B * x, A * B x=(l a m ~ b d a){ }^{*} x\),
or * \(^{\mathrm{A}}\) * \(\mathrm{x}=(\operatorname{lam} . \mathrm{bda}){ }^{\mathrm{x}} \mathrm{x}\)
```


## SYNOPSIS

```
SU BROUT\mathbb{NE CHPGV (TTYPE,JOBZ,UPLO,N,A,B,W ,Z,LD Z,W ORK,W ORK2,}
    \mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
COM PLEXA (*),B (*),Z (LD Z,*),W ORK (*)
INTEGER ITYPE,N,LD Z,INFO
REALW (*),W ORK 2 (*)
SUBROUTINE CHPGV_64 (TTYPE,JOBZ,UPLO,N,A,B,W,Z,LD Z,W ORK,
    WORK2, \mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
COM PLEX A (*),B (*),Z (LD Z,*),W ORK (*)
\mathbb{NTEGER*8 ITYPE,N,LDZ,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REALW (*),W ORK2 (*)
```


## F95 INTERFACE

 [W ORK2], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1):: JOBZ, UPLO
COM PLEX,D $\mathbb{I}$ ENSION (:) ::A,B,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::Z
$\mathbb{N} T E G E R:: \mathbb{I T} Y P E, N, L D Z, \mathbb{N} F O$
REAL,D IM ENSION (:) ::W ,W ORK2

SU BROUTINE HPGV_64 (TTYPE, $\mathcal{O}$ BZ, UPLO , N, A, B, W, Z, [LD Z], [W ORK ], [W ORK 2], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1):: JOBZ, UPLO
COMPLEX,D $\mathbb{I M} E N S I O N(:):: A, B, W$ ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) :: Z
$\mathbb{N}$ TEGER (8) :: $\mathbb{T} Y$ PE, $N, L D Z, \mathbb{N F O}$
REAL,D $\mathbb{I M}$ ENSION (:) ::W ,W ORK2

## C INTERFACE

\#include < sunperfh>
void chpgv (int itype, char jobz, charuplo, int n, com plex
*a, com plex *b, float *w, com plex *z, int ldz, int
*info);
void chpgv_64 (long itype, char jobz, char uplo, long n, com plex *a, complex *b, float*w, com plex *z, long ldz, long *info);

## PURPOSE

chpgv com putes all the eigenvalues and, optionally, the eigenvectors of a com plex generalized H erm itian-definite eigenproblem, of the form $A * x=(l a m ~ b d a) * B * x$, $\mathrm{A} * \mathrm{~B} \mathrm{x}=(\operatorname{lam} . \mathrm{bda}){ }^{\star} \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=\left(\mathrm{lam}\right.$ bda) ${ }^{*} \mathrm{x}$. H ere A and B are assum ed to be $H$ erm tian, stored in packed form at, and $B$ is also positive definite.

## ARGUMENTS

ITYPE (input)
Specifies the problem type to be solved:
$=1: A{ }^{*} \mathrm{x}=(\operatorname{lam} \mathrm{bda}){ }^{\mathrm{B}}{ }^{*} \mathrm{x}$
$=2: \mathrm{A} * \mathrm{~B} * \mathrm{x}=\left(\operatorname{lam}\right.$ bda) ${ }^{*} \mathrm{x}$
$=3: B * A * X=(l a m ~ b d a){ }^{*} \mathrm{x}$

JOBZ (input)
$=N^{\prime}:$ C om pute eigenvalues only;
= V ': C om pute eigenvalues and eigenvectors.
UPLO (input)
$=\mathrm{U}$ : : U pper triangles of $A$ and $B$ are stored;
$=\mathrm{L}$ ': Low ertriangles of $A$ and $B$ are stored.

N (input) The order of the m atrioes A and $\mathrm{B} . \mathrm{N}>=0$.

A (input/output)
O $n$ entry, the upper or low er triangle of the Her $m$ itian $m$ atrix $A$, packed colum nw ise in a linear aray. The jth column of A is stored in the array A as follows: if UPLO $=\mathrm{U}^{\prime}, \mathrm{A}(i+(j$
 $(j-1)^{\star}(2 \star n-j / 2)=A(i, 7)$ for $j=i<=n$.

On exit, the contents ofA are destroyed.
B (input/output)
O n entry, the upper or low er triangle of the Her $m$ tian $m$ atrix $B$, packed colum nw ise in a linear anray. The jth column of $B$ is stored in the array B as follows: if UPLO $=\mathrm{U}$ ', $\mathrm{B}(\mathrm{i}+(j$ 1) $\star j 2$ ) $=\mathrm{B}(i, 7)$ for $1<=i<=j$ if $U P L O=L$ ', $B(i+$ $(j-1) *(2 * n-j / 2)=B(i, 7)$ for $j=i<=n$.

On exit, the triangular factor $U$ or $L$ from the Cholesky factorization $B=U * * H * U$ or $B=L * L * * H$, in the sam e storage form atas $B$.

W (output)
If $\mathbb{N}$ FO $=0$, the eigenvalues in ascending order.
$Z$ (input) If $J O B Z=V$ ', then if $\mathbb{N F O}=0, Z$ contains the m atrix Z of eigenvectors. The eigenvectors are norm alized as follow s : if ITYPE $=1$ or 2 , $Z * * H * B * Z=I$; if IT $Y P E=3, Z * * H * i n v(B) * Z=I$. If JO BZ $=N$ ', then $Z$ is not referenced.

## LD $Z$ (input)

The leading $d i m$ ension of the array $Z . L D Z>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD $Z>=\max (1, N)$.

W ORK (w orkspace)
dim ension M AX ( $1,2 * \mathrm{~N}-1)$ )
W ORK 2 (w orkspace)
dim ension MAX ( $1,3 * N-2)$ )
$\mathbb{N} F O$ (output)
$=0$ : successfulexit
<0: if $\mathbb{N N}$ FO = -i, the i-th argum ent had an illegalvahue
> 0: CPPTRF orCHPEV retumed an error code:
$<=\mathrm{N}:$ if $\mathbb{N F O}=\mathrm{i}, \mathrm{CH}$ PEV failed to converge; i
off-diagonalelem ents of an interm ediate tridiagonal form did not convergeto zero; > N : if $\mathbb{N} F O=$
$\mathrm{N}+\mathrm{i}$, for $1<=\mathrm{i}<=\mathrm{n}$, then the leading m inorof orderiofB is notpositive definite. The factorization of B could not be com pleted and no eigenvalues or eigenvectors w ere com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

chpgvd - com pute allthe eigenvalues and, optionally, the eigenvectors of a com plex generalized H erm itian-definite eigenproblem, of the form $\mathrm{A} * \mathrm{x}=\left(\mathrm{lam}\right.$ bda) ${ }^{\mathrm{B}} \mathrm{B}$ x, $\mathrm{A} * \mathrm{~B} x=\left(\operatorname{lam}\right.$ bda) ${ }^{\mathrm{x}}$, or $B{ }^{*} A * X=\left(l a m\right.$ bda) ${ }^{*} X$

## SYNOPSIS

```
SU BROUTINE CHPGVD (TTYPE,JOBZ,UPLO,N,AP,BP,W,Z,LD Z,W ORK,
    LW ORK,RW ORK,LRW ORK,IN ORK,LIN ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
COM PLEX AP (*),BP (*),Z (LD Z,*),W ORK (*)
```



```
INTEGER IN ORK (*)
REALW (*),RW ORK (*)
SUBROUTINE CHPGVD_64(TTYPE,NOBZ,UPLO,N,AP,BP,W ,Z,LD Z,W ORK,
        LW ORK,RW ORK,LRW ORK,INORK,LIN ORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
COM PLEX AP (*),BP (*),Z (LD Z,*),W ORK (*)
```



```
INTEGER*8 IN ORK (*)
REALW (*),RW ORK (*)
```


## F95 INTERFACE

SU BROUTINE HPGVD (ITYPE, JOBZ, UPLO ,N,AP, BP, W, Z, [LD Z], [W ORK], [LW ORK], RW ORK ], [LRW ORK], [IW ORK], [LINORK], [ $\mathbb{N} F O$ ])

COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::Z
$\mathbb{N} T E G E R$ :: $\mathbb{T Y}$ PE,N,LD Z,LW ORK,LRW ORK,LIW ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: $\mathbb{I N}$ ORK
REAL,D IM ENSION (:) ::W ,RW ORK

SU BROUTINE HPGVD_64 (TTYPE, JOBZ, UPLO ,N,AP,BP,W,Z, [LD Z], $\mathbb{W}$ ORK ], [LW ORK ], RW ORK ], [LRW ORK ], [IW ORK ], [LIN ORK ], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1)::JOBZ,UPLO
COM PLEX,D $\mathbb{M}$ ENSION (:) ::AP,BP,W ORK
COMPLEX,D $\mathbb{M}$ ENSION (: : : : : Z
$\mathbb{N}$ TEGER (8) :: ITYPE,N,LDZ,LW ORK,LRW ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N}$ TEGER (8),D $\mathbb{I M}$ ENSION (:) :: $\mathbb{I N}$ ORK
REAL,DIM ENSION (:) ::W ,RW ORK

## C INTERFACE

\#include <sunperfh>
void chpgvd (int itype, char jobz, char uplo, intn, com plex *ap, com plex *bp, float*w , com plex *z, int ldz, int*info);
void chpgvd_64 (long itype, char jobz, char uplo, long n, com plex *ap, complex *bp, float*w, com plex *z, long ldz, long *info);

## PURPOSE

chpgrd com putes all the eigenvalues and, optionally, the eigenvectors of a com plex generalized H erm itian-definite eigenproblem, of the form $A * x=(l a m ~ b d a) * B * x$, $\mathrm{A} * \mathrm{~B} x=(\operatorname{lam} \operatorname{bda}) * \mathrm{x}$, or $\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\mathrm{lam} . \mathrm{bda}) * \mathrm{x}$. H ere A and B are assum ed to be $H$ erm itian, stored in packed form at, and $B$ is also positive definite.
Ifeigenvectors are desired, ituses a divide and conquer algorithm .

The divide and conquer algorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on $m$ achines w ith a guard digit in add/subtract, or on those binary $m$ achines $w$ thout guard digits w hich subtract like the $C$ ray X M P , C ray Y M P , C ray C -90, orC ray-2. It could conceivably fail on hexadecim al or decim al machines $w$ ithout guard digits, butw e know of none.

## ARGUMENTS

ITYPE (input)

Specifies the problem type to be solved:
$=1: \mathrm{A}{ }^{*} \mathrm{x}=\left(\mathrm{lam}\right.$ bda) ${ }^{\mathrm{B}} \mathrm{B}_{\mathrm{x}}$
$=2: \mathrm{A} * \mathrm{~B} * \mathrm{x}=\left(\mathrm{lam}\right.$ bda) ${ }^{*} \mathrm{x}$
$=3: B{ }^{*} \mathrm{~A} * \mathrm{X}=\left(\right.$ lam bda) ${ }^{*} \mathrm{X}$

JOBZ (input)
$=\mathrm{N}^{\prime}$ : C om pute eigenvalues only;
$=\mathrm{V}$ : C om pute eigenvalues and eigenvectors.
UPLO (input)
$=U$ ': U ppertriangles of $A$ and $B$ are stored;
$=\mathrm{L}$ ': Low ertriangles of $A$ and $B$ are stored.
N (input) The order of the $m$ atrices $A$ and $B . N>=0$.
AP (input/output)
O $n$ entry, the upper or low er triangle of the H er-
$m$ itian matrix A, packed colum nw ise in a linear
array. The jth colum n of A is stored in the
array AP as follows: ifUPLO = U', AP (i+ (j

$\left.+(j-1)^{\star}(2 \star n-j) / 2\right)=A(i, 7)$ for $j=i<=n$.
On exit, the contents of AP are destroyed.

BP (input/output)
O $n$ entry, the upper or low ertriangle of the Her $m$ tian $m$ atrix $B$, packed colum nw ise in a linear array. The jth column of $B$ is stored in the array BP as follows: if UPLO $=\mathrm{U}^{\prime}, \mathrm{BP}(i+(j$
 $+(j-1)^{\star}(2 \star n-j / 2)=B(i, 7)$ for $j=i<=n$.

On exit, the triangular factor $U$ or $L$ from the Cholesky factorization $B=U * * H * U$ or $B=L * L * * H$, in the sam e storage form at as $B$.

W (output)
If $\mathbb{N}$ FO $=0$, the eigenvalues in ascending order.
$Z$ (input) If $\mathcal{O D B Z}=\mathrm{V}^{\prime}$, then if $\mathbb{N} F O=0, \mathrm{Z}$ contains the m atrix Z of eigenvectors. The eigenvectors are norm alized as follows: if ITYPE $=1$ or 2 , $Z * * H * B * Z=I$; if IT $Y P E=3, Z * * H * i n v(B) * Z=I$. If Jo $\mathrm{BZ}=\mathrm{N}$ ', then Z is not referenced.

LD $Z$ (input)
The leading $d i m$ ension of the array $Z . L D Z>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD $Z>=\mathrm{max}(1, N)$.

W ORK (w orkspace)

On exit, if $\mathbb{N F} F=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of array $\mathrm{W} O R \mathrm{~K}$. If $\mathrm{N}<=1$, LW ORK $>=1$. If $\mathrm{JOBZ}=\mathrm{N}$ 'and $N>1$, LW $O R K>=N$. If $\mathrm{OOBZ}=\mathrm{V}$ 'and $\mathrm{N}>1$, LW ORK $>=2 \star \mathrm{~N}$.

IfLW ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of theW ORK ancay, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

RW ORK (w orkspace)
On exit, if $\mathbb{N}$ FO $=0, R W$ ORK (1) retums the optim al LRW ORK.

LRW ORK (input)
The dimension of array RW ORK. If $\mathrm{N}<=1$, LRW ORK >=1. IfJOBZ $=N$ 'and $N>1$,LRW ORK $>=$ N. If JOBZ $=\mathrm{V}$ 'and $\mathrm{N}>1$,LRW ORK >=1 + 5*N + $2 * N * * 2$ 。

If LRW ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the RW ORK array, retums this value as the first entry of the RW ORK aray, and no enrorm essage related to LRW ORK is issued by X ERBLA.

IN ORK (w orkspace/output)
On exit, if $\mathbb{N} F O=0, \mathbb{I N}$ ORK (1) retums the optim al
LIN ORK.

LIV ORK (input)
The dim ension of array $\mathbb{I N}$ ORK. If JOBZ $=\mathrm{N}$ 'orN
$<=1$, LIV ORK >= 1. If JOBZ $=V$ 'and $N>1$, LIN ORK >= $3+5 * N$ 。

If $L \mathbb{I V} O R K=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the $\mathbb{I V}$ ORK array, retums this value as the first entry of the $\mathbb{I W} O R K$ aray, and no errorm essage related to $L \mathbb{I N} O R K$ is issued by X ERBLA.
$\mathbb{I N F O}$ (output)
= 0 : successfulexit
< 0: if $\mathbb{N}$ FO = -i, the i-th argum ent had an illegalvalue
>0: CPPTRF orCHPEVD retumed an enrorcode:
<= N: if $\mathbb{N F F O}=\mathrm{i}, \mathrm{CHPEVD}$ failed to converge; i off-diagonalelem ents of an interm ediate tridiagonal form did notconvergeto zero; > N : if $\mathbb{N} F O=$ $\mathrm{N}+\mathrm{i}$, for $1<=\mathrm{i}<=\mathrm{n}$, then the leading m inorof orderiofB is notpositive definite. The factorization of B could not be com pleted and no eigenvalues or eigenvectors w ere com puted.

## FURTHER DETAILS

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv. of K entucky, USA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

chpgrx - com pute selected eigenvalues and, optionally, eigenvectors of a com plex generalized $H$ erm itian-definite eigenproblem, of the form $\mathrm{A} * \mathrm{x}=\left(\mathrm{lam}\right.$ bda) ${ }^{\mathrm{B}} \mathrm{B}$ x, $\mathrm{A} * \mathrm{~B} x=\left(\operatorname{lam}\right.$ bda) ${ }^{\mathrm{x}}$, or $B * A * X=\left(l a m\right.$ bda) ${ }^{*} X$

## SYNOPSIS

```
SUBROUTINE CHPGVX (TTYPE,NOBZ,RANGE,UPLO,N,AP,BP,VL,VU,IL,
    \mathbb{U},ABSTOL,M,W ,Z,LDZ,W ORK,RW ORK,IN ORK,\mathbb{FA}\mathbb{I},\mathbb{N}FO)
CHARACTER * 1 JOBZ,RANGE,UPLO
COM PLEX AP (*),BP (*),Z (LD Z,*),W ORK (*)
\mathbb{NTEGER ITYPE,N,\mathbb{N},\mathbb{U},M,LDZ,INFO}
INTEGER IN ORK (*),\mathbb{FA}|(\mp@subsup{|}{}{*})
REALVL,VU,ABSTOL
REALW (*),RWORK (*)
SU BROUT\mathbb{NE CHPGVX_64 (TTYPE,JOBZ,RANGE,UPLO,N,AP,BP,VL,VU,IL,}
```


CHARACTER * 1 JOBZ,RANGE,UPLO
COM PLEX AP (*), BP (*), Z (LD Z,*), W ORK (*)
$\mathbb{N} T E G E R * 8 \mathbb{I T} Y P E, N, \mathbb{L}, \mathbb{U}, M, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I N} O R K(*), \mathbb{F A} \mathbb{L}(*)$
REALVL,VU,ABSTOL
REALW (*),RWORK (*)

## F95 INTERFACE

SU BROUTINE HPGVX (TTYPE, JOBZ,RANGE,UPLO,N,AP,BP,VL,VU, IL, $\mathbb{U}, A B S T O L, M, W, Z,[L D Z],[W O R K],[R W$ ORK ], [IW ORK], $\mathbb{F} A \mathbb{I}$, [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COM PLEX,D $\mathbb{I}$ ENSION (:) ::AP,BP,W ORK
COM PLEX, D $\mathbb{M}$ ENSION (: : : : : Z

$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: $\mathbb{I W}$ ORK, $\mathbb{F} A \mathbb{I}$
REAL ::VL,VU,ABSTOL
REAL,D $\mathbb{I M} E N S I O N(:):: W, R W O R K$

SU BROUTINE HPGVX_64 (TTYPE, JOBZ,RANGE, UPLO, N, AP, BP, VL, VU, $\mathbb{H}, \mathbb{U}, A B S T O L, M, W, Z,[L D Z],[W O R K],[R W$ ORK $],[\mathbb{W} O R K], \mathbb{F} A \mathbb{L}$, [ $\mathbb{N} \mathrm{FO}]$ )

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COMPLEX,D $\mathbb{M} E N S I O N(:):: A P, B P, W$ ORK
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : Z
$\mathbb{N}$ TEGER (8) :: ITYPE, $N, \mathbb{I}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, \mathbb{N}$ FO
$\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{I W} O R K, \mathbb{F A} \mathbb{I}$
REAL ::VL,VU,ABSTOL
REAL,D $\mathbb{I M}$ ENSION (:) ::W ,RW ORK

## C INTERFACE

\#include <sunperfh>
void chpgvx (int itype, char jंjbz, char range, charuple, int n, com plex *ap, com plex *bp, floatvl, floatvu, int il, int in, floatabstol, int $\star_{m}$, float ${ }^{*}$, com plex *z, int ldz, int *ifail, int *info);
void chpgvx_64 (long type, char jobz, char range, charuplo, long n, com plex *ap, com plex *bp, floatvl, float vu, long il, long in, float abstol, long *m, float ${ }^{*}{ }_{W}$, complex *z, long ldz, long *ifail, long *info);

## PURPOSE

chpgvx com putes selected eigenvalues and, optionally, eigenvectors of a com plex generalized $H$ erm itian-definite eigenproblem, of the form $A * x=(\operatorname{lam} \operatorname{bda}) * B * x, A * B x=(\operatorname{lam} b d a){ }^{*} x$, or $B * A * x=\left(l a m\right.$ bda) ${ }^{*} x$. H ere $A$ and $B$ are assum ed to be $H$ erm itian, stored in packed form at, and $B$ is also positive definite.
E igenvahues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

## ARGUMENTS

## ITYPE (input)

Specifies the problem type to be solved:
$=1: A{ }^{*} \mathrm{X}=\left(\mathrm{lam}\right.$ bda) ${ }^{\mathrm{B}} \mathrm{B}_{\mathrm{X}}$
$=2: \mathrm{A} * \mathrm{~B} * \mathrm{X}=\left(\mathrm{lam}\right.$ bda) ${ }^{*} \mathrm{X}$
$=3: B * A * X=(l a m ~ b d a){ }^{*} \mathrm{x}$
JOBZ (input)
$=N^{\prime}:$ C om pute eigenvalues only;
$=\mathrm{V}$ : C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvalues w ill be found;
= V ::alleigenvalues in the half-open interval
( $\mathrm{NL}, \mathrm{VU}$ ] will be found; = I ': the I -th through $\mathbb{I U}$-th eigenvaluesw illbe found.
UPLO (input)
$=\mathrm{U}$ ': U ppertriangles of $A$ and $B$ are stored;
$=\mathrm{L}$ ': Low ertriangles of $A$ and $B$ are stored.
N (input) The order of the m atrices A and $\mathrm{B} . \mathrm{N}>=0$.

AP (input/output)
On entry, the upper or low ertriangle of the Her $m$ itian matrix A, packed colum nw ise in a linear array. The $j$ th colum $n$ of $A$ is stored in the aray AP as follows: ifUPLO = U',AP (i+ (j

$\left.+(j-1)^{\star}(2 \star n-j) / 2\right)=A(i, 7)$ for $j<=i<=n$.
On exit, the contents ofAP are destroyed.

BP (input/output)
O $n$ entry, the upper or low ertriangle of the Her -
$m$ itian matrix B, packed colum nw ise in a linear
array. The jth colum n of $B$ is stored in the array BP as follows: if UPLO $=\mathrm{U}, \mathrm{BP}(\mathrm{i}+(j$
$1) \star j 2)=\mathrm{B}(i, 7$ for $1<=i<=j$ if $U P L O=\mathrm{L}, \mathrm{BP}(i$
$+(j-1)^{\star}(2 \star n-j / 2)=B(i, j)$ for $j=i<=n$.
On exit, the triangular factor $U$ or $L$ from the
Cholesky factorization $B=U * * H * U$ orB $=\mathrm{L} * \mathrm{~L} * * \mathrm{H}$,
in the sam e storage form atas $B$.
VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE=A'or I'.

IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE = A' 'or I'.

II (input)
IfRA N G E= I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{L}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N otreferenced ifRANGE= A 'or V'.
IU (input)
IfRA N G E= I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}$, if $\mathrm{N}>0 ; \mathbb{L}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N otreferenced ifRANGE= A 'or V'.

ABSTOL (input)
The absolute error tolerance for the eigenvalues. A $n$ approxim ate eigenvalue is accepted as converged when it is determ ined to $l i e$ in an interval $[a, b]$ of w idth less than orequal to

ABSTOL + EPS * max ( $\mid$ |, $\mid$ |) ,
where EPS is them achine precision. IfABSTOL is less than or equal to zero, then EPS* $\mid$ |w illbe used in its place, where $T$ | is the 1-norm of the tridiagonalm atrix obtained by reducing AP to tridiagonal form .

E igenvalues w illbe com puted m ost accurately when ABSTOL is set to tw ioe the underflow threshold $2 *$ SLAM CH ( $\mathrm{S}^{7}$ ), notzero. If this routine retums w ith $\mathbb{N}$ FO >0, indicating that som e eigenvectors did not converge, try setting ABSTOL to $2 *$ SLAM CH (S ).

M (output)
The total num berofeigenvalues found. $0<=\mathrm{M}$ <= N . IfRANGE $=A \prime$ ', $\mathrm{M}=\mathrm{N}$, and ifRANGE $=\mathrm{I}^{\prime}, \mathrm{M}=$ $\mathbb{U}-\mathbb{L}+1$.

W (output)
O n norm alexit, the firstM elem ents contain the selected eigenvalues in ascending order.
$Z$ (input) If $\mathrm{OO} \mathrm{BZ}=\mathrm{N}$ ', then Z is not referenced. If JO BZ
$=\mathrm{V}$ ', then if $\mathbb{N} F O=0$, the firstM colum ns of $Z$ contain the orthonorm aleigenvectors of the $m$ atrix A comesponding to the selected eigenvalues, w ith
the i-th colum $n$ of $Z$ holding the eigenvectorassociated w ith W (i). The eigenvectors are norm alized as follow s : if $I T Y \mathrm{PE}=1$ or $2, \mathrm{Z} * * \mathrm{H} * \mathrm{~B} * \mathrm{Z}=\mathrm{I}$; if ITYPE $=3, Z * * H * \operatorname{inv}(B) * Z=I$.

If an eigenvector fails to converge, then that colum $n$ of $Z$ contains the latestapproxim ation to the eigenvector, and the index of the eigenvector is retumed in $\mathbb{F A} \mathbb{I}$. N ote: the userm ustensure that at leastm ax $(1, M)$ ) 0 lum ns are supplied in the array $Z$; ifRANGE = $V$ ', the exactvalue of $M$ is not know $n$ in advance and an upper bound $m$ ust be used.

LD $Z$ (input)
The leading dim ension of the array $Z$. LD $Z>=1$, and if $\mathrm{OB} \mathrm{B}=\mathrm{V}$ ', LD Z $>=\max (1, \mathbb{N})$.

W ORK (w orkspace)
dim ension ( $2 \star \mathrm{~N}$ )

RW ORK (w orkspace)
dim ension ( $7 * \mathrm{~N}$ )

IW ORK (w orkspace)
dim ension ( $5 * \mathrm{~N}$ )

IFA II (output)
If $J 0 B Z=V$ ', then if $\mathbb{N} F O=0$, the firstM ele$m$ ents of $\mathbb{F} A \mathbb{I}$ are zero. If $\mathbb{N} F O>0$, then $\mathbb{F} A \mathbb{I}$ contains the indices of the eigenvectors that failed to converge. If $\mathrm{JOBZ}=\mathrm{N}$ ', then $\mathbb{F} A \mathbb{I}$ is not referenced.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{N N}$ FO $=-$ i, the i-th argum enthad an illegalvalue
> 0: CPPTRF orCH PEVX retumed an errorcode:
$<=\mathrm{N}:$ if $\mathbb{N} F \mathrm{O}=\mathrm{i}, \mathrm{CHPEVX}$ failed to converge; i eigenvectors failed to converge. Their indices are stored in amray $\mathbb{F} A \mathbb{I} .>N$ : if $\mathbb{N} F O=N+$ $i$, for $1<=i<=n$, then the leading $m$ inorof orderiofB is notpositive definite. The factorization of B could not be com pleted and no eigenvalues or eigenvectors w ere com puted.

## FURTHER DETAILS

M ark Fahey, D epartm entofM athem atics, U niv. ofK entucky, USA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

## chpm $v$-perform the $m$ atrix-vector operation $y:=a l p h a * A * x$

+ beta*y


## SYNOPSIS

```
SUBROUT\mathbb{NE CHPMV (UPLO,N,ALPHA,A,X, INCX,BETA,Y,INCY)}
CHARACTER * 1 UPLO
COM PLEX ALPHA,BETA
COM PLEX A (*),X (*),Y (*)
\mathbb{NTEGERN,\mathbb{NCX,INCY}}\mathbf{N}=\mp@code{N}
SU BROUT\mathbb{NE CHPM V_64 (UPLO,N,ALPHA,A ,X , INCX,BETA,Y,INCY)}
CHARACTER * 1 UPLO
COM PLEX ALPHA,BETA
COM PLEX A (*),X (*),Y (*)
INTEGER*8N,\mathbb{NCX,INCY}
```


## F95 INTERFACE




```
CHARACTER (LEN=1) ::UPLO
COMPLEX ::ALPHA,BETA
COM PLEX,D IM ENSION (:) ::A,X,Y
\mathbb{NTEGER ::N,\mathbb{NCX,}\mathbb{N}CY}\\mp@code{M}
```



```
CHARACTER (LEN=1) ::UPLO
COMPLEX ::ALPHA,BETA
COM PLEX,D IM ENSION (:) ::A,X,Y
```

$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{I N C X}, \mathbb{N} C Y$

## C INTERFACE

\#include <sunperfh>
void chpm v (charuplo, intn, com plex *alpha, complex *a, com plex *x, int incx, com plex *beta, com plex *y, intincy);
void chpm v_64 (char uplo, long n, com plex *alpha, com plex *a, com plex *x, long incx, com plex *beta, com plex *y, long incy);

## PURPOSE

chpm $v$ perform $s$ the $m$ atrix-vector operation $y:=a l p h a * A * x+$ beta* $y$ where alpha and beta are scalars, $x$ and $y$ are $n$ ele$m$ ent vectors and $A$ is an $n$ by $n$ herm itian $m$ atrix, supplied in packed form .

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the $m$ atrix A is supplied in the packed array A as follow s:

UPLO = U'or U ' The uppertriangularpartofA is supplied in A.

UPLO = L'or I' The low ertriangularpartofA is supplied in A.

U nchanged on exit.

N (input)
O n entry, N specifies the order of the m atrix A . $\mathrm{N}>=0$. U nchanged on exit.

A LPH A (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
$\left(\left(\mathrm{n}^{*}(\mathrm{n}+1)\right) / 2\right)$. Before entry w ith UPLO $=$ U' or L', the array A mustcontain the upper triangularpartof the herm itian matrix packed
sequentially, column by colum $n$, so that A (1) containsa(1,1), A (2) and A (3) contain a( 1,2 ) and a( 2,2 ) respectively, and so on. Before entry w ith UPLO = L'or I', the array A m ust contain the low er triangularpart of the her$m$ tiian matrix packed sequentially, column by colum n, so thatA (1) contains a (1,1), A (2) and $A(3)$ contain a $(2,1)$ and a $(3,1)$ respectively, and so on. N ote that the im aginary parts of the diagonalelem ents need notbe set and are assum ed to be zero. U nchanged on exit.

X (input)
$(1+(n-1) * a b s(\mathbb{N} C X))$. Before entry, the increm ented array $X \mathrm{~m}$ ust contain the n elem ent vectorx. U nchanged on exit.
$\mathbb{N} C X$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents ofX. $\mathbb{N C X}$ <> 0 . U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen
BETA is supplied as zero then $Y$ need notbe seton input. U nchanged on exit.

Y (input/output)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))$. Before entry, the increm ented array $Y$ must contain the $n$ elem ent vectory. On exit, Y is overw rilten by the updated vectory.
$\mathbb{N} C Y$ (input)
O n entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents ofY. $\mathbb{N} C Y$ <> 0 . U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chpr-perform the herm itian rank 1 operation $\mathrm{A}:=$ alpha*x*conjg ( $x^{\prime}$ ) + A

## SYNOPSIS

```
SUBROUT\mathbb{NE CHPR(UPLO,N,ALPHA,X,\mathbb{NCX,A)}}\mathbf{N}\mathrm{ (UN}
CHARACTER * 1 UPLO
COM PLEX X (*),A (*)
\mathbb{NTEGER N,}\mathbb{N}CX
REAL A LPHA
SU BROUTINE CHPR_64(UPLO,N,ALPHA,X, NNCX,A)
CHARACTER * 1 UPLO
COM PLEX X (*),A (*)
INTEGER*8 N, INCX
REALALPHA
F95 INTERFACE
SUBROUT\mathbb{NE HPR (UPLO, NN ],ALPHA,X, [\mathbb{NCX ],A)}}\mathbf{(})=
CHARACTER (LEN=1) ::UPLO
COM PLEX,D IM ENSION (:) ::X,A
\mathbb{NTEGER::N,\mathbb{NCX}}\mathbf{}=1
REAL ::ALPHA
SU BROUT\mathbb{NE HPR_64 (UPLO, N ],ALPHA, X , [NCX ],A)}
CHARACTER (LEN=1) ::UPLO
COM PLEX,D IM ENSION (:) ::X,A
INTEGER (8)::N,\mathbb{NCX}
```

REAL ::ALPHA

## C INTERFACE

\#include <sunperfh>
void chpr(charuplo, intn, float alpha, complex *x, int incx, com plex *a);
void chpr_64 (charuplo, long n, float alpha, com plex *x, long incx, com plex *a);

## PURPOSE

chpr perform s the herm titian rank 1 operation $A:=$ alpha* $x^{*}$ con $\bar{j}\left(x^{\prime}\right)+A$ where alpha is a realscalar, $x$ is an $n$ elem entvector and $A$ is an $n$ by $n$ herm titian $m$ atrix, supplied in packed form .

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangularpart of the $m$ atrix $A$ is supplied in the packed array A as follow s:
$\mathrm{UPLO}=\mathrm{U}$ 'or $\mathrm{L}^{\prime}$ The uppertriangularpartofA is supplied in A.

UPLO = L 'or I' The low ertriangularpart of A is supplied in A.

U nchanged on exit.

N (input)
O n entry, $N$ specifies the order of the $m$ atrix A. $\mathrm{N}>=0$. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
$(1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C X))$. Before entry, the increm ented array $X$ must contain the $n$ elem ent vectorx. U nchanged on exit.

On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents ofX. $\mathbb{N} C X<>0$. U nchanged on exit.

A (input/output)
$\left(\left(n^{*}(n+1)\right) / 2\right)$. Before entry $w$ ith UPLO $=$ U' or L', the array A mustcontain the upper triangularpart of the herm itian matrix packed sequentially, column by colum n, so thatA (1) containsa(1,1), A (2) and A (3) contain a( 1,2 ) and a ( 2,2 ) respectively, and so on. On exit, the amay A is overw ritten by the uppertriangular partof the updated $m$ atrix. Before entry w ith UPLO = L 'or I', the array A m ust contain the low er triangularpart of the herm itian $m$ atrix packed sequentially, colum $n$ by collm $n$, so that A ( 1 ) contains a ( 1,1 ), A (2) and A (3) contain $a(2,1)$ and $a(3,1)$ respectively, and so on. On exit, the array A is overw rilten by the low er triangularpart of the updated $m$ atrix. N ote that the im aginary parts of the diagonalelem ents need notbe set, they are assum ed to be zero, and on exit they are set to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chpr2 -perform the Herm tian rank 2 operation $A:=$ alpha*x*conjg( $y^{\prime}$ ) + conjg (alpha ) ${ }^{\star} y^{\star}$ conjg $\left(x^{\prime}\right)+$ A

## SYNOPSIS

```
SUBROUTINE CHPR2(UPLO,N,ALPHA,X,\mathbb{NCX,Y,INCY,AP)}
CHARACTER * 1 UPLO
COM PLEX ALPHA
COM PLEX X (*),Y (*),AP (*)
```



```
SU BROUT\mathbb{NE CHPR2_64 (UPLO,N,ALPHA,X, NNCX,Y,INCY,AP)}
CHARACTER * 1 UPLO
COM PLEX ALPHA
COM PLEX X (*),Y (*),AP (*)
INTEGER*8N,\mathbb{NCX,INCY}
```


## F95 INTERFACE

```
SU BROUTINE HPR2 (UPLO, \(\mathbb{N}], A L P H A, X,[\mathbb{N} C X], Y,[\mathbb{N C Y}], A P)\)
CHARACTER (LEN=1) ::UPLO
COMPLEX ::ALPHA
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::X,Y,AP
\(\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N C Y}\)
SU BROUTINE HPR2_64 (UPLO, \(\mathbb{N}], A L P H A, X,[\mathbb{N} C X], Y,[\mathbb{N} C Y], A P)\)
CHARACTER (LEN=1) ::UPLO
COMPLEX ::ALPHA
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::X,Y,AP
```

$\mathbb{N} T E G E R(8):: N, \mathbb{N} C X, \mathbb{N} C Y$

## C INTERFACE

\#include <sunperfh>
void chpr2 (char uplo, intn, com plex *alpha, com plex *x, int incx, com plex *y, int incy, com plex *ap);
void chpr2_64 (charuple, long n, com plex *alpha, com plex *x, long incx, com plex *y, long incy, com plex *ap);

## PURPOSE

chpr2 performs the Herm tian rank 2 operation $A:=$ alpha*x*conjg ( $\mathrm{y}^{\prime}$ ) + conjg (alpha ) $\mathrm{y}^{\star}$ con $\dot{g}\left(\mathrm{x}^{\prime}\right)$ + A where alpha is a scalar, $x$ and $y$ are $n$ elem ent vectors and $A$ is an $n$ by $n$ herm itian $m$ atrix, supplied in packed form .

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the $m$ atrix $A$ is supplied in the packed amay AP as follow s:

UPLO = U 'or L ' The uppertriangularpartof $A$ is supplied in AP .

UPLO = L'or I' The low ertriangularpartof A is supplied in A P .

U nchanged on exit.

N (input)
O n entry, N specifies the order of the m atrix A . $\mathrm{N}>=0$. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
$(1+(n-1) \star a b s(\mathbb{N} C X))$. Before entry, the increm ented array X m ust contain the n elem ent vectorx. U nchanged on exit.
$\mathbb{N C X}$ (input)

On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents ofX. $\mathbb{N} C X<>0$. U nchanged on exit.

Y (input)
$(1+(n-1) \star \operatorname{abs}(\mathbb{N} C Y))$. Before entry, the increm ented array $Y$ must contain the $n$ elem ent vectory. U nchanged on exit.
$\mathbb{N} C Y$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $Y$. $\mathbb{N} C Y$ <> 0 . Unchanged on exit.
AP (input/output)
$\left(\left(n^{*}(n+1)\right) / 2\right)$. Before entry $w$ ith UPLO $=$
$U$ ' or $L$ ', the array AP m ust contain the upper
triangularpartof the herm tian $m$ atrix packed
sequentially, column by colum n, so thatAP (1) contains a (1, 1), AP (2) and AP (3) contain a ( 1,2 ) and a (2,2) respectively, and so on. On exit, the array AP is overw ritten by the upper triangular part of the updated $m$ atrix. Before entry with UPLO = L 'or I', the array AP must contain the low er triangular part of the herm itian $m$ atrix packed sequentially, colum $n$ by colum $n$, so thatA P (1) contains a (1, 1), AP (2) and AP (3 ) contain a(2,1) and a (3, 1 ) respectively, and so on. On exit, the array AP is overw ritten by the low ertriangularpart of the updated $m$ atrix. N ote that the im aginary parts of the diagonalele$m$ ents need notbe set, they are assum ed to be zero, and on exit they are set to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

> chprefs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is H erm itian indefinite and packed, and provides emorbounds and backw ard emor estim ates for the solution

## SYNOPSIS

```
SUBROUT\mathbb{NE CHPRFS (UPLO,N,NRHS,A,AF,\mathbb{PIVOT,B,LDB,X,LDX,FERR,}}\mathbf{N},\textrm{L}
    BERR,W ORK,W ORK2,INFO)
CHARACTER * 1 UPLO
COM PLEX A (*),AF (*),B (LDB,*),X (LDX ,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
\mathbb{NTEGER \mathbb{PIVOT (*)}}\mathbf{(})
REAL FERR(*),BERR(*),WORK2 (*)
SUBROUT\mathbb{NE CHPRFS_64 (UPLO,N,NRHS,A,AF,\mathbb{PIVOT,B,LDB,X,LDX,}}\mathbf{N},\textrm{N},\textrm{N}
    FERR,BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 UPLO
COM PLEX A (*),AF (*),B (LD B ,*),X (LDX,*),W ORK (*)
\mathbb{NTEGER*8N,NRHS,LDB,LDX,INFO}
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
REAL FERR (*),BERR (*),W ORK2 (*)
```


## F95 INTERFACE

SU BROUTINE HPRFS (UPLO,N, $\mathbb{N} R H S], A, A F, \mathbb{P} \mathbb{I} O T, B,[L D B], X,[L D X]$, FERR, BERR, [W ORK], [W ORK 2], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1) ::UPLO
COMPLEX,D $\mathbb{I M} E N S I O N(:):: A, A F, W$ ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : $: B, X$
$\mathbb{N}$ TEGER : $: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} \mathrm{B}, \mathrm{LD} \mathrm{X}, \mathbb{N} F \mathrm{O}$
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{I} O T$
REAL,D $\mathbb{M} E N S I O N(:):: F E R R, B E R R, W$ ORK 2

SU BROUT $\mathbb{N} E$ HPRFS_64 (UPLO, N, $\mathbb{N} R H S], A, A F, \mathbb{P} \mathbb{I V O T}, B,[L D B], X$, [ $L$ D X ], FERR, BERR, [WORK], [WORK2], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::UPLO
COM PLEX , D $\mathbb{M}$ ENSION (:) ::A ,AF,W ORK
COM PLEX, D IM ENSION (: : : : : B, X
$\mathbb{N}$ TEGER (8) :: N , NRHS,LDB,LDX, $\mathbb{N}$ FO
$\mathbb{N}$ TEGER (8), D $\mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$
REAL,D $\mathbb{M} E N S I O N(:):: F E R R, B E R R, W$ ORK2

## C INTERFACE

\#include <sunperfh>
void chprfs (charuplo, intn, intnrhs, com plex *a, com plex *af, int *ipivot, com plex *b, int ldb, com plex *x, int ldx, float * ferr, float *berr, int *info);
void chprfs_64 (charuple, long n, long nıhs, com plex *a, complex *af, long *ípívot, com plex *b, long ldb, com plex *x, long ldx, float * ferr, float *berr, long *info);

PURPOSE
chpris im proves the com puted solution to a system of linear equations w hen the coefficientm atrix is H erm itian indefinte and packed, and provides emorbounds and backw ard error estim ates for the solution.

## ARGUMENTS

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low er triangle ofA is stored.

N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber
of collm ns of the matrioes $B$ and $X$. NRHS $>=0$.

A (input) The upper or low er triangle of the H erm itian
$m$ atrix A, packed colum nw ise in a linear array.
The jth column ofA is stored in the array A as
follows: if UPLO $=U$ ',A $(i+(j-1) * j 2)=A(i, 7)$ for $1<=i<=j$ if $U P L O=L ', A(i+(j-1) *(2 * n-7 / 2)$
=A $(i, j)$ for $j=i<=n$.

## AF (input)

The factored form of them atrix A. AF contains the block diagonal matrix D and the multipliers used to obtain the factor $U$ orL from the factorization $A=U * D * U * * H$ orA $=L * D * L * * H$ as com puted by CHPTRF, stored as a packed triangularm atrix.

PIVOT (input)
D etails of the interchanges and the block structure ofD as determ ined by CHPTRF.
$B$ (input) The righthand side m atrix $B$.
LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, \mathbb{N})$.

X (input/output)
On entry, the solution $m$ atrix $X$, as com puted by CHPTRS. On exit, the im proved solution $m$ atrix $X$.

## LD X (input)

The leading dim ension of the array X . LD X >= $\max (1, N)$.

## FERR (output)

The estim ated forw ard errorbound for each solution vector $X()$ ) the $j$ th column of the solution $m$ atrix X). If XTRUE is the true solution comesponding to $X(\neg)$ FERR $(\neg)$ is an estim ated upperbound for the $m$ agnitude of the largest ele$m$ ent in ( $X(\mathcal{D})-X$ TRU $E$ ) divided by the $m$ agnitude of the largestelem ent in $X(\mathcal{)}$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

## BERR (output)

The com ponentw ise relative backw ard error of each
solution vector $X$ ( $j$ ) (ie., the sm allest relative change in any elem entofA orB thatm akes $X$ ( $\mathcal{j}$ ) an exactsolution).

W ORK (w orkspace)
dim ension ( $2 * \mathrm{~N}$ )
W ORK 2 (w orkspace)
dim ension $\mathbb{N}$ )
$\mathbb{I N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N ~ F O ~}=-$ i, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

chpsv - com pute the solution to a com plex system of linear equations $A * X=B$,

## SYNOPSIS



```
CHARACTER * 1 UPLO
COM PLEX A (*),B (LD B,*)
INTEGERN,NRHS,LDB,INFO
INTEGER \mathbb{PIVOT (*)}
SUBROUT\mathbb{NECHPSV_64(UPLO,N,NRHS,A,\mathbb{PIVOT,B,LDB,INFO )}}\mathbf{N}\mathrm{ (N,N}
CHARACTER * 1 UPLO
COM PLEX A (*),B (LDB,*)
INTEGER*8N,NRHS,LDB,INFO
INTEGER*8 \mathbb{PIVOT (*)}
```


## F95 INTERFACE

SU BROUTINE HPSV (UPLO ,N, $\mathbb{N} R H S], A, \mathbb{P} \mathbb{I V} O T, B,[L D B],[\mathbb{N} F O])$
CHARACTER (LEN=1)::UPLO
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::A
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::B
$\mathbb{N}$ TEGER ::N,NRHS,LDB, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T$
SUBROUTINE HPSV_64 (UPLO,N, $\mathbb{N} R H S], A, \mathbb{P} \mathbb{I V O T}, \mathrm{~B},[\mathrm{LD} B],[\mathbb{N} F O])$
CHARACTER (LEN=1)::UPLO

COM PLEX , D $\mathbb{M}$ ENSION (:) ::A
COM PLEX , D IM ENSION (:,:) ::B
$\mathbb{N}$ TEGER (8) :: N , NRHS,LDB, $\mathbb{N} F O$
$\mathbb{N}$ TEGER (8), D $\mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$

## C INTERFACE

\#include <sunperfh>
void chpsv (charuple, int n, int nrhs, com plex *a, int *ịivot, com plex *b, int ldlo, int *info);
void chpsv_64 (char uplo, long n, long nrhs, com plex *a, long *ípivot, com plex *b, long ldb, long *info);

## PURPOSE

chpsv com putes the solution to a com plex system of linear equations
$A$ * $X=B, w h e r e A$ is an $N$ boy-N H erm tian $m$ atrix stored in packed form atand X and B are N Hoy-N RH S $m$ atrices.

The diagonalpivoting $m$ ethod is used to factorA as
$A=U * D * U * * H$, if $U P L O=U$ ', or
$A=L * D * L * * H$, if $U P L O=L^{\prime}$,
w here U (orL) is a productof perm utation and unit upper (low er) triangularm atrices, $D$ is $H$ erm itian and block diagonalw ith 1 boy -1 and 2 -by-2 diagonal blocks. The factored form of A is then used to solve the system of equations A * $X=B$.

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': Upper triangle ofA is stored;
= $\mathbb{L}$ ': Low er triangle ofA is stored.

N (input) The num ber of linearequations, ie., the order of them atrix $A . N>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS $>=0$.

A (input/output)
O n entry, the upper or low ertriangle of the H er$m$ itian $m$ atrix $A$, packed colum nw ise in a linear anray. The $j$ th column of $A$ is stored in the
array A as follows: if UPLO = U',A (i+ $(\mathfrak{j}$
 $(j-1) *(2 n-j / 2)=A(i, j)$ for $j=i<=n$. See below for further details.

On exit, the block diagonalm atrix D and the multipliers used to obtain the factorU orL from the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}$ orA $=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}$ as com puted by CHPTRF, stored as a packed triangular $m$ atrix in the sam e storage form atas $A$.

IPIVOT (output)
D etails of the interchanges and the block structure ofD, as determ ined by CHPTRF. If IPIVOT (k) $>0$, then row sand columnsk and $\mathbb{P} \mathbb{V} O T(k)$ were interchanged, and $\mathrm{D}(\mathrm{k}, \mathrm{k})$ is a 1 -by -1 diagonal block. If UPLO $=\mathrm{U}$ 'and $\mathbb{P} \mathbb{I V O T}(\mathrm{k})=\mathbb{P} \mathbb{V} O T(\mathrm{k}-1)$ $<0$, then row s and colum ns $k-1$ and $-\mathbb{P} \mathbb{I V}$ O ( $k$ ) w ere interchanged and $D(k-1 * k, k-1 k)$ is a 2 -by-2 diagonal block. If UPLO $=\mathrm{L}^{\prime}$ and $\mathbb{P} \mathbb{I V O T}(\mathrm{k})=$ $\mathbb{P I V O T}(k+1)<0$, then row $s$ and colum ns $k+1$ and $-\mathbb{P}$ IV O T (k) w ere interchanged and $D(k \cdot k+1, k \cdot k+1)$ is a 2-by-2 diagonalblock.

B (input/output)
On entry, the N -by -N RH S righthand side $m$ atrix B . On exit, if $\mathbb{N F O}=0$, the $N$-by-NRHS solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the aray $B$. LD B >= $\max (1, \mathbb{N})$.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an illegalvalue
$>0:$ if $\mathbb{N F O}=\mathrm{i}, \mathrm{D}(\mathrm{i}, \mathrm{i})$ is exactly zero. The factorization has been com pleted, but the block diagonalm atrix $D$ is exactly singular, so the solution could notbe com puted.

## FURTHER DETAILS

The packed storage schem e is illustrated by the follow ing exam ple when $\mathrm{N}=4, \mathrm{UPLO}=\mathrm{U}$ ':

Tw o-dim ensional storage of the $H$ erm itian $m$ atrix A:
al1 a12 a13 a14
a33 a34 (aij= con $\dot{j}(a \ddot{j}))$ a44

Packed storage of the upper triangle ofA :

$$
A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]
$$

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

chpsvx - use the diagonal pivoting factorization $\mathrm{A}=$ $\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}$ to com pute the solution to a com plex system of linearequations $A * X=B$, where $A$ is an $N$ by -N H erm tian $m$ atrix stored in packed form at and $X$ and $B$ are N -by-N R H S m atrices

## SYNOPSIS

```
SUBROUT\mathbb{NECHPSVX (FACT,UPLO,N,NRHS,A,AF,\mathbb{PIVOT,B,LDB,X,LDX,}}\mathbf{N},\textrm{L},\textrm{L}
    RCOND,FERR,BERR,W ORK,W ORK2,INFO)
CHARACTER * 1 FACT,UPLO
COM PLEX A (*),AF (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,INFO
INTEGER \mathbb{PIVOT (*)}
REALRCOND
REAL FERR (*),BERR (*),WORK2 (*)
SUBROUT\mathbb{NE CHPSVX_64(FACT,UPLO,N,NRHS,A,AF, \mathbb{PIVOT,B,LDB,X,}}\mathbf{N},\textrm{N},\textrm{N}
    LDX,RCOND,FERR,BERR,WORK,W ORK 2, INFO)
CHARACTER * 1 FACT,UPLO
COMPLEX A (*),AF (*),B (LDB,*),X (LDX,*),W ORK (*)
\mathbb{NTEGER*8N,NRHS,LDB,LDX,INFO}
INTEGER*8 \mathbb{P IVOT (*)}
REALRCOND
REALFERR (*),BERR (*),WORK2 (*)
```


## F95 INTERFACE

SU BROUTINE HPSVX (FACT,UPLO,N, $\mathbb{N} R H S], A, A F, \mathbb{P} \mathbb{V} O T, B,[L D B], X$, [LDX],RCOND,FERR,BERR, [W ORK], [W ORK2], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) ::FACT,UPLO
COMPLEX,D $\mathbb{I M} E N S I O N(:):: A, A F, W$ ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : $\mathrm{B}, \mathrm{X}$
$\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T$
REAL ::RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::FERR,BERR,W ORK 2
SUBROUTINE HPSVX_64 (FACT,UPLO,N, $\mathbb{N R H S ] , A , A F , \mathbb { P } I V O T , B , [ L D B ] , ~}$ $\mathrm{X},[\operatorname{LD} \mathrm{X}], R C O N D, F E R R, B E R R, \mathbb{W} O R K], \mathbb{W} O R K 2],[\mathbb{N} F O])$

CHARACTER (LEN=1)::FACT,UPLO
COMPLEX,D $\mathbb{M}$ ENSION (:) ::A,AF,WORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : : B, X
$\mathbb{N}$ TEGER (8) ::N,NRHS,LDB,LDX, $\mathbb{N}$ FO
$\mathbb{N} T E G E R(8), D \mathbb{M}$ ENSION(:):: $\mathbb{P} \mathbb{I V O T}$
REAL ::RCOND
REAL,D IM ENSION (:) ::FERR,BERR,W ORK 2

## C INTERFACE

\#include <sunperfh>
void chpssx (char fact, charuplo, intn, int nrhs, com plex *a, com plex *af, int *ipivot, com plex *b, int ldb, com plex *x, int ldx, float *rcond, float *ferr, float*ber, int*info);
void chpssx_64 (char fact, charuplo, long n, long nrhs, com plex *a, com plex *af, long *ipivot, com plex *b, long lab, com plex *x, long ldx, float *roond, float *ferr, float *berr, long *info);

## PURPOSE

chpsvx uses the diagonalpivoting factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}$ to com pute the solution to a com plex system of linear equations $A * X=B$, where $A$ is an $N$ by $-N$ Herm itian $m$ atrix stored in packed form at and $X$ and $B$ are $N$-byNRH S m atrices.

Errorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:

1. IfFACT $=N$ ', the diagonalpivoting $m$ ethod is used to factorA as

$$
A=U * D * U * * H, \text { if } U P L O=U ' \text { or }
$$

$$
A=L * D * L * * H, \text { if } U P L O=L \prime,
$$

where $U$ (orL) is a product of perm utation and unit upper (low er)
triangularm atrices and $D$ is Herm itian and block diagonal w ith
1-by-1 and 2-by-2 diagonalblocks.
2. If som eD $(i, i)=0$, so thatD is exactly singular, then the routine
retums w ith $\mathbb{N} F O=$ i. O therw ise, the factored form of $A$ is used
to estim ate the condition num ber of the $m$ atrix $A$. If the reciprocal of the condition num ber is less than $m$ achine precision,
$\mathbb{N} F O=N+1$ is retumed as a waming, but the routine stillgoes on
to solve for $X$ and com pute error bounds as described below.
3.The system of equations is solved for $X$ using the factored form of A.
4. Iterative refinem ent is applied to im prove the com puted solution
$m$ atrix and calculate error bounds and backw ard enror estim ates
for it.

## ARGUMENTS

FACT (input)
Specifies w hether ornot the factored form of A has been supplied on entry. = F ': On entry, A F and $\mathbb{P I V O T}$ contain the factored form of A. AF and $\mathbb{P} \mathbb{I V}$ O T w illnotbe m odified. $=\mathrm{N}$ ': Them atrix A
w ill.be copied to AF and factored.

UPLO (input)
$=\mathrm{U}:$ : U ppertriangle ofA is stored;
= L': Low er triangle ofA is stored.
N (input) The num ber of linearequations, i.e., the order of them atrix $A . N>=0$.

NRHS (input)
The num ber of right hand sides, i.e., the num ber of collm ns of the m atrices B and X. NRHS >=0.

A (input) The upper or low er triangle of the H erm itian $m$ atrix A, packed colum nw ise in a linear anray. The $j$ th column of A is stored in the array A as follows: if UPLO $=U U^{\prime}, A(i+(j-1) * j 2)=A(i, 7)$ for $1<=i<=j$ if $U P L O=L ', A(i+(j-1) *(2 * n-) / 2)$
$=A(i, y)$ for $j=i<=n$. See below forfurther details.

AF (input/output)
If $F A C T=F$ ', then $A F$ is an input argum ent and on entry contains the block diagonalm atrix $D$ and the m ultipliers used to obtain the factor $U$ orl from the factorization $A=U * D * U * * H$ orA $=L * D * L * * H$ as com puted by CH PTRF, stored as a packed triangular $m$ atrix in the sam e storage form atasA.
IfFACT = N ', then AF is an output argum ent and on exit contains the block diagonalm atrix $D$ and the m ultipliers used to obtain the factorU or L from the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}$ or $\mathrm{A}=$ L*D *L**H as com puted by CHPTRF, stored as a packed triangularm atrix in the sam e storage form at as A.

## IPIVOT (inputoroutput)

If $F A C T=F '$, then $\mathbb{P I V O T}$ is an input argum ent and on entry contains details of the interchanges and the block structure of D, as determ ined by CHPTRF. If $\mathbb{P I V O T}(k)>0$, then row sand colum nsk and $\mathbb{P} \mathbb{I V O T}(k)$ w ere interchanged and $D(k, k)$ is a 1 -by-1 diagonal block. If UPLO = U'and $\mathbb{P} \mathbb{V} O T(k)=\mathbb{P} \mathbb{I V} O T(k-1)<0$, then row sand colum ns $k-1$ and $-\mathbb{P} \mathbb{V O T}(k)$ were interchanged and $D(k-$ $1 \mathrm{k}, \mathrm{k}-1 \mathrm{k})$ is a 2 -by-2 diagonalblock. IfUPLO $=$ L 'and $\mathbb{P} \mathbb{I V O T}(k)=\mathbb{P} \mathbb{I V O T}(k+1)<0$, then row s and colum nsk+1 and - $\mathbb{P}$ IV OT (k) w ere interchanged and D $(k *+1, k k+1)$ is a 2 -by-2 diagonalblock.

IfFACT = $N$ ', then $\mathbb{P I V O T}$ is an output argum ent and on exit contains details of the interchanges and the block structure of D, as determ ined by CHPTRF.

B (input) The N -by-N RH S righthand side m atrix B .
LD B (input)
The leading dim ension of the aray $\mathrm{B} . \operatorname{LD} B>=$ $\max (1, N)$.

X (output)
If $\mathbb{N} F O=0$ or $\mathbb{N} F O=\mathrm{N}+1$, the N -by-NRH S solution $m$ atrix $X$.

LD X (input)
The leading dim ension of the array X . LD X >= $\max (1, N)$.

## RCOND (output)

The estim ate of the reciprocal condition num berof the matrix A. IfRCOND is less than them achine precision (in particular, if RCOND $=0$ ), the $m$ atrix is singular to working precision. This condition is indicated by a retum code of $\mathbb{N}$ FO > 0.

## FERR (output)

The estim ated forw ard enrorbound for each solution vector $X(\mathcal{)})$ (the $j$ th column of the solution $m$ atrix X). If XTRUE is the true solution corresponding to $X(\mathcal{H})$, FERR ( $)$ ) is an estim ated upperbound for the $m$ agnitude of the largest ele$m$ ent in ( $X(\mathcal{y})-X$ TRUE) divided by the $m$ agnitude of the largestelem ent in $X(\mathcal{J})$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

## BERR (output)

The com ponentw ise relative backw ard error of each solution vector $X(\mathcal{j})$ (i.e., the sm allest relative change in any elem entofA orB thatm akes X (i) an exactsolution).

W ORK (w orkspace)
dim ension ( $2 * \mathrm{~N}$ )

W ORK 2 (w orkspace)
dim ension (N)
$\mathbb{I N} F O$ (output)
= 0: successfulexit
< 0 : if $\mathbb{N N F O}=-$ i, the $i$-th argum ent had an illegalvalue
> 0 : if $\mathbb{N F O}=i$, and $i$ is
<= N : D (i,i) is exactly zero. The factorization has been com pleted but the factorD is exactly singular, so the solution and error bounds could not be com puted. RCOND $=0$ is retumed. $=\mathrm{N}+1$ : D is nonsingular, butRCOND is less than machine precision, $m$ eaning that the $m$ atrix is singular to w orking precision. N evertheless, the solution and error bounds are com puted because there are a num berof situationsw here the com puted solution
can bem ore accurate than the value ofRCOND w ould suggest.

## FURTHER DETAILS

The packed storage schem e is illustrated by the follow ing exam plewhen $\mathrm{N}=4, \mathrm{UPLO}=\mathrm{U}$ ':

Tw o-dim ensional storage of the H erm itian m atrix A :
al1 a12 a13 a14
a22 a23 a24
a33 a34 (aij=conjg (ä̈)) a44

Packed storage of the upper triangle ofA :
A = [a11,a12,a22,a13,a23,a33,a14,a24, a34,a44 ]

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

chptrd-reduce a complex Herm itian matrix A stored in packed form to realsym $m$ etric tridiagonal form $T$ by a unitary sim ilarity transform ation

## SYNOPSIS

```
SUBROUT\mathbb{NECHPTRD(UPLO,N,AP,D,E,TAU,INFO)}
CHARACTER * 1 UPLO
COM PLEX AP (*),TAU (*)
INTEGER N, \mathbb{NFO}
REALD (*),E (*)
SUBROUT\mathbb{NE CHPTRD_64(UPLO,N,AP,D,E,TAU, INFO)}
CHARACTER * 1 UPLO
COM PLEX AP (*),TAU (*)
INTEGER*8 N, INFO
REALD (*),E (*)
```


## F95 INTERFACE

```
SU BROUTINE HPTRD (UPLO,N,AP,D,E,TAU, [ \(\mathbb{N} F O\) ])
CHARACTER (LEN=1) ::UPLO
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::AP,TAU
\(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
REAL,D \(\mathbb{I M}\) ENSION (:) ::D , E
SU BROUTINE HPTRD_64 (UPLO,N,AP,D ,E,TAU, [ \(\mathbb{N} F \mathrm{FO}\) ])
CHARACTER (LEN=1)::UPLO
```


## C INTERFACE

\#include <sunperfh>
void chptrd (charuplo, intn, com plex *ap, float *d, float *e, com plex *tau, int *info);
void chptro_ 64 (charuplo, long n, com plex *ap, float *d, float *e, com plex *tau, long *info);

## PURPOSE

chptrd reduces a com plex H erm itian m atrix A stored in packed form to real sym $m$ etric tridiagonal form $T$ by a unitary sim ilarity transform ation: Q ** H * A * $\mathrm{Q}=\mathrm{T}$.

## ARGUMENTS

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.
N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

AP (input)
O $n$ entry, the upper or low ertriangle of the Her $m$ tian $m$ atrix $A$, packed colum nw ise in a linear array. The jth column of A is stored in the array AP as follows: ifUPLO = U',AP (i+ (j $1) \star j 2$ ) $=A(i, \gamma)$ for $1<=i<=j$ ifUPLO $=L^{\prime}$, AP $(i$ $+(j-1)^{\star}(2 * n-j / 2)=A(i, 7)$ for $j<=i<=n$. On exit, if $\mathrm{UPLO}=\mathrm{U}$ ', the diagonal and first superdiagonalofA are overw ritten by the comesponding ele$m$ ents of the tridiagonalm atrix $T$, and the ele$m$ ents above the first superdiagonal, $w$ ith the array TAU, represent the unitary $m$ atrix $Q$ as a product of elem entary reflectors; if $\mathrm{i} P L O=\mathrm{L}$ ', the diagonaland firstsubdiagonalofA are overw ritten by the comesponding elem ents of the tridiagonalm atrix $T$, and the elem ents below the first subdiagonal, w ith the amay TAU, represent the unitary $m$ atrix $Q$ as a product of elem entary reflectors. See FurtherD etails.

D (output)
The diagonalelem ents of the tridiagonalm atrix T :
$D(i)=A(i, i)$.

E (output)
The off-diagonal elem ents of the tridiagonal $m$ atrix $T: E(i)=A(i, i+1)$ if $U P L O=U^{\prime}, E(i)=$ A $(i+1, i)$ if $\mathrm{UPLO}=\mathrm{L}$.

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails) .
$\mathbb{I N F O}$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N F O}=-i$, the $i-$ th argum enthad an illegalvalue

## FURTHER DETAILS

IfU PLO $=U$ ', the m atrix $Q$ is represented as a product of elem entary reflectors

$$
Q=H(n-1) \ldots H(2) H(1)
$$

Each H (i) has the form
$H(i)=I-\tan { }^{*} V^{*} V^{\prime}$
$w$ here tau is a com plex scalar, and $v$ is a com plex vector w ith $\mathrm{v}(\mathrm{i}+1 \mathrm{n})=0$ and $\mathrm{v}(\mathrm{i})=1 ; \mathrm{v}(1: i-1)$ is stored on exit in $A P$, overw riting $A(1: i-1, i+1)$, and tau is stored in TAU (i).

If U PLO $=\mathrm{L}$ ', them atrix Q is represented as a product of elem entary reflectors

$$
Q=H(1) H(2) \ldots H(n-1)
$$

Each H (i) has the form

$$
H(i)=I-\tan * v^{\star} v^{\prime}
$$

$w$ here tau is a com plex scalar, and $v$ is a com plex vector w ith $\mathrm{v}(1: i)=0$ and $v(i+1)=1 ; v(i+2 \mathrm{~m})$ is stored on exit in AP, overw riting A (i+2 $\mathrm{n}, \mathrm{i}$ ), and tau is stored in TA U (i).

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

chptrf-com pute the factorization of a com plex Herm tian packed $m$ atrix A using the Bunch $-K$ aufn an diagonalpívoting m ethod

## SYNOPSIS



```
CHARACTER * 1 UPLO
COM PLEX A (*)
\mathbb{NTEGER N,}\mathbb{N}FO
INTEGER \mathbb{PIVOT (*)}
```



```
CHARACTER * 1 UPLO
COMPLEX A (*)
\mathbb{NTEGER*8 N,\mathbb{NFO}}\mathbf{~}+
\mathbb{NTEGER *8 \mathbb{PIVOT (*)}}\mathbf{(})
```


## F95 INTERFACE

```
SU BROUTINE HPTRF (UPLO ,N,A, \(\mathbb{P} \mathbb{I V O T},[\mathbb{N} F O]\) )
CHARACTER (LEN=1)::UPLO
COMPLEX,D \(\mathbb{I M}\) ENSION (:) ::A
\(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{V} O T\)
SU BROUTINE HPTRF_64 (UPLO, N, A, \(\mathbb{P} \mathbb{I V O T},[\mathbb{N} F O]\) )
CHARACTER (LEN=1) ::UPLO
```


## C INTERFACE

\#include <sunperfh>
void chptrf(charuple, intn, com plex *a, int *ipivot, int *info);
void chptrf_64 (charuplo, long n, com plex *a, long *ipivot, long *info);

## PURPOSE

chptrf com putes the factorization of a com plex H erm itian packed $m$ atrix $A$ using the Bunch-K aufm an diagonalpivoting $m$ ethod:

$$
A=U * D * U * * H \text { or } A=L * D * L * * H
$$

where U (orL) is a productofperm utation and unit upper (low er) triangular matrices, and $D$ is $H$ erm itian and block diagonalw ith 1 boy-1 and 2 -by-2 diagonalblocks.

## ARGUMENTS

UPLO (input)
$=U^{\prime}:$ U ppertriangle of $A$ is stored;
$=1$ ': Low er triangle ofA is stored.

N (input) The order of the $m$ atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
O n entry, the upper or low er triangle of the H er$m$ itian $m$ atrix A, packed colum nw ise in a linear array. The $j$ th colum n of $A$ is stored in the aray $A$ as follows: if UPLO = U',A (i+ (j $1) \star j 2$ ) $=A(i, j)$ for $1<=i<=j$ ifUPLO $=L^{\prime}, A(i+$ $(j-1) *(2 n-1) / 2)=A(i, 7)$ for $\dot{j}=i<=n$.

O n exit, the block diagonalm atrix $D$ and the $m u l$ tipliers used to obtain the factorU orL, stored as a packed triangularm atrix overw riting A (see below for further details).

D etails of the interchanges and the block structure of $D$. If $\mathbb{P I V O T}(k)>0$, then row sand columnsk and $\mathbb{P I V O T}(k)$ were interchanged and $D(k, k)$ is a $1-b y-1$ diagonalblock. If $U P L O=U '$ and $\mathbb{P} \mathbb{I V O T}(k)=\mathbb{P} \mathbb{V} O T(k-1)<0$, then row $s$ and colum ns $k-1$ and - $\mathbb{P I V O T}(k)$ were interchanged and D $(k-1 k, k-1 k)$ is a $2-b y-2$ diagonal block. If UPLO $=\mathrm{L}$ 'and $\mathbb{P I V O T}(k)=\mathbb{P} \mathbb{I V O T}(k+1)<0$, then row s and colum nsk+1 and - $\mathbb{P}$ IV OT (k) were interchanged and $D(k, k+1, k \cdot k+1)$ is a 2 -by -2 diagonal block.
$\mathbb{I N} F O$ (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N F O}=-$ i, the $i$-th argum enthad an illegalvalue $>0:$ if $\mathbb{N} F O=i, D(i, i)$ is exactly zero. The factorization has been com pleted, but the block diagonalm atrix $D$ is exactly singular, and division by zero w ill occur if it is used to solve a system ofequations.

## FURTHER DETAILS

## 5-96-B ased on m odifications by J.Lew is, Boeing C om puter

 ServicesCom pany
If $U P L O=U$ ', then $A=U * D * U$ ', where
$U=P(n) \star U(n) * \ldots{ }^{\star}(k) U(k)^{\star} \ldots$,
i.e., $U$ is a product of term $S P(k) * U(k)$, where $k$ decreases from $n$ to 1 in steps of 1 or 2 , and $D$ is ablock diagonal $m$ atrix $w$ ith 1 -by -1 and 2 -by-2 diagonalblocks $D(k) . P(k)$ is a perm utation $m$ atrix as defined by $\mathbb{P} \mathbb{I V O T}(k)$, and $U(k)$ is a unituppertriangularm atrix, such that if the diagonal block $D(k)$ is of orders ( $s=1$ or 2 ), then

$$
\begin{aligned}
& \text { ( I v 0 ) k-s } \\
& U(k)=(0 \quad 1 \quad 0) s \\
& \text { ( } 00 \text { I ) n-k } \\
& \mathrm{k}-\mathrm{s} \mathrm{~s} \mathrm{n}-\mathrm{k}
\end{aligned}
$$

If $s=1, D(k)$ overw rites $A(k, k)$, and $v$ overw rites $A(1 k-$ $1, k$ ). If $s=2$, the upper triangle ofD ( $k$ ) overw rites $A(k-$ $1, k-1)$, A $(k-1, k)$, and $A(k, k)$, and $v$ overw rites A $(1 k-2, k-$ 1 k).

If $U P L O=L$ ', then $A=L * D * L$ ', where

$$
L=P(\mathbb{1}) \star L(1) \star \ldots * P(k) \star L(k)^{\star} \ldots,
$$

i.e., $L$ is a product of term $s P(k) * L(k)$, where $k$ increases
from 1 to $n$ in steps of 1 or 2, and $D$ is a block diagonal $m$ atrix $w$ ith 1 -by- 1 and 2 -by-2 diagonalblocks $D(k)$. $P(k)$ is a perm utation $m$ atrix as defined by $\mathbb{P} \mathbb{I V} O T(k)$, and $L(k)$ is a unit low ertriangularm atrix, such that if the diagonal
block D (k) is of orders ( $s=1$ or2), then

```
    ( I 0 0 ) k-1
L (k)=( 0 I 0 ) s
    ( 0 v I ) n-k-s+1
        k-1 s n-k-s+1
```

If $s=1, D(k)$ overw rites $A(k, k)$, and $v$ overw rites A ( $k+1 n, k$ ). If $s=2$, the low ertriangle ofD ( $k$ ) overw rites $A(k, k), A(k+1, k)$, and $A(k+1, k+1)$, and $v$ overw rites A ( $k+2 n, k k+1$ ).

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chptri-com pute the inverse of a com plex H erm itian indefinthe $m$ atrix $A$ in packed storage using the factorization $A=$ $\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}$ orA $=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}$ com puted by CHPTRF

## SYNOPSIS



```
CHARACTER * 1 UPLO
COM PLEX A (*),W ORK (*)
\mathbb{NTEGER N,\mathbb{NFO}}\mathbf{N}=0
INTEGER \mathbb{PIVOT (*)}
```

SU BROUTINE CHPTRI_64 (UPLO,N,A, $\mathbb{P} \mathbb{I V O T}, \mathrm{W} O R K, \mathbb{N} F O)$
CHARACTER * 1 UPLO
COMPLEX A (*), W ORK (*)
$\mathbb{N}$ TEGER*8 $\mathrm{N}, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{V} O T\left({ }^{*}\right)$
F95 INTERFACE
SU BROUTINE HPTRI(UPLO,N,A, $\mathbb{P} \mathbb{I V O T},[\mathbb{W} O R K],[\mathbb{N} F O])$
CHARACTER (LEN=1) ::UPLO
COMPLEX,D $\mathbb{M}$ ENSION (:) ::A,W ORK
$\mathbb{N} T E G E R:: N, \mathbb{N F O}$
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V} O T$
SU BROUTINE HPTRI_64 (UPLO,N,A, $\mathbb{P} \mathbb{I V O T},[\mathbb{W} O R K],[\mathbb{N} F O])$
CHARACTER (LEN=1) ::UPLO
COMPLEX,D $\mathbb{M}$ ENSION (:) ::A,W ORK
$\mathbb{N}$ TEGER (8) ::N, $\mathbb{N} F O$
$\mathbb{N}$ TEGER (8),D $\mathbb{I M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V}$ OT

## C INTERFACE

\#include < sunperfh>
void chptri(charuplo, intn, com plex *a, int *ipivot, int *info);
void chptri_ 64 (char uplo, long n, com plex *a, long *ipijvot, long *info);

## PURPOSE

chptri computes the inverse of a complex Herm itian indefinite $m$ atrix $A$ in packed storage using the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}$ orA $=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}$ com puted by CHPTRF.

## ARGUMENTS

UPLO (input)
Specifies w hether the details of the factorization are stored as an upper or low er triangularm atrix.
$=\mathrm{U}$ ': U ppertriangular, form is $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}$;
= L ': Low ertriangular, form is $\mathrm{A}=\mathrm{L} * \mathrm{D}$ * $\mathrm{L} * * \mathrm{H}$.

N (input) The order of the m atrix A. $\mathrm{N}>=0$.
A (input/output)
O n entry, the block diagonalm atrix D and them ultipliers used to obtain the factorU orL as com puted by CHPTRF, stored as a packed triangular $m$ atrix.

On exit, if $\mathbb{N} F O=0$, the ( H em itian) inverse of the originalm atrix, stored as a packed triangular $m$ atrix. The $j$ th colum $n$ of inv ( $A$ ) is stored in the array A as follows: if UPLO = U',A (i+ (j 1) $* j 2)=\operatorname{inv}(A)(i, \gamma)$ for $1<=i<=j$; ifUPLO $=L^{\prime}$, A $(i+(j-1) *(2 n-1) / 2)=\operatorname{inv}(A)(i, 7)$ for $j<=i<=n$.
$\mathbb{P I V O T}$ (input)
D etails of the interchanges and the block structure ofD as determ ined by CHPTRF.

W ORK (w orkspace)
dim ension (N)
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the $i$-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N} F O=i, D(i, i)=0$; the $m$ atrix is singular and its inverse could notbe com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
chptrs-solve a system of linear equationsA*X = B w ith a
com plex H erm itian m atrix A stored in packed form at using the
factorization A = U *D *U**H or A = L*D *L**H computed by
CHPTRF
```


## SYNOPSIS

```
SUBROUT\mathbb{NE CHPTRS (UPLO,N,NRHS,A,\mathbb{PIVOT,B,LDB,INFO)}}\mathbf{N}\mathrm{ (N,N}
```

CHARACTER * 1 UPLO
COMPLEX A (*), B (LD B,*)
$\mathbb{N}$ TEGER N,NRHS,LDB, $\mathbb{N} F O$
$\mathbb{I N} T E G E R \mathbb{P} \mathbb{I V} O T\left({ }^{*}\right)$
SU BROUTINECHPTRS_64(UPLO,N,NRHS,A, $\mathbb{P} \mathbb{I V O T}, \mathrm{B}, \mathrm{LD}, \mathrm{N}, \mathbb{N} F \mathrm{~F})$
CHARACTER * 1 UPLO
COMPLEX A (*), B (LDB,*)
$\mathbb{N} T E G E R * 8 N, N R H S, L D B, \mathbb{N} F O$
$\mathbb{N}$ TEGER*8 $\mathbb{P} \mathbb{I V O T}$ ( ${ }^{*}$ )

## F95 INTERFACE

SU BROUT $\mathbb{N} E$ HPTRS (UPLO ,N, $\mathbb{N} R H S], A, \mathbb{P} \mathbb{I V} O T, B,[L D B],[\mathbb{N} F O])$
CHARACTER (LEN=1) ::UPLO
COMPLEX,D IM ENSION (:) ::A
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::B
$\mathbb{N}$ TEGER :: N, NRHS,LDB, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{V} O T$
SU BROUTINE HPTRS_64 (UPLO,N, $\mathbb{N} R H S], A, \mathbb{P} \mathbb{I} O T, B,[L D B],[\mathbb{N} F O])$

CHARACTER (LEN=1) ::UPLO
COM PLEX,D $\operatorname{IM}$ ENSION (:) ::A
COM PLEX, D IM ENSION (:,:) ::B
$\mathbb{N}$ TEGER (8) :: N , NRHS,LD B, $\mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT}$

## C INTERFACE

\#include <sunperfh>
void chptrs (charuplo, intn, int nrhs, com plex *a, int *ipivot, com plex *b, int ldb, int *info);
void chptrs_64 (charuplo, long n, long nrhs, com plex *a, long *ipivot, com plex *b, long ldlo, long *info);

## PURPOSE

chptrs solves a system of linear equations $A * X=B$ with $a$ com plex $H$ erm itian $m$ atrix A stored in packed form atusing the factorization $A=U * D * U * * H$ or $A=L * D * L * * H$ computed by CHPTRF.

## ARGUMENTS

## UPLO (input)

Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
$=\mathrm{U}$ : U ppertriangular, form is $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}$;
$=\mathrm{L}^{\prime}:$ Low ertriangular, form is $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}$.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS $>=0$.

A (input) The block diagonalm atrix D and the m ultipliers
used to obtain the factorU orL as com puted by CH PTRF, stored as a packed triangularm atrix.

IPIVOT (input)
D etails of the interchanges and the block structure ofD as determ ined by CHPTRF.

B (input/output)
O n entry, the right hand side m atrix B. On exit, the solution $m$ atrix X .

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.
$\mathbb{N}$ FO (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N F O}=-$ i, the $i$-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

chsein-use inverse tieration to find specified right and/or left eigenvectors of a com plex upperH essenberg matrix H

## SYNOPSIS



```
        LDVL,VR,LDVR,MM,M,W ORK,RW ORK,\mathbb{FA}|L,\mathbb{FA}\mathbb{L}R,\mathbb{NNFO)}
CHARACTER * 1 S\mathbb{DE,EIG SRC, IN ITV}
COM PLEX H (LDH,*),W (*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
INTEGERN,LDH,LDVL,LDVR,MM,M, INFO
\mathbb{NTEGER \mathbb{FA}\mathbb{L}(*),\mathbb{FA}\mathbb{L}R(*)}
LOGICAL SELECT (*)
REAL RW ORK (*)
```



```
        LDVL,VR,LDVR,MM,M,W ORK,RW ORK,\mathbb{FA}\mathbb{LL},\mathbb{FA}|\mathbb{H},\mathbb{N}FO)
CHARACTER * 1SIDE,EIGSRC,IN ITV
COM PLEX H (LDH ,*),W (*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
INTEGER*8N,LDH,LDVL,LDVR,MM,M,INFO
\mathbb{NTEGER*8 \mathbb{FA}|L(*),\mathbb{FA}\mathbb{LR}(*)}\=(*)
LOG ICAL*8 SELECT (*)
REAL RW ORK (*)
```


## F95 INTERFACE

SU BROUTINE HSE $\mathbb{N}$ (SDE, EIG SRC, $\mathbb{N} \mathbb{I T V}, \operatorname{SELECT}, \mathbb{N}], H,[L D H], W, V L$, [LDVL], VR, [LDVR],MM,M,[WORK], RW ORK], $\mathbb{F A} \mathbb{L} L, \mathbb{F} A \mathbb{H},[\mathbb{N} F O])$

COM PLEX,D $\mathbb{M}$ ENSION (:) ::W ,W ORK
COM PLEX,D $\mathbb{I}$ ENSION (:,:) ::H,VL,VR
$\mathbb{N} T E G E R:: N, L D H, L D V L, L D V R, M M, M, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: $\mathbb{F} A \mathbb{L} L, \mathbb{F} A \mathbb{L} R$
LOG ICAL,D IM ENSION (:) ::SELECT
REAL,D $\mathbb{M}$ ENSION (:) ::RW ORK

SU BROUTINE HSEIN_64 (SDE, E $\mathbb{I} G R C, \mathbb{N} \operatorname{ITV}, S E L E C T, \mathbb{N}], H,[L D H], W$, VL, [LDVL], VR, [LDVR], M M , M, [W ORK ], [RW ORK], $\mathbb{F} A \mathbb{L} L, \mathbb{F} A \mathbb{L} R$, [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::SDE,EIGSRC, $\mathbb{N}$ ITV
COM PLEX,D $\mathbb{M}$ ENSION (:) ::W ,W ORK
COMPLEX,D IM ENSION (:,:) ::H,VL,VR
$\mathbb{N}$ TEGER (8) ::N ,LD H,LDVL,LDVR,M M , M , $\mathbb{N} F O$
$\mathbb{N}$ TEGER (8),D $\mathbb{I M}$ ENSION (:) :: $\mathbb{F} A \mathbb{L} L, \mathbb{F} A \mathbb{I} R$
LOG ICAL (8), D IM ENSION (:) ::SELECT
REAL,D $\mathbb{I M} E N S I O N(:):: R W$ ORK

## C INTERFACE

\#include <sunperfh>
void chsein (char side, chareigsrc, char initv, int *select, int n , com plex *h, int ldh, com plex *w, com plex ${ }^{*} \mathrm{vl}$, int ldvl, com plex $\mathrm{V}_{\mathrm{Vr}}$, int ldvr, intm m , int *m , int *ifaill, int *ifailr, int *info);
void chsein_64 (charside, char eigsrc, char initv, long *select, long n, com plex *h, long ldh, com plex *w , com plex *vl, long ldvl, com plex *vr, long ldvr, long m m, long *m , long *ifaill, long *ifailr, long *info);

## PURPOSE

chsein uses inverse teration to find specified rightand/or lefteigenvectors of a com plex upper $H$ essenberg $m$ atrix $H$.

The righteigenvectorx and the lefteigenvector $y$ of the $m$ atrix $H$ comesponding to an eigenvalue $w$ are defined by:

$$
H * x=w * x, \quad y^{* * h} * H=w * y^{* * h}
$$

$w$ here $y^{\star *} h$ denotes the conjugate transpose of the vectory.

## ARGUMENTS

= R ': com pute righteigenvectors only;
= L ': com pute lefteigenvectors only;
= B ': com pute both right and lefteigenvectors.

## E IG SRC (input)

Specifies the souroe of eigenvalues supplied in $W$ :
= Q': the eigenvalueswere found using CHSEQR;
thus, if H has zero subdiagonalelem ents, and so is block-triangular, then the $j$ th eigenvalue can be assum ed to be an eigenvalue of the block containing the jth row /oolum $n$. This property allow $s$ CHSEIN to perform inverse iteration on justone diagonalblock. = N ': no assum ptions are $m$ ade on the correspondence betw een eigenvalues and diagonalblocks. In this case, CH SE $\mathbb{I N}$ m ustalw ays perform inverse iteration using the $w$ hole $m$ atrix $H$.
$\mathbb{N}$ ITV (input)
= N ': no initialvectors are supplied;
= U ': user-supplied initial vectors are stored in the arrays V L and/orV R .

SELECT (input)
Specifies the eigenvectors to be com puted. To select the eigenvector corresponding to the eigenvalueW ( $\mathcal{j}$ ) SELECT ( $\mathbf{j}$ ) m ustbe set to TRUE..

N (input) The order of the m atrix H. N >=0.
$H$ (input) The upper $H$ essenberg $m$ atrix $H$.

LD H (input)
The leading dim ension of the array $H$. LD H >= $\max (1, N)$.

W (input/output)
On entry, the eigenvalues of H . On exit, the real parts of $W$ may have been altered since close eigenvalues are perturbed slightly in searching for independenteigenvectors.

VL (input/output)
On entry, if $\mathbb{N} \mathbb{I T V}=\mathrm{U}$ 'and $S \mathbb{D} E=\mathrm{L}$ 'or B ', VL $m$ ust contain starting vectors for the inverse iteration for the lefteigenvectors; the starting vector for each eigenvectorm ustbe in the sam e colum $n$ in which the eigenvector will be stored. On exit, if $S \mathbb{D} E=$ 'L or B', the lefteigenvectors specified by SELEC T w illbe stored consecutively in the colum ns of $V L$, in the sam e order as
theireigenvalues. If $S \mathbb{D} E=R \prime V L$ is not referenced.

LDVL (input)
The leading dim ension of the array VL . LDVL >= $\max (1, \mathbb{N})$ if $\mathrm{S} \mathbb{D} E=\mathrm{L}$ 'or B'; LDVL >= 1 otherw ise.

VR (input/output)
On entry, if $\mathbb{N} \operatorname{ITV}=U$ 'and $S \mathbb{D} E=R$ 'or $B^{\prime}, V R$ $m$ ust contain starting vectors for the inverse iteration for the righteigenvectors; the starting vector for each eigenvectorm ustbe in the sam e colum $n$ in which the eigenvector will be stored. On exit, if $S \mathbb{D} E=R$ 'or $B$ ', the righteigenvectors specified by SELEC T w illbe stored consecutively in the colum ns of $V R$, in the sam e order as theireigenvalues. If $S \mathbb{D} E=L^{\prime}, V R$ is not referenced.

LDVR (input)
The leading dim ension of the array $V R$. LDVR >= $\max (1, \mathbb{N})$ if $S \mathbb{D} E=R$ 'or $B^{\prime} ; L D V R>=1$ otherw ise.

M M (input)
The num berof colum ns in the arrays VL and/or VR. $M M>=M$.

M (output)
The num ber of colum ns in the arrays VL and/or VR required to store the eigenvectors $\vDash$ the num ber of TRUE . elem ents in SELECT).

W ORK (w orkspace)
dim ension $(\mathbb{N} * N)$
RW ORK (w orkspace)
dim ension (N)
FAILL (output)
IfSDE $=\mathrm{L}$ 'or B ', $\mathbb{F A} A L L(i)=j>0$ if the left eigenvector in the $i$-th column of VL (comesponding to the eigenvalue w ( $j$ ) failed to converge; $\mathbb{F A} \mathbb{I L}($ (i) $=0$ if the eigenvectorconverged satisfactorily. If $S \mathbb{D} E=R$ ', $\mathbb{F A} \mathbb{I L}$ is not referenced.

FA IIR (output)
If $S \mathbb{D} E=R$ 'or $B^{\prime}, \mathbb{F} A \mathbb{L} R(i)=j>0$ if the
right eigenvector in the i-th colum $n$ of VR (comesponding to the eigenvalue w ( $~(~) ~ f a i l e d ~ t o ~$ converge; $\mathbb{F A}$ IIR (i) = 0 ifthe eigenvectorconverged satisfactorily. IfSDE $E=\mathbb{L}, \mathbb{F} A \Pi R$ is notreferenced.

INFO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N} F O=i$, is the num ber of eigenvectors which failed to converge; see $\mathbb{F} A \mathbb{I} L$ and $\mathbb{F} A \mathbb{I} R$ for further details.

## FURTHER DETAILS

E ach eigenvector is norm alized so that the elem entof largest $m$ agnitude has $m$ agnitude 1 ; here the $m$ agnitude of a com plex num ber $(x, y)$ is taken to be $|x|+|y|$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

chseqr - com pute the eigenvalues of a com plex upper H essenberg $m$ atrix $H$, and, optionally, the $m$ atrices $T$ and $Z$ from the Schurdecom position $H=Z$ T $\mathrm{Z} * * \mathrm{H}$, where T is an upper triangular $m$ atrix (the Schur form ), and $Z$ is the unitary $m$ atrix of Schurvectors

## SYNOPSIS

```
SU BROUT\mathbb{NE CHSEQR(JOB,COMPZ,N,IO, HHI,H,LDH,W,Z,LD Z,W ORK,}
    LW ORK,\mathbb{NFO)}
CHARACTER * 1 JOB,COMPZ
COM PLEX H (LDH,*),W (*),Z (LDZ,*),W ORK (*)
\mathbb{NTEGER N,\mathbb{LO,}\mathbb{H}I,LDH,LD Z,LW ORK,INFO}
SUBROUT\mathbb{NE CHSEQR_64(JOB,COM PZ,N, HO, HHI,H,LDH,W,Z,LD Z,}
    W ORK,LW ORK,\mathbb{NFO)}
CHARACTER * 1 OB,COM PZ
COM PLEX H (LDH,*),W (*),Z (LDZ,*),W ORK (*)
\mathbb{NTEGER*8N,}\mathbb{NO},\mathbb{H}I,LDH,LDZ,LW ORK,\mathbb{NFO}
```


## F95 INTERFACE

 $\mathbb{W}$ ORK ],LW ORK, $[\mathbb{N} F O]$ )

CHARACTER (LEN=1)::JOB,COM PZ
COM PLEX,D $\mathbb{I}$ ENSION (:) ::W ,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) :: H, Z
$\mathbb{N}$ TEGER :: $\mathrm{N}, \mathbb{I} \mathrm{O}, \mathbb{H} \mathrm{I}, \mathrm{LD} \mathrm{H}, \mathrm{LD} \mathrm{Z}, \mathrm{LW}$ ORK, $\mathbb{N} F O$
SU BROUTINE HSEQR_64 (JOB,COMPZ,N, $\mathbb{L O}, \mathbb{H} I, H,[L D H], W, Z,[L D Z]$,
[W ORK],LW ORK, [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1):: JOB ,COM PZ
COM PLEX,D $\mathbb{M}$ ENSION (:) ::W ,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) :: H,Z
$\mathbb{N}$ TEGER (8) ::N, $\mathbb{L O}, \mathbb{H}$ I, LD H ,LD Z, LW ORK , $\mathbb{N}$ FO

## C INTERFACE

\#include <sunperfh>
void chseqr (char, char, int, int, int, com plex*, int, com plex^, com plex*, int, int^);
void chseqr_64 (char, char, long, long, long, com plex*, long, com plex*, com plex*, long, long*);

## PURPOSE

chseqr com putes the eigenvalues of a com plex upper H essenberg $m$ atrix $H$, and, optionally, the $m$ atrices $T$ and $Z$ from the Schurdecom position $H=Z \mathrm{~T} \mathrm{Z**H}$, where T is an upper triangular $m$ atrix (the Schur form ), and $Z$ is the unitary $m$ atrix of Schurvectors.

O ptionally Z may be postm ultiplied into an input unitary $m$ atrix $Q$, so that this routine can give the Schur factorization of a m atrix A which has been reduced to the $H$ essenberg form $H$ by the unitary $m$ atrix $Q: A=Q * H * Q * H=$ (Q Z ) * $\mathrm{T}^{*}(\mathrm{Q} \mathrm{Z}) * * \mathrm{H}$.

## ARGUMENTS

$J O B$ (input)
= E ': com pute eigenvalues only;
$=S$ ': com pute eigenvalues and the Schur form $T$.

COM PZ (input)
= N ': no Schurvectors are com puted;
$=$ ' $I$ ': Z is initialized to the unitm atrix and the m atrix Z of Schurvectors of H is retumed; $=\mathrm{V}$ : $Z \mathrm{~m}$ ust contain an unitary m atrix Q on entry, and the product $\mathrm{Q} * \mathrm{Z}$ is retumed.

N (input) The order of the m atrix $\mathrm{H} . \mathrm{N}>=0$.
\#O (input)
It is assum ed that $H$ is already upper triangular
in row $s$ and colum ns 1: $\mathrm{HO}-1$ and $\mathbb{H} \mathrm{I}+1 \mathrm{~N} . \mathbb{H O}$ and IH I are norm ally setby a previous call to C G EBA L, and then passed to CGEHRD when the $m$ atrix output by CGEBAL is reduced to $H$ essenberg form . O therw ise HO and $\mathbb{H}$ I should be set to 1 and $N$ respectively. $1<=\mathbb{H O}<=\mathbb{H} \mathrm{I}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{H} \mathrm{O}=1$ and $\mathbb{H} \mathrm{I}=0$, if $\mathrm{N}=0$.

IH I (input)
See the description of IIO .

H (input/output)
On entry, the upper $H$ essenberg $m$ atrix $H$. On exit, if $\mathrm{JOB}=\mathrm{S}^{\prime}, \mathrm{H}$ contains the uppertriangular $m$ atrix $T$ from the Schurdecom position the Schur form). If $J O B=E$ ', the contents of $H$ are unspecified on exit.

LD H (input)
The leading dim ension of the aray H. LD H >= $\max (1, \mathbb{N})$.

W (output)
The com puted eigenvalues. If $J O B=S$ ', the eigenvalues are stored in the sam e order as on the diagonal of the Schur form retumed in H , w th $W(i)=H(i, i)$.

Z (input) IfCOM PZ = N ': Z is not referenced.
If COMPZ = I': on entry, Z need notbe set, and on exit, $Z$ contains the unitary $m$ atrix $Z$ of the Schurvectors of H . IfCOM PZ = V : on entry Z $m$ ust contain an $N$ by $-\mathrm{N} m$ atrix $Q$, which is assum ed to be equal to the unitm atrix except for the sub$m$ atrix $Z$ ( $\mathbb{H} O: \mathbb{H} I, \mathbb{I} O: \mathbb{H} I$ ); on exitZ contains $Q$ *Z . N orm ally $Q$ is the unitary $m$ atrix generated by CUNGHR after the call to CGEHRD which form ed the H essenberg $m$ atrix $H$.

LD $Z$ (input)
The leading dim ension of the aray $Z$. LD $Z>=$ $\max (1, N)$ if COMPZ = I'orV';LD Z >= 1 otherw ise.

W ORK (w orkspace)
On exit, if $\mathbb{N F F O}=0, \mathrm{~W}$ ORK (1) retums the optim al LW ORK.

LW ORK (output)
The dimension of the aray $W$ ORK. LW ORK >=
$\max (1, N)$.

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N F O}=-i$, the $i$-th argum ent had an illegalvalue
$>0:$ if $\mathbb{N} F O=$ i, CHSEQR failed to compute all the eigenvalues in a total of $30 *$ ( $\mathbb{H}$ I- $\mathrm{HO}+1$ ) terations; elem ents 1 :ilo-1 and i+1 1 of $W$ contain those eigenvalues which have been successfully com puted.

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

cjadm m -Jagged diagonalm atrix-m atrix m ultiply (m odified Ellpack)

## SYNOPSIS

```
SUBROUT\mathbb{NECJADMM(TRANSA,M,N,K,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,PNTR,MAXNZ,\mathbb{PERM,}}\mathbf{N}=,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,M,N,K,DESCRA (5),MAXNZ,
* LDB,LDC,LWORK
\mathbb{NTEGER INDX NNZ),PNTR MAXNZ+1),\mathbb{PERM M)}}\mathbf{M}\mathrm{ (N)}
COM PLEX ALPHA,BETA
COMPLEX VALNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NECJADMM_64(TRANSA,M,N,K,A LPHA,DESCRA,}
* VAL,\mathbb{NDX,PNTR,MAXNZ,\mathbb{PERM,}}\mathbf{N},
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,M,N,K,DESCRA (5),M AXNZ,
* LDB,LDC,LW ORK
```



```
COMPLEX ALPHA,BETA
COM PLEX VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
```

where NN Z=PN TR M AXNZ+1)-PN TR (1)+1 is the num berofnon-zero elem ents

## F95 INTERFACE

SUBROUTINE JADMM (TRANSA, M, $\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N} D X$, * PNTR,MAXNZ, $\mathbb{P} E R M, B,[L D B], B E T A, C,[L D C],[W$ ORK], [LW ORK])
$\mathbb{N} T E G E R$ TRANSA, M, K, MAXNZ
$\mathbb{N} T E G E R, D \mathbb{I}$ ENSION (:) :: DESCRA, $\mathbb{N} D \mathrm{X}, \mathrm{PN} T \mathrm{R}, \mathbb{P} E R M$
COMPLEX ALPHA,BETA
COM PLEX,D $\mathbb{M}$ ENSION (:) :: VAL

SUBROUTINE JADMM_64(TRANSA,M, $\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N} D X$, * PNTR,MAXNZ, $\mathbb{P} E R M, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])$

INTEGER*8 TRANSA, M, K, MAXNZ
$\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O}(:):: D E S C R A, \mathbb{N} D X, P N T R, \mathbb{P E R M}$
COMPLEX ALPHA,BETA
COM PLEX ,D $\mathbb{I M}$ ENSION (:) :: VAL
COM PLEX , D $\mathbb{M}$ ENSION (:, :) :: B , C

## DESCRIPTION

$$
C<- \text { alpha op (A ) B + beta C }
$$

where A LPHA and BETA are scalar, $C$ and $B$ are dense $m$ atrioes, A is a m atrix represented in jagged-diagonal form at and $o p(A)$ is one of $o p(A)=A$ or $o p(A)=A^{\prime}$ or $o p(A)=\operatorname{conjg}\left(A^{\prime}\right)$.
( 'indicates m atrix transpose)

## ARGUMENTS

TRANSA Indicates how to operate $w$ th the sparse $m$ atrix
0 : operate $w$ ith $m$ atrix
1 : operate $w$ th transpose $m$ atrix
2 : operate $w$ ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M $\quad$ um berof row $s$ in matrix A

N $N$ um berof colum ns in matrix C

K $\quad \mathrm{N}$ um berof colum ns in m atrix A

A LPH A Scalarparam eter

D ESCRA () D escriptor argum ent. Fíve elem ent integer anay
DESCRA (1) m atrix structure
0 : general
1 : symm etric ( $A=A$ )
2 : Herm itian ( $A=\operatorname{CONJ}(A))$
3 :Triangular
4 : Skew (Anti)-Symm etric ( $A=-A$ )
5 :D iagonal
6 : Skew Herm itian ( $\mathrm{A}=-\mathrm{CONJ}(\mathrm{A})$ )
D ESCRA (2) upper/low er triangular indicator
1 : low er

2 :upper
DESCRA (3) main diagonal type
0 :non-unit
1 :unit
DESCRA (4) A ray base $\mathbb{N} O T \mathbb{I}$ PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
D ESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 :no repeated indices

VAL () array of length NN Z consisting of entries of A. VA L can be view ed as a colum $n m$ ajorordering of a row perm utation of the Ellpack representation of , where the Ellpack representation is perm uted so that the row s are non-increasing in the num ber of nonzero entries. V alues added forpadding in E llpack are not included in the Jagged-D iagonal form at.
$\mathbb{I N D X}$ () array of length NNZ consisting of the colum n indices of the comesponding entries in VAL.
PN TR () array of length M AXNZ+1, where PNTR (I)-PN TR (1)+1 points to the location in VAL of the firstelem ent in the row -perm uted E llpack represenation of $A$.

MAXNZ max num berofnonzeros elem ents per row .
$\mathbb{P E R M} 0$ integer array of length $M$ such that $I=\mathbb{P E R M}$ ( $I$ ), where row I in the originalE llpack representation corresponds to row I' in the perm uted representation. If $\operatorname{PERM}(1)=0$, it is assum ed by convention that $\mathbb{P E R M}(\mathbb{I})=I . \mathbb{P E R M}$ is used to determ ine the order in which row s ofC are updated.

B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension of $B$
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension of $C$
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is notreferenced in the current version.

## SEE ALSO

N IST FO RTRA N Sparse B las U ser's G uide available at: http://m ath nistgov/n cso/Staff/k Rem ington/Ispblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

c’adup - right perm utation of a jagged diagonalm atrix

## SYNOPSIS

SUBROUT $\mathbb{N} E C J A D R P(T R A N S P, M, K, V A L, ~ \mathbb{N D X}$, PNTR, MAXNZ, * $\quad$ PERM, WORK,LW ORK )
$\mathbb{N}$ TEGER TRANSP, M, K, MAXNZ,LW ORK
$\mathbb{N} T E G E R \quad \mathbb{N} D X(*), \operatorname{PNTR}(M A X N Z+1), \mathbb{P E R M}(\mathbb{K}), W$ ORK (LW ORK) COMPLEX VAL (*)

SUBROUTINECJADRP_64 (TRANSP, M, K, VAL, $\mathbb{N} D X, P N T R, M A X N Z$, * $\quad \mathbb{P E R M}, \mathrm{WORK}$,LWORK)
$\mathbb{N}$ TEGER*8 TRANSP, M, K, MAXNZ,LW ORK
$\mathbb{N T E G E R * 8} \mathbb{N} D X(*), \operatorname{PNTR}(M A X N Z+1), \mathbb{P E R M}(K), W$ ORK (LWORK) COMPLEX VAL (*)

## F95 INTERFACE

SUBROUTINE JADRP (TRANSP, M, K, VAL, $\mathbb{N} D X, P N T R, M A X N Z$, * $\quad \mathbb{P} E R M,[W O R K],[L W O R K])$
$\mathbb{N}$ TEGER TRANSP, M, K, MAXNZ
$\mathbb{N}$ TEGER,D $\mathbb{M}$ ENSION (:) :: $\mathbb{N D} \mathrm{X}, \mathrm{PNTR}, \mathbb{P E R M}$
COM PLEX , D $\mathbb{M}$ ENSION (:) ::VAL

SUBROUTINE JADRP_64 (TRANSP, M, K, VAL, $\mathbb{N} D \mathrm{X}, \mathrm{PN} T \mathrm{R}, \mathrm{M} A X N Z$, * $\quad \mathbb{P} E R M,[W O R K],[L W O R K])$

IN TEGER * 8 TRANSP, M, K, M AXNZ
$\mathbb{N}$ TEGER*8,D $\mathbb{M}$ ENSION (:) :: $\mathbb{N} D \mathrm{X}, \mathrm{PNTR}, \mathbb{P E R M}$
COM PLEX , D $\mathbb{M}$ ENSION (:) ::VAL

DESCRIPTION

A $<-A P$
$A<-A P^{\prime}$
( 'indicates m atrix transpose)
$w$ here perm utation $P$ is represented by an integervector $\mathbb{P} E R M$, such that $\mathbb{P E R M}(I)$ is equal to the position of the only nonzero elem entin row Iofperm utation $m$ atrix $P$.

N O TE : In orderto get a sym etrically perm uted jagged diagonal $m$ atrix P A P', one can explicitly perm ute the colum ns P A by calling

SJADRP ( $0, M, M, V A L, \mathbb{N} D X, P N T R, M A X N Z, \mathbb{P} E R M, W$ ORK,LW ORK)
where param eters $V A L, \mathbb{N D X}, P N T R, M A X N Z, \mathbb{P E R M}$ are the representation of $A$ in the jagged diagonal form at. The operation $m$ akes sense if the originalm atrix $A$ is square.

## ARGUMENTS

TRAN SP Indicates how to operate $w$ ith the perm utation $m$ atrix
0 : operate w ith m atrix
1 : operate $w$ ith transpose $m$ atrix

M $\quad \mathrm{N}$ um berof row $s$ in matrix A

K $\quad \mathrm{N}$ um ber of colum ns in matrix A

VAL () amay of length PNTR MAXNZ+1)-PNTR (1) consisting of entries ofA. VA L can be view ed as a colum $n m$ ajor ordering of a row perm utation of the E llpack representation of A, w here the Ellpack representation is perm uted so that the row $s$ are non-increasing in the num ber of nonzero entries. $V$ alues added for padding in Ellpack are not included in the Jagged - D iagonal form at.

INDX () array of length PN TR MAXNZ+1)-PNTR (1) consisting of the colum $n$ indices of the corresponding entries in VAL.

PNTR () array of length M AXNZ+1, where PNTR (I) PNTR (1)+1 points to the location in VA L of the firstelem ent in the row -perm uted E lhpack represenation of .

M A X N Z max num ber ofnonzeros elem ents per row.
$\mathbb{P} E R M$ ( integeramay of length $K$ such that $I=\mathbb{P} E R M$ ( $I$ ).

A ray $\mathbb{P} E R M$ represents a perm utation $P$, such that $\mathbb{P E R M}$ ( I ) is equal to the position of the only nonzero elem ent in row Iofperm utation $m$ atrix $P$.
Forexam ple, if
|001|
$\mathrm{P}=\left|\begin{array}{lll}1 & 0 & 0\end{array}\right|$
|010|
then $\mathbb{P E R M}=(3,1,2)$.

W ORK () scratch array of length LW ORK. LW ORK should be at leastK.

LW ORK length ofW ORK aray

If LW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

## SEE ALSO

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/K Rem ington/tspblas/
"D ocum ent for the B asic $L$ inearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib org/utk/papers/sparse _ps

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

```
cj̇dsm -Jagged-diagonal form at triangular solve
```


## SYNOPSIS

```
SUBROUTINE CJADSM(TRANSA,M ,N,UN ITD,DV,ALPHA,DESCRA,
* VAL,\mathbb{NDX,PNTR,MAXNZ,\mathbb{PERM,}}\mathbf{N},
* B,LDB,BETA,C,LDC,W ORK,LWORK)
INTEGER TRANSA,M,N,UNITD,DESCRA (5),MAXNZ,
* LDB,LDC,LW ORK
\mathbb{NTEGER INDX NNZ),PNTR MAXNZ+1),\mathbb{PERM M)}}\mathbf{M}\mathrm{ (N)}
COMPLEX ALPHA,BETA
COM PLEX DV M),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NECJADSM_64(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,}
* VAL,NDX,PNTR,MAXNZ,\mathbb{PERM,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,M,N,UNITD,DESCRA (5),MAXNZ,}
* LDB,LDC,LWORK
INTEGER*8 INDX (NNZ),PNTR MAXNZ+1),\mathbb{PERM M)}
COM PLEX ALPHA,BETA
COM PLEX DV M),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
```

where NN Z=PN TR M A XNZ+1)-PN TR (1)+1 is the num berofnon-zero elem ents

## F95 INTERFACE

SUBROUTINE JADSM (TRANSA, M, $\mathbb{N}], U N \mathbb{I T D}, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X$,

* PNTR,MAXNZ, $\mathbb{P E R M}, \mathrm{B},[\mathrm{LD} B], B E T A, C,[L D C],[W$ ORK ], [LW ORK ])
$\mathbb{N} T E G E R$ TRANSA, M, MAXNZ
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: DESCRA, $\mathbb{N} D \mathrm{X}, \mathrm{PN} T \mathrm{R}, \mathbb{P} E R M$
COM PLEX ALPHA,BETA
COM PLEX,D $\mathbb{M}$ ENSION (:) :: VAL,DV
COMPLEX,D $\mathbb{I M} E N S I O N(:,:):$ B, C

SUBROUT $\mathbb{N} E \operatorname{JADSM}$ _64 (TRANSA, M, $\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X$,

* PNTR,MAXNZ, $\mathbb{P E R M}, \mathrm{B},[\mathrm{LDB}], B E T A, C,[L D C],[W O R K],[L W O R K])$
$\mathbb{N} T E G E R * 8$ TRANSA, M, MAXNZ
$\mathbb{N} T E G E R * 8, D \mathbb{I M} E N S \mathbb{O}(:):: D E S C R A, \mathbb{N} D X, P N T R, \mathbb{P E R M}$
COMPLEX ALPHA,BETA
COM PLEX, D $\mathbb{I M} E N S I O N(:):: V A L, D V$
COM PLEX , D $\mathbb{M}$ ENSION (:, :) :: B , C


## DESCRIPTION

$$
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { Op (A) B + BETA C } \\
& C<-A L P H A \text { Op (A)D B + BETA } C
\end{aligned}
$$

where A LPH A and BETA are scalar, $C$ and $B$ are $m$ by $n$ dense $m$ atrices, $D$ is a diagonalscaling $m$ atrix, $A$ is a unit, ornon-unit, upper or low ertriangularm atrix represented in jagged-diagonal form at and $o p(A)$ is one of op (A) $)=\operatorname{inv}(A)$ or op (A $)=\operatorname{inv}(A)$ or op (A) $=\operatorname{inv}\left(\infty n \dot{g}\left(A^{\prime}\right)\right)$ (inv denotesm atrix inverse, 'indicates m atrix transpose)

## ARGUMENTS

TRAN SA Indicates how to operate $w$ th the sparse $m$ atrix 0 : operate $w$ ith $m$ atrix 1 : operate $w$ ith transpose $m$ atrix
2 : operate $w$ ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M $\quad N$ um berof row $s$ in $m$ atrix A

N $\quad \mathrm{N}$ um berof colum ns in $m$ atrix $C$

UN ITD Type of scaling:
1 : Identity matrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum $n$ scaling)
4 : A utom atic row scaling (see section N O TES for further details)

DV () A rray of length M containing the diagonalentries of the scaling diagonalm atrix D .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay D ESCRA (1) m atrix structure

0 : general
1 : symm etric ( $A=A$ )
2 : Herm ( $\mathrm{A}=\mathrm{CONJG}$ (A))
3 :Triangular
4 : Skew (A nti)-Symm etric ( $A=-A$ )
5 :D iagonal
6 : Skew Herm itian ( $\mathrm{A}=-\mathrm{CON}$ J ( A ) )
N ote:For the routine, DESCRA (1)=3 is only supported.
DESCRA (2) upper/low er triangular indicator
1: low er
2 : upper
DESCRA (3) m ain diagonaltype
0 :non-unit
1 : unit
DESCRA (4) A ray base $\mathbb{N} O T \mathbb{M}$ PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? $\mathbb{N} O T \mathbb{I}$ PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL 0 array of length NNZ consisting of entries of A. VA L can be view ed as a colum $n m$ ajorordering of a row perm utation of the Ellpack representation of A, where the Ellpack representation is perm uted so that the row s are non-increasing in the num ber of nonzero entries. V alues added forpadding in E llpack are not included in the Jagged-D iagonal form at.
$\mathbb{N} D \mathrm{X} 0 \quad$ array of length NNZ consisting of the colum n indices of the comesponding entries in VAL.

PNTR 0) array of length M AXNZ +1 , where PNTR ( 1 ) PNTR (1) +1 points to the location in VA L of the firstelem ent in the row -perm uted $E$ lipack represenation of $A$.

MAXNZ max num berofnonzeros elem ents per row .
$\mathbb{P E R M}$ ) integer array of length M such that $\mathrm{I}=\mathbb{P} E R M$ ( I ), where row I in the originalE llpack representation corresponds to row I' in the perm uted representation. If $\operatorname{PERM}(\mathbb{1})=0$, its assum ed by convention that $\mathbb{P E R M}(\mathbb{I})=I . \mathbb{P E R M}$ is used to determ ine the order in which row s ofC are updated.

B 0 rectangular array w th first dim ension LD B.

LD B leading dim ension ofB
BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension of $C$

W ORK () scratch array of length LW ORK.
On exit, ifLW ORK $=-1, W$ ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK aray. LW ORK should be at least2*M.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on multiple processors, LW ORK $>=2 * \mathrm{M}$ *N_CPUS where N_CPUS is the maxim um num berof processors available to the program.

If LW ORK $=0$, the routine is to allocate $w$ orkspace needed.
IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

## SEE ALSO

N IST FORTRAN Sparse B las U ser's G uide available at:
http://m ath nistgov/m csd/Staff/k Rem ington/Ispblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

## NOTES/BUGS

1. No test for singularity ornear-singularity is included in this routine. Such tests $m$ ust.be perform ed before calling this routine.
2. If U N ITD $=4$, the routine scales the row s ofA such that their 2 -norm s are one. The scaling $m$ ay im prove the accuracy of the com puted solution. C orresponding entries ofV A L are changed only in the particular case. On retum DV m atrix stored as a vector contains the diagonalm atrix by w hich the row s have been scaled. U N ITD = 2 should be used for the next calls to the routine w ith overw ritten VAL and DV.

WORK (1)=0 on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here $i$ is the row num berw hich 2 -norm is exactly zero.
3. If $\operatorname{DESCRA}(3)=1$ and UN ITD < 4, the unitdiagonalelem ents $m$ ightorm ightnotbe referenced in the JA D representation of a sparse m atrix. They are notused anyw ay in these cases. ButifUN ITD=4, the unit diagonalelem ents M U ST be referenced in the $\sqrt{A} D$ representation.
4.The routine can be applied for solving triangular system $s$ w hen the upper or low er triangle of the general sparse m atrix A is used. H ow ever DESCRA (1) m ust.be equal to 3 in this case.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

clarz -applie a com plex elem entary reflector $H$ to a com plex M -by-N m atrix C, from either the leftor the right

## SYNOPSIS



```
CHARACTER * 1 SDE 
COMPLEX TAU
COM PLEX V (*),C (LDC,*),W ORK (*)
INTEGER M,N,L, INCV,LDC
```



```
CHARACTER * 1SDE
COMPLEX TAU
COM PLEX V (*),C (LDC,*),W ORK (*)
INTEGER*8M,N,L,INCV,LDC
```


## F95 INTERFACE

SU BROUT $\mathbb{N} E$ LARZ (S $\mathbb{D} E, \mathbb{M}], \mathbb{N}], L, V,[\mathbb{N} C V], T A U, C,[L D C],[W O R K])$
CHARACTER (LEN=1) ::SDE
COM PLEX ::TAU
COM PLEX,D $\mathbb{M}$ ENSION (:) ::V,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) :: C
$\mathbb{N} T E G E R:: M, N, L, \mathbb{N} C V, L D C$
SU BROUTINE LARZ_64 (SDE, $\mathbb{M}], \mathbb{N}], L, V,[\mathbb{N} C V], T A U, C,[L D C],[W O R K])$

CHARACTER (LEN=1) ::SDE

COM PLEX ::TAU
COM PLEX,D $\mathbb{M}$ ENSION (:) ::V ,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::C
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{L}, \mathbb{N} C V, L D C$

## C INTERFACE

\#include <sunperfh>
void clarz (char side, intm, intn, intl, com plex *v, int incv, com plex *tau, com plex *c, int lddc);
void clarz_64 (char side, long m , long n, long l, com plex *v, long incv, com plex *tau, com plex *c, long ldc);

## PURPOSE

clarz applies a com plex elem entary reflectorH to a com plex M -by -N m atrix C , from either the leftor the right. H is represented in the form

$$
\mathrm{H}=\mathrm{I}-\tan * \mathrm{~V}^{*} \mathrm{~V}^{\prime}
$$

$w$ here tau is a com plex scalar and $v$ is a com plex vector.

If tau $=0$, then H is taken to be the unitm atrix.

To apply H ' the conjugate transpose of H), supply con'j (tau) instead tau.

H is a product ofk elem entary reflectors as retumed by CTZRZF.

## ARGUMENTS

```
S\mathbb{DE (input)}
\(=\mathrm{L}:\) form \(\mathrm{H} * \mathrm{C}\)
\(=R\) : form \(C\) * \(H\)
```

M (input) The num ber of row s of the $m$ atrix $C$.

N (input) The num ber of colum ns of the $m$ atrix C .

L (input) The num berofentries of the vector $V$ containing the $m$ eaningful part of the $H$ ouseholdervectors. If $S \mathbb{D} E=L \prime, M>=L>=0$, if $S D E=R \prime, N>=L$ $>=0$.

V (input) The vector v in the representation of H as
retumed by CTZRZF.V is notused ifTAU $=0$.
$\mathbb{N} C V$ (input)
The increm entbetw een elem ents of $v . \mathbb{N} C V<>0$.

TAU (input)
The value tau in the representation of $H$.

C (input/output)
On entry, the $M$-by $N$ matrix C. On exit, $C$ is overw ritten by the $m$ atrix $H$ * $C$ if $S \mathbb{D E}=\mathrm{L}$ ', or $C * H$ if $S \mathbb{D} E=R$ '.

LDC (input)
The leading dim ension of the aray C. LD C >= $m a x(1, M)$.

W ORK (w orkspace)
$(\mathbb{N})$ if $S \mathbb{D} E=L^{\prime}$ or $\left.M\right)$ if $S \mathbb{D} E=R^{\prime}$

## FURTHER DETAILS

B ased on contributions by
A.Petitet, C om puter Science D ept., U niv . of Tenn., K noxville, U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

clarzb - applie a com plex block reflectort or its transpose H **H to a com plex distributed M -by-N C from the leftor the right

## SYNOPSIS

SUBROUTINE CLARZB (SDE,TRANS,D $\mathbb{R E C T}, S T O R E V, M, N, K, L, V, L D V, T$, LDT, C,LDC,W ORK,LDWORK)

CHARACTER * 1 SDE, TRANS,D $\mathbb{R E C T}, S T O R E V$

$\mathbb{N}$ TEGER M , N , K, L, LDV ,LD T,LD C ,LDW ORK
SU BROUTINECLARZB_64 (SDE,TRANS,D $\mathbb{R E C T}, S T O R E V, M, N, K, L, V, L D V$, T,LDT, C,LDC,W ORK,LDW ORK)

CHARACTER * $1 \mathrm{~S} \mathbb{D} E, T R A N S, D \mathbb{R E C T}, S T O R E V$
COM PLEX V (LDV , $)$, T (LDT,*), C (LDC,$\star), W$ ORK (LDW ORK,$\star)$
$\mathbb{N}$ TEGER*8M,N,K,L,LDV,LD T,LDC,LDW ORK

## F95 INTERFACE

SU BROUTINE LARZB (SDE,TRANS,D $\mathbb{R E C T}$,STOREV, $\mathbb{M}], \mathbb{N}], K, L, V,[L D V]$, T, [LDT], C, [LDC], [W ORK], [LDW ORK])

CHARACTER (LEN=1) ::SDE,TRANS,D $\mathbb{R E C T}$, STOREV
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::V,T,C,W ORK
$\mathbb{N}$ TEGER :: M , N , K, L, LDV ,LD T, LD C, LD W ORK
SU BROUTINE LARZB_64 (SDE,TRANS,D $\mathbb{R E C T}, S T O R E V, \mathbb{M}], \mathbb{N}], K, L, V$, [LDV],T, [LDT], C, [LDC], [W ORK], [LDW ORK])

CHARACTER (LEN=1) ::SDE,TRANS,D $\mathbb{R E C T}$, STO REV
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::V,T,C,W ORK
$\mathbb{N}$ TEGER (8) ::M , N, K,L,LDV,LD $, ~ L D C, L D W$ ORK

## C INTERFACE

\#include <sunperfh>
void clarzb (char side, char trans, char direct, char storev, int $m$, intn, intk, intl, com plex *v, int ldv, com plex *t, intldt, com plex ${ }^{*}$ c, int ldc, int ldw ork);
void clarzb_64 (charside, char trans, char direct, char storev, long m , long n , long k , long l , com plex *v, long ldv, com plex *t, long ldt, com plex *c, long ldc, long ldw ork);

## PURPOSE

clarzb applies a com plex block reflector H or its transpose $\mathrm{H} * * \mathrm{H}$ to a com plex distributed M -by -N C from the leftorthe right.

Cumently, only STOREV = R'and D $\mathbb{R E C T}=\mathrm{B}$ 'are supported.

## ARGUMENTS

STDE (input)
= L': apply H orH 'from the Left
= R':apply H orH 'from the Right
TRANS (input)
= N ': apply H N o transpose)
= C ': apply H ' (C onjugate transpose)

D $\mathbb{R E C T}$ (input)
Indicates how H is form ed from a product of elem entary reflectors = F ': H = H (1) H (2) . . . H (k) (Forw ard, not supported yet)
= $\mathrm{B}^{\prime}: \mathrm{H}=\mathrm{H}(\mathrm{k}) \ldots \mathrm{H}(2) \mathrm{H}(1)$ (Backw ard)

STOREV (input)
Indicates how the vectors w hich define the elem en-
tary reflectors are stored:
= C':Colum nw ise (notsup-
ported yet)
= R ':Rowwise

M (input) The num ber of row s of the $m$ atrix $C$.

N (input) The num ber of colum ns of the m atrix C .

K (input) The order of the $m$ atrix T ( $=$ the num ber of elem entary reflectors whose product defines the block reflector).

L (input) The num berof colum ns of the $m$ atrix $V$ containing the $m$ eaningfilpart of the $H$ ouseholder reflectors. If $S \mathbb{D} E=L ', M>=L>=0$, if $S \mathbb{D} E=R \prime, N>=L$ $>=0$ 。
V (input) If $S T O R E V=C^{\prime}, N V=K$; if $S T O R E V=R \prime, N V=L$.

LDV (input)
The leading dim ension of the aray $V$. IfSTOREV = C',LDV >=L; ifSTOREV = R',LDV >=K .

T (input) The triangular K -by K m atrix T in the representation of the block reflector.

LD T (input)
The leading dim ension of the anay $\mathrm{T} \cdot \mathrm{LD} \mathrm{T}>=\mathrm{K}$.

C (input/output)
On entry, the M -by -N m atrix C . On exit, C is overw ritten by H * C orH * C or $\mathrm{C}^{*} \mathrm{H}$ orC *H '.

LD C (input)
The leading dim ension of the array $C . \operatorname{LDC}>=$ $\max (1, M)$.

W ORK (w orkspace)
dim ension ( $\mathrm{M} A X(M, N), K)$

LDW ORK (input)
The leading dim ension of the array $W$ ORK. If $S \mathbb{D} E$
$=\mathrm{L} \prime$, LDW ORK >= max $(1, N)$; ifS $\mathbb{D} E=R \prime$ LDW ORK $>=\max (1, M)$.

## FURTHER DETAILS

B ased on contributions by
A. Petitet, C om puter Science D ept., U niv . ofTenn ., K noxville, U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

clarzt-form the triangular factor $T$ of a com plex block reflector $H$ of order> $n$, which is defined as a product ofk elem entary reflectors

## SYNOPSIS

```
SUBROUT\mathbb{NE CLARZT D RECT,STOREV,N,K,V,LDV,TAU,T,LDT)}
CHARACTER * 1 D RECT,STOREV
COM PLEX V (LDV,*),TAU (*),T (LDT,*)
INTEGERN,K,LDV,LDT
SUBROUT\mathbb{NECLARZT_64(D RECT,STOREV,N,K,V,LDV,TAU,T,LDT)}
CHARACTER * 1D RECT,STOREV
COM PLEX V (LDV,*),TAU (*),T (LDT,*)
INTEGER*8N,K,LDV,LDT
```

F95 INTERFACE
SU BROUTINE LARZT © $\mathbb{R E C T}, \operatorname{STOREV}, \mathrm{N}, \mathrm{K}, \mathrm{V},[\operatorname{LDV}], T A U, T,[L D T])$
CHARACTER (LEN=1) ::D RECT,STOREV
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::TAU
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::V,T
$\mathbb{N} T E G E R:: N, K, L D V, L D T$
SU BROUTINE LARZT_64 D $\mathbb{R E C T}, \operatorname{STOREV}, N, K, V,[L D V], T A U, T,[L D T])$
CHARACTER (LEN=1) ::D $\mathbb{R E C T}$, STOREV
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::TAU
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::V,T
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{K}, \mathrm{LD} V$,LD T

## C INTERFACE

\#include <sunperfh>
void clarzt(chardirect, char storev, intn, intk, com plex ${ }^{*} \mathrm{v}$, int $l d \mathrm{v}$, com plex *tau, com plex *t, int ldt);
void clarzt 64 (chardirect, char storev, long n, long k, complex ${ }^{*}$ v, long $l d v, ~ c o m p l e x ~ * t a u, ~ c o m p l e x ~ * t, ~$ long ldt);

## PURPOSE

clarzt form s the triangular factor T of a com plex block reflectorH of order> n,which is defined as a product ofk elem entary reflectors.

IfD $\mathbb{R E C T}=\mathrm{F}^{\prime}, \mathrm{H}=\mathrm{H}(1) \mathrm{H}(2) \ldots \mathrm{H}(\mathrm{k})$ and T is upper triangular;

IfD $\mathbb{R E C T}=\mathrm{B}^{\prime}, \mathrm{H}=\mathrm{H}(\mathrm{k}) \ldots \mathrm{H}(2) \mathrm{H}(1)$ and T is lower triangular.

IfSTOREV = C', the vector which defines the elem entary reflector $\mathrm{H}(\mathrm{i})$ is stored in the $i$-th colum n of the amray V , and

$$
H=I-V \star T * V^{\prime}
$$

IfSTOREV = R', the vector which defines the elem entary reflectorH (i) is stored in the $i$-th row of the array $V$, and

$$
\mathrm{H}=\mathrm{I}-\mathrm{V}^{\prime} \star \mathrm{T} \star \mathrm{~V}
$$

Curently, only STOREV = R'and D $\mathbb{R E C T}=B$ 'are supported.

## ARGUMENTS

D $\mathbb{R E C T}$ (input)
Specifies the order in which the elem entary
reflectors are multiplied to form the block
reflector:
$=F ': H=H(1) H(2) \ldots H(k)$ (Forw ard, notsup-
ported yet)
$=B^{\prime}: H=H(k) \ldots H(2) H(1)$ (Backw ard)

## STOREV (input)

Specifies how the vectors w hich define the elem entary reflectors are stored (see also Further
D etails):
= R ': row w ise
N (input) The order of the block reflector $\mathrm{H} . \mathrm{N}>=0$.
$K$ (input) The order of the triangular factor $T \vDash$ the num ber of elem entary reflectors). $\mathrm{K}>=1$.
$V$ (input) ( $L D V, K$ ) if $S T O R E V=C^{\prime}(L D V, N)$ if $S T O R E V=R^{\prime}$ Them atrix $V$. See furtherdetails.

LD V (input)
The leading dim ension of the array V . If $\operatorname{STOREV}=$ $C$ ', LDV $>=\max (1, N)$; ifSTOREV = $R$ ', LDV $>=K$.

TAU (input)
TAU (i) must contain the scalar factor of the elem entary reflectort (i).
$T$ (input) The $k$ by $k$ triangular factor $T$ of the block reflector. If $\mathrm{D} \mathbb{R E C T}=\mathrm{F}^{\prime}, \mathrm{T}$ is upper triangular; if $\mathrm{D} \mathbb{R E C T}=\mathrm{B}$ ', T is low er triangular. The restof the aray is notused.

LD T (input)
The leading dim ension of the array T.LD T >=K.

## FURTHER DETAILS

B ased on contributions by
A. Petitet, C om puter Science D ept., U niv. of Tenn., K noxville, U SA

The shape of the $m$ atrix $V$ and the storage of the vectors which define the $H$ (i) is bestillustrated by the follow ing exam ple w th $\mathrm{n}=5$ and $\mathrm{k}=3$. The elem ents equal to 1 are not stored; the comesponding aray elem ents arem odified but restored on exit. The restof the array is notused.

D $\mathbb{R E C T}=\mathrm{F}^{\prime}$ and STOREV $=\mathrm{C}^{\prime}: \quad \mathrm{D} \mathbb{R E C T}=\mathrm{F}^{\prime}$ and STOREV = R':

```
                                    V
```



```
(v1 v2 v3) (v1 v1 v1 v1 v1 \ldots..1
)
    V = (v1 v2 v3 ) (v2 v2 v2 v2 v2 .
```

```
..1 )
    (v1 v2 v3 ) (v3 v3 v3 v3 v3 .
    .1 )
        (v1 v2 v3 )
        . . .
        1..
            1.
                1
D\mathbb{RECT}=\mp@subsup{B}{}{\prime}\mathrm{ 'andSTOREV = C': D PRECT = B' and}
STOREV = R':
    1
```



```
    . 1
        (1 . . ..v1 v1 v1 v1 v1 )
        . . }
v2 v2 v2 )
    ...
v3 v3 v3 )
        •••
    (v1 v2 v3 )
    V = (v1 v2 v3)
        (v1 v2 v3 )
```


## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

clatzm -routine is deprecated and has been replaced by routine CUNMRZ

## SYNOPSIS



```
CHARACTER * 1SDE
COMPLEX TAU
COM PLEX V (*),C1 (LDC ,*),C2 (LD C ,*),W ORK (*)
INTEGERM,N,\mathbb{NCV,LDC}
```



```
CHARACTER * 1SDE 
COMPLEX TAU
COM PLEX V (*),C1 (LD C ,*),C2 (LD C ,*),W ORK (*)
INTEGER*8M,N,INCV,LDC
```


## F95 INTERFACE

SU BROUTINE LATZM (SDE, $\mathbb{M}], \mathbb{N}], V,[\mathbb{N} C V], T A U, C 1, C 2,[L D C],[W$ ORK])
CHARACTER (LEN=1) ::SDE
COM PLEX ::TAU
COM PLEX,D $\mathbb{M}$ ENSION (:) ::V,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : ::C1,C2
$\mathbb{N} T E G E R:: M, N, \mathbb{N} C V, L D C$
SU BROUTINE LATZM_64 (SDE, $\mathbb{M}], \mathbb{N}], V,[\mathbb{N} C V], T A U, C 1, C 2,[\operatorname{DC}]$, [ W ORK])

CHARACTER (LEN=1)::SDE

COM PLEX ::TAU
COM PLEX,D $\mathbb{M}$ ENSION (:) ::V ,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::C1,C2
$\mathbb{N} T E G E R(8):: M, N, \mathbb{N} C V, L D C$

## C INTERFACE

\#include <sunperfh>
void clatzm (charside, intm, intn, com plex *v, int incv, com plex *tau, com plex *c1, com plex *c2, int ldc);
void clatzm _64 (char side, long m , long n, com plex *v, long incv, com plex *tau, com plex *c1, com plex *c2, long ldc);

## PURPOSE

clatzm routine is deprecated and has been replaced by routine CUNM RZ .

CLA TZM applies a H ouseholderm atrix generated by CTZRQF to a $m$ atrix.

LetP = I-tau*u*u', u=(1),
(v)
where $v$ is an ( $m-1$ ) vector if $S \mathbb{D} E=$ ' ', ora ( $n-1$ ) vector if $S \mathbb{D} E=R$.

If $S \mathbb{D} E$ equals $\mathbb{L}$ ', let
$C=[C 1] 1$
[C2]m-1
n
Then $C$ is overw ritten by $P * C$.

If $S \mathbb{D} E$ equals $R$ ', let $C=[C 1, C 2] m$
$1 \mathrm{n}-1$
Then C is overw rilten by C *P.

## ARGUMENTS

```
S\mathbb{DE (input)}
    = L': form P * C
    = R':form C * P
```

$M$ (input) The num ber of row s of the $m$ atrix $C$.

N (input) The num ber of colum ns of the $m$ atrix $C$.
$V$ (input) $(1+\mathbb{M}-1) * a b s(\mathbb{N C V}))$ if $S \mathbb{D} E=L^{\prime}(1+\mathbb{N}-$ 1)*abs ( $\mathbb{N} C V)$ ) if $S \mathbb{D} E=R$ 'The vectorv in the representation ofP. $V$ is notused if $T A U=0$.
$\mathbb{N} C V$ (input)
The increm entbetw een elem ents of v. $\mathbb{N} C V<>0$
TAU (input)
The value tau in the representation ofP.

C1 (input/output)
$(L D C, N)$ if $S \mathbb{D} E=L^{\prime}(M, \mathbb{1})$ if $S \mathbb{D} E=R^{\prime} O n$ entry, the $n$-vector $C 1$ if $S \mathbb{D E}=\mathrm{L}$ ', or the $m-$ vectorC 1 if $S \mathbb{D} E=R$ '.

On exit, the first row ofP *C if $S \mathbb{D} E=$ ' ', or the first colum $n$ of $C * P$ if $S I D E=R$ '.

C2 (input/output)
$(\mathbb{L D} C, N)$ if $S \mathbb{D} E=\mathbb{L}^{\prime}(\mathrm{LD} C, N-1)$ if $S \mathbb{D} E=R^{\prime}$ On entry, the $(m-1) x n m$ atrix $C 2$ if $S \mathbb{D} E=\mathbb{L}$ ', or them $x(n-1) m$ atrix $C 2$ if $S D E=R$.

Onexit, rows 2 m ofP*C if $S \mathbb{D} E=\mathrm{L}$ ', orcolum ns 2 m of $\mathrm{C} *$ if $S \mathbb{D} E=R$.

LD C (input)
The leading dim ension of the arrays C1 and C 2 . $\operatorname{LDC}>=\max (1, M)$.

W ORK (w orkspace)
$(\mathbb{N})$ if $S \mathbb{D} E=L^{\prime}(M)$ if $S \mathbb{D} E=R^{\prime}$

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

cosqb - synthesize a Fourier sequence from its representation in term s of a cosine series $w$ th odd $w$ ave num bers. The CO SQ operations are unnorm alized inverses of them selves, so a call to COSQF follow ed by a call to CO SQB w illm ultiply the input sequence by $4 * N$.

## SYNOPSIS

```
SUBROUT\mathbb{NE COSQB N,X,W SAVE)}
```

$\mathbb{N}$ TEGER N
REALX (*), W SAVE (*)
SUBROUTINECOSQB_64 $\mathbb{N}, \mathrm{X}, \mathrm{W}$ SAVE)
$\mathbb{N}$ TEGER*8N
REALX (*), W SAVE (*)

F95 INTERFACE
SU BROUTINE COSQB $\mathbb{N}, \mathrm{X}, \mathrm{W}$ SAVE)
$\mathbb{N} T E G E R:: N$
REAL,D $\mathbb{I M}$ ENSION (:) ::X,W SAVE

SU BROUTINECOSQB_64 $\mathbb{N}, \mathrm{X}, \mathrm{W}$ SAVE)
$\mathbb{N}$ TEGER (8) :: N
REAL,D $\mathbb{I}$ ENSION (:) ::X,W SAVE

## C INTERFACE

\#include <sunperfh>
void cosqb (intn, float *x, float *w save);

## ARGUMENTS

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product of sm allprim es. $\mathrm{N}>=0$.

X (input/output)
On entry, an array of length N containing the sequence to be transform ed. On exit, the quarterw ave cosine synthesis of the input.
W SAVE (input)
O n entry, an array with dim ension of at least (3 *
$\mathrm{N}+15$ ) that has been initialized by COSQ I.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

cosqf-com pute the Fourier coefficients in a cosine series representation w ith only odd w ave num bers. The COSQ operations are unnorm alized inverses of them selves, so a call to COSQF follow ed by a call to COSQB w illm ultiply the input sequence by $4 * N$.

## SYNOPSIS

```
SUBROUT\mathbb{NE COSQF N,X,W SAVE)}
INTEGER N
REAL X (*),W SAVE (*)
SUBROUT\mathbb{NE COSQF_64N,X,W SAVE)}
INTEGER*8N
REALX (*),W SAVE (*)
F95 INTERFACE
SU BROUT\mathbb{NE COSQF N,X,W SAVE)}
\mathbb{NTEGER ::N}
REAL,D IM ENSION (:) ::X,W SAVE
SU BROUT\mathbb{NE COSQF_64 N,X,W SAVE)}
\mathbb{NTEGER (8) ::N}
REAL,D IM ENSION (:) ::X,W SAVE
```


## C INTERFACE

```
\#include <sunperfh>
void cosqf(intn, float *x, float *w save);
```


## ARGUMENTS

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product
of sm allprim es. $\mathrm{N}>=0$.
X (input/output)
On entry, an array of length N containing the sequence to be transform ed. On exit, the quarter-w ave cosine transform of the input.
W SAVE (input)
O n entry, an amay with dim ension of at least (3

* $N+15$ ) that has been initialized by COSQ I.


## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

cosqi-initialize the array W SAVE, which is used in both $C O S Q F$ and $C O S Q B$.

## SYNOPSIS

> SUBROUTINE COSQIN,W SAVE)
$\mathbb{N}$ TEGER N
REALWSAVE (*)
SUBROUTINECOSQI_64 N,W SAVE)
$\mathbb{N}$ TEGER*8 N
REALW SAVE (*)
F95 INTERFACE
SUBROUTINECOSQIN,W SAVE)
$\mathbb{N} T E G E R:: N$
REAL,D $\mathbb{M}$ ENSION (:) ::W SAVE

SUBROUTINE COSQI_64 $\mathbb{N}, W$ SAVE)
$\mathbb{N}$ TEGER (8) :: N
REAL,D $\mathbb{I M}$ ENSION (:) ::W SAVE

## C INTERFACE

\#include <sunperfh>
void cosqi(intn, float * ${ }_{\text {w save }}$ );
void cosqi_ 64 (long n, float *W save);

## ARGUMENTS

N (input) Length of the sequence to be transform ed. The $m$ ethod is $m$ ost efficientw hen $N$ is a productof sm allprim es.

W SAVE (input)
On entry, an array ofdim ension ( 3 * $\mathrm{N}+15$ ) or greater. CO SQ I needs to be called only once to intialize W SAVE before calling COSQF and/orCOSQB
if N and W SAVE rem ain unchanged betw een these calls. Thus, subsequent transform s or inverse transforms of same size can be obtained faster than the first since they do not require initialization of the w orkspace.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

cost-com pute the discrete Fourier cosine transform of an even sequence. The COST transform s are unnorm alized inverses of them selves, so a call of COST follow ed by another call of C O ST w illm ulliply the input sequence by 2 * (N-1).

## SYNOPSIS

```
SUBROUT\mathbb{NE COST N,X,W SAVE)}
```

$\mathbb{N}$ TEGER N
REALX (*), W SAVE (*)
SU BROUTINECOST_64 $\mathbb{N}, \mathrm{X}, \mathrm{W}$ SAVE)
$\mathbb{N}$ TEGER*8N
REALX ( ${ }^{\star}$ ), $\mathrm{W} \operatorname{SAVE}\left({ }^{\star}\right)$

F95 INTERFACE
SU BROUTINE COST $\mathbb{N}, \mathrm{X}, \mathrm{W}$ SAVE)
$\mathbb{N} T E G E R:: N$
REAL,D $\mathbb{M}$ ENSION (:) ::X,W SAVE

SU BROUTINE COST_64 $\mathbb{N}, \mathrm{X}, \mathrm{W}$ SAVE)
$\mathbb{N}$ TEGER (8) :: N
REAL,D $\mathbb{M}$ ENSION (:) ::X,W SAVE

## C INTERFACE

\#include <sunperfh>
void cost(intn, float *x, float *w save);

## ARGUMENTS

N (input) Length of the sequence to be transform ed. These subroutines are most efficient when $\mathrm{N}-1$ is a productofsm allprim es. $\mathrm{N}>=2$.

X (input/output)
On entry, an aray of length $N$ containing the sequence to be transform ed. On exit, the cosine transform of the input.
W SAVE (input)
O n entry, an array with dim ension of at least (3

* $\mathrm{N}+15$ ), initialized by CO STI.


## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

costi-initialize the array W SA VE, which is used in COST .

## SYNOPSIS

```
    SUBROUT\mathbb{NE COSTIN,W SAVE)}
```

    \(\mathbb{I N}\) TEGER N
    REALW SAVE (*)
    SUBROUTINECOSTI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
    \(\mathbb{N}\) TEGER*8 N
    REALW SAVE (*)
    F95 INTERFACE
SUBROUTINE COSTIN,W SAVE)
$\mathbb{N} T E G E R:: N$
REAL,D $\mathbb{I M}$ ENSION (:) ::W SAVE
SUBROUTINECOSTI_64 $\mathbb{N}, \mathrm{W}$ SAVE)
$\mathbb{N}$ TEGER (8) :: N
REAL,D $\mathbb{M}$ ENSION (:) ::W SAVE
C INTERFACE
\#include <sunperfh>
void costi(intn, float *w save);
void costi_ 64 (long n, float *w save);

## ARGUMENTS

N (input) Length of the sequence to be transform ed. The $m$ ethod is $m$ ostefficientw hen $N-1$ is a product of sm allprim es. $\mathrm{N}>=2$.

W SAVE (input)
On entry, an array ofdim ension ( 3 * $N+15$ ) or greater. COST I is called once to initializeW SA VE before calling COST and need notbe called again between calls to COST if N and W SAVE rem ain unchanged. Thus, subsequent transform $s$ of sam e size can be obtained faster than the first since they do not require initialization of the w orkspace.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cpbcon -estim ate the reciprocalof the condition num ber (in the 1 -norm ) of a com plex H erm tian positive definite band $m$ atrix using the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}$ or $\mathrm{A}=$ L*L**H com puted by CPBTRF

## SYNOPSIS

```
SU BROUT\mathbb{NE CPBCON (UPLO,N,KD,A,LDA,ANORM,RCOND,W ORK,W ORK2,}
    \mathbb{NFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGER N,KD,LDA, INFO}
REAL ANORM,RCOND
REALW ORK2(*)
SUBROUTINE CPBCON_64 (UPLO,N,KD,A,LDA,ANORM,RCOND,W ORK,
    W ORK2,\mathbb{NFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),W ORK (*)
INTEGER*8N,KD,LDA,INFO
REAL ANORM,RCOND
REAL W ORK2 (*)
```


## F95 INTERFACE

```
SUBROUTINE PBCON (UPLO, \(\mathbb{N}], K D, A,[L D A], A N O R M, R C O N D,[\mathbb{O R K}]\), [W ORK2], [ \(\mathbb{N} F \mathrm{O}\) ])
CHARACTER (LEN=1) ::UPLO
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : ::A
```

$\mathbb{N} T E G E R:: N, K D, L D A, \mathbb{N F O}$
REAL ::ANORM,RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::W ORK 2

SU BROUTINE PBCON_64 (UPLO, $\mathbb{N}], K D, A,[L D A], A N O R M, R C O N D,[W O R K]$, [ W ORK2], $[\mathbb{N} \mathrm{FO}])$

CHARACTER (LEN=1) ::UPLO
COM PLEX , D $\mathbb{M}$ ENSION (:) ::W ORK
COM PLEX , D $\mathbb{M}$ ENSION (: : : : : A
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{KD}, \mathrm{LD} \mathrm{A}, \mathbb{N} F O$
REAL ::ANORM,RCOND
REAL,D $\mathbb{I}$ ENSION (:) ::W ORK2

## C INTERFACE

\#include <sunperfh>
void qpbcon (charuple, intn, intkd, com plex *a, int lda, floatanorm, float*rcond, int*info);
void qpbcon_64 (charuplo, long n, long kd, com plex *a, long lda, float anorm , float *roond, long *info);

## PURPOSE

cpbcon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a com plex H erm itian positive definite band $m$ atrix using the Cholesky factorization $A=U * * H * U$ or $A=$ L*L**H com puted by CPB TRF .

A $n$ estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND = 1 / (ANORM * norm (inv (A))).

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': U pper triangular factorstored in A ;
= IL ': Low er triangular factor stored in A .

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

K D (input)
The num ber of superdiagonals of the $m$ atrix $A$ if U PLO $=\mathrm{U}$ ', or the num ber of sub-diagonals if UPLO
$=\mathrm{L}^{\prime} . \mathrm{KD}>=0$ 。

A (input) The triangular factorU or $L$ from the Cholesky
factorization $\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}$ orA $=\mathrm{L} * \mathrm{~L} * * \mathrm{H}$ of the band
$m$ atrix $A$, stored in the first KD +1 row $s$ of the array. The jth colum n ofU orL is stored in the $j$ th colum $n$ of the array A as follows: if UPLO
$=U ', A(k d+1+i-j)=U(i, j)$ for $m a x(1, j$
$\mathrm{kd})<=\mathrm{i}<=\dot{j}$ if UPLO $=\mathrm{L}$ ', $\mathrm{A}(1+i-j)=\mathrm{L}(i, j)$
for $j=i<=m$ in $(n, j+k d)$.

LD A (input)
The leading dim ension of the array A. LDA >= K D +1 .
ANORM (input)
The 1-norm (or infinity-norm ) of the Herm tian band $m$ atrix $A$.

## RCOND (output)

The reciprocal of the condition number of the $m$ atrix $A$, com puted asRCOND = $1 /(A N O R M * A \mathbb{N} V N M)$, where $A \mathbb{N} V N M$ is an estim ate of the 1 -nom of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension $(2 * N)$

W ORK 2 (w orkspace)
dim ension (N)
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the i-th argum enthad an ille-
galvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cpbequ - com pute row and colum n scalings intended to equilibrate a $H$ erm itian positive definite band $m$ atrix $A$ and reduce its condition num ber (w ith respect to the tw o-norm )

## SYNOPSIS

```
SUBROUT\mathbb{NE CPBEQU (UPLO,N,KD,A,LDA,SCALE,SCOND,AMAX,INFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*)
\mathbb{NTEGER N,KD,LDA,}\mathbb{NFO}
REAL SCOND,AMAX
REAL SCA LE (*)
SUBROUTINE CPBEQU_64(UPLO,N,KD,A,LDA,SCALE,SCOND,AMAX,
    \mathbb{NFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*)
\mathbb{NTEGER*8N,KD,LDA,}\mathbb{N}FO
REAL SCOND,AMAX
REAL SCALE (*)
F95 INTERFACE
SU BROUT\mathbb{NE PBEQU (UPLO, N ],KD,A,[LDA],SCA LE,SCOND,AMAX,}
    [\mathbb{NFO])}
CHARACTER (LEN=1)::UPLO
COM PLEX,D IM ENSION (:,:) ::A
INTEGER ::N,KD,LDA,}\mathbb{N}F
REAL ::SCOND,AMAX
REAL,DIM ENSION (:) ::SCALE
```

SU BROUTINE PBEQU_64 (UPLO, $\mathbb{N}], K D, A,[L D A], S C A L E, S C O N D, A M A X$, [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::UPLO
COM PLEX,D $\mathbb{I}$ ENSION (: : : : : A
$\mathbb{N} T E G E R(8):: N, K D, L D A, \mathbb{N} F O$
REAL ::SCOND,AMAX
REAL,D $\mathbb{I M} E N S I O N(:):: S C A L E$

## C INTERFACE

\#include < sunperfh>
void qpbequ (charuplo, intn, int kd, com plex *a, int lda, float *scale, float *scond, float *am ax, int *info);
void qpbequ_64 (charuplo, long n, long kd, com plex *a, long lda, float * scale, float * scond, float * am ax, long *info);

## PURPOSE

cpbequ com putes row and colum n scalings intended to equilibrate a H erm itian positive definite band $m$ atrix A and reduce its condition num ber (w ith respect to the tw o-norm ). S contains the scale factors, $S(i)=1 /$ squt $(A)(i, i))$, chosen so that the scaled matrix B w th elem ents $B(i, 7)=$ $S(i) * A(i, j) * S(j)$ has ones on the diagonal. This choioe of $S$ puts the condition num berofB $w$ ithin a factor $N$ of the sm allest possible condition num ber over allpossible diagonal scalings.

## ARGUMENTS

UPLO (input)
$=U$ : U ppertriangularofA is stored;
= L': Low er triangularofA is stored.
N (input) The order of the matrix $\mathrm{A} . \mathrm{N}>=0$.
KD (input)
The num ber of superdiagonals of the $m$ atrix A if UPLO $=\mathrm{U}$ ', or the num berof subdiagonals ifU PLO $=\mathrm{L} \cdot \mathrm{KD}>=0$.

A (input) The upper or low er triangle of the H erm Itian band
$m$ atrix $A$, stored in the firstK $D+1$ row sof the array. The $j$ th colum n of A is stored in the $j$ th column of the amay A as follow s: if UPLO = U', A $(k d+1+i-j)=A(i, j)$ for $\max (1, j k d)<=i<=j$ if UPLO $=L \prime$ ', $A(1+i-j)=A(i, 7)$ for $\dot{j}=i<=m$ in $(n, \dot{j}+k d)$.

LD A (input)
The leading dim ension of the array A. LDA >= K D +1 .

SCALE (output)
If $\mathbb{N} F O=0$, SCA LE contains the scale factors for A.

SCOND (output)
If $\mathbb{N} F O=0, S C A$ LE contains the ratio of the sm allest SCA LE (i) to the largestSCA LE (i). If SCOND $>=0.1$ and AMAX is neither too large nor too sm all, it is notw orth scaling by SCA LE .

AMAX (output)
A bsolute value of largestm atrix elem ent. IfA M AX is very close to overflow orvery close to underflow , the $m$ atrix should be scaled.
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an illegalvahue.
$>0$ : if $\mathbb{N F O}=$ i, the $i$-th diagonal elem ent is nompositive.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

cpbrfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is H em itian positive definite and banded, and provides errorbounds and backw ard errorestim ates for the solution

## SYNOPSIS

```
SUBROUT\mathbb{NE CPBRFS (UPLO,N,KD,NRHS,A,LDA,AF,LDAF,B,LDB,X,}
    LDX,FERR,BERR,W ORK,W ORK 2, INFO)
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),AF (LDAF,*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,KD,NRHS,LDA,LDAF,LDB,LDX, INFO
REAL FERR (*),BERR (*),W ORK2 (*)
SUBROUT\mathbb{NE CPBRFS_64 (UPLO,N,KD,NRHS,A,LDA,AF,LDAF,B,LDB,}
    X,LDX,FERR,BERR,W ORK,W ORK 2, INFO)
```

CHARACTER * 1 UPLO
COM PLEX A (LDA,*), AF (LDAF,*), B (LDB, $\left.{ }^{*}\right), \mathrm{X}(\mathrm{LDX}, \star), \mathrm{W}$ ORK (*)
$\mathbb{N}$ TEGER*8N,KD,NRHS,LDA,LDAF,LDB,LDX, $\mathbb{N} F O$
REAL FERR ( ${ }^{*}$ ), BERR ( ${ }^{*}$ ), $\mathrm{W} O \operatorname{OR} 2$ ( ${ }^{*}$ )

## F95 INTERFACE

SU BROUTINE PBRFS (UPLO, $\mathbb{N}], K D, \mathbb{N} R H S], A,[L D A], A F,[L D A F], B$, [LD B], $\mathrm{X},[\mathrm{LD} \mathrm{X}], F E R R, B E R R,[W$ ORK], [W ORK 2], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1) ::UPLO
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::A,AF,B,X
$\mathbb{N} T E G E R:: N, K D, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O$
REAL,D IM ENSION (:) ::FERR,BERR,W ORK 2

SU BROUTINE PBRFS_64 (UPLO, $\mathbb{N}], K D, \mathbb{N} R H S], A,[L D A], A F,[L D A F]$, B, [LDB],X, [LDX],FERR,BERR, [W ORK], [W ORK2], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1) ::UPLO
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : A, AF, B, X
$\mathbb{N}$ TEGER (8) ::N, KD,NRHS,LDA,LDAF,LDB,LDX, $\mathbb{N} F O$
REAL,D $\mathbb{I M}$ ENSION (:) ::FERR,BERR,W ORK 2

## C INTERFACE

\#include < sunperfh>
void qpbrfs (charuplo, intn, intkd, intnrhs, com plex *a, int lda, com plex *af, int ldaf, com plex *b, int ldb, com plex *x, int ldx, float *ferr, float *berr, int *info);
void qpbrfs_64 (charuplo, long n, long kd, long nrhs, com plex *a, long lda, com plex *af, long ldaf, com plex *b, long lalb, com plex *x, long ldx, float *ferr, float*berr, long *info);

## PURPOSE

cpbrfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is $H$ erm itian positive definite and banded, and provides errorbounds and backw ard error estim ates for the solution.

## ARGUMENTS

UPLO (input)
= U ': U pper triangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrix A. N >=0.
KD (input)
The num ber of superdiagonals of the $m$ atrix A if UPLO = U',orthe num berof subdiagonals ifU PLO
$=L^{\prime} . \mathrm{KD}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber
of collm ns of the m atrices B and X. NRHS >=0.

A (input) The upper or low er triangle of the $H$ erm itian band $m$ atrix $A$, stored in the firstK $D+1$ row s of the array. The jth colum n of A is stored in the jth column of the anay A as follow s: if UPLO = U',
$\mathrm{A}(\mathrm{kd}+1+i-j, j)=A(i, 7)$ for $\max (1, j \mathrm{jkd})<=i<=j$ if $\mathrm{UPLO}=\mathrm{L}^{\prime}, \mathrm{A}(1+i-j)=A(i, 7)$ for
j $=$ i< $=m$ in $(n, j+k d)$.

LD A (input)
The leading dim ension of the array A. LDA >= K D +1 .

AF (input)
The triangular factor $U$ or $L$ from the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}$ orA $=\mathrm{L} * \mathrm{~L} * * \mathrm{H}$ of the band $m$ atrix A as com puted by CPBTRF, in the same storage form atas A (see A).

LDAF (input)
The leading dim ension of the array AF. LDAF >= $K D+1$.
$B$ (input) The righthand side $m$ atrix $B$.

LD B (input)
The leading dim ension of the aray $B$. LD B >= $\max (1, \mathbb{N})$.

X (input/output)
On entry, the solution $m$ atrix $X$, as com puted by CPBTRS. On exit, the im proved solution $m$ atrix $X$.

LD $X$ (input)
The leading dim ension of the array X . LD X >= $\max (1, N)$.

## FERR (output)

The estim ated forw ard enrorbound for each solution vector $X(\mathcal{)}$ ) the $j$ th colum $n$ of the solution $m$ atrix X). If XTRUE is the true solution corresponding to $X(\mathcal{O}), \operatorname{FERR}(\mathcal{O})$ is an estim ated upperbound for the $m$ agnitude of the largest ele$m$ ent in ( $X(\mathcal{H})$-XTRUE) divided by the $m$ agnitude of the largestelem ent in $X(\mathcal{J})$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

## BERR (output)

The com ponentw ise relative backw ard error of each solution vectorX (i) (i.e., the sm allest relative
change in any elem entofA orB thatm akes X ( 7 ) an exactsolution).

W ORK (w orkspace)
dim ension ( $2 * N$ )

W ORK2 (w orkspace)
dim ension $(\mathbb{N})$

INFO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the $i$ th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

qpbstf - com pute a splitC holesky factorization of a com plex H erm itian positive definite band $m$ atrix A

## SYNOPSIS

```
SUBROUT\mathbb{NE CPBSTF (UPLO,N,KD,AB,LDAB, INFO)}
CHARACTER * 1 UPLO
COM PLEX AB (LDAB,*)
INTEGERN,KD,LDAB,INFO
SUBROUT\mathbb{NE CPBSTF_64 (UPLO ,N,KD,AB,LDAB,INFO )}
CHARACTER * 1 UPLO
COM PLEX AB (LDAB,*)
INTEGER*8N,KD,LDAB,INFO
```


## F95 INTERFACE

SU BROUTINE PBSTF (UPLO, $\mathbb{N}], K D, A B,[L D A B],[\mathbb{N F O}])$

CHARACTER (LEN=1)::UPLO
COM PLEX,D IM ENSION (:,:)::AB
$\mathbb{N} T E G E R:: N, K D, L D A B, \mathbb{N F O}$

SU BROUTINE PBSTF_64 (UPLO, $\mathbb{N}], K D, A B,[L D A B],[\mathbb{N} F O])$

CHARACTER (LEN=1) ::UPLO
COMPLEX,D $\mathbb{I M}$ ENSION (:,:) ::AB
$\mathbb{N} T E G E R(8):: N, K D, L D A B, \mathbb{N} F O$
void qpbstf(char uplo, intn, int kd, com plex *ab, int ldab, int*info);
void qpbstf_ 64 (charuplo, long n, long kd, com plex *ab, long ldab, long *info);

## PURPOSE

cpbstf com putes a split C holesky factorization of a com plex H erm itian positive definite band $m$ atrix A.

This routine is designed to be used in conjunction with CHBGST.
The factorization has the form $A=S * * H * S$ where $S$ is a band $m$ atrix of the sam e bandw idth as A and the follow ing structure:

$$
\begin{aligned}
& S=(U \quad) \\
& \text { (M L ) }
\end{aligned}
$$

w here U is upper triangular of orderm $=(\mathrm{n}+\mathrm{kd}) / 2$, and L is low er triangular of ordern-m .

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': U ppertriangle ofA is stored;
$=\mathrm{L}$ ': Low er triangle of A is stored.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

KD (input)
The num ber of superdiagonals of the $m$ atrix $A$ if
$\mathrm{UPLO}=\mathrm{U}$ ', or the num ber of subdiagonals if UPLO
$=\mathbb{L}^{\prime} . \mathrm{KD}>=0$ 。

A B (input/output)
O n entry, the upper or low er triangle of the Her $m$ itian band $m$ atrix A, stored in the firstkd+1 row s of the amay. The $j$ th colum n of A is stored in the $j$ th colum $n$ of the array A B as follow s: if $\mathrm{UPLO}=\mathrm{U}^{\prime}, \mathrm{AB}(\mathrm{kd}+1+i-j, j)=A(i, j)$ for $\mathrm{max}(1, j$ $\mathrm{kd})<=i<=\dot{j}$ ifU PLO $=L^{\prime}, A B(1+i-j, j)=A(i, j)$ for $j=i<=m$ in $(n, j+k d)$.

On exit, if $\mathbb{N F O}=0$, the factors from the split Cholesky factorization $A=S * * H * S$. See Further D etails.

LD A B (input)
The leading dim ension of the array AB. LD A B >= K D +1 .
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue $>0:$ if $\mathbb{N} F O=$ i, the factorization could not be com pleted, because the updated elem enta (i,i) w as negative; the m atrix $A$ is notposilive definite.

## FURTHER DETAILS

The band storage schem e is illustrated by the follow ing exam ple, w hen $N=7, K D=2$ :

| $\mathrm{S}=(\mathrm{s} 11 \mathrm{~s} 12 \mathrm{~s} 13$ |  |
| :---: | :---: |
| ( | s22 s23 s24 ) |
| ( | s33 s34 ) |
| ( | s44 ) |
| ( | s53 s54 s55 |
| ( | s64 s65 s66 ) |
| ( | s75 s76 s77) |

If U PLO $=\mathrm{U}$ ', the amay A B holds:
on entry: on exit:

*     * a13 a24 a35 a46 a57 * * s13 s24 s53' s64's75'
* a12 a23 a34 a45 a56 a67 * s12 s23 s34 s54' s65's76' a11 a22 a33 a44 a55 a66 a77 s11 s22 s33 s44 s55 s66 s77

IfU PLO = L', the anay AB holds:
on entry: on exit:

```
a11 a22 a33 a44 a55 a66 a77 s11 s22 s33 s44 s55
s66 s77 a21 a32 a43 a54 a65 a76 * s12's23's34'
s54 s65 s76 * a31 a42 a53 a64 a64 * * s13'
s24's53 s64 s75 * *
```

A ray elem entsm arked * are notused by the routine; s12' denotes con $\mathfrak{j}(s 12)$; the diagonalelem ents of $S$ are real.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cpbsv - com pute the solution to a com plex system of linear equations $A * X=B$,

## SYNOPSIS

```
SUBROUT\mathbb{NE CPBSV (UPLO,N,ND IAG,NRHS,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LDB,*)
INTEGERN,NDIAG,NRHS,LDA,LDB,INFO
SU BROUT\mathbb{NE CPBSV_64 (UPLO ,N,ND IAG ,NRH S,A,LDA ,B,LD B,INFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LD B,*)
\mathbb{NTEGER*8N,ND IAG,NRHS,LDA,LDB,INFO}
```


## F95 INTERFACE

SU BROUTINE PBSV (UPLO, $\mathbb{N}], N D \mathbb{I A} G, \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N} F O])$

CHARACTER (LEN=1)::UPLO
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A,B
$\mathbb{N} T E G E R:: N, N D \mathbb{A} G, N R H S, L D A, L D B, \mathbb{N} F O$
SU BROUTINE PBSV_64 (UPLO, $\mathbb{N}], N D \mathbb{I} G, \mathbb{N} R H S], A,[L D A], B,[L D B]$, [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::UPLO
COM PLEX,D $\operatorname{IM}$ ENSION (: :: : ::A,B
$\mathbb{N} \operatorname{TEGER}(8):: N, N D \mathbb{I} G, N R H S, L D A, L D B, \mathbb{N} F O$

## C INTERFACE

\#include <sunperfh>
void qpbsv (charuplo, intn, intndiag, int nrhs, com plex
*a, int lda, com plex *b, int ldlo, int *info);
void qpbsv_64 (charuplo, long n, long ndiag, long nrhs, com plex *a, long lda, com plex *b, long ldb, long
*info);

## PURPOSE

cpbsv com putes the solution to a com plex system of linear equations
$A * X=B, w h e r e A$ is an $N$ boy $N$ H erm itian positive defintie band $m$ atrix and X and B are N -by-N R H S m atrices. The Cholesky decom position is used to factorA as
$A=U * * H * U$, if $U P L O=U '$, or
$A=L * L \star * H$, if U PLO $=\mathrm{L}$ ',
$w$ here $U$ is an uppertriangularband $m$ atrix, and $L$ is a low er triangular band $m$ atrix, w ith the sam e num ber of superdiagonals or subdiagonals as A. The factored form of A is then used to solve the system ofequations $A * X=B$.

## ARGUMENTS

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low er triangle ofA is stored.

N (input) The num ber of linearequations, i.e., the order of them atrix $A . N>=0$.

ND IA G (input)
The num ber of superdiagonals of the $m$ atrix A if UPLO $=\mathrm{U}$ ', or the num berof subdiagonals if P PLO
$=L^{\prime} \cdot \mathrm{ND} \mathbb{I} G>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS $>=0$.

A (input/output)
O $n$ entry, the upper or low er triangle of the Her -
$m$ itian band m atrix A, stored in the firstND IA G +1
row s of the array. The $j$ th colum $n$ of $A$ is stored
in the jth colum $n$ of the array A as follow s: if
 $\max (1, j \mathrm{jND} \mathrm{IAG})<=i<=j$ if $\mathrm{UPLO}=\mathrm{L}^{\prime}, \mathrm{A}(1+i-j)$
$=A(i, j)$ for $j<i<=m$ in $(\mathbb{N}, j+N D$ IA G ). See below for furtherdetails.

On exit, if $\mathbb{N} F O=0$, the triangular factor $U$ orL from the Cholesky factorization $A=U * * H * U$ or $A=$ $\mathrm{L} * \mathrm{~L} * * H$ of the band $m$ atrix $A$, in the sam e storage form atas A.

LD A (input)
The leading dim ension of the aray A. LD A >= N D IA G +1.
B (input/output)
O n entry, the N foy-NRHS righthand sidem atrix B. On exit, if $\mathbb{N F O}=0$, the N boy -N RH S solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the array $\mathrm{B} . \operatorname{LDB}>=$ $\max (1, N)$.

IN FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum ent had an illegalvałue $>0:$ if $\mathbb{N F O}=i$, the leading $m$ inoroforder iof $A$ is notpositive definite, so the factorization could notbe com pleted, and the solution has not been com puted.

## FURTHER DETAILS

The band storage scheme is illustrated by the follow ing exam ple, w hen $N=6, N D I A G=2$, and $U P L O=U ':$

On entry: On exit:

*     * a13 a24 a35 a46 * * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55 u66

Sim ilarly, ifU PLO = 'L 'the form atofA is as follow s:

On entry: On exit:

a31 a42 a53 a64 * * 131142153164 * *

A may elem entsm arked * are notused by the routine.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cpbsvx -use the C holesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}$ or $\mathrm{A}=$ $\mathrm{L} * \mathrm{~L} * * \mathrm{H}$ to com pute the solution to a com plex system of linear equations $A * X=B$,

## SYNOPSIS

```
SUBROUT\mathbb{NE CPBSVX EACT,UPLO,N,NDIAG,NRHS,A,LDA,AF,LDAF,}
    EQUED,S,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK 2,
    INFO)
CHARACTER * 1FACT,UPLO,EQUED
COM PLEX A (LDA,*),AF (LDAF,*),B (LDB,*),X (LDX,*),W ORK (*)
NTEGERN,ND IAG,NRHS,LDA,LDAF,LDB,LDX,\mathbb{NFO}
REALRCOND
REALS (*),FERR (*),BERR (*),W ORK ( (*)
SUBROUTINE CPBSVX_64 FACT,UPLO,N,NDIAG,NRHS,A,LDA,AF,LDAF,
    EQUED,S,B,LDB,X,LDX,RCOND ,FERR,BERR,W ORK,W ORK2,
    \mathbb{NFO)}
```

CHARACTER * 1 FACT, UPLO, EQUED
COM PLEX A (LDA,*),AF (LDAF,*), B (LDB,*), X (LDX,*), W ORK (*)
$\mathbb{N} T E G E R * 8 N, N D \mathbb{I A} G, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O$
REALRCOND


## F95 INTERFACE

SU BROUTINE PBSVX (FACT,UPLO, $\mathbb{N}], N D \mathbb{I} G, \mathbb{N R H S}], A,[L D A], A F,[L D A F]$, EQUED, S, B, [LDB],X, [LDX],RCOND,FERR,BERR, [W ORK], [W ORK 2], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::FACT,UPLO,EQUED
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A,AF,B,X
$\mathbb{N} T E G E R:: N, N D \mathbb{A} G, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O$
REAL: :RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::S,FERR,BERR,W ORK 2
SUBROUTINE PBSVX_64 (FACT,UPLO, $\mathbb{N}], N D \mathbb{I A G}, \mathbb{N} R H S], A,[L D A], A F$, [LDAF],EQUED ,S,B, [LDB],X, [LDX],RCOND,FERR,BERR, [W ORK], [W ORK2], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::FACT,UPLO,EQUED
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : A, AF, B, X
$\mathbb{I N}$ TEGER (8) ::N ,ND $\mathbb{I A} G, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N F O}$
REAL ::RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::S,FERR,BERR,W ORK 2

## C INTERFACE

\#include < sunperfh>
void qpbsvx (char fact, charuplo, int n, int ndiag, int nihs, com plex *a, int lda, com plex *af, int ldaf, char equed, float *s, com plex *b, int ldb, com plex *x, int ldx, float *roond, float *ferr, float *bers, int*info);
void qpbsvx_64 (char fact, char uplo, long n, long ndiag, long nrhs, com plex *a, long lda, com plex *af, long ldaf, charequed, float *s, com plex *b, long ldb, com plex *x, long ldx, float *roond, float *ferr, float *berr, long *info);

## PURPOSE

cpbsvx uses the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}$ or $\mathrm{A}=$ $\mathrm{L} * \mathrm{~L} * * \mathrm{H}$ to com pute the solution to a com plex system of linear equations
$A * X=B$, where $A$ is an $N$-by $-N$ H erm itian positive definthe band m atrix and X and B are N -by-N RH S m atrices.

Errorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:

1. IfFACT = E', real scaling factors are com puted to equilibrate
the system :

$$
\operatorname{diag}(S) * A * \operatorname{diag}(S) * \operatorname{inv}(\operatorname{diag}(S)) * X=\operatorname{diag}(S) * B
$$

W hether or not the system willbe equilibrated depends on the
scaling of the m atrix A , but ifequilibration is used, A is
overw ritten by diag $(S) * A * \operatorname{diag}(S)$ and $B$ by diag $(S) * B$.
2. IfFACT $=\mathrm{N}$ 'or E ', the Cholesky decom position is used to
factorthem atrix A (afterequilibration ifFACT = E) as
$A=U * * H * U$, if $U P L O=U$ ', or
$\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{H}$, if $\mathrm{UPLO}=\mathrm{L}$ ',
$w$ here $U$ is an upper triangularband $m$ atrix, and $L$ is a low er
triangularband $m$ atrix.
3. If the leading i-by-iprincipal $m$ inor is not positive definite,
then the routine retums w ith $\mathbb{N} F O=$ i. O therw ise, the factored
form of A is used to estim ate the condition num ber of the $m$ atrix
A. If the reciprocal of the condition num ber is less than $m$ achine
precision, $\mathbb{I N}$ FO $=\mathrm{N}+1$ is retumed as a w aming, but the routine
still goes on to solve for X and com pute errorbounds as described below .
4. The system of equations is solved for $X$ using the factored form
of A.
5. Iterative refinem ent is applied to im prove the com puted solution
$m$ atrix and calculate error bounds and backw ard error estim ates
for it.
6. If equilibration w as used, the $m$ atrix $X$ is prem ultiplied by
diag (S) so that it solves the original system before equilibration.

## ARGUMENTS

FACT (input)
Specifies w hether ornot the factored form of the
$m$ atrix $A$ is supplied on entry, and ifnot, whether them atrix A should be equilibrated before it is factored. = F ': On entry, AF contains the factored form ofA. IfEQUED = Y', them atrix A has been equilibrated $w$ ith scaling factors given by $S$.
A and A F w illnotbe m odified. = N ': Them atrix
A w ill.be copied to AF and factored.
$=\mathrm{E}$ ': The matrix A w ill be equilibrated if necessary, then copied to AF and factored.

UPLO (input)
$=\mathrm{U}$ ': Upper triangle ofA is stored;
= $\mathbb{L}$ ': Low ertriangle ofA is stored.

N (input) The num ber of linearequations, ie., the order of them atrix $A . N>=0$.

ND IA G (input)
The num ber of superdiagonals of the $m$ atrix $A$ if $\mathrm{UPLO}=\mathrm{U}$ ', orthe num berof subdiagonals ifUPLO $=\mathbb{L}^{\prime} . \mathrm{NDIAG}>=0$.

NRHS (input)
The num ber of right-hand sides, ie., the num ber
of colum ns of the $m$ atriges $B$ and X. NRH S $>=0$.

A (input/output)
O $n$ entry, the upper or low er triangle of the Her $m$ itian band $m$ atrix A, stored in the firstND IA G +1 row s of the amay, exceptifFACT = $\mathrm{F}^{\prime}$ and EQUED $=Y \prime$, then A m ustcontain the equilibrated $m$ atrix diag $(S) * A$ *diag $(S)$. The $j$ th colum n ofA is stored in the $j$ th colum $n$ of the array $A$ as follow $s$ : if $\left.\mathrm{UPLO}=\mathrm{U}^{\prime}, \mathrm{A} \mathbb{N D I A G + 1 + i - j}\right)=A(i, j$ for $\max (1, j \mathrm{j} D \mathrm{IA} \mathrm{G})<=\dot{i}=\dot{j}$ if $\mathrm{UPLO}=\mathrm{L}$ ', A ( $1+i-j$ ) $=A(i, j)$ for $\dot{j}=i<=m$ in $(\mathbb{N}, \dot{j}+N D I A G)$. See below for furtherdetails.

On exit, ifFACT = E' and EQUED = Y', A is overw ritten by diag $(\mathrm{S}) \star A$ *diag $(\mathrm{S})$.

LD A (input)
The leading dim ension of the array A. LDA >= ND IA G +1.

AF (input/output)
If FACT $=F^{\prime}$, then $A F$ is an inputargum entand on entry contains the triangular factor $U$ or $L$ from the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}$ or $\mathrm{A}=$ $L * L * * H$ of the band $m$ atrix $A$, in the sam e storage
form atas A (see A). IfEQUED = Y', then AF is the factored form of the equilibrated $m$ atrix A.

If $F A C T=N$ ', then $A F$ is an output argum ent and on exitretums the triangular factor $U$ orL from the Cholesky factorization $A=U * * H * U$ or $A=$ L*L**H.

IfFACT = E', then AF is an output argum ent and on exit retums the triangular factor $U$ orL from the Cholesky factorization $A=U * * H * U$ or $A=$ L*L**H of the equilibrated $m$ atrix $A$ (see the description of $A$ for the form of the equilibrated m atrix) .

LD AF (input)
The leading dim ension of the amay AF. LDAF >= N D IA G +1.

EQUED (input)
Specifies the form of equilibration thatw as done.
$=\mathrm{N}^{\prime}$ : N o equilibration (alw ays true ifFACT = N 7 .
$=$ Y ': Equilibration w as done, i.e., A has been replaced by diag $(\mathrm{S})$ * A * diag $(\mathrm{S})$. EQUED is an inputargum entifFACT = F ; otherw ise, it is an outputargum ent.

S (input/output)
The scale factors forA; notaccessed if EQUED = $N^{\prime} . S$ is an inputargum entifFACT=F'; otherw ise, S is an outputargum ent. IfFACT $=\mathrm{F}^{\prime}$ and EQUED $=Y$ ', each elem entof m ustbe positive.

B (input/output)
O n entry, the N $-b y-N R H S$ righthand side m atrix B .
On exit, if EQUED $=\mathrm{N}$ ', B is notm odified; if EQUED $=Y$ ', $B$ is overw ritten by diag $(S)$ * $B$.

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.

X (output)
If $\mathbb{N} F O=0$ or $\mathbb{I N} F O=\mathrm{N}+1$, the N -by-NRHS solution
$m$ atrix $X$ to the original system ofequations.
$N$ ote that if EQ UED $=Y$ ', A and B arem odified on exit, and the solution to the equilibrated system is inv (diag (S ) ) *X .

## LD X (input)

The leading dim ension of the anay X . LD X >= $\max (1, N)$.

## RCOND (output)

The estim ate of the reciprocal condition num ber of the $m$ atrix $A$ after equilibration (if done). If RCOND is less than them achine precision (in particular, ifRCOND $=0$ ), the $m$ atrix is singular to w orking precision. This condition is indicated by a retum code of $\mathbb{N} F O>0$.
FERR (output)
The estim ated forw ard errorbound for each solution vector $X()$ ) the $j$ th column of the solution $m$ atrix X). If XTRUE is the true solution corresponding to $X(\mathcal{O})$, FERR ( $)$ ) is an estim ated upperbound for the $m$ agnitude of the largest ele$m$ ent in (X ( $)$-X TRUE) divided by the magnitude of the largestelem entin $\mathrm{X}(\mathcal{J})$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

## BERR (output)

The com ponentw ise relative backw ard emror of each
solution vectorX (j) (ie., the sm allest relative change in any elem entofA orB thatm akes X ( $\mathcal{j}$ ) an exactsolution).

W ORK (w orkspace)
dim ension ( $2 \star \mathrm{~N}$ )

W ORK 2 (w orkspace)
dim ension $\mathbb{N}$ )
$\mathbb{N} F O$ (output)
= 0 : successfulexit
< 0 : if $\mathbb{N} F O=-$ i, the $i$-th argum enthad an ille-
galvalue
> 0 : if $\mathbb{N F F O}=\mathrm{i}$, and i is
$<=N$ : the leading $m$ inoroforderiof $A$ is not positive definite, so the factorization could not be com pleted, and the solution has not been com puted. RCOND $=0$ is retumed. $=\mathrm{N}+1$ : U is nonsingular, butRCOND is less than machine precision, $m$ eaning that the $m$ atrix is singular to $w$ orking precision. Nevertheless, the solution and error bounds are com puted because there are a num ber of situations w here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

## FURTHER DETAILS

The band storage scheme is illustrated by the follow ing exam ple, when $N=6, N D \mathbb{I A}=2$, and UPLO $=U$ ':

Tw o-dim ensional storage of the $H$ erm itian m atrix A:

```
al1 a12 al3
    a22 a23 a24
            a33 a34 a35
            a44 a45 a46
            a55 a56
(ai=congg(a\ddot{i}) a66
```

$B$ and storage of the upper triangle of $A$ :

*     * a13 a24 a35 a46
* a12 a23 a34 a45 a56
a11 a22 a33 a44 a55 a66
Sim ilarly, if U PLO = L 'the form atofA is as follow s:

```
a11 a22 a33 a44 a55 a66
```

a21 a32 a43 a54 a65 *
a31 a42 a53 a64 * *

A ray elem entsm arked * are notused by the routine.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cpbtf2 -com pute the C holesky factorization of a com plex H erm itian positive definite band $m$ atrix A

## SYNOPSIS

```
SUBROUT\mathbb{NE CPBTF2(UPLO,N,KD,AB,LDAB, INFO)}
CHARACTER * 1 UPLO
COM PLEX AB (LDAB,*)
INTEGERN,KD,LDAB,INFO
SUBROUT\mathbb{NE CPBTF2_64(UPLO,N,KD,AB,LDAB,INFO)}
CHARACTER * 1 UPLO
COM PLEX AB (LDAB,*)
INTEGER*8N,KD,LDAB,INFO
```


## F95 INTERFACE

SU BROUTINE PBTF2 (UPLO, $\mathbb{N}], K D, A B,[L D A B],[\mathbb{N} F O])$

CHARACTER (LEN=1) ::UPLO
COM PLEX,D IM ENSION (:,:)::AB
$\mathbb{N} T E G E R:: N, K D, L D A B, \mathbb{N F O}$

SU BROUTINE PBTF2_64 (UPLO, N ],KD ,AB, [LDAB], [NFO ])

CHARACTER (LEN=1) ::UPLO
COMPLEX,D $\mathbb{I M}$ ENSION (:,:) ::AB
$\mathbb{N} T E G E R(8):: N, K D, L D A B, \mathbb{N} F O$
void qpbtf2 (charuple, intn, intkd, com plex *ab, int ldab, int*info);
void qpbtf2_64 (charuplo, long n, long kd, com plex *ab, long ldab, long *info);

## PURPOSE

cpbtff com putes the Cholesky factorization of a com plex H er$m$ itian positive definite band $m$ atrix A.

The factorization has the form

```
A = U'* U , ifUPLO = U',or
A = L * L', ifUPLO = L',
```

$w$ here $U$ is an uppertriangularm atrix, $U$ 'is the conjugate transpose ofU , and $L$ is low er triangular.

This is the unblocked version of the algorithm, calling Level2 BLAS.

## ARGUMENTS

## UPLO (input)

Specifies w hether the upper or low er triangular
part of the $H$ erm itian $m$ atrix $A$ is stored:
$=\mathrm{U}$ ': Upper triangular
= LL: Low ertriangular

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

KD (input)
The num ber of super-diagonals of the m atrix A if $\mathrm{UPLO}=\mathrm{U}$ ', or the num ber of sub-diagonals if UPLO
$=\mathbb{L} . \mathrm{KD}>=0$ 。

A B (input/output)
O n entry, the upper or low er triangle of the H er$m$ tiian band $m$ atrix $A$, stored in the firstKD +1 row s of the amay. The $j$ th colum n of A is stored in the $j$ th colum n of the array A B as follow s: if $\mathrm{UPLO}=\mathrm{U}$ ', AB $(k d+1+i-j, j)=A(i, j)$ for $\max (1, j$ $\mathrm{kd})<=i<=\dot{j}$ ifUPLO $=L ', A B(1+i-j, j)=A(i, j)$ for $\dot{j}=i<=m$ in $(n, j+k d)$.

On exit, if $\mathbb{N} F O=0$, the triangular factor $U$ orL
from the Cholesky factorization $\mathrm{A}=\mathrm{U}$ * U or $\mathrm{A}=$ L * L ' of the band m atrix A , in the sam e storage form atas A.

LDAB (input)
The leading dim ension of the array AB. LD AB >= KD+1.
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N} F O=-k$, the $k$-th argum enthad an illegalvalue $>0:$ if $\mathbb{N} F O=k$, the leading $m$ inoroforderk is notpositive definite, and the factorization could notbe com pleted.

## FURTHER DETAILS

The band storage scheme is illustrated by the follow ing exam ple, when $\mathrm{N}=6, \mathrm{KD}=2$, and $\mathrm{U} P L O=\mathrm{U}$ ':

On entry: On exit:

*     * a13 a24 a35 a46 * * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55 u66

Sim ilarly, if UPLO = L'the form atofA is as follow s:
On entry: On exit:

166
a21 a32 a43 a54 a65 * $121 \quad 132143154165$
*
a31 a42 a53 a64 * * 131142153164 *

A ray elem entsm arked * are not used by the routine.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

cpbtrf-com pute the C holesky factorization of a com plex H erm itian positive definite band $m$ atrix A

## SYNOPSIS

```
SUBROUTINE CPBTRF(UPLO,N,KD,A,LDA, INFO)
CHARACTER * 1 UPLO
COM PLEX A (LDA,*)
\mathbb{NTEGERN,KD,LDA,}\mathbb{NFO}
SU BROUT\mathbb{NE CPBTRF_64 (UPLO,N ,KD,A,LDA, INFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*)
INTEGER*8N,KD,LDA,INFO
```


## F95 INTERFACE

```
SU BROUTINE PBTRF (UPLO, \(\mathbb{N}], K D, A,[L D A],[\mathbb{N F O}])\)
CHARACTER (LEN=1)::UPLO
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: N, K D, L D A, \mathbb{N} F O\)
SU BROUTINE PBTRF_64 (UPLO, \(\mathbb{N}], K D, A,[L D A],[\mathbb{N} F O])\)
CHARACTER (LEN=1)::UPLO
COM PLEX,D \(\mathbb{I M}\) ENSION (: : : : : : A
\(\mathbb{N}\) TEGER (8) ::N,KD,LDA, \(\mathbb{N} F O\)
void qpbtrf(charuplo, intn, intkd, com plex *a, int lda, int*info);
void qpbtrf_64 (charuplo, long n, long kd, com plex *a, long lda, long *info);

\section*{PURPOSE}
cpbtrf com putes the C holesky factorization of a com plex H er\(m\) itian positive definite band \(m\) atrix A.

The factorization has the form
\[
\begin{aligned}
& A=U * * H * U, \text { if } U P L O=U^{\prime}, \text { or } \\
& A=L * L * * H, \text { if } U P L O=L ',
\end{aligned}
\]
\(w\) here \(U\) is an uppertriangularm atrix and \(L\) is low er triangular.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': Uppertriangle of \(A\) is stored;
\(=1 \mathrm{~L}\) ': Low er triangle of A is stored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

KD (input)
The num berof superdiagonals of the \(m\) atrix \(A\) if
UPLO = U', orthe num berof subdiagonals ifUPLO
\(=L^{\prime} . K D>=0\) 。

A (input/output)
O \(n\) entry, the upper or low er triangle of the H er\(m\) tian band \(m\) atrix \(A\), stored in the firstKD +1 row s of the array. The \(j\) th colum n of A is stored in the \(j\) th colum \(n\) of the amay \(A\) as follow \(s\) : if \(\mathrm{UPLO}=U ', A(k d+1+i-j, j)=A(i, j)\) for \(m a x(1, j\) \(\mathrm{kd})<=i<=j\) if \(\mathrm{UPLO}=\mathbb{L}, \mathrm{A}(1+i-j, j)=A(i, j)\) for \(j<=i<=m\) in \((n, j+k d)\).

On exit, if \(\mathbb{N} F O=0\), the triangular factor \(U\) orL from the Cholesky factorization \(A=U * * H * U\) orA \(=\) L*L**H of the band \(m\) atrix A, in the sam e storage form atas A.

The leading dim ension of the array A. LDA >= K D +1 .
\(\mathbb{N F O}\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the leading \(m\) inoroforder is notpositive definite, and the factorization could notbe com pleted.

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, when \(N=6, K D=2\), and \(U P L O=U:\)
On entry: On exit:
```

    * a13 a24 a35 a46 * * u13 u24 u35
    u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55
u66

```
Sim ilarly, if UPLO = L 'the form atofA is as follow s:
On entry: Onexit:
    a11 a22 a33 a44 a55 a66 \(111 \quad 122 \quad 133144 \quad 155\)
166
    a21 a32 a43 a54 a65 * \(121 \quad 132143154165\)
*
    a31 a42 a53 a64 * * 131142153164 *
*

A rray elem entsm arked * are notused by the routine.
C ontributed by
PeterM ayes and G inseppe Radicati, \(\mathbb{B M}\) EC SEC , Rom e, M arch 23,1989

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cpbtrs - solve a system of linear equations A *X \(=\mathrm{B}\) w th a \(H\) erm tian positive definite band \(m\) atrix \(A\) using the Cholesky factorization \(A=U * * H * U\) orA \(=L * L * * H\) com puted by CPBTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CPBTRS (UPLO,N,KD,NRHS,A,LDA,B,LDB, INFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LD B,*)
\mathbb{NTEGER N,KD,NRHS,LDA,LDB,INFO}
SUBROUTINE CPBTRS_64(UPLO,N,KD,NRHS,A,LDA,B,LDB,INFO)
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LDB,*)
\mathbb{NTEGER*8 N,KD,NRHS,LDA,LDB,INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE PBTRS (UPLO, \(\mathbb{N}], K D, \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N} F O])\)
CHARACTER (LEN=1) ::UPLO
COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::A,B
\(\mathbb{N} T E G E R:: N, K D, N R H S, L D A, L D B, \mathbb{N} F O\)
SU BROUTINE PBTRS_64 (UPLO, \(\mathbb{N}], K D, \mathbb{N} R H S], A,[L D A], B,[L D B]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::UPLO
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,B
\(\mathbb{N}\) TEGER (8) ::N,KD,NRHS,LDA,LDB, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void qpbtrs (char uplo, intn, intkd, intnrhs, com plex *a, int lda, com plex *b, int ldl, int *info);
void qpbtrs_64 (charuple, long n, long kd, long nrhs, com plex *a, long lda, com plex *b, long ldb, long
*info);

\section*{PURPOSE}
cpbtre solyes a system of linear equations \(A * X=B\) with a H erm itian positive definite band \(m\) atrix A using the C holesky factorization \(A=U * * H * U\) orA \(=L * L * * H\) com puted by CPBTRF .

\section*{ARGUMENTS}

\section*{UPLO (input)}
\(=\mathrm{U}\) : U Upertriangular factorstored in A;
\(=1\) ': Low er triangular factorstored in A.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

KD (input)
The num berof superdiagonals of the \(m\) atrix \(A\) if \(\mathrm{UPLO}=\mathrm{U}\) ', or the num ber of subdiagonals if \(\mathrm{U} P L O\)
\(=L^{\prime} . K D>=0\) 。

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the m atrix B. NRH S \(>=0\).

A (input) The triangular factor \(U\) or L from the Cholesky factorization \(A=U * * H * U\) orA \(=L * L * * H\) of the band \(m\) atrix \(A\), stored in the first \(K D+1\) row \(s\) of the array. The jth colum n of \(U\) orL is stored in the jth colum \(n\) of the anay \(A\) as follow s: if UPLO \(=U \prime, A(k d+1+i-j)=U(i, j)\) for \(m a x(1, j\) \(\mathrm{kd})<=\mathrm{i}<=\dot{j}\) ifUPLO \(=\mathrm{L}\) ', A \((1+i-j, j=\mathrm{j}(i, j)\) for \(\dot{j}=i<=m\) in \((n, j+k d)\).

LD A (input)
The leading dim ension of the aray A. LDA >= K D +1 .

B (input/output)

O \(n\) entry, the right hand side \(m\) atrix \(B\). On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cpocon -estim ate the reciprocal of the condition num ber (in the 1-norm ) of a com plex H erm tian positive definitem atrix using the C holesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) com puted by CPO TRF

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGERN,LDA,INFO}
REALANORM,RCOND
REAL W ORK2 (*)
SUBROUTINE CPOCON_64 (UPLO,N,A,LDA,ANORM,RCOND,W ORK,W ORK2,
\mathbb{NFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),W ORK (*)
INTEGER*8N,LDA,INFO
REAL ANORM,RCOND
REAL W ORK2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE POCON (UPLO, \(\mathbb{N}], A,[L D A], A N O R M, R C O N D,[W\) ORK ], [W ORK2], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO
COMPLEX,D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX,D IM ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: N, L D A, \mathbb{N} F O\)

REAL ::ANORM,RCOND
REAL,D \(M\) ENSION (:) ::W ORK 2

SUBROUTINE POCON_64 (UPLO, \(\mathbb{N}], A,[L D A], A N O R M, R C O N D,[W O R K],[W O R K 2]\), [ \(\mathbb{N} \mathrm{FO}]\) )

CHARACTER (LEN=1) :: UPLO
COM PLEX , D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX , D \(\mathbb{M}\) ENSION (: : : : : A
\(\mathbb{N} T E G E R(8):: N, L D A, \mathbb{N F O}\)
REAL ::ANORM,RCOND
REAL,D \(\mathbb{I}\) ENSION (:) ::W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void qpocon (charuplo, intn, com plex *a, int lda, float anorm, float *rcond, int*info);
void qpocon_64 (charuplo, long n, com plex *a, long lda, floatanorm , float *rcond, long *info);

\section*{PURPOSE}
cpocon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a com plex \(H\) erm titian posilive definitem atrix using the Cholesky factorization \(A=U * * H * U\) or \(A=L \star L * * H\) com puted by CPO TRF .

A \(n\) estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND \(=1 /\) (ANORM * norm (inv (A))).

\section*{ARGUMENTS}

\section*{UPLO (input)}
\(=U\) ': U ppertriangle of A is stored;
\(=\mathrm{L}\) ': Low er triangle ofA is stored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input) The triangular factor \(U\) or \(L\) from the Cholesky factorization \(A=U * * H * U\) orA \(=L * L * * H\), as com puted by CPOTRF.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

ANORM (input)
The 1-norm (or infinity-norm ) of the Herm titian \(m\) atrix A.

\section*{RCOND (output)}

The reciprocal of the condition number of the \(m\) atrix \(A\), com puted as RCOND \(=1\) ( \(A N O R M * A \mathbb{N} V N M\) ), where \(A \mathbb{N} V N M\) is an estim ate of the 1 -nom of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{N N}\) FO \(=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cpoequ - com pute row and colum n scalings intended to equilibrate a H erm itian positive definite \(m\) atrix \(A\) and reduce its condition num ber (w ith respect to the tw o-norm )

\section*{SYNOPSIS}
```

SUBROUTINE CPOEQU N,A,LDA,SCALE,SCOND,AMAX,INFO)
COM PLEX A (LDA,*)
NNTEGER N,LDA,}\mathbb{N}F
REAL SCOND,AMAX
REALSCALE (*)
SUBROUT\mathbb{NE CPOEQU_64 N,A,LDA,SCALE,SCOND,AM AX,NNFO)}
COM PLEX A (LDA,*)
\mathbb{NTEGER*8 N,LDA,INFO}
REAL SCOND,AMAX
REAL SCALE (*)
F95 INTERFACE

```

```

    COM PLEX,D IM ENSION (:,:) ::A
    \mathbb{NTEGER ::N,LDA,}\mathbb{NFO}
    REAL ::SCOND,AMAX
    REAL,D IM ENSION (:) ::SCALE
    ```

```

    COM PLEX,D IM ENSION (:,:) ::A
    \mathbb{NTEGER (8)::N,LDA,\mathbb{NFO}}\mathbf{N}=\mp@code{L}
    ```

REAL :: SCOND,AMAX
REAL,D \(\mathbb{I M}\) ENSION (:) ::SCALE

\section*{C INTERFACE}
\#include <sunperfh>
void qpoequ (intn, com plex *a, int lda, float *scale, float
*scond, float *am ax, int *info);
void qpoequ_64 (long n, com plex *a, long lda, float *scale, float*scond, float *am ax, long *info);

\section*{PURPOSE}
cpoequ computes row and colum n scalings intended to equilibrate a Herm itian positive definite matrix A and reduce its condition num ber (w th respect to the tw o-norm ). \(S\) contains the scale factors, \(S(i)=1 /\) sqrt (A \((i, i))\), chosen so that the scaled matrix B w ith elem ents \(\mathrm{B}(1,7)=\) S (i)*A \((i, 7) * S(i)\) has ones on the diagonal. This choige of \(S\) puts the condition num ber ofB w ithin a factor N of the sm allest possible condition num ber over allpossible diagonalscalings.

\section*{ARGUMENTS}

N (input) The order of the m atrix A. N >= 0 .

A (input) The N -by-N Herm itian positive definite m atrix whose scaling factors are to be com puted. Only the diagonalelem ents ofA are referenced.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

SCALE (output)
If \(\mathbb{N} F O=0, S C A L E\) contains the scale factors for A.

\section*{SCOND (output)}

If \(\mathbb{N} F O=0, S C A L E\) contains the ratio of the sm allest SCA LE (i) to the largestSCA LE (i). IfSC O ND >= 0.1 and AMAX is neither too large nor too sm all, it is notw orth scaling by SCA LE .

A bsolute value of largestm atrix elem ent. If A M A X is very close to overflow orvery close to underflow , the \(m\) atrix should be scaled.
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue
> 0 : if \(\mathbb{N F O}=\) i, the \(i\)-th diagonal elem ent is nonpositive.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cporfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is \(H\) erm tian positive definite,

\section*{SYNOPSIS}
```

SUBROUTINE CPORFS (UPLO,N,NRHS,A,LDA,AF,LDAF,B,LDB,X,LDX,
FERR,BERR,W ORK,W ORK2,INFO)
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),AF (LDAF,*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDA,LDAF,LDB,LDX,NNFO
REAL FERR (*),BERR (*),W ORK (*)
SUBROUT\mathbb{NE CPORFS_64 UPLO,N,NRHS,A,LDA,AF,LDAF,B,LDB,X,LDX,}
FERR,BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),AF (LDAF,*),B (LDB,*),X (LDX,*),W ORK (*)
\mathbb{NTEGER*8N,NRHS,LDA,LDAF,LDB,LDX, INFO}
REAL FERR (*),BERR (*),WORK2 (*)

```

\section*{F95 INTERFACE}

SUBROUTINE PORFS (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F], B,[L D B]\), \(\mathrm{X},[\operatorname{LDX}], F E R R, B E R R,[\mathbb{W} O R K],[\mathbb{W} O R K 2],[\mathbb{N} F O])\)

CHARACTER (LEN=1)::UPLO
COMPLEX,D IM ENSION (:) ::W ORK
COMPLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,AF,B,X
\(\mathbb{N}\) TEGER :: N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N}\) FO
REAL,D IM ENSION (:) ::FERR,BERR,W ORK2

SUBROUTINE PORFS_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F], B,[L D B]\), \(\mathrm{X},[\operatorname{LDX}], F E R R, B E R R,[\mathbb{W}\) ORK \(],[\mathbb{W}\) ORK2], [ \(\mathbb{N} F O])\)

CHARACTER (LEN=1)::UPLO
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::A,AF,B,X
\(\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
REAL,D IM ENSION (:) ::FERR,BERR,WORK2

\section*{C INTERFACE}
\#include < sunperfh>
void qporfs (charuple, intn, intnrhs, com plex *a, int lda, com plex *af, int Idaf, com plex *b, int ldb, com plex *x, intldx, float*ferr, float *berr, int *info);
void qporfs_64 (charuplo, long n, long nihs, com plex *a, long lda, com plex *af, long ldaf, com plex *b, long ldb, com plex *x, long ldx, float *ferr, float *berr, long *info);

\section*{PURPOSE}
cporfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is H em itian positive definite, and provides errorbounds and backw ard emoresti\(m\) ates for the solution.

\section*{ARGUMENTS}

\section*{UPLO (input)}
\(=\mathrm{U}\) : U ppertriangle of A is stored;
= L': Low ertriangle ofA is stored.
N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the m atrices B and X. NRHS >=0.

A (input) The \(H\) erm itian \(m\) atrix \(A\). If \(\mathrm{PLO}=\mathrm{U}\) ', the leading \(N\) by -N uppertriangularpartofA contains the upper triangularpart of the \(m\) atrix \(A\), and the strictly low ertriangularpartofA is not referenced. IfUPLO = L', the leading N -by N lower triangularpartofA contains the low er triangular
partof the \(m\) atrix \(A\), and the strictly upper triangularpart ofA is not referenced.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, \mathbb{N})\).

AF (input)
The triangular factor \(U\) or \(L\) from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\), as com puted by CPO TRF.

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, \mathbb{N})\).
\(B\) (input) The righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the anay \(B\). LD B \(>=\) \(\max (1, \mathbb{N})\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by CPO TRS. On exit, the im proved solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the array X . LDX >= \(\max (1, \mathbb{N})\).

FERR (output)
The estim ated forw ard enrorbound for each solution vector \(X(\mathcal{)}\) ) the \(j\) th column of the solution matrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{H}), \operatorname{FERR}(\mathcal{)})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(\mathrm{X}(\mathcal{1})-\mathrm{XTRUE}\) ) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{j})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector X ( \()\) (i.e., the sm allest relative change in any elem entofA orB thatm akes X (i) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )

W ORK2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cposv - com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CPOSV (UPLO,N,NRHS,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LDB,*)
INTEGERN,NRHS,LDA,LDB,}\mathbb{N}F
SUBROUT\mathbb{NE CPOSV_64(UPLO,N,NRHS,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LDB,*)
INTEGER*8N,NRHS,LDA,LDB,INFO

```
F95 INTERFACE
    SU BROUTINEPOSV (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N F O}])\)
    CHARACTER (LEN=1) ::UPLO
    COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,B
    \(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, \mathbb{N} F O\)
    SUBROUTINEPOSV_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N F O}])\)
    CHARACTER (LEN=1) ::UPLO
    COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : A, B
    \(\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D B, \mathbb{N} F O\)
C INTERFACE
    \#include <sunperfh>
void qposv (charuple, intn, intnrhs, com plex *a, int lda, com plex *b, int ldb, int *info);
void qposv_64 (char uplo, long n, long nrhs, com plex *a, long lda, com plex *b, long ldb, long *info);

\section*{PURPOSE}
cposv com putes the solution to a com plex system of linear equations
\(A * X=B, w h e r e A\) is an \(N\) boy \(-N\) H erm itian positive defintie \(m\) atrix and \(X\) and \(B\) are \(N\) foy-N R H S m atrices.

The Cholesky decom position is used to factorA as
\(A=U{ }^{* *} H^{*} \mathrm{U}\), if \(\mathrm{UPLO}=\mathrm{U}\) ', or
\(A=L * L * * H\), if \(U P L O=L '\),
where \(U\) is an upper triangularm atrix and \(L\) is a low ertriangular \(m\) atrix. The factored form of \(A\) is then used to solve the system ofequations \(A * X=B\).

\section*{ARGUMENTS}

UPLO (input)
\(=U\) ': U pper triangle of \(A\) is stored;
\(=\mathbb{L}\) ': Low ertriangle of \(A\) is stored.

N (input) The num ber of linearequations, i.e., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of the m atrix B. NRHS \(>=0\).

A (input/output)
O n entry, the \(H\) erm itian m atrix A. If UPLO = \(U^{\prime}\), the leading N -oy N uppertriangularpart of \(A\) contains the upper triangular part of the \(m\) atrix \(A\), and the strictly low er triangularpart of A is not referenced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading N boy- N low er triangularpart of A contains the low er triangularpart of the m atrix A, and the strictly upper triangularpart of A is notreferenced.

On exit, if \(\mathbb{N F O}=0\), the factor \(U\) orL from the Cholesky factorization \(A=U * * H * U\) or \(A=L * L * * H\).

LD A (input)
The leading dim ension of the anay A. LD A >= \(\max (1, N)\).

B (input/output)
On entry, the N -by-NRHS righthand side matrix B.
On exi, if \(\mathbb{N} F O=0\), the \(N\) by \(N\) RH S solution
matrix \(X\).

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO = -i, the i-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N F O}=\mathrm{i}\), the leading m inoroforderiof
A is notpositive definite, so the factorization could not.be com pleted, and the solution has not been com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cposvx - use the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) to com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CPOSVX (FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,EQUED,}
S,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK2,INFO)
CHARACTER * 1 FACT,UPLO,EQUED
COM PLEX A (LDA,*),AF (LDAF,*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDA,LDAF,LDB,LDX,\mathbb{NFO}
REALRCOND
REALS (*),FERR (*),BERR (*),W ORK 2 (*)
SUBROUT\mathbb{NE CPOSVX_64(FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,EQUED,}
S,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK2,INFO)

```
CHARACTER * 1 FACT, UPLO, EQUED
COM PLEX A (LDA,*),AF (LDAF,*), B (LDB,*),X (LDX,*),W ORK (*)
\(\mathbb{N} T E G E R * 8 N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
REALRCOND
REALS (*), FERR (*), BERR (*), W ORK 2 (*)

\section*{F95 INTERFACE}

SU BROUTINE POSVX \(\mathbb{F A C T}, \mathrm{UPLO}, \mathbb{N}], \mathbb{N R H} \mathrm{S}], \mathrm{A},[\mathrm{LDA}], A F,[L D A F]\), EQUED, S, B, [LDB],X, [LDX],RCOND,FERR,BERR, [W ORK], [W ORK2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1)::FACT,UPLO,EQUED
COMPLEX,D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : A, AF, B, X
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
REAL ::RCOND
REAL,D \(\mathbb{I M}\) ENSION (:) ::S,FERR,BERR,W ORK 2

SU BROUTINE POSVX_64 (FACT,UPLO, \(\mathbb{N}], \mathbb{N R H S ] , A , [ L D A ] , A F , [ L D A F ] , ~}\) EQUED, S, B, [LDB],X, [LDX],RCOND,FERR,BERR, [W ORK], [W ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::FACT,UPLO,EQUED
COMPLEX,D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : A, AF, B, X
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::S,FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void qposvx (char fact, charuplo, intn, int nrhs, com plex
*a, int lda, com plex *af, intldaf, charequed, float *s, com plex *b, int ldb, com plex *x, int ldx, float *roond, float *ferr, float *berr, int *info);
void qposvx_64 (char fact, charuplo, long n, long nrhs, com plex *a, long lda, com plex *af, long ldaf, char equed, float*s, com plex *b, long ldb, com plex *x, long ldx, float *roond, float * ferr, float *berr, long *info);

\section*{PURPOSE}
cposvx uses the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) to com pute the solution to a com plex system of linear equations
\(A * X=B\), where \(A\) is an \(N\)-by-N H erm tian positive definthe \(m\) atrix and \(X\) and \(B\) are \(N\)-by-N RH S m atrices.

Errorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT \(=\mathrm{E}\) ', real scaling factors are computed to equilibrate the system :
\(\operatorname{diag}(S) * A * \operatorname{diag}(S) * \operatorname{inv}(\operatorname{diag}(S)) * X=\operatorname{diag}(S) * B\)
W hether or not the system willbe equilibrated depends on the
scaling of the m atrix A , but if equilibration is used, A
overw rilten by diag \((\mathrm{S}) \star A\) *diag \((\mathrm{S})\) and B by diag \((\mathrm{S}) \star\) B .
2. IfFACT = N 'or E', the Cholesky decom position is used to
factor them atrix A (afterequilibration ifFACT =E)
as
\[
\begin{aligned}
& A=U \star * H * U, \text { if } U P L O=U ' \text {, or } \\
& A=L * L * * H, \text { if } U P L O=L^{\prime},
\end{aligned}
\]
w here U is an upper triangularm atrix and L is a low er triangular
\(m\) atrix.
3. If the leading i-by-iprincipal m inor is not positive definite,
then the routine retums \(w\) ith \(\mathbb{N F O}=\) i. O therw ise, the factored
form ofA is used to estim ate the condition num ber of the \(m\) atrix
A. If the reciprocal of the condition num ber is less than \(m\) achine
precision, \(\mathbb{N} F O=N+1\) is retumed as a \(w\) aming, but the routine
stillgoes on to solve forX and com pute emorbounds as described below .
4. The system ofequations is solved forX using the factored form
of A.
5. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for it.
6. If equilibration \(w\) as used, the \(m\) atrix \(X\) is prem ultiplied by
diag (S) so that it solves the original system before equilibration.

\section*{ARGUMENTS}

FACT (input)
Specifies w hether ornotthe factored form of the \(m\) atrix \(A\) is supplied on entry, and if not, whether them atrix A should be equilibrated before it is factored. = F': On entry, AF contains the fac-
tored form ofA. IfEQUED \(=Y\) ', them atrix \(A\) has been equilibrated \(w\) ith scaling factors given by \(S\). A and A F w illnotbe m odified. \(=\mathrm{N}\) ': The m atrix A w illle copied to A F and factored.
= E : The m atrix A w ill be equilibrated if necessary, then copied to AF and factored.

UPLO (input)
= U :: Uppertriangle ofA is stored;
= L': Low er triangle ofA is stored.

N (input) The num ber of linearequations, ie., the order of the matrix A. \(N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the \(m\) atrices \(B\) and X. NRHS \(>=0\).
A (input/output)
O n entry, the \(H\) erm itian \(m\) atrix A, except ifFACT =
\(\mathrm{F}^{\prime}\) and EQUED = Y ', then A mustcontain the equilibrated \(m\) atrix diag \((S) \star A * d i a g(S)\). If U PLO \(=\) U', the leading N -by-N uppertriangular partofA contains the upper triangularpart of the \(m\) atrix A, and the strictly low ertriangularpartofA is not referenced. If \(\mathrm{PLO}=\mathrm{L}\) ', the leading N -by N low er triangularpart ofA contains the low ertriangularpart of the m atrix A, and the strictly upper triangular part of A is not referenced. A is notm odified ifFACT = F or N', or ifFACT = E'and EQUED = N 'on exit.

Onexit, ifFACT=E' and EQUED = Y', A is overw rilten by diag ( S ) A A *diag ( S ) .

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

AF (input/output)
If \(F A C T=F\) ', then \(A F\) is an inputargum entand on entry contains the triangular factor \(U\) or \(L\) from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{H}\), in the sam e storage form at as A. IfEQUED ne. \(N\) ', then \(A F\) is the factored form of the equilibrated \(m\) atrix diag \((S) * A * \operatorname{diag}(S)\).

If \(F A C T=N\) ', then \(A F\) is an output argum ent and on exit retums the triangular factorU orL from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) of the originalm atrix A.

If \(F A C T=E\) ', then \(A F\) is an output argum ent and on exit retums the triangular factorU orL from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(L * L * * H\) of the equilibrated \(m\) atrix \(A\) (see the description of \(A\) for the form of the equilibrated \(m\) atrix).

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, N)\).

EQUED (input)
Specifies the form of equilibration thatw as done.
\(=\mathrm{N}^{\prime}\) : N o equilibration (alw ays true ifFACT = N).
\(=Y^{\prime}:\) Equilibration \(w\) as done, i.e., A has been replaced by diag (S) * A * diag (S). EQUED is an inputargum ent if \(\mathrm{FACT}=\mathrm{F}\) '; otherw ise, it is an outputargum ent.

S (input/output)
The scale factors forA ; not accessed if EQUED = \(\mathrm{N}^{\prime} . \mathrm{S}\) is an inputargum ent if \(\mathrm{FACT}=\mathrm{F}\) '; otherw ise, S is an outputargum ent. IfFACT = \(\mathrm{F}^{\prime}\) and EQUED = Y', each elem entofS m ustbe positive.

B (input/output)
On entry, the \(N-b y-N R H S\) righthand side \(m\) atrix \(B\).
On exit, if EQUED \(=N\) ', \(B\) is notm odified; if EQUED \(=Y\) ', \(B\) is overw ritten by diag \((S) * B\).

LD B (input)
The leading dim ension of the anay \(B\). LD B \(>=\) \(\max (1, \mathbb{N})\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=\mathrm{N}+1\), the N -by-NRH S solution
\(m\) atrix \(X\) to the original system of equations.
\(N\) ote that if EQUED \(=Y\) ', A and \(B\) are m odified on exit, and the solution to the equilibrated system is inv (diag (S))*X .

LD X (input)
The leading dim ension of the array X . LDX >= \(\max (1, \mathbb{N})\).

RCOND (output)
The estim ate of the reciprocal condition num berof the \(m\) atrix \(A\) after equilibration (if done). If

RCOND is less than them achine precision (in particular, ifRCOND \(=0\) ), the \(m\) atrix is singular to w orking precision. This condition is indicated by a retum code of \(\mathbb{N} F O>0\).

\section*{FERR (output)}

The estim ated forw ard errorbound for each solution vector \(X(\mathcal{)}\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution comesponding to \(X(\mathcal{O})\), FERR ( \()\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in (X ( \()\)-X TRU E) divided by the magnitude of the largestelem ent in \(\mathrm{X}(\mathcal{j})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vectorX (j) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(j\) ) an exact solution).

W ORK (w orkspace)
dim ension ( \(2 \star \mathrm{~N}\) )

W ORK 2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{N}\) FO (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an illegalvalue
> 0 : if \(\mathbb{N F F O}=i\), and \(i\) is
<= \(N\) : the leading \(m\) inor oforderiof \(A\) is not positive definite, so the factorization could not be com pleted, and the solution has not been com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1\) : U is nonsingular, butRCOND is less than machine precision, \(m\) eaning that the \(m\) atrix is singular to w orking precision. N evertheless, the solution and error bounds are com puted because there are a num ber of situations w here the com puted solution can bem ore accurate than the value ofRCOND would suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cpotf2 -com pute the Cholesky factorization of a com plex Herm itian positive definite \(m\) atrix A

\section*{SYNOPSIS}

SU BROUTINE CPOTF2 (UPLO, N, A, LDA, \(\mathbb{N} F O\) )
CHARACTER * 1 UPLO
COM PLEX A (LDA,*)
\(\mathbb{N}\) TEGER \(N, L D A, \mathbb{N F O}\)
SU BROUTINE CPOTF2_64 (UPLO ,N,A,LDA, \(\mathbb{N} F O\) )
CHARACTER * 1 UPLO
COM PLEX A (LDA,*)
\(\mathbb{N}\) TEGER*8N,LDA, \(\mathbb{N}\) FO

\section*{F95 INTERFACE}

SU BROUTINE POTF2 (UPLO, \(\mathbb{N}], A,[L D A],[\mathbb{N F O}])\)
CHARACTER (LEN=1)::UPLO
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathrm{LDA}, \mathbb{N} F O\)
SU BROUTINE POTF2_64 (UPLO, \(\mathbb{N}], A,[L D A],[\mathbb{N F O}])\)
CHARACTER (LEN=1)::UPLO
COMPLEX,D \(\mathbb{M}\) ENSION (: : : : : A
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} A, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void qpotf2 (charuple, int \(n\), complex *a, int lda, int *info);
void qpotf2_64 (char uple, long n, com plex *a, long lda, long *info);

\section*{PURPOSE}
cpotf2 com putes the Cholesky factorization of a com plex H er\(m\) itian posilive definite \(m\) atrix A.

The factorization has the form
\(A=U^{\prime} * U\), if \(U P L O=U '\), or
\(A=L\) * L', ifUPLO = L',
\(w\) here \(U\) is an uppertriangularm atrix and \(L\) is low er triangular.

This is the unblocked version of the algorithm, calling Level2 BLAS.

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the upper or low er triangular
part of the \(H\) erm itian \(m\) atrix \(A\) is stored. \(=U\) ':
U pper triangular
= IL ': Low ertriangular

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the \(H\) erm itian matrix A. If \(\mathrm{UPLO}=\mathrm{U}\) ', the leading \(n\) by \(n\) upper triangularpart of A contains the upper triangularpart of the \(m\) atrix \(A\), and the strictly low ertriangularpartof A is not referenced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading n by n low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of A is notreferenced.

On exit, if \(\mathbb{N F O}=0\), the factor \(U\) orL from the
Cholesky factorization \(A=U\) * U orA \(=\mathrm{L} * \mathrm{~L}{ }^{\prime}\).

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-\mathrm{k}\), the k -th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N} F O=k\), the leading \(m\) inor oforder \(k\) is notpositive definite, and the factorization could notbe com pleted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
qpotrf-com pute the C holesky factorization of a complex Herm itian positive definite m atrix A

\section*{SYNOPSIS}

SUBROUTINE CPOTRF (UPLO, N, A ,LDA, \(\mathbb{N} F O\) )
CHARACTER * 1 UPLO
COM PLEX A (LDA,*)
\(\mathbb{N}\) TEGER \(N, L D A, \mathbb{N F O}\)
SU BROUTINE CPOTRF_64 (UPLO,N,A,LDA, \(\mathbb{N} F O\) )
CHARACTER * 1 UPLO
COM PLEX A (LDA,*)
\(\mathbb{N}\) TEGER*8N,LDA, \(\mathbb{N} F O\)

\section*{F95 INTERFACE}

SU BROUTINE POTRF (UPLO, \(\mathbb{N}], A,[\operatorname{LDA}],[\mathbb{N F O}])\)
CHARACTER (LEN=1)::UPLO
COM PLEX,D \(\mathbb{I}\) ENSION (: : : : : : A
\(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathrm{LDA}, \mathbb{N} F O\)
SU BROUTINE POTRF_64 (UPLO, \(\mathbb{N}], A,[L D A],[\mathbb{N} F O])\)
CHARACTER (LEN=1)::UPLO
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : ::A
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} A, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void qpotrf(charuplo, int \(n\), complex *a, int lda, int *info);
void qpotrf_64 (charuplo, long n, com plex *a, long lda, long *info);

\section*{PURPOSE}
cpotrf com putes the Cholesky factorization of a com plex H er\(m\) itian posilive definite m atrix A.

The factorization has the form
\(A=U * * H * U\), if \(U P L O=U '\), or
\(A=L * L \star * H\), if \(\mathrm{UPLO}=\mathrm{L}\) ',
\(w\) here \(U\) is an upper triangularm atrix and \(L\) is low er triangular.

This is the block version of the algorithm, calling Level 3 BLAS .

\section*{ARGUMENTS}

UPLO (input)
\(=U\) ': U pper triangle of \(A\) is stored;
\(=\mathbb{L}\) ': Low er triangle of \(A\) is stored.

N (input) The order of them atrix A. N \(>=0\).

A (input/output)
O n entry, the H erm itian m atrix A. If UPLO = U', the leading N -by -N uppertriangularpartof A contains the uppertriangularpart of the \(m\) atrix \(A\), and the strictly low er triangularpart of \(A\) is not referenced. If UPLO = 'L', the leading N -by N low er triangularpant of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of A is notreferenced.

On exit, if \(\mathbb{N F O}=0\), the factor \(U\) orL from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\).

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).
= 0: successfulexit
< 0 : if \(\mathbb{N}\) FO \(=-\)-i, the i-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the leading \(m\) inoroforder \(i\) is not positive definite, and the factorization could notbe com pleted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cpotri-com pute the inverse of a com plex H em itian positive definite \(m\) atrix \(A\) using the Cholesky factorization \(A=\) \(\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) com puted by CPO TRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CPOTRI(UPLO,N,A,LDA, IN FO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*)
NNTEGER N,LDA,\mathbb{NFO}
SU BROUT\mathbb{NE CPOTRI_64 (UPLO,N,A,LDA, INFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*)
\mathbb{N TEGER*8 N,LDA,INFO}

```
F95 INTERFACE
    SU BROUTINE POTRI(UPLO, \(\mathbb{N}\) ], A, [LDA ], [ \(\mathbb{N F O}\) ])
    CHARACTER (LEN=1) ::UPLO
    COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::A
    \(\mathbb{N}\) TEGER ::N,LDA, \(\mathbb{N}\) FO
    SU BROU TINE POTRI_64 (UPLO, \(\mathbb{N}], A,[L D A],[\mathbb{N} F O])\)
    CHARACTER (LEN=1)::UPLO
    COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::A
    \(\mathbb{N}\) TEGER (8) ::N,LDA, \(\mathbb{N}\) FO
void qpotri(charuplo, int \(n\), com plex *a, int lda, int
    *info);
void qpotri_ 64 (char uplo, long n, com plex *a, long lda, long
    *info);

\section*{PURPOSE}
cpotricom putes the inverse of a com plex H erm itian positive definite \(m\) atrix \(A\) using the Cholesky factorization \(A=\) \(\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) com puted by CPO TRF .

\section*{ARGUMENTS}

UPLO (input)
= U : U Upertriangle ofA is stored;
= L': Low er triangle ofA is stored.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
A (input/output)
O n entry, the triangular factor \(U\) or \(L\) from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\), as com puted by CPOTRF. On exit, the upper or low er triangle of the (Herm itian) inverse ofA, overw riting the input factorU orL.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, \mathbb{N})\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue
\(>0\) : if \(\mathbb{I N F O}=\mathrm{i}\), the ( \(i, i\) i) elem entof the factor
U orL is zero, and the inverse could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cpotrs -solve a system of linearequations \(A * X=B\) w th a Herm itian positive definite \(m\) atrix \(A\) using the Cholesky factorization \(A=U * * H * U\) orA \(=L * L * * H\) com puted by CPO TRF

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE CPOTRS (UPLO,N,NRHS,A ,LDA,B,LDB, NNO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LD B,*)
INTEGERN,NRHS,LDA,LDB,INFO
SUBROUTINE CPOTRS_64 (UPLO,N,NRHS,A,LDA,B,LDB,INFO)
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LDB,*)
\mathbb{NTEGER*8N,NRHS,LDA,LDB,INFO}

```
F95 INTERFACE
    SU BROUTINE POTRS (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N F O}])\)
    CHARACTER (LEN=1) ::UPLO
    COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::A,B
    \(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, \mathbb{N} F O\)
    SU BROUTINE POTRS_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N F O}])\)
    CHARACTER (LEN=1) ::UPLO
    COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::A, B
    \(\mathbb{N}\) TEGER (8) ::N,NRHS,LDA,LDB, \(\mathbb{N}\) FO
void qpotrs (char uple, intn, intnrhs, com plex *a, int lda, com plex *b, int ldb, int *info);
void qpotrs_64 (charuplo, long n, long nrhs, com plex *a, long lda, com plex *b, long ldb, long *info);

\section*{PURPOSE}
cpotrs solves a system of linear equations \(A * X=B\) with a H erm itian posilive definite m atrix A using the Cholesky factorization \(A=U * * H * U\) orA \(=L * L * * H\) com puted by CPOTRF .

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': Uppertriangle of \(A\) is stored;
\(=\mathbb{L}\) ': Low er triangle ofA is stored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRH S \(>=0\).

A (input) The triangular factor \(U\) or \(L\) from the Cholesky
factorization \(A=U * * H * U\) orA \(=L * L * * H\), as com puted by CPO TRF .

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

B (input/output)
On entry, the right hand side m atrix B. On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cppcon -estim ate the reciprocal of the condition num ber (in the 1 -norm ) of a com plex H erm itian positive definite packed \(m\) atrix using the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) L*L**H com puted by CPPTRF

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
COM PLEX A (*),W ORK (*)
INTEGERN,\mathbb{NFO}
REALANORM,RCOND
REALW ORK2 (*)
SUBROUT\mathbb{NE CPPCON_64(UPLO,N,A,ANORM,RCOND,W ORK,W ORK2, INFO)}
CHARACTER * 1 UPLO
COM PLEX A (*),W ORK (*)
INTEGER*8 N, INFO
REAL ANORM,RCOND
REALW ORK2 (*)
F95 INTERFACE

```

```

    CHARACTER (LEN=1) ::UPLO
    COMPLEX,DIM ENSION (:) ::A,W ORK
    \mathbb{NTEGER ::N,\mathbb{NFO}}\mathbf{N}=0
    REAL ::ANORM,RCOND
    REAL,DIM ENSION (:) ::W ORK2
    ```

CHARACTER (LEN=1) ::UPLO
COM PLEX,D \(\mathbb{I}\) ENSION (:) ::A,W ORK
\(\mathbb{N} T E G E R(8):: N, \mathbb{N} F O\)
REAL ::ANORM,RCOND
REAL,D \(\mathbb{I M}\) ENSION (:) ::W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void qppcon (charuple, intn, com plex *a, floatanorm , float *rcond, int *info);
void qppcon_64 (charuplo, long n, com plex *a, float anorm , float *roond, long *info);

\section*{PURPOSE}
cppcon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a com plex H erm itian positive definite packed \(m\) atrix using the Cholesky factorization \(A=U * * H * U\) or \(A=\) L*L**H com puted by CPPTRF .

A \(n\) estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND = 1 / (ANORM * norm (inv (A))).

\section*{ARGUMENTS}

UPLO (input)
= U : U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
A (input) The triangular factorU or \(L\) from the Cholesky
factorization \(A=U * * H * U\) orA \(=L * L * * H\), packed colum nw ise in a linearanay. The jth colum \(n\) of \(U\) or \(L\) is stored in the array A as follow s: if UPLO = U', A \((i+(j-1) * j 2)=U(i, j)\) for \(1<=i<=j\) if \(\mathrm{UPLO}=\mathrm{L}, \mathrm{A}(i+(j-1) *(2 n-7 / 2)=\mathrm{L}(i, 7)\) for j=i<=n.

ANORM (input)
The 1-norm (or infinity-norm) of the Herm tian \(m\) atrix A.

\section*{RCOND (output)}

The reciprocal of the condition num ber of the
\(m\) atrix \(A\), com puted as RCOND = 1/(ANORM *A \(\mathbb{N} V N M)\),
where \(A \mathbb{N} V N M\) is an estim ate of the 1 -nom of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an ille-
galvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cppequ - com pute row and colum n scalings intended to equilibrate a Herm itian positive definite \(m\) atrix A in packed storage and reduce its condition num ber (w ith respect to the tw o-norm )

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CPPEQU (UPLO,N,A,SCALE,SCOND,AM AX,INFO)}
CHARACTER * 1 UPLO
COM PLEX A (*)
INTEGERN,\mathbb{NFO}
REAL SCOND,AMAX
REAL SCA LE (*)
SU BROUT\mathbb{NE CPPEQU_64(UPLO,N,A,SCALE,SCOND,AMAX,INFO)}
CHARACTER * 1 UPLO
COM PLEX A (*)
INTEGER*8 N, INFO
REAL SCOND,AMAX
REAL SCALE (*)
F95 INTERFACE
SUBROUT\mathbb{NE PPEQU (UPLO, N ],A,SCALE,SCOND,AMAX,[NFO])}
CHARACTER (LEN=1) ::UPLO
COMPLEX,DIMENSION (:) ::A
\mathbb{NTEGER::N,\mathbb{NFO}}\mathbf{N}=0
REAL ::SCOND,AMAX
REAL,DIM ENSION (:) ::SCALE

```

CHARACTER (LEN=1) ::UPLO
COM PLEX,D \(\mathbb{I}\) ENSION (:) ::A
\(\mathbb{N} T E G E R(8):: N, \mathbb{N} F O\)
REAL :: SCOND,AMAX
REAL,D \(\mathbb{I M} E N S I O N(:):: S C A L E\)

\section*{C INTERFACE}
\#include <sunperfh>
void qppequ (charuplo, int n, com plex *a, float *scale, float*scond, float*am ax, int*info);
void qppequ_64 (char uplo, long n, com plex *a, float *scale, float*scond, float *am ax, long *info);

\section*{PURPOSE}
cppequ com putes row and colum n scalings intended to equilibrate a Herm itian positive definite \(m\) atrix A in packed storage and reduce its condition num ber (w ith respect to the tw o-norm ). S contains the scale factors, \(S(i)=1 /\) sqrt \((A\) ( \(i, i))\), chosen so that the scaled \(m\) atrix \(B\) w ith elem ents \(B(i, j)=S(i) * A(i, j) * S(i)\) has ones on the diagonal. This choige ofs puts the condition num ber of B w ithin a factor \(N\) of the sm allestpossible condition num berover all possible diagonal scalings.

\section*{ARGUMENTS}

UPLO (input)
= U : U ppertriangle ofA is stored;
= L': Low er triangle of A is stored.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).

A (input) The upper or low er triangle of the H erm itian \(m\) atrix A, packed colum nw ise in a linear anray. The \(j\) th column of A is stored in the array A as follows: if UPLO = U',A ( \(i+(j-1) * j 2)=A(i, 7)\) for \(1<=i<=j\) ifUPLO = L', A ( \(\left.i+(j-1)^{*}(2 n-j) / 2\right)\) \(=A(i, 7)\) for \(j=i<=n\).

SCALE (output)
If \(\mathbb{N} F O=0, S C A L E\) contains the scale factors for A.

\section*{SCOND (output)}

If \(\mathbb{N} F O=0, S C A\) LE contains the ratio of the sm allest SCA LE (i) to the largestSCA LE (i). If SC OND \(>=0.1\) and \(A M A X\) is neither too large nor too sm all, it is notw orth scaling by SC A LE .

AMAX (output)
A bsolute value of largestm atrix elem ent. If A M A X is very close to overflow orvery close to underflow , the m atrix should be scaled.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue
> 0 : if \(\mathbb{N} F O=i\), the \(i\) th diagonal elem ent is nonpositive.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cppris -im prove the com puted solution to a system of linear equations when the coefficientm atrix is H em itian positive definite and packed, and provides emrorbounds and backw ard errorestim ates for the solution

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CPPRFS (UPLO,N,NRHS,A,AF,B,LDB,X,LDX,FERR,BERR,}
W ORK,W ORK 2, INFO)
CHARACTER * 1 UPLO
COM PLEX A (*),AF (*),B (LDB,*),X (LDX ,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
REAL FERR (*),BERR (*),W ORK2 (*)
SU BROUT\mathbb{NE CPPRFS_64 (UPLO,N,NRHS,A,AF,B,LDB,X,LDX,FERR,}
BERR,W ORK,W ORK2,INFO)
CHARACTER * 1 UPLO
COM PLEX A (*),AF (*),B (LDB,*),X (LDX,*),W ORK (*)
\mathbb{NTEGER*8N,NRHS,LDB,LDX,}\mathbb{N}FO
REAL FERR (*),BERR (*),WORK2 (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE PPRFS (UPLO,N, NRHS],A,AF,B,[LDB],X, [LDX],FERR, BERR, [W ORK], [W ORK2], [ $\mathbb{N} F O]$ )
CHARACTER (LEN=1) ::UPLO
COMPLEX,D $\mathbb{I M} E N S I O N(:):: A, A F, W$ ORK
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::B,X
$\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::FERR,BERR,W ORK 2

```

SU BROUTINE PPRFS_64 (UPLO,N, NRHS],A,AF,B,[LDB],X,[LDX],FERR, BERR, [W ORK], [W ORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::UPLO
COMPLEX,D \(\mathbb{M}\) ENSION (:) ::A,AF,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : \(B, \mathrm{X}\)
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDB,LDX, \(\mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include < sunperfh>
void qppris (charuple, intn, intnrhs, com plex *a, com plex *af, com plex *b, int ldo, com plex *x, int ldx, float * ferr, float *berr, int *info);
void qpprfs_64 (charuplo, long n, long nihs, com plex *a, com plex *af, com plex *b, long ldb, com plex *x, long ldx, float * ferrs, float *berr, long *info);

\section*{PURPOSE}
cpprfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is \(H\) erm itian positive definite and packed, and provides errorbounds and backw ard errorestim ates for the solution.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}:\) : U pper triangle ofA is stored;
\(=\mathrm{L}\) ': Low er triangle of A is stored.
N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the m atrices B and X. NRH S >=0.

A (input) The upper or low er triangle of the H erm tian \(m\) atrix \(A, p a c k e d\) colum nw ise in a linear array. The \(j\) th column of A is stored in the array A as follows: if UPLO = U',A \((i+(j 1) * j 2)=A(i, j)\) for \(1<=i<=j\) ifUPLO \(=\mathrm{L}\) ', A ( \(i+(j-1) *(2 n-j) / 2)\)
\(=A(i, j)\) for \(j=i<=n\).

\section*{AF (input)}

The triangular factorU or L from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\), as com puted by SPPTRF \(/ C P P T R F\), packed collm nw ise in a linear aray in the sam e form at as A (see A).
\(B\) (input) The righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the anay B. LD B >= \(\max (1, N)\).
X (input/output)
O \(n\) entry, the solution \(m\) atrix \(X\), as com puted by CPPTRS. On exit, the im proved solution \(m\) atrix \(X\).

\section*{LD X (input)}

The leading din ension of the array X . LD X >= max (1,N).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X(\mathcal{)}\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution comesponding to \(X(\mathcal{O})\), FERR ( \()\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{H})\)-X TRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{H})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each
solution vector \(X\) (i) (i.e., the sm allest relative change in any elem entofA orB thatm akes X (i) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 *\) N )

W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
cppsv - com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CPPSV (UPLO,N,NRHS,A,B,LDB, INFO)}
CHARACTER * 1 UPLO
COM PLEX A (*),B (LD B,*)
INTEGERN,NRHS,LDB,INFO
SU BROUTINE CPPSV_64 (UPLO,N,NRHS,A,B,LDB,\mathbb{NFO)}
CHARACTER * 1 UPLO
COM PLEX A (*),B (LD B,*)
INTEGER*8N,NRHS,LDB,INFO

```

\section*{F95 INTERFACE}
```

SU BROUTINE PPSV (UPLO,N, NRHS],A,B, [LDB], [ $\mathbb{N} F O]$ )
CHARACTER (LEN=1)::UPLO
COM PLEX,D $\mathbb{M}$ ENSION (:) ::A
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : B
$\mathbb{N}$ TEGER :: N,NRHS,LDB, $\mathbb{N}$ FO
SU BROUTINE PPSV_64 (UPLO,N, $\mathbb{N} R H S], A, B,[L D B],[\mathbb{N F O}])$
CHARACTER (LEN=1) ::UPLO
COMPLEX,D $\mathbb{M}$ ENSION (:) ::A
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::B
$\mathbb{N}$ TEGER (8) :: N,NRHS,LDB, $\mathbb{N}$ FO

```

\section*{C INTERFACE}
\#include <sunperfh>
void qppsv (charuplo, intn, intnrhs, com plex *a, com plex *b, int ldb, int *info);
void qppsv_64 (char uple, long n, long nrhs, com plex *a, com plex *b, long ldb, long *info);

\section*{PURPOSE}
cppsv com putes the solution to a com plex system of linear equations
\(A * X=B\), where \(A\) is an \(N\)-by-N Herm itian positive defintie \(m\) atrix stored in packed form at and \(X\) and \(B\) are \(N\) by \(-N\) RH \(S\) \(m\) atrices.

The Cholesky decom position is used to factorA as
\[
\begin{aligned}
& A=U * * H * U, \text { if } U P L O=U ' \text { or } \\
& A=L * L * * H, \text { if } U P L O=L \prime
\end{aligned}
\]
where \(U\) is an upper triangularm atrix and \(L\) is a low er triangular \(m\) atrix. The factored form ofA is then used to solve the system ofequations \(A * X=B\).

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) : : U pper triangle ofA is stored;
= L': Low er triangle ofA is stored.
N (input) The num ber of linearequations, ie., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS \(>=0\).

A (input/output)
O \(n\) entry, the upper or low ertriangle of the Her \(m\) itian \(m\) atrix A, packed colum nw ise in a linear array. The jth column of A is stored in the array A as follows: if UPLO = U',A (i+ (j)
 \((j-1)^{*}(2 n-j / 2)=A(i, 7)\) for \(\dot{j}=i<=n\). See below for further details.

On exit, if \(\mathbb{N} F O=0\), the factor \(U\) orL from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\), in the sam e storage form at as A.

B (input/output)
On entry, the N -by -NRH S righthand side m atrix B. On exit, if \(\mathbb{N F O}=0\), the N boy \(-\mathrm{NRH} S\) solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the i-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N}\) FO \(=i\), the leading \(m\) inoroforder iof \(A\) is notposilive definite, so the factorization could notbe com pleted, and the solution has not been com puted.

\section*{FURTHER DETAILS}

The packed storage schem e is illustrated by the follow ing
exam ple w hen \(N=4, \mathrm{UPLO}=\mathrm{U}\) ':

Tw o-dim ensionalstorage of the H erm itian m atrix A :
```

a11 a12 a13 a14
a22 a23 a24
a33 a34 (aij= congj (aï))
a44

```

Packed storage of the upper triangle ofA:
\(A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]\)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
cppsvx -use the C holesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) to com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CPPSVX (FACT,UPLO,N,NRHS,A,AF,EQUED,S,B,LDB,}
X,LDX,RCOND,FERR,BERR,WORK,W ORK2,INFO)
CHARACTER * 1FACT,UPLO,EQUED
COM PLEXA (*),AF (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
REALRCOND
REALS (*),FERR (*),BERR(*),WORK2 (*)
SUBROUT\mathbb{NECPPSVX_64(FACT,UPLO,N,NRHS,A,AF,EQUED,S,B,}
LDB,X,LDX,RCOND,FERR,BERR,WORK,W ORK 2, INFO)
CHARACTER * 1 FACT,UPLO,EQUED
COMPLEX A (*),AF (*),B (LDB,*),X (LDX,*),W ORK (*)
\mathbb{NTEGER*8N,NRHS,LDB,LDX,}\mathbb{N}FO
REAL RCOND
REALS (*),FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE PPSVX (EACT,UPLO, \(\mathbb{N}], \mathbb{N} R H S], A, A F, E Q U E D, S, B\), [LDB], \(\mathrm{X},[\mathrm{LD} \mathrm{X}], R C O N D, F E R R, B E R R,[W O R K],[W\) ORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::FACT,UPLO,EQUED
COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::A,AF,W ORK
COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::B,X
\(\mathbb{N}\) TEGER ::N,NRHS,LDB,LDX, \(\mathbb{N}\) FO
REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) :: S,FERR,BERR,W ORK 2

SU BROUTINE PPSVX_64 (FACT,UPLO, \(\mathbb{N}], \mathbb{N} R H S], A, A F, E Q U E D, S, B\), [ [D B ], X , [LDX ], RCOND ,FERR, BERR, [WORK], [WORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::FACT, UPLO, EQUED
COMPLEX,D \(\mathbb{M}\) ENSION (:) ::A,AF,W ORK
COM PLEX , D \(\mathbb{I M} E N S I O N(:,:):\) B , X
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDB,LDX, \(\mathbb{N}\) FO
REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) :: S,FERR,BERR,W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void qppsvx (char fact, charuple, intn, int nrhs, com plex *a, com plex *af, char equed, float *s, com plex *b, int ldlo, com plex *x, int ldx, float *rcond, float * ferrr, float *berr, int *info);
void qppsvx_64 (char fact, char uple, long n, long nrhs, com plex *a, com plex *af, charequed, float*s, com plex *b, long ldb, com plex *x, long ldx, float *rcond, float *ferr, float *berr, long *info);

\section*{PURPOSE}
cppsvx uses the Cholesky factorization \(A=U * * H * U\) or \(A=\) L*L**H to com pute the solution to a com plex system of linear equations
\(A * X=B, w h e r e A\) is an \(N\) boy \(-N\) H erm itian positive defintie \(m\) atrix stored in packed form atand \(X\) and \(B\) are \(N\) boy \(-N\) RH S m atrices.

Emorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1.IfFACT = E', real scaling factors are computed to equilibrate
the system :
diag \((\mathrm{S})\) * A * diag \((\mathrm{S})\) * inv \((\operatorname{diag}(\mathrm{S}))\) * X \(=\operatorname{diag}(\mathrm{S})\) * B
W hether or not the system w illbe equilibrated depends on the
scaling of them atrix A , but ifequilibration is used, A is
overw ritten by diag \((\mathrm{S}) \star A\) *diag \((\mathrm{S})\) and B by diag \((\mathrm{S}) \star\) B .
2. IfFACT = N 'or E', the Cholesky decom position is used to
factor them atrix A (afterequilibration ifFACT = E ) as
\(A=U\) * \(U\), iff \(U P L O=U\) ', or
\(A=L * L \prime\), ifUPLO \(=L \prime\),
where \(U\) is an upper triangularm atrix, \(I\) is a low er triangular
\(m\) atrix, and 'indicates conjugate transpose.
3. Ifthe leading iboy-iprincipal \(m\) inor is not positive definite, then the routine retums w ith \(\mathbb{N N F O}=i\). O therw ise, the factored
form ofA is used to estim ate the condition num ber of the \(m\) atrix
A. If the reciprocal of the condition num ber is less than \(m\) achine
precision, \(\mathbb{N N} F O=N+1\) is retumed as a w aming, but the routine
stillgoes on to solve for X and com pute emorbounds as described below .
4.The system ofequations is solved forX using the factored form
of A.
5. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for it.
6. If equilibration w as used, the \(m\) atrix \(X\) is prem ultiplied by diag (S) so that it solves the original system before equilibration.

\section*{ARGUMENTS}

FACT (input)
Specifies w hether ornotthe factored form of the \(m\) atrix \(A\) is supplied on entry, and if not, w hether the m atrix A should be equilibrated before it is factored. = F': On entry, AF contains the factored form of . IfEQUED \(=Y\) ', them atrix A has been equilibrated w ith scaling factors given by \(S\).

A and AF w illnotbe m odified. \(=\mathrm{N}\) ': The m atrix
A w ill.be copied to A F and factored.
= E': The matrix A w ill be equilibrated if necessary, then copied to A F and factored.

UPLO (input)
= U ': U pper triangle ofA is stored;
= L': Low er triangle ofA is stored.
N (input) The num ber of linear equations, i.e., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collum ns of the \(m\) atrioes \(B\) and X. NRH \(S>=0\).

A (input/output)
On entry, the upper or low er triangle of the H erm itian \(m\) atrix A, packed colum nw ise in a linear array, except ifFACT = F'andEQUED = \(Y\) ', then
A \(m\) ust contain the equilibrated \(m\) atrix diag \((S) * A * \operatorname{diag}(S)\). The jth column ofA is stored in the array A as follow s: if UPLO = U', A (i+ \((j-1) * j 2)=A(i, 7)\) for \(1<=i<=j\) if \(U P L O=L\), A \((i+(j-1) *(2 n-j) / 2)=A(i, j)\) for \(j=i<=n\). See below for further details. A is not \(m\) odified if FACT = F' or \(N\) ', orifFACT = E'andEQUED = N 'on exit.

On exit, ifFACT=E' and EQUED = \(\mathrm{Y}^{\prime}, \mathrm{A}\) is overw ritten by diag \((S) * A * d i a g(S)\).

AF (input/output)
If \(F A C T=F\) ', then \(A F\) is an inputargum entand on entry contains the triangular factor \(U\) or \(L\) from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(L \star L * * H\), in the sam e storage form at as \(A\). IfEQ \(U E D\) ne. \(N\) ', then \(A F\) is the factored form of the equilibrated \(m\) atrix \(A\).

If \(F A C T=N\) ', then \(A F\) is an output argum ent and on exit retums the triangular factor \(U\) orL from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) of the originalm atrix A.

IfFACT \(=\mathrm{E}\) ', then \(A F\) is an output argum ent and on exit retums the triangular factor \(U\) orl from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) of the equilibrated \(m\) atrix A (see the description of \(A\) for the form of the equilibrated
matrix).

EQUED (input)
Specifies the form of equilibration thatw as done.
\(=\mathrm{N}^{\prime}\) : N o equilibration (alw ays true ifFACT = N 7 .
\(=Y\) ': Equilibration w as done, i.e., A has been replaced by diag \((\mathrm{S}) ~ * A * \operatorname{diag}(\mathrm{~S})\). EQUED is an inputargum entifFACT = F'; otherw ise, it is an outputargum ent.
\(S\) (input/output)
The scale factors forA; not accessed if EQUED = \(\mathrm{N}^{\prime} . \mathrm{S}\) is an inputargum entifFACT=F'; otherw ise, S is an outputargum ent. IfFACT=\(=\mathrm{F}^{\prime}\) and
EQUED \(=Y\) ', each elem entof m ustbe posilive.
B (input/output)
On entry, the N -by-NRHS righthand side m atrix B.
On exit, if EQUED = \(N^{\prime}\) ', B is notm odified; if
\(E Q U E D=Y ', B\) is overw ritten by diag \((S) * B\).

LD B (input)
The leading dim ension of the array B . LD B \(>=\) \(\max (1, N)\).

X (output)
If \(\mathbb{N}\) FO \(=0\) or \(\mathbb{N}\) FO \(=N+1\), the \(N\) boy \(-N\) RH S solution
\(m\) atrix \(X\) to the original system ofequations.
N ote that ifEQUED = Y',A and B arem odified on exit, and the solution to the equilibrated system is inv \((\) diag \((S)) \star X\).

LD X (input)
The leading dim ension of the array X . LD X >= \(\max (1, N)\).

RCOND (output)
The estim ate of the reciprocal condition num ber of the \(m\) atrix A after equilibration (ifdone). If
RCOND is less than the \(m\) achine precision (in particular, ifRCOND \(=0\) ), the \(m\) atrix is singularto w orking precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO \(>0\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X\) ( \()\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix \(X)\). If \(X T R U E\) is the true solution corresponding to \(X(\mathcal{J}), \operatorname{FERR}(\mathcal{)}\) ) is an estim ated upperbound forthe \(m\) agnitude of the largest ele-
\(m\) ent in \((X(\mathcal{O})\) X TRUE \()\) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector \(X(\mathcal{)}\) ) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).
W ORK (w orkspace)
dím ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N} F O=i\), and \(i\) is
\(<=\mathrm{N}\) : the leading m inor oforderiof A is not positive definite, so the factorization could not be com pleted, and the solution has not been com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1: \mathrm{U}\) is nonsingular, butRCOND is less than machine precision, \(m\) eaning that the \(m\) atrix is singularto working precision. Nevertheless, the solution and error bounds are com puted because there are a num berof siluations w here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{FURTHER DETAILS}

The packed storage schem e is illustrated by the follow ing exam ple when \(N=4, U P L O=U\) ':

Tw o-dim ensional storage of the H erm tian m atrix A :
```

al1 a12 a13 a14
a22 a23 a24
a33 a34 (aij= conjg (aji))

```
            a44

Packed storage of the upper triangle ofA :
\(A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]\)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
cpptrf-com pute the C holesky factorization of a com plex H erm itian positive definite m atrix A stored in packed form at

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CPPTRF (UPLO,N,A , INFO )}
CHARACTER * 1 UPLO
COMPLEX A (*)
\mathbb{NTEGER N, INFO}
SU BROUTINE CPPTRF_64 (UPLO ,N,A , INFO)
CHARACTER * 1 UPLO
COM PLEX A (*)
INTEGER*8 N, INFO
F95 INTERFACE
SUBROUT\mathbb{NE PPTRF (UPLO ,N,A , [NFO ])}
CHARACTER (LEN=1)::UPLO
COMPLEX,D IM ENSION (:) ::A
INTEGER::N,\mathbb{NFO}
SUBROUTINE PPTRF_64 (UPLO,N,A,[\mathbb{NFO ])}
CHARACTER (LEN=1) ::UPLO
COMPLEX,D IM ENSION (:) ::A
\mathbb{NTEGER (8)::N,\mathbb{NFO}}\mathbf{~}\mathrm{ ( }

```
void qpptrf(charuplo, intn, com plex *a, int*info);
void qpptrf_64 (charuplo, long n, com plex *a, long *info);

\section*{PURPOSE}
cpptrf com putes the Cholesky factorization of a com plex H er\(m\) tian positive definite \(m\) atrix A stored in packed form at.

The factorization has the form
\[
A=U * * H * U, \text { if } U P L O=U ' \text {, or }
\]
\(A=L * L * * H\), if \(U P L O=L \prime\),
\(w\) here \(U\) is an upper triangularm atrix and \(L\) is low er triangular.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': Upper triangle ofA is stored;
= L': Low er triangle of A is stored.

N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O \(n\) entry, the upper or low er triangle of the H er\(m\) itian \(m\) atrix A, packed colum nw ise in a linear amray. The jth colum n of \(A\) is stored in the array \(A\) as follow s: if UPLO \(=U '\) 'A (i+ (j \(1)^{\star} j 2\) ) \(=A(i, j)\) for \(1<=i<=j\) ifUPLO \(=\mathbb{L}\) ', A ( \(i+\) \((j-1) *(2 n-j / 2)=A(i, j)\) for \(j=i<=n\). See below for further details.

On exit, if \(\mathbb{N} F O=0\), the triangular factor \(U\) orL from the Cholesky factorization \(A=U * * H * U\) or \(A=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{H}\), in the sam e storage form atas A.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvałue
\(>0:\) if \(\mathbb{N} F O=i\), the leading \(m\) inoroforder is notpositive definite, and the factorization could notbe com pleted.

\section*{FURTHER DETAILS}

The packed storage schem e is illustrated by the follow ing exam ple w hen \(N=4, \mathrm{UPLO}=\mathrm{U}\) ':

Tw o-dim ensionalstorage of the \(H\) erm itian m atrix A :
```

a11 a12 a13 a14
a22 a23 a24
a33 a34 (aij= congg(a\ddot{i})
a44

```

Packed storage of the upper triangle of A :
\[
A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]
\]

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cpptri-com pute the inverse of a com plex H erm itian positive definite \(m\) atrix \(A\) using the Cholesky factorization \(A=\) \(\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) com puted by CPPTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CPPTRI(UPLO,N,A, INFO)}
CHARACTER * 1 UPLO
COM PLEX A (*)
\mathbb{NTEGER N,\mathbb{NFO}}\mathbf{~}=0
SU BROUT\mathbb{NE CPPTRI_64(UPLO,N,A,NNFO )}
CHARACTER * 1 UPLO
COM PLEX A (*)
INTEGER*8 N,\mathbb{NFO}

```
F95 INTERFACE
    SU BROUTINE PPTRI(UPLO , N, A, [ \(\mathbb{N F O}\) ])
    CHARACTER (LEN=1)::UPLO
    COMPLEX,D IM ENSION (:) ::A
    \(\mathbb{N} T E G E R:: N, \mathbb{N F O}\)
    SU BROUTINE PPTRI_64 (UPLO,N,A, [NFO ])
    CHARACTER (LEN=1) ::UPLO
    COMPLEX,D \(\mathbb{I M}\) ENSION (:) ::A
    \(\mathbb{N}\) TEGER (8) ::N, \(\mathbb{N} F O\)
void qpptri(charuplo, intn, com plex *a, int*info);
void qpptri_64 (char uplo, long n, com plex *a, long *info);

\section*{PURPOSE}
cpptricom putes the inverse of a com plex H erm itian positive definite \(m\) atrix \(A\) using the Cholesky factorization \(A=\) \(\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) com puted by CPPTRF.

\section*{ARGUMENTS}

UPLO (input)
\(=U^{\prime}:\) U pper triangular factor is stored in \(A\);
\(=L^{\prime}:\) Low er triangular factor is stored in A.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the triangular factor \(U\) or \(L\) from the Cholesky factorization \(A=U * * H * U\) or \(A=L * L * * H\), packed colum nw ise as a linear array. The jth colum \(n\) ofU orL is stored in the amay A as fol low s: if UPLO \(=U^{\prime}, A(i+(j 1) \star j 2)=U(i, 1)\) for \(1<=i<=j\) if UPLO \(=L\) ', A \((i+(j 1) *(2 n-j) / 2)\)
\(=L(i, j)\) for \(\dot{j}=i<=n\).

On exit, the upper or low er triangle of the (Her\(m\) itian) inverse of \(A\), overw riting the input factor U orL.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the i-th argum enthad an illegalvałue
\(>0:\) if \(\mathbb{N} F O=i\), the \((i, i)\) elem entof the factor
U orL is zero, and the inverse could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cpptrs - solve a system of linear equations A *X \(=\mathrm{B}\) w ith a Herm itian positive definite \(m\) atrix \(A\) in packed storage using the Cholesky factorization \(A=U * * H * U\) orA \(=L * L * * H\) com puted by CPPTRF

\section*{SYNOPSIS}
```

SUBROUTINE CPPTRS (UPLO,N,NRHS,A,B,LDB, INFO)
CHARACTER * 1 UPLO
COM PLEX A (*),B (LD B,*)
INTEGER N,NRHS,LDB,\mathbb{NFO}
SU BROUT\mathbb{NE CPPTRS_64 (UPLO,N,NRHS,A , B,LD B , INFO )}
CHARACTER * 1 UPLO
COM PLEX A (*),B (LD B,*)
INTEGER*8N,NRHS,LDB,INFO

```

\section*{F95 INTERFACE}
```

SU BROUTINE PPTRS (UPLO,N, NRHS],A,B,[LDB],[NFO])
CHARACTER (LEN=1)::UPLO
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::A
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : : B
$\mathbb{N}$ TEGER :: $\mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} \mathrm{B}, \mathbb{N}$ FO
SU BROUTINE PPTRS_64 (UPLO ,N, $\mathbb{N} R H S], A, B,[L D B],[\mathbb{N} F O])$
CHARACTER (LEN=1)::UPLO
COMPLEX,D $\mathbb{M}$ ENSION (:) ::A
COM PLEX,D $\mathbb{I M}$ ENSION (: : : : : B

```
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDB, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void qpptrs (char uplo, intn, intnris, com plex *a, com plex *b, int ldb, int *info);
void qpptrs_64 (charuplo, long n, long nrhs, com plex *a, com plex *b, long ldb, long *info);

\section*{PURPOSE}
cpptrs solves a system of linear equations A *X \(=\mathrm{B}\) w th a H erm itian positive definite \(m\) atrix A in packed storage using the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) com puted by CPPTRF.

\section*{ARGUMENTS}

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The orderof the m atrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of them atrix B. NRHS \(>=0\).

A (input) The triangular factor \(U\) or L from the Cholesky
factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\), packed
colum nw ise in a lineararay. The \(j\) th colum \(n\) of
U or L is stored in the array A as follow s: if UPLO = U', A \((i+(j-1) * j 2)=U(i, j)\) for \(1<=i<=j\) if UPLO \(=L \prime\), A \(\left(i+(j-1)^{*}(2 n-j) / 2\right)=L(i, j)\) for j=i<=n。

B (input/output)
On entry, the righthand side m atrix B. On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B . LD B >= \(\max (1, N)\).
= 0: successfulexit
< 0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
cptcon - com pute the reciprocal of the condition num ber (in the 1 -norm ) of a com plex H erm titian positive definite tridiagonalm atrix using the factorization \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\) or \(\mathrm{A}=\) \(\mathrm{U} * * \mathrm{H} * \mathrm{D} * \mathrm{U}\) com puted by CPTTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CPTCON N,DIAG,OFFD,ANORM,RCOND,W ORK,INFO)}
COM PLEX OFFD (*)
\mathbb{NTEGER N, \mathbb{NFO}}\mathbf{~}=0
REAL ANORM,RCOND
REALDIAG (*),WORK (*)
SU BROUT\mathbb{NE CPTCON_64 N,D IAG,OFFD,ANORM ,RCOND,W ORK, INFO)}
COM PLEX OFFD (*)
\mathbb{NTEGER*8 N, INFO}
REAL ANORM,RCOND
REALDIAG (*),WORK (*)
F95 INTERFACE

```

```

    COM PLEX,DIM ENSION (:) ::OFFD
    \mathbb{NTEGER ::N,\mathbb{NFO}}0=0
    REAL ::ANORM,RCOND
    REAL,D IM ENSION (:) ::D IA G,W ORK
    ```
    SUBROUTINE PTCON_64 ( \(\mathbb{N}\) ],D \(\mathbb{A} G, O F F D, A N O R M, R C O N D,[W O R K],[\mathbb{N} F O])\)

\section*{C INTERFACE}
\#include <sunperfh>
void qptoon (intn, float *diag, com plex *offd, float anorm , float *roond, int*info);
void qptcon_64 (long n, float *diag, com plex *offd, float anorm, float *rcond, long *info);

\section*{PURPOSE}
cptcon com putes the reciprocal of the condition num ber (in the 1-norm ) of a com plex H erm itian positive definite tridiagonalm atrix using the factorization \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\) or \(\mathrm{A}=\) \(\mathrm{U}{ }^{* *} \mathrm{H} * \mathrm{D}\) *U com puted by CPTTRF .

Norm (inv (A )) is com puted by a direct method, and the reciprocal of the condition num ber is com puted as
```

RCOND = 1 / (ANORM * norm (inv(A))).

```

\section*{ARGUMENTS}

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).

D IA G (input)
The \(n\) diagonalelem ents of the diagonal \(m\) atrix
D IA G from the factorization of \(A\), as com puted by
CPTTRF.

OFFD (input)
The ( \(n-1\) ) off-diagonalelem ents of the unit bidiagonal factorU orL from the factorization ofA, as com puted by CPTTRF.

ANORM (input)
The 1-norm of the originalm atrix A.
RCOND (output)
The reciprocal of the condition num ber of the \(m\) atrix \(A\), com puted as RCOND \(=1 /(A N O R M * A \mathbb{N} V N M)\), where \(A \mathbb{N} V N M\) is the 1 -nom of \(\operatorname{inv}(A)\) com puted in this routine.

W ORK (w orkspace)
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

Them ethod used is described in N icholas J . H igham, "E fficient A lgorithm s for C om puting the C ondition N um berof a TridiagonalM atrix", SIA M J.Sci.Stat. C om put., V ol. 7, No. 1, January 1986.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cpteqr-com pute alleigenvalues and, optionally, eigenvectors of a sym \(m\) etric positive definite tridiagonalm atrix by first factoring the \(m\) atrix using SPTTRF and then calling CBD SQ R to com pute the singularvahues of the bidiagonal factor

\section*{SYNOPSIS}

```

CHARACTER * 1 COMPZ
COMPLEX Z (LDZ,*)
INTEGER N,LD Z, \mathbb{NFO}
REALD (*),E (*),W ORK (*)
SUBROUT\mathbb{NE CPTEQR_64(COMPZ,N,D,E,Z,LDZ,W ORK,INFO)}
CHARACTER * 1 COMPZ
COM PLEX Z (LDZ,*)
\mathbb{NTEGER*8N,LD Z,NNFO}
REALD (*),E (*),W ORK (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE PTEQR COMPZ, $\mathbb{N}], D, E, Z,[L D Z],[W O R K],[\mathbb{N} F O])$
CHARACTER (LEN=1): :COM PZ
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::Z
$\mathbb{N}$ TEGER : : N, LD Z, $\mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::D ,E,W ORK
SU BROUTINE PTEQR_64 (COMPZ, $\mathbb{N}], D, E, Z,[L D Z],[W O R K],[\mathbb{N} F O])$

```

CHARACTER (LEN=1): : COMPZ
COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::Z
\(\mathbb{N}\) TEGER (8) :: N,LD Z, \(\mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void qpteqr(charcom pz, intn, float *d, float *e, com plex
*z, intldz, int *info);
void qpteqr_64 (charcom pz, long n, float *d, float *e, com plex *z, long ldz, long *info);

\section*{PURPOSE}
cpteqr com putes alleigenvalues and, optionally, eigenvectors of a sym \(m\) etric posilive definite tridiagonalm atrix by first factoring the \(m\) atrix using SPTTRF and then calling CBD SQR to com pute the singularvalues of the bidiagonal factor.

This routine com putes the eigenvalues of the positive definte tridiagonal m atrix to high relative accuracy. This m eans that if the eigenvalues range overm any orders ofm agnitude in size, then the sm alleigenvalues and comesponding eigenvectors \(w\) illbe com puted m ore accurately than, for exam ple, w ith the standard \(Q R\) m ethod.

The eigenvectors of a fullorband positive definite Herm itian \(m\) atrix can also be found ifCHETRD, CHPTRD, orCHBTRD has been used to reduce this \(m\) atrix to tridiagonal form . (The reduction to tridiagonal form, how ever, \(m\) ay preclude the possibility of obtaining high relative accuracy in the sm all eigenvalues of the originalm atrix, if these eigenvalues range overm any orders ofm agnitude.)

\section*{ARGUMENTS}

COM PZ (input)
= N ': C om pute eigenvalues only .
\(=\mathrm{V}\) : C om pute eigenvectors of original H erm itian \(m\) atrix also. A may \(Z\) contains the unitary \(m\) atrix used to reduce the originalm atrix to tridiagonal form . = I': C om pute eigenvectors of tridiagonal \(m\) atrix also.

N (input) The order of the m atrix. \(\mathrm{N}>=0\).
D (input/output)
O n entry, the \(n\) diagonalelem ents of the tridiagonalm atrix. On norm alexit, D contains the eigenvalues, in descending order.

E (input/output)
O \(n\) entry, the ( \(n-1\) ) subdiagonal elem ents of the tridiagonal matrix. On exit, E hasbeen destroyed.

Z (input) \(O n\) entry, if \(C O M P Z=V\) ', the unitary \(m\) atrix used in the reduction to tridiagonal form. On exit, if \(C O M P Z=V\) ', the orthonorm aleigenvectors of the original Herm tian matrix; if COM PZ = I', the orthonorm al eigenvectors of the tridiagonal \(m\) atrix. If \(\mathbb{N} F O>0\) on exit, \(Z\) contains the eigenvectors associated with only the stored eigenvalues. If COMPZ \(=N\) ', then \(Z\) is not referenced.

LD Z (input)
The leading dim ension of the amray Z . LD \(\mathrm{Z}>=1\), and ifCOM PZ \(=V\) 'or \(I\) ', LD \(Z>=\max (1, N)\).

W ORK (w orkspace)
dim ension ( \(4 * \mathrm{~N}\) )
\(\mathbb{N} F O\) (output)
= 0: successfulexit.
< \(0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue.
> 0 : if \(\mathbb{N} F O=i\), and \(i\) is: <= \(N\) the Cholesky factorization of the \(m\) atrix could notbe perform ed because the \(i\)-th principalm inorw as not positive definite. > N the SVD algorithm failed to converge; if \(\mathbb{N} F O=N+\) i, ioff-diagonal elem ents of the bidiagonal factor did not converge to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cptrfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is \(H\) erm itian positive definite and tridiagonal, and provides error bounds and backw ard errorestim ates for the solution

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE CPTRFS (UPLO,N,NRHS,D IAG,OFFD,D IA GF,OFFDF,B,LDB,X,}
LDX,FERR,BERR,W ORK,W ORK 2, INFO)
CHARACTER * 1 UPLO
COM PLEX OFFD (*),OFFDF (*),B (LDB,*),X (LDX ,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
REALDIAG (*),DIAGF (*),FERR (*),BERR (*),W ORK 2 (*)
SU BROUTINE CPTRFS_64 (UPLO,N,NRHS,DIAG,OFFD,D IAGF,OFFDF,B,LD B,
X,LDX,FERR,BERR,W ORK,W ORK 2, INFO)
CHARACTER * 1 UPLO
COM PLEX OFFD (*),OFFDF (*),B (LDB,*),X (LDX ,*),W ORK (*)
INTEGER*8N,NRHS,LDB,LDX,\mathbb{NFO}
REALDIAG (*),DIAGF (*),FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINEPTRFS (UPLO, \(\mathbb{N}], \mathbb{N} R H S], D \mathbb{A} G, O F F D, D \mathbb{A} G F, O F F D F, B,[L D B]\),


CHARACTER (LEN=1)::UPLO
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::OFFD,OFFDF,W ORK
COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::B,X
\(\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::D \(\mathbb{A} G, D \mathbb{A} G F, F E R R, B E R R, W\) ORK 2

SU BROUTINE PTRFS_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], D \mathbb{I A}\), OFFD, D IA GF, OFFDF, B, [LD B ], \(\mathrm{X},[\) LDX ],FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1)::UPLO
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::OFFD,OFFDF,W ORK
COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) :: B, X
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDB,LDX, \(\mathbb{N}\) FO
REAL,D \(\mathbb{M}\) ENSION (:) ::D \(\mathbb{A} G, D \mathbb{I} G F, F E R R, B E R R, W\) ORK 2

\section*{C INTERFACE}
\#include < sunperfh>
void qptrfs (char uplo, intn, intnrhs, float *diag, com plex *offd, float *diagf, com plex *offdf, com plex *b, int ldb, com plex *x, int ldx, float *ferr, float *berr, int *info);
void qptrfs_64 (charuplo, long n, long nrhs, float *diag, com plex *offd, float*diagf, com plex *offdf, com plex *b, long ldb, com plex *x, long ldx, float *ferr, float *berr, long *info);

\section*{PURPOSE}
cptrfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is H erm itian positive definite and tridiagonal, and provides error bounds and backw ard errorestim ates for the solution.

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the superdiagonal or the subdiagonal of the tridiagonalm atrix A is stored and the form of the factorization:
\(=\mathrm{U}\) ': OFFD is the superdiagonalofA, and A = \(\mathrm{U} * * \mathrm{H} * \mathrm{D}\) IA G * U ;
\(=\mathrm{L}\) : : OFFD is the subdiagonal of A , and \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{D} \operatorname{IA} \mathrm{G} * \mathrm{~L} * * \mathrm{H}\). (The two form s are equivalent if A is real.)

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS \(>=0\).

D IA G (input)
The n realdiagonalelem ents of the tridiagonal matrix A.

OFFD (input)
The ( \(n-1\) ) off-diagonalelem ents of the tridiagonal \(m\) atrix A (see UPLO).

D IA GF (input)
Then diagonalelem ents of the diagonal matrix D IA G from the factorization com puted by CPTTRF.
OFFDF (input)
The ( \(n-1\) ) off-diagonalelem ents of the unit bidiagonal factor \(U\) orL from the factorization computed by CPTTRF (see UPLO).
\(B\) (input) The righthand side \(m\) atrix \(B\).
LD B (input)
The leading dim ension of the array \(B\). LD B \(>=\) \(\max (1, \mathbb{N})\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by CPTTRS. On exit, the im proved solution \(m\) atrix \(X\).

\section*{LD X (input)}

The leading dim ension of the array X . LD X >= \(\max (1, \mathbb{N})\).

FERR (output)
The forw ard errorbound foreach solution vector \(X\) ( \()\) (the \(j\) th colum \(n\) of the solution \(m\) atrix \(X\) ). IfXTRUE is the true solution comesponding to \(X(\mathcal{J})\), FERR \((\mathcal{F})\) is an estim ated upperbound for the \(m\) agnitude of the largestelem ent in ( \(X(\mathcal{\jmath})\) XTRUE) divided by the \(m\) agnitude of the largestelem ent in X ( 7 ).

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector X (j) (ie., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension (N)
W ORK 2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{I N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cptsv - com pute the solution to a com plex system of linear equations \(A * X=B\), where \(A\) is an \(N\) by \({ }^{N}\) H erm itian positive definite tridiagonalm atrix, and X and B are N -by-NRHS m atrices.

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE CPTSV N,NRHS,D IAG,SUB,B,LDB,INFO)}
COM PLEX SUB (*),B (LDB,*)
\mathbb{NTEGER N,NRHS,LDB,INFO}
REALDIAG (*)
SU BROUT\mathbb{NE CPTSV_64 N ,NRHS,D IA G ,SUB,B,LDB,INFO)}
COM PLEX SUB (*),B (LDB,*)
\mathbb{NTEGER*8N,NRHS,LDB,INFO}
REALDIAG (*)
F95 INTERFACE

```

```

COM PLEX,D IM ENSION (:) ::SUB
COM PLEX,D IM ENSION (:,:) ::B
\mathbb{NTEGER ::N,NRHS,LDB,NNFO}
REAL,DIM ENSION (:) ::D IAG
SUBROUTINEPTSV_64(\mathbb{N ], NRHS],D IAG,SUB,B,[LDB],[NFO])}
COM PLEX,D IM ENSION (:) ::SUB
COM PLEX,D IM ENSION (:,:) ::B
\mathbb{NTEGER (8)::N,NRHS,LD B,\mathbb{NFO}}\mathbf{N}=\mp@code{N}

```

\section*{C INTERFACE}
\#include <sunperfh>
void qptsv (intn, intnrhs, float *diag, com plex *sub, com plex *b, int ldlo, int *info);
void qptsv_64 (long n, long nrhs, float *diag, com plex *sub, com plex *b, long ldlo, long *info);

\section*{PURPOSE}
cptsv com putes the solution to a com plex system of linear equations \(A * X=B\), where \(A\) is an \(N\) boy \(-N\) H erm itian positive definite tridiagonalm atrix, and X and B are N boy-NRHS \(m\) atrices.
\(A\) is factored as \(A=L * D * L * * H\), and the factored form of \(A\) is then used to solve the system ofequations.

\section*{ARGUMENTS}

N (input) The order of the matrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRH S >=0.

D IA G (input/output)
O n entry, the \(n\) diagonalelem ents of the tridiago-
nalm atrix A. On exit, the \(n\) diagonalelem ents of the diagonalm atrix D IA G from the factorization \(A\) \(=\mathrm{L} * \mathrm{D}\) IA G *L**H.

SU B (input/output)
O \(n\) entry, the \((n-1)\) subdiagonal elem ents of the tridiagonalm atrix A. On exit, the ( \(\mathrm{n}-1\) ) subdiagonalelem ents of the unitbidiagonal factorL from the L*D IA G *L**H factorization ofA. SUB can also be regarded as the superdiagonal of the unitbidiagonalfactor \(U\) from the \(U * * H * D I A G * U\) factorization of A.

B (input/output)
On entry, the N -by -NRH S righthand side m atrix B.
On ex弌, if \(\mathbb{N F O}=0\), the N boy \(-\mathrm{NRH} S\) solution
m atrix X .

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

IN FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N F O}=i\), the leading \(m\) inoroforder \(i\) is not positive definite, and the solution has not been com puted. The factorization has not been com pleted unless \(i=N\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cptsvx - use the factorization \(A=L * D * L * * H\) to com pute the solution to a com plex system of linear equations \(A * X=B\), w here A is an N -by N H erm itian positive definite tridiagonal \(m\) atrix and \(X\) and \(B\) are \(N\)-by \(-N\) R H S \(m\) atrices

\section*{SYNOPSIS}
```

SUBROUTINE CPTSVX (FACT,N,NRHS,DIAG,SUB,D IA GF,SUBF,B,LDB,X,
LDX,RCOND,FERR,BERR,WORK,W ORK2, INFO)
CHARACTER * 1 FACT
COM PLEX SUB (*),SUBF (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX, INFO
REALRCOND
REALDIAG (*),DIAGF (*),FERR (*),BERR (*),WORK 2 (*)

```
SU BROUTINE CPTSVX_64 (FACT,N,NRHS,DIAG,SUB,DIAGF,SUBF,B,LDB,
    \(X, L D X, R C O N D, F E R R, B E R R, W O R K, W O R K 2, \mathbb{N} F O)\)
CHARACTER*1FACT
COM PLEX SUB (*) , \(\operatorname{SUBF}(*), B(L D B, *), X(L D X, *), W O R K(*)\)
\(\mathbb{N}\) TEGER*8N,NRHS,LDB,LDX, \(\mathbb{N} F O\)
REALRCOND


\section*{F95 INTERFACE}

SUBROUTINEPTSVX \(\mathbb{E} A C T, \mathbb{N}], \mathbb{N} R H S], D \mathbb{A} G, S U B, D \mathbb{A} G F, S U B F, B,[L D B]\), X, [LDX],RCOND ,FERR, BERR, [W ORK], [W ORK 2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::FACT
COMPLEX,D \(\mathbb{M}\) ENSION (:) ::SUB,SUBF,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : \(\mathrm{B}, \mathrm{X}\)
\(\mathbb{N}\) TEGER ::N,NRHS,LDB,LDX, \(\mathbb{N}\) FO
REAL ::RCOND
REAL,D \(\mathbb{I}\) ENSION (:) :: D \(\mathbb{A} G, D \mathbb{A} G, F E R R, B E R R, W\) ORK 2

SU BROUTINE PTSVX_64 (FACT, \(\mathbb{N}], \mathbb{N R H S}], D \mathbb{I A} G, S U B, D \mathbb{A} G F, S U B F, B\), [ [D B ], X , [LDX ], RCOND ,FERR, BERR, [WORK], [WORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::FACT
COM PLEX ,D \(\mathbb{I}\) ENSION (:) :: SUB, SUBF,W ORK
COM PLEX, D \(\mathbb{M}\) ENSION (: : : : : B , X
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDB,LDX, \(\mathbb{N}\) FO
REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::D IA G ,D IA GF,FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void qptsvx (char fact, intn, intnrhs, float*diag, com plex *sub, float *diagf, com plex *subf, com plex *b, int ldb, com plex *x, int ldx, float *roond, float * ferr, float *berr, int *info);
void qptsvx_64 (char fact, long n, long nrhs, float *diag, com plex *sub, float *diagf, com plex *subf, com plex *b, long ldlo, com plex *x, long ldx, float *rcond, float * ferr, float *berr, long *info);

\section*{PURPOSE}
cptsvx uses the factorization \(A=L * D *\) \({ }^{*} * H\) to com pute the solution to a com plex system of linear equations \(A * X=B\), where A is an N -by N H erm itian positive definite tridiagonal \(m\) atrix and X and B are N -by-N R H S m atrioes.

Emorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT \(=N^{\prime}\), them atrix \(A\) is factored as \(A=L * D{ }^{*} \mathrm{~L} * * H\), w here L
is a unit low erbidiagonalm atrix and D is diagonal. The factorization can also be regarded as having the form
\(A=U * * H * D * U\).
2. If the leading iloy-iprincipal \(m\) inor is not posilive definite,
then the routine retums w ith \(\mathbb{N N F O}=\) i. O therw ise, the factored
form ofA is used to estim ate the condition num ber of the
m atrix
A. If the reciprocal of the condition num ber is less than \(m\) achine
precision, \(\mathbb{N}\) FO \(=\mathrm{N}+1\) is retumed as a w aming, but the routine
still goes on to solve for X and com pute emorbounds as described below .
3.The system ofequations is solved forX using the factored form of A.
4. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for it.

\section*{ARGUMENTS}

FACT (input)
Specifies w hether ornotthe factored form of the \(m\) atrix \(A\) is supplied on entry \(=\mathrm{F}^{\prime}\) : On entry, D IA GF and SUBF contain the factored form of A. D IA G , SUB, D IA G F , and SUBF w illnotbe m odified.
\(=N\) ': Them atrix A w illbe copied to D IA GF and SUBF and factored.

N (input) The order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber
of colum ns of the m atriges B and X. NRH S \(>=0\).

D IA G (input)
The \(n\) diagonalelem ents of the tridiagonal matrix A.

SUB (input)
The ( \(n-1\) ) subdiagonalelem ents of the tridiagonal \(m\) atrix A.

D IA GF (input/output)
IfFACT = \(\mathrm{F}^{\prime}\), then \(D \mathrm{IAGF}\) is an inputargum entand on entry contains the \(n\) diagonalelem ents of the diagonalm atrix D IA G from the \(L * D\) IA G *L **H factorization ofA. IfFACT \(=N\) ', then D IA GF is an outputargum entand on exitcontains the \(n\) diagonal
elem ents of the diagonal \(m\) atrix \(D \mathbb{I A}\) from the \(\mathrm{L} * \mathrm{D} \mathbb{I A}\) G \(\mathrm{L}^{* * *}{ }^{\mathrm{H}}\) factorization ofA.

\section*{SU BF (input/output)}

If \(F A C T=F\) ', then \(S U B F\) is an inputargum ent and on entry contains the ( \(n-1\) ) subdiagonalelem ents of the unit bidiagonal factor \(L\) from the \(\mathrm{L} * \mathrm{D} \operatorname{IAG} \mathrm{L}_{\mathrm{L} * *} \mathrm{H}\) factorization of A . IfFACT=N', then SUBF is an outputargum entand on exit contains the ( \(n-1\) ) subdiagonalelem ents of the unit bidiagonal factorl from the \(L * D I A G * L * * H\) factorization ofA.
\(B\) (input) The \(N\) toy \(-N\) RH S righthand side \(m\) atrix \(B\).
LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=N+1\), the N -by-NRH \(S\) solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the amay X . LD X >= \(\max (1, N)\).

\section*{RCOND (output)}

The reciprocalcondition num berof the \(m\) atrix \(A\). If RCOND is less than the machine precision (in particular, ifRCOND \(=0\) ), the \(m\) atrix is singular to working precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO \(>0\).

FERR (output)
The forw ard emorbound foreach solution vector \(X\) (j) (the \(j\) th colum \(n\) of the solution \(m\) atrix \(X\) ). IfXTRUE is the true solution comesponding to \(X(\mathcal{j})\), FERR ( \(\mathcal{j})\) is an estim ated upperbound for the \(m\) agnitude of the largestelem ent in (X ( \()\)-X TRU E) divided by the \(m\) agnitude of the largestelem ent in \(\mathrm{X}(\underset{)}{ }\).

BERR (output)
The com ponentw ise relative backw ard emorof each solution vector \(X(\mathcal{j})\) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension \((\mathbb{N})\)

W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{I N F O}\) (output)
= 0: successfiulexit
\(<0\) : if \(\mathbb{N F O}=-\) i, the \(i\)-th argum ent had an illegal value
\(>0\) : if \(\mathbb{N} F O=i\), and \(i\) is
\(<=N\) : the leading \(m\) inor of orderiof \(A\) is not positive definite, so the factorization could not be com pleted, and the solution has not been com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1\) : U is nonsingular, butRCOND is less than \(m\) achine precision, \(m\) eaning that the \(m\) atrix is singular to w orking precision. N evertheless, the solution and error bounds are com puted because there are a num ber of situations w here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cpttrf-com pute the L *D *L 'factorization of a com plex Herm itian positive definite tridiagonalm atrix A

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE CPTTRF N,D IAG,OFFD,NNFO)}
COM PLEX OFFD (*)
INTEGERN,\mathbb{NFO}
REALDIAG (*)
SU BROUT\mathbb{NE CPTTRF_64 N,D IAG,OFFD,INFO)}
COM PLEX OFFD (*)
INTEGER*8N,INFO
REALDIAG (*)
F95 INTERFACE
SUBROUT\mathbb{NEPTTRF (N ],D IAG ,OFFD,[\mathbb{NFO ])}}\mathbf{(})
COM PLEX,DIM ENSION (:) ::OFFD
\mathbb{NTEGER ::N,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REAL,D IM ENSION (:) ::D IA G

```

```

    COM PLEX,DIM ENSION (:) ::OFFD
    \mathbb{NTEGER (8)::N,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
    REAL,D IM ENSION (:) ::D IA G
    ```
C INTERFACE
    \#include < sunperfh>
void qpttrf(intn, float *diag, com plex *offd, int *info);
void qpttrf_64 (long n, float *diag, com plex *offd, long
*info);

\section*{PURPOSE}
cpttrf com putes the L *D *L 'factorization of a com plex H erm itian positive definite tridiagonalm atrix A. The factorization \(m\) ay also be regarded as having the form \(A=U{ }^{*} D * U\).

\section*{ARGUMENTS}

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

D IA G (input/output)
O n entry, the \(n\) diagonalelem ents of the tridiago-
nalm atrix A. On exit, the \(n\) diagonalelem ents of the diagonalm atrix D IA G from the \(L * D\) IA G *L' factorization of A.

OFFD (input/output)
O \(n\) entry, the \((n-1)\) subdiagonal elem ents of the tridiagonalm atrix A. On exit, the \((n-1)\) subdiagonalelem ents of the unitbidiagonal factorL from the L*D IA G *L' factorization ofA. OFFD can also be regarded as the superdiagonal of the unitbidiagonal factor \(U\) from the \(U\) *D \(\mathbb{A} G * U\) factorization of A.
\(\mathbb{I N}\) FO (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\mathrm{k}\), the k -th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=k\), the leading \(m\) inor oforder \(k\) notpositive definite; if \(k<N\), the factorization
could notbe com pleted, while ifk \(=\mathrm{N}\), the factorization w as com pleted, butD \(\mathbb{A} G \mathbb{N})=0\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cpters - solve a tridiagonalsystem of the form \(A * X=B\) using the factorization \(A=U\) *D *U orA \(=L * D * L\) 'com puted by CPTTRF

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE CPTTRS (UPLO,N,NRHS,D IAG,OFFD,B,LDB,INFO )}
CHARACTER * 1 UPLO
COM PLEX OFFD (*),B (LDB,*)
INTEGERN,NRHS,LDB,NNFO
REALDIAG (*)
SU BROUT\mathbb{NE CPTTRS_64 (UPLO ,N ,NRHS,D IA G ,OFFD ,B,LDB,INFO )}
CHARACTER * 1 UPLO
COM PLEX OFFD (*),B (LDB,*)
INTEGER*8N,NRHS,LDB,INFO
REALDIAG(*)

```
F95 INTERFACE
    SU BROUTINE PTTRS (UPLO, \(\mathbb{N}], \mathbb{N} R H S], D \mathbb{A} G, O F F D, B,[L D B],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO
    COM PLEX,D \(\mathbb{M}\) ENSION (:) ::OFFD
    COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::B
    \(\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O\)
    REAL,D \(\mathbb{I M} E N S I O N(:):: D \mathbb{I A G}\)
    SU BROUTINE PTTRS_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], D \mathbb{I} G, O F F D, B,[L D B],[\mathbb{N} F O])\)
    CHARACTER (LEN=1)::UPLO

COM PLEX,D \(\mathbb{M}\) ENSION (:) ::OFFD
COM PLEX , D \(\mathbb{M}\) ENSION (:,:) ::B
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDB, \(\mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::D IA G

\section*{C INTERFACE}
\#include <sunperfh>
void qpttrs (charuple, intn, intnrhs, float *diag, com plex *offd, com plex *b, int ldb, int *info);
void qpttrs_64 (charuplo, long n, long nrhs, float *diag, com plex *offd, com plex *b, long ldb, long *info);

\section*{PURPOSE}
qpttrs solves a tridiagonal system of the form
\(A * X=B\) using the factorization \(A=U\) *D *U or \(A=\) L *D *L 'com puted by CPTTRF. D is a diagonalm atrix specified in the vectorD, U (orL) is a unitbidiagonalm atrix whose superdiagonal (subdiagonal) is specified in the vector E , and X and B are N by NRH S m atriges.

\section*{ARGUMENTS}

UPLO (input)
Specifies the form of the factorization and w hether the vector OFFD is the superdiagonal of the upperbidiagonal factorU or the subdiagonal of the low er bidiagonal factorL. \(=\mathrm{U}\) ': A \(=\) U *D IA G *U, OFFD is the superdiagonal of \(U\) \(=\mathbb{L} ': A=\mathrm{L} * D \mathbb{A} G * L\) ', OFFD is the subdiagonal of L

N (input) The order of the tridiagonalm atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of them atrix B. NRHS \(>=0\).

D IA G (input)
The \(n\) diagonalelem ents of the diagonal \(m\) atrix
\(D I A G\) from the factorization \(A=U * D I A G * U\) orA \(=\) L *D IA G *L '.

OFFD (input/output)
If \(\mathrm{U} P \mathrm{O} O=\mathrm{U}\) ', the \((\mathrm{n}-1)\) superdiagonalelem ents of
the unit bidiagonal factor \(U\) from the factorization \(A=U\) *D IA G*U. If UPLO = \(L^{\prime}\) ', the ( \(n-1\) ) subdiagonal elem ents of the unitbidiagonal factorL from the factorization \(A=L \star D\) IA G \(* \mathrm{~L}\). .

B (input/output)
O n entry, the righthand side vectors \(B\) for the system of linearequations. On exit, the solution vectors, X .

LD B (input)
The leading dim ension of the array B. LD B >= max (1,N).
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0\) : if \(\mathbb{N} F O=-\mathrm{k}\), the k -th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cptts2 - solve a tridiagonalsystem of the form \(A * X=B\) using the factorization \(A=U{ }^{*} D * U\) orA \(=L * D * L\) 'com puted by CPTTRF

\section*{SYNOPSIS}

```

COM PLEX E (*),B (LDB,*)
INTEGER IUPLO,N,NRHS,LDB
REALD (*)
SUBROUTINE CPTTS2_64(IUPLO,N,NRHS,D ,E,B,LDB)
COM PLEX E (*),B (LDB,*)
INTEGER*8 \mathbb{UPLO,N,NRHS,LDB}
REALD (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE CPTTS2 (UPLO,N,NRHS,D,E,B,LDB)
COM PLEX,D $\mathbb{M}$ ENSION (:) ::E
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::B
$\mathbb{N} T E G E R:: \mathbb{I} P L O, N, N R H S, L D B$
REAL,D $\mathbb{M}$ ENSION (:) ::D
SU BROUTINE CPTTS2_64 (UUPLO ,N,NRHS,D ,E,B,LDB)
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::E
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : B
$\mathbb{N} T E G E R(8):: \mathbb{I U P L O}, N, N R H S, L D B$
REAL,D $\mathbb{M}$ ENSION (:) ::D

```

\section*{C INTERFACE}
\#include < sunperfh>
void qptts2 (int iuplo, intn, intnrhs, float *d, com plex
*e, com plex *b, int ldb);
void qptts2_64 (long iuplo, long n, long nrhs, float *d, com plex *e, com plex *b, long ldb);

\section*{PURPOSE}
cptts2 solves a tridiagonal system of the form
\(A * X=B\) using the factorization \(A=U{ }^{*} D * U\) or \(A=\) \(\mathrm{L} * \mathrm{D} * \mathrm{~L}\) 'com puted by CPTTRF. D is a diagonalm atrix specified in the vectorD, U (orL) is a unitbidiagonalm atrix whose superdiagonal (subdiagonal) is specified in the vectore, and \(X\) and \(B\) are N by NRH S m atrioes.

\section*{ARGUMENTS}

IUPLO (input)
Specifies the form of the factorization and whether the vectorE is the superdiagonal of the upperbidiagonal factor \(U\) or the subdiagonal of the low erbidiagonal factor \(L .=1: A=U * D * U\), \(E\) is the superdiagonal of \(U\)
\(=0: A=L * D\) *L' E is the subdiagonalof L
N (input) The order of the tridiagonalm atrix \(\mathrm{A} . \mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS \(>=0\).
\(D\) (input) The \(n\) diagonalelem ents of the diagonal \(m\) atrix \(D\)
from the factorization \(A=U{ }^{*} D * U\) orA \(=L * D * L '\).
\(E\) (input) If \(\mathbb{U P L O}=1\), the \((n-1)\) superdiagonalelem ents of the unit bidiagonal factor \(U\) from the factorization \(A=U\) *D *U. If \(\mathbb{I} P L O=0\), the ( \(n-1\) ) subdiagonalelem ents of the unitbidiagonal factorL from the factorization \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L}\).

B (input/output)
On entry, the righthand side vectors \(B\) for the
system of linearequations. On exit, the solution
vectors, X .
LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, N)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
crot -apply a plane rotation, where the cos (C) is realand the sin (S) is com plex, and the vectors \(X\) and \(Y\) are com plex

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CROTN,X,INCX,Y, INCY,C,S)}
COM PLEX S
COM PLEX X (*),Y (*)
\mathbb{NTEGERN,}\mathbb{NCX,\mathbb{NCY}}\mathbf{N}=\mp@code{N}
REALC

```

```

COM PLEX S
COM PLEX X (*),Y (*)
INTEGER*8N,\mathbb{NCX,}\mathbb{NCY}
REALC
F95 INTERFACE

```

```

COM PLEX ::S
COM PLEX,D IM ENSION (:) ::X,Y
\mathbb{NTEGER::N,\mathbb{NCX,INCY}}\mathbf{N}={
REAL ::C

```

```

COM PLEX ::S
COM PLEX,D IM ENSION (:) ::X,Y
\mathbb{NTEGER (8)::N,\mathbb{NCX,INCY}}\mathbf{~}=\mp@code{N}

```

REAL ::C

\section*{C INTERFACE}
\#include <sunperfh>
void crot(intn, com plex *x, intincx, com plex *y, intincy, floatc, com plex *s);
void crot 64 (long n, com plex *x, long incx, com plex *y, long incy, float c, com plex *s);

\section*{PURPOSE}
crot applies a plane rotation, w here the cos (C) is real and the \(\sin (S)\) is com plex, and the vectors \(X\) and \(Y\) are com plex.

\section*{ARGUMENTS}

N (input)
The num ber of elem ents in the vectors X and Y .

X (input/output)
O n input, the vectorX. O n output, X is overw ritten \(w\) th \(C * X+S * Y\).
\(\mathbb{N C X}\) (input)
The increm ent betw een successive values of \(Y\).
\(\mathbb{N} C X<>0\).

Y (input/output)
O n input, the vector \(Y\). O n output, \(Y\) is overw nitten \(w\) th \(-C O N J G(S) * X+C * Y\).
\(\mathbb{N} C Y\) (input)
The increm ent betw een successive values of \(Y\).
\(\mathbb{N} C Y<>0\).

C (input)
\(S\) (input)
\(C\) and \(S\) define a rotation
[ C S ]
\([-\infty \cap \dot{g}(\mathrm{~S}) \mathrm{C}]\)
where \(\mathrm{C} * \mathrm{C}+\mathrm{S} * \mathrm{CONJG}(\mathrm{S})=1.0\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

crotg -C onstructa G iven S plane rotation

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CROTG (A,B,C,S)}
COM PLEX A,B,S
REALC
SUBROUT\mathbb{NECROTG_64(A,B,C,S)}
COM PLEX A,B,S
REALC

```
F95 INTERFACE
    SU BROUTINEROTG (A, B, C, S)
    COM PLEX ::A,B,S
    REAL ::C
    SU BROUTINEROTG_64 (A,B,C,S)
    COM PLEX ::A,B,S
    REAL ::C

\section*{C INTERFACE}
\#include <sunperfh>
void crotg (com plex *a, com plex *b, float *c, com plex *s);
void crotg_64 (com plex *a, com plex *b, float *c, com plex *s);

\section*{PURPOSE}
crotg C onstructa G iven Splane rotation that will annihilate an elem entof a vector.

\section*{ARGUMENTS}

A (input/output)
On entry, A contains the entry in the firstvector that comesponds to the elem ent to be annihilated in the second vector. On exit, contains the nonzero elem ent of the rotated vector.
B (input)
On entry, B contains the entry to be annihilated in the second vector. U nchanged on exit.

C (output)
On exit, C and S are the elem ents of the rotation \(m\) atrix thatw illibe applied to anninilate \(B\).

S (output)
On exit, \(C\) and \(S\) are the elem ents of the rotation \(m\) atrix thatw ill be applied to annihilate \(B\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

cscal-C om pute y := alpha * y

```

\section*{SYNOPSIS}
```

SUBROUTINECSCAL N,ALPHA,Y,\mathbb{NCY)}
COM PLEX ALPHA
COM PLEX Y (*)
\mathbb{NTEGER N, INCY}
SU BROUTINE CSCAL_64 N,ALPHA,Y,INCY)
COM PLEX ALPHA
COM PLEX Y (*)
INTEGER*8N,\mathbb{NCY}
F95 INTERFACE

```

```

    COMPLEX ::ALPHA
    COMPLEX,D IM ENSION (:) ::Y
    \mathbb{NTEGER ::N,\mathbb{NCY}}\mathbf{N}=\mp@code{N}
    SU BROUTINE SCAL_64 (N ],ALPHA,Y,[\mathbb{N CY ])}
    COMPLEX ::ALPHA
    COM PLEX,D IM ENSION (:) ::Y
    \mathbb{NTEGER (8)::N,INCY}
    C INTERFACE
\#include <sunperfh>

```
void cscal(intn, com plex *alpha, com plex *y, int incy);
void cscal 64 (long n, complex *alpha, complex *y, long incy);

\section*{PURPOSE}
cscalC om pute \(y:=\) alpha * \(y\) w here alpha is a scalar and \(y\) is an \(n\)-vector.

\section*{ARGUMENTS}

N (input)
O n entry, \(N\) specifies the num ber of elem ents in the vector. N m ustbe at leastone for the subroutine to have any visible effect. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha.
U nchanged on exit.

Y (input/output)
(1+(n-1)*abs( \(\mathbb{N} C Y\) ) ). On entry, the increm ented array \(Y\) m ust contain the vectory. On exit, \(Y\) is overw rilten by the updated vectory.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y\). \(\mathbb{N} C\) Y m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS

\section*{NAME}
csctr-Scatters elem ents from \(x\) into \(y\).

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CSCTR NZ,X,NDD,Y)}

```
COM PLEX X (*), Y (*)
\(\mathbb{N}\) TEGER NZ
\(\mathbb{N} T E G E R \mathbb{N} D X(*)\)
SUBROUTINECSCTR_64 \(\mathbb{N} Z, X, \mathbb{N} D X, Y)\)
COMPLEXX \({ }^{(*)}\) ) Y (*)
\(\mathbb{N} T E G E R * 8 N Z\)
\(\mathbb{N} T E G E R * 8 \mathbb{N} D X(*)\)
F95 \(\mathbb{I N}\) TERFACE
SUBROUTINE SCTR (NZ],X, \(\mathbb{N} D X, Y\) )
COMPLEX,D \(\mathbb{I}\) ENSION (:) ::X,Y
\(\mathbb{N}\) TEGER ::NZ
\(\mathbb{N}\) TEGER,D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{N} D \mathrm{X}\)
SUBROUTINE SCTR_64 ( \(\mathbb{N} Z], X, \mathbb{N} D X, Y)\)
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::X,Y
\(\mathbb{N}\) TEGER (8) ::NZ
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{N D} \mathrm{X}\)

\section*{PURPOSE}

CSC TR -Scatters the com ponents of a sparse vector \(x\) stored in com pressed form into specified com ponents of a vectory
in fullstorage form .
do \(i=1, n\)
\(y(\) indx (i) \()=x(i)\)
enddo

\section*{ARGUMENTS}

N Z (input) - \(\mathbb{N}\) TEGER
N um ber of elem ents in the com pressed form .
U nchanged on exit.

X (input)
V ector containing the values to be scattered from com pressed form into fill storage form. U nchanged on exit.
\(\mathbb{N} D X\) (input) \(-\mathbb{N} T E G E R\)
\(V\) ector containing the indiges of the com pressed form. It is assum ed that the elem ents in \(\mathbb{N} D \mathrm{X}\) are distinctand greater than zero. U nchanged on exit.

Y (output)
V ectorw hose elem ents specified by indx have been set to the corresponding entries ofx. Only the elem ents corresponding to the indices in indx have been m odified.

\section*{Contents}
- NAME
- SYNOPSIS

\title{
- F95 INTERFACE
}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
cskym m -Skyline form atm atrix-m atrix \(m\) ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CSKYMM(TRANSA,M,N,K,ALPHA,DESCRA,}

* VAL,PNTR,B,LDB,BETA,C,LDC,WORK,LWORK)
INTEGER TRANSA,M,N,K,DESCRA (5),
* LDB,LDC,LW ORK
INTEGER PNTR(*),
COMPLEX ALPHA,BETA
COM PLEX VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NECSKYMM_64(TRANSA,M,N,K,ALPHA,DESCRA,}
* VAL,PNTR,B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,M,N,K,DESCRA (5),
* LDB,LDC,LW ORK
INTEGER*8 PNTR (*),
COMPLEX ALPHA,BETA
COMPLEX VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
w here NN Z = PN TR (K +1)PN TR (1) (upper triangular)
NNZ = PNTR (M +1)PNTR (1) (low ertriangular)
PN TR 0 size = (K+1) (uppertriangular)
PN TR () size = M +1) (low ertriangular)

```

\section*{F95 INTERFACE}

SUBROUTINE SKYMM (TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L\), * PNTR, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
\(\mathbb{N} T E G E R\) TRANSA, M, K
\(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION (:) :: DESCRA, PNTR
COM PLEX ALPHA,BETA
COMPLEX,D \(\mathbb{I M}\) ENSION (:) :: VAL

SUBROUTINESKYMM_64(TRANSA,M, N],K,ALPHA,DESCRA,VAL, * PNTR, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
\(\mathbb{I N T E G E R *}\) TRANSA, M, K
\(\mathbb{N} T E G E R * 8, D \mathbb{I}\) ENSION (:) :: DESCRA, PNTR
COM PLEX ALPHA,BETA
COM PLEX,D \(\mathbb{M}\) ENSION (:) :: VAL
COM PLEX,D \(\mathbb{I M}\) ENSION (: : : :: B,C

\section*{DESCRIPTION}
\[
C \text { <-alpha op (A) B + beta C }
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, \(A\) is a \(m\) atrix represented in skyline form at and op(A) is one of \(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=c o n j\left(A^{\prime}\right)\).
( 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate with m atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in \(m\) atrix A

N \(\quad N\) um berof colum ns in \(m\) atrix \(C\)

K \(\quad N\) um berof colum \(n s\) in \(m\) atrix A

ALPHA Scalarparam eter

DESCRA () D escriptor argum ent. Five elem ent integer array
DESCRA (1) m atrix structure
0 : general \(\mathbb{N} O T\) SUPPORTED)
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm Aian ( \(\mathrm{A}=\mathrm{CONJG}\) (A))
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJ}\) ( A ) )
D ESCRA (2) upper/low er triangular indicator
1: low er

2 :upper
DESCRA (3) main diagonal type
0 :non-unit
1 : unit
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED)
0 :C C ++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT \(\mathbb{I}\) PLEM ENTED)
0 : unknown
1 : no repeated indices

VAL () array contain the nonzeros of in in skyline profile form . Row -oriented ifD ESCRA (2) = 1 (low er triangular), Colum \(n\) oriented ifD ESCRA (2) \(=2\) (upper triangular).

PN TR ( integer anray of length M +1 (low er triangular) or \(\mathrm{K}+1\) (uppertriangular) such thatPN TR (I) PN TR (1)+1 points to the location in VAL of the firstelem ent of the skyline profile in row (colum n) I.

B 0 rectangular anray w ith first dim ension LD B .
LD B leading dim ension ofB
BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC
W ORK () scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK anay. LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/n cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse .ps

\section*{NOTES/BUGS}

The SK Y data structure is not supported for a generalm atrix structure (DESCRA (1)=0).

A lso not supported:
1. low ertriangularm atrix \(A\) of size \(m\) by \(n\) where \(m>n\)
2. uppertriangularm atrix A of size m by n wherem < n

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
cskysm -Skyline form at triangular solve

\section*{SYNOPSIS}
```

SUBROUTINE CSKYSM(TRANSA,M ,N ,UN ITD,DV,ALPHA,DESCRA,

* VAL,PNTR,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,M,N,UNITD,DESCRA (5),}
* LDB,LDC,LWORK
\mathbb{NTEGER PNTR (*),}
COM PLEX ALPHA,BETA
COM PLEX DV M),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SU BROUTINE CSKY SM _64(TRANSA,M ,N,UNTID,DV,A LPHA,DESCRA,
* VAL,PNTR,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,M,N,UNITD,DESCRA (5),}
* LDB,LDC,LW ORK
INTEGER*8 PNTR (*),
COM PLEX ALPHA,BETA
COM PLEX DV M),VAL NNZ),B (LDB,*),C (LDC,\star),W ORK (LW ORK)
where NN Z = PN TR (M+1)PN TR (1) (uppertriangular)
NN Z = PNTR (K +1)-PNTR (1) (low ertriangular)
PN TR 0) size = M +1) (uppertriangular)
PNTR 0 size = (K+1) (low ertriangular)

```

\section*{F95 INTERFACE}

SUBROUTINE SKYSM (TRANSA, M, \(\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L\), * PNTR, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
\(\mathbb{N} T E G E R \quad\) TRANSA, M,UNITD
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:)::\) DESCRA, PNTR

SU BROUTINE SKYSM _64 (TRANSA, M, \(\mathbb{N}]\), UN ITD , DV,ALPHA, DESCRA,
* VAL, PNTR, B, [LDB],BETA,C, [LDC], [W ORK], [LW ORK])
\(\mathbb{N} T E G E R * 8\) TRANSA, M, UNITD
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}\) (:) :: DESCRA, PNTR
COMPLEX ALPHA,BETA
COM PLEX, D \(\mathbb{I M} E N S I O N(:):: V A L, D V\)
COM PLEX , D \(\mathbb{M}\) ENSION (: :) :: B , C

\section*{DESCRIPTION}
\(C<-A L P H A \quad\) Op (A) B + BETA \(C \quad C<-A L P H A D\) op (A) B + BETA C C <-ALPHA Op (A) D B + BETA C where A LPHA and BETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense \(m\) atrices, \(D\) is a diagonalscaling \(m\) atrix, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix represented in skyline form at and op (A ) is one of
```

op (A ) = inv (A ) or op (A ) = inv (A ) or op (A ) = inv (oonjg (A')).

``` (inv denotesm atrix inverse, 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRAN SA Indicates how to operate \(w\) ith the sparse \(m\) atrix 0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M \(\quad \mathrm{N}\) um berof row s in \(m\) atrix \(A\)
\(N \quad N\) um berof colum \(n s\) in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity \(m\) atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)
4 :A utom atic row or colum \(n\) scaling (see section N O TES for further details)

DV () A rray of length M containing the diagonalentries of the scaling diagonalm atrix D .
```

D ESCRA () D escriptor argum ent. Fi̇ve elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric (A=A)
2:H erm itian (A = CON JG (A ))
3:Triangular
4 : Skew (A nti)-Symm etric (A=-A )
5 :D iagonal
6:Skew Herm itian (A= CON JG (A ) )
N ote: For the routine, D ESCRA (1)=3 is only supported.
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 :non-unit
1 :unit
DESCRA (4) A nay base $\mathbb{N} O T \mathbb{M}$ PLEM ENTED)
$0: C / C++$ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? (NOT $\mathbb{M}$ PLEM ENTED)
0 : unknown
1 : no repeated indices

```

VAL () array contain the nonzeros ofA in skyline profile form .
Row -oriented ifD ESCRA (2) = 1 (low er triangular), colum n oriented ifD ESCRA (2) \(=2\) (upper triangular) .

PN TR () integer array of length \(M+1\) (low ertriangular) or
\(\mathrm{K}+1\) (upper triangular) such thatPN TR (I)-PN TR (1)+1 points to the location in VAL of the firstelem ent of the skyline profile in row (colum n) I.

B 0 rectangular anay w ith first dim ension LD B .

LD B leading dim ension ofB

BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch amay of length LW ORK.
On exit, ifLW ORK \(=-1, W\) ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK array. LW ORK should be at leastM .

Forgood perform ance, LW O RK should generally be larger.

For optim um perform ance on \(m\) ultiple processors, LW ORK \(>=M\) *N _CPU S where N_CPUS is the m axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser'S G uide available at:
http://m ath nist.gov/m cso/Staff/K Rem ington/Espblas/
"D ocum ent forthe B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlio .org/utk/papers/sparse _ps

\section*{NOTES /BUGS}
1.A lso notsupported:
a. low er triangularm atrix A ofsizem by n wherem \(>n\)
b. upper triangularm atrix \(A\) of size \(m\) by \(n\) where \(m<n\)
2. N o test for singularity ornear-singularity is included in this routine. Such tests m ust.be perform ed before calling this routine.
3. If U N ITD \(=4\), the routine scales the row s of \(A\) if \(D E S C R A(2)=1\) and the colum ns ofA if \(D E S C R A(2)=2\) such that their 2 -norm s are one. The scaling \(m\) ay im prove the accuracy of the com puted solution. C orresponding entries of V A L are changed only in this particular case. O n retum D V m atrix stored as a vector contains the diagonalm atrix by w hich the row \(s\) (colum ns) have been scaled. U N ITD = 2 if \(D E S C R A(2)=1\) and UN ITD \(=3\) if \(D E S C R A(2)=2\) should be used for the next calls to the routine \(w\) ith overw rilten \(V A L\) and \(D V\).

WORK \((1)=0\) on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here i is the row (colum n) num berw hich 2 -norm is exactly zero.
4. If \(D E S C R A(3)=1\) and \(U \mathrm{~N}\) ITD \(<4\), the unit diagonalelem ents
\(m\) ightorm ightnotbe referenced in the SK Y representation of a sparse m atrix. They are not used anyw ay in these cases. ButifUN ITD = 4, the unit diagonalelem ents M U ST be referenced in the SK Y representation.
5.The routine can be applied for solving triangular system \(s\) w hen the upper or low er triangle of the general sparse \(m\) atrix \(A\) is used. H ow ever DESCRA (1) m ustbe equal to 3 in this case.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cspcon -estim ate the reciprocalof the condition num ber (in the 1-norm ) of a com plex sym \(m\) etric packed \(m\) atrix \(A\) using the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by CSPTRF

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
COM PLEX AP (*),W ORK (*)
INTEGERN,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}
REAL ANORM,RCOND

```

```

CHARACTER * 1 UPLO
COM PLEX AP (*),W ORK (*)
\mathbb{NTEGER*8 N,\mathbb{NFO}}\mathbf{N}+
INTEGER*8 \mathbb{PIVOT (*)}
REALANORM,RCOND

```

\section*{F95 INTERFACE}
```

SU BROUTINE SPCON (UPLO,N,AP, $\mathbb{P} \mathbb{I V O T}, A N O R M, R C O N D,[\mathbb{O R K}],[\mathbb{N} F O])$
CHARACTER (LEN=1) ::UPLO
COMPLEX,D $\mathbb{M}$ ENSION (:) ::AP,W ORK
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
$\mathbb{N}$ TEGER,D $\mathbb{I}$ ENSION (:) :: $\mathbb{P} \mathbb{I V}$ OT
REAL ::ANORM,RCOND

```

SU BROUTINE SPCON_64 (UPLO ,N,AP, \(\mathbb{P} \mathbb{I V O T}, A N O R M, R C O N D,[W O R K],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::AP,W ORK
\(\mathbb{N}\) TEGER ( 8 ):: \(\mathrm{N}, \mathbb{I N F O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V O T}\)
REAL ::ANORM,RCOND

\section*{C INTERFACE}
\#include <sunperfh>
void cspcon (char uplo, int n, com plex *ap, int *ípivot, float anorm , float*rcond, int *info);
void cspcon_64 (charuplo, long n, com plex *ap, long *ipívot, float anorm, float *rcond, long *info);

\section*{PURPOSE}
cspcon estim ates the reciprocal of the condition num ber (in the 1-norm ) of a com plex sym \(m\) etric packed \(m\) atrix A using the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by CSPTRF.

A n estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND = 1 / (ANORM * norm (inv (A))).

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) ': U ppertriangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\);
= L ': Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D}\) * \(\mathrm{L} * * \mathrm{~T}\).

N (input) The order of the matrix A. \(\mathrm{N}>=0\).

AP (input)
Com plex array, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2\) ) The block diagonal \(m\) atrix \(D\) and the \(m\) ultipliers used to obtain the factorU orL as com puted by CSPTRF, stored as a packed triangularm atrix.

PIVOT (input)
Integer array, dim ension \(\mathbb{N}\) ) D etails of the interchanges and the block structure ofD as determ ined by CSPTRF.

ANORM (input)
The 1-norm of the originalm atrix A.

\section*{RCOND (output)}

The reciprocal of the condition num ber of the \(m\) atrix \(A\), com puted as RCOND \(=1 /(A N O R M * A \mathbb{N} V N M)\), \(w\) here \(A \mathbb{N} V N M\) is an estim ate of the 1 -norm of inv (A) com puted in this routine.

W ORK (w orkspace)
C om plex array, dim ension ( \(2 \star \mathrm{~N}\) )
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
csprfs - im prove the com puted solution to a system of linear equations when the coefficientm atrix is sym \(m\) etric indefinite and packed, and provides errorbounds and backw ard error estim ates for the solution

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CSPRFS (UPLO,N,NRHS,A,AF, \mathbb{PIVOT,B,LDB,X,LDX,FERR,}}\mathbf{N},\textrm{N},\textrm{N}
BERR,W ORK,W ORK2,INFO)
CHARACTER * 1 UPLO
COM PLEX A (*),AF (*),B (LDB,*),X (LDX ,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
\mathbb{NTEGER \mathbb{PIVOT (*)}}\mathbf{(})
REAL FERR (*),BERR (*),WORK2 (*)
SU BROUTINE CSPRFS_64 (UPLO,N,NRHS,A,AF, PPIV OT,B,LD B,X,LDX,
FERR,BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 UPLO
COM PLEX A (*),AF (*),B (LD B ,*),X (LDX,*),W ORK (*)
\mathbb{NTEGER*8N,NRHS,LDB,LDX,INFO}
INTEGER*8 \mathbb{PIVOT (*)}
REAL FERR (*),BERR (*),W ORK2 (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE SPRFS (UPLO, N, $\mathbb{N} R H S], A, A F, \mathbb{P} \mathbb{I} O T, B,[L D B], X,[L D X]$, FERR, BERR, [W ORK], [W ORK 2], [ $\mathbb{N F O}$ ])
CHARACTER (LEN=1) ::UPLO
COMPLEX,D $\mathbb{I M} E N S I O N(:):: A, A F, W$ ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : $\mathrm{B}, \mathrm{X}$

```
\(\mathbb{N}\) TEGER : \(: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} \mathrm{B}, \mathrm{LD} \mathrm{X}, \mathbb{N} F \mathrm{O}\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{I} O T\)
REAL,D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK2

SU BROUTINE SPRFS_64 (UPLO,N, \(\mathbb{N R H S}], A, A F, \mathbb{P} \mathbb{I} O T, B,[L D B], X\), [LDX],FERR,BERR, [WORK],[WORK2], [ \(\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO
COM PLEX , D \(\mathbb{M}\) ENSION (:) ::A ,AF,W ORK
COM PLEX, D IM ENSION (: : : : : B, X
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL,D \(\mathbb{I}\) ENSION (:) ::FERR,BERR,W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void csprfs (charuple, intn, intnihs, com plex *a, com plex *af, int *ipivot, com plex *b, int ldb, com plex *x, intldx, float * ferr, float *berr, int *info);
void csprfs_64 (charuple, long n, long nrhs, complex *a, complex *af, long *ipivot, com plex *b, long ldb, com plex *x, long ldx, float * ferr, float *berr, long *info);

\section*{PURPOSE}
csprfs im proves the com puted solution to a system of linear equations w hen the coefficientm atrix is sym \(m\) etric indefinte and packed, and provides emorbounds and backw ard error estim ates for the solution.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': U ppertriangle of A is stored;
= L': Low ertriangle ofA is stored.
N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber
of collm ns of the matrioes \(B\) and \(X\). NRHS \(>=0\).

A (input) C om plex array, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2\) ) The upper or low er triangle of the symm etricm atrix A, packed colum nw ise in a linearanay. The jth colum \(n\) of

A is stored in the array A as follow s: ifU PLO = \(U^{\prime}, A(i+(j-1) * j 2)=A(i, 7)\) for \(1<=i<=j\) if UPLO = L', A (i+ (j-1)* (2*n-j) 2 ) =A (i, \(\bar{i})\) for \(j=\mathrm{i}<=\mathrm{n}\).

AF (input)
C om plex array, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2\) ) The factored form of them atrix A. A F contains the block diagonalm atrix D and the \(m\) ultipliers used to obtain the factor \(U\) or \(L\) from the factorization \(A=\) \(U * D * U * * T\) orA \(=L * D * L * * T\) as com puted by CSPTRF, stored as a packed triangularm atrix.
PIVOT (input)
Integer array, dim ension (N) D etails of the interchanges and the block structure ofD as determ ined by CSPTRF.

B (input) Com plex array, dim ension (LDB,NRHS) The righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).

X (input/output)
Com plex array, dim ension (LD X NRHS) On entry, the solution matrix X , as com puted by CSPTRS. On exit, the im proved solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the aray X . LD X >= \(\max (1, N)\).

FERR (output)
Realarray, dim ension (NRHS) The estim ated forw ard error bound foreach solution vector \(X(\mathcal{)}\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix \(X)\). IfX TRUE is the true solution comesponding to \(X(\mathcal{J}), \operatorname{FERR}(\mathcal{)})\) is an estim ated upperbound for the \(m\) agnitude of the largest elem entin \((X(\mathcal{j})-X T R U E)\) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is am ostalw ays a slightoverestim ate of the true error.

BERR (output)
Realarray, dim ension (NRHS) The com ponentw ise relative backw and enror ofeach solution vector
\(\mathrm{X}(\mathcal{)})\) (i.e., the sm allestrelative change in any elem ent of A orB thatm akes X ( \(\mathcal{I}\) ) an exact solu-
tion).

W ORK (w orkspace)
C om plex array, dim ension ( \(2 \star \mathrm{~N}\) )
W ORK 2 (w orkspace)
Integer array, dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO = -i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
cspsv - com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
COM PLEX AP (*),B (LDB,*)
\mathbb{N TEGER N,NRHS,LDB, NNFO}
INTEGER \mathbb{PIVOT (*)}
SU BROUTINE CSPSV_64(UPLO,N,NRHS,AP,\mathbb{P IVOT,B,LDB, IN FO )}
CHARACTER * 1 UPLO
COM PLEX AP (*),B(LDB,*)
INTEGER*8N,NRHS,LDB, INFO
\mathbb{NTEGER*8 \mathbb{P IVOT (*)}}\mathbf{(})=

```

\section*{F95 INTERFACE}

SU BROUTINE SPSV (UPLO,N, NRHS],AP, \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}],[\mathbb{N} F O])\)
CHARACTER (LEN=1) ::UPLO
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::AP
COM PLEX,D \(\mathbb{I}\) ENSION (:,:) ::B
\(\mathbb{N}\) TEGER ::N,NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
SU BROUTINE SPSV_64 (UPLO,N, NRHS],AP, \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\operatorname{LDB}],[\mathbb{N} F O\) ])
CHARACTER (LEN=1) ::UPLO

COM PLEX,D \(\mathbb{M}\) ENSION (:) ::AP
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::B
\(\mathbb{N}\) TEGER (8) :: N,NRHS,LD B, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)

\section*{C INTERFACE}
\#include <sunperfh>
void cspsv (charuplo, intn, int nrhs, com plex *ap, int *ịíjot, com plex *b, int ldb, int *info);
void cspsv_64 (charuplo, long n, long nihs, com plex *ap, long *ịíivot, com plex *b, long ldb, long *info);

\section*{PURPOSE}
cspsv com putes the solution to a com plex system of linear equations
\(A\) * \(X=B\), where \(A\) is an \(N\)-by-N symm etric \(m\) atrix stored in packed form at and \(X\) and \(B\) are \(N\)-by \(-N\) RH \(S m\) atrices.

The diagonal pivoting \(m\) ethod is used to factorA as
\(A=U * D * U * *\), if \(U P L O=U\) ', or
\(A=L * D * L * T\), if \(U P L O=L '\),
where \(U\) (orL) is a product of perm utation and unit upper (low er) triangularm atrices, \(D\) is sym \(m\) etric and block diagonalw th 1 -by -1 and 2 -by -2 diagonal blocks. The factored form ofA is then used to solve the system of equations A * \(X=B\).

\section*{ARGUMENTS}

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The num ber of linearequations, ie., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS \(>=0\).

AP (input/output)
C om plex array, dim ension \(\mathbb{N}^{*}(\mathbb{N}+1) / 2\) ) On entry, the upper or low er triangle of the sym \(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linear array. The jth
column of A is stored in the array AP as follow s : ifUPLO \(=U\) ',AP \((i+(j-1) * j 2)=A(i, j)\) for \(1<=\mathrm{i}<=\dot{j}\) if UPLO = L',AP (i+ (j-1)* (2n-j)/2)= A \((i, j)\) for \(j=i<=n\). See below for further details.

On exit, theblock diagonalm atrix \(D\) and the \(m\) ultipliers used to obtain the factor \(U\) orL from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) as com puted by CSPTRF, stored as a packed triangular \(m\) atrix in the sam e storage form atas \(A\).

\section*{PIVOT (output)}

Integer array, dim ension \(\mathbb{N}\) ) D etails of the interchanges and the block structure of \(D\), as determ ined by CSPTRF. If IPIV OT \((k)>0\), then row \(s\) and colum nsk and \(\mathbb{P} \mathbb{I V} O T(k)\) w ere interchanged, and \(D(k, k)\) is a 1 -by-1 diagonalblock. If P PLO \(=\mathrm{U}^{\prime}\) and \(\mathbb{P I V O T}(k)=\mathbb{P} \mathbb{I V O T}(k-1)<0\), then row sand colum nsk-1 and - \(\mathbb{P}\) IV OT (k) were interchanged and D ( \(k-1 * k, k-1 k)\) is a \(2-b y-2\) diagonalblock. If \(\mathrm{UPLO}=\mathrm{L}\) 'and \(\mathbb{P} \mathbb{I V O T}(k)=\mathbb{P} \mathbb{I V O T}(k+1)<0\), then row s and colum ns \(k+1\) and \(-\mathbb{P}\) IV OT ( \(k\) ) were interchanged and \(D(k: k+1, k: k+1)\) is a \(2-b y-2\) diagonal block.

\section*{B (input/output)}

Com plex array, dim ension (LD B , NRHS) On entry, the N -by-N RH S right hand sidem atrix B. On exit, if \(\mathbb{N} F O=0\), the \(N\) by \(-N\) RH \(S\) solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the anay B. LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO = -i, the i-th argum enthad an illegalvalue \(>0:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{D}(\mathrm{i}, \mathrm{i})\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, so the solution could notbe com puted.

\section*{FURTHER DETAILS}

The packed storage scheme is illustrated by the follow ing exam ple when \(N=4\), UPLO = U':

Tw o-dim ensional storage of the sym m etric m atrix A:
a11 a12 a13 a14
```

a22 a23 a24
a33 a34 (aij= aji)
a44

```

Packed storage of the upper triangle ofA :
\[
A P=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]
\]

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
cspsvx - use the diagonal pivoting factorization \(\mathrm{A}=\) \(\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) to com pute the solution to a com plex system of linearequations \(A * X=B\), where \(A\) is an \(N\) by -N symm etric \(m\) atrix stored in packed form at and \(X\) and \(B\) are N -by-N RH S m atrices

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CSPSVX FACT,UPLO,N,NRHS,A,AF,\mathbb{PIVOT,B,LDB,X,LDX,}}\mathbf{N},\textrm{N},\textrm{N}
RCOND,FERR,BERR,W ORK,W ORK2, INFO)
CHARACTER * 1FACT,UPLO
COM PLEX A (*),AF (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,INFO
INTEGER \mathbb{PIVOT (*)}
REALRCOND
REAL FERR (*),BERR (*),WORK2 (*)
SUBROUT\mathbb{NECSPSVX_64(EACT,UPLO,N,NRHS,A,AF, \mathbb{PIVOT,B,LDB,X,}}\mathbf{N},\textrm{N},\textrm{N}
LDX,RCOND,FERR,BERR,WORK,W ORK 2, INFO)
CHARACTER * 1 FACT,UPLO
COMPLEX A (*),AF (*),B (LDB,*),X (LDX,*),W ORK (*)
\mathbb{NTEGER*8N,NRHS,LDB,LDX,INFO}
INTEGER*8 \mathbb{PIVOT (*)}
REALRCOND
REALFERR (*),BERR (*),WORK2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE SPSVX (FACT,UPLO,N, NRHS],A,AF, \(\mathbb{P} I V O T, B,[L D B], X\), [LDX],RCOND,FERR,BERR, [W ORK], [W ORK2], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1) ::FACT,UPLO
COMPLEX,D \(\mathbb{I M} E N S I O N(:):: A, A F, W\) ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : : B, X
\(\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T\)
REAL ::RCOND
REAL,D IM ENSION (:) ::FERR,BERR,W ORK 2
SU BROUTINE SPSVX_64 (FACT,UPLO,N, NRHS],A,AF, \(\mathbb{P} \mathbb{N} O T, B,[L D B]\), \(\mathrm{X},[\mathrm{LD} \mathrm{X}], R C O N D, F E R R, B E R R,[\mathbb{W}\) ORK], \(\mathbb{W}\) ORK 2], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1)::FACT,UPLO
COMPLEX,D \(\mathbb{M}\) ENSION (:) ::A,AF,WORK
COMPLEX,D \(\mathbb{M}\) ENSION (: : : : : \(:\), X
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S \mathbb{I N}(:):: \mathbb{P} \mathbb{I V O T}\)
REAL ::RCOND
REAL,D IM ENSION (:) ::FERR,BERR,W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void cspsvx (char fact, charuplo, intn, int nrhs, com plex *a, com plex *af, int *ipivot, com plex *b, int ldb, com plex *x, int ldx, float *rcond, float *ferr, float*berr, int*info);
void cspsvx_64 (char fact, charuplo, long n, long nihs, com plex *a, com plex *af, long *ipivot, com plex *b, long lab, com plex *x, long ldx, float *roond, float * ferr, float *berr, long *info);

\section*{PURPOSE}
cspsvx uses the diagonalpivoting factorization \(A=U * D * U * * T\) or \(A=L * D * L * * T\) to com pute the solution to a com plex system of linear equations \(A\) * \(X=B\), where \(A\) is an \(N\)-by \(-N\) sym\(m\) etric \(m\) atrix stored in packed form at and \(X\) and \(B\) are \(N\)-byN RH S m atrices.

Errorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT \(=N\) ', the diagonalpivoting \(m\) ethod is used to factorA as
\[
A=U * D * U * * T, \text { if } U P L O=U ' \text {,or }
\]
\[
A=L * D * L \star * T \text {, if } U P L O=L '
\]
where \(U\) (orL) is a productofperm utation and unitupper (low er)
triangularm atrioes and D is sym m etric and block diagonal w ith

1-by-1 and 2-by-2 diagonalblocks.
2. If som eD \((i, i)=0\), so thatD is exactly singular, then the routine
retums w ith \(\mathbb{N N F O}=\) i. O therw ise, the factored form of \(A\) is used
to estim ate the condition num ber of the \(m\) atrix A. If the reciprocal of the condition num ber is less than \(m\) achine precision,
\(\mathbb{N} F O=N+1\) is retumed as a w aming, but the routine stillgoes on
to solve for \(X\) and com pute error bounds as described below.
3.The system ofequations is solved forX using the factored form of A.
4. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for it.

\section*{ARGUMENTS}

\section*{FACT (input)}

Specifies w hether ornot the factored form of \(A\) has been supplied on entry.\(=F^{\prime}\) : On entry, A F and \(\mathbb{P} \mathbb{I V O T}\) contain the factored form of A. A, AF and \(\mathbb{P} \mathbb{I V O T} w\) ill not be modified. \(=\mathrm{N}\) ': The m atrix A w illibe copied to AF and factored.

UPLO (input)
\(=\mathrm{U}\) : U ppertriangle of A is stored;
\(=\mathbb{L}\) ': Low ertriangle of \(A\) is stored.

N (input) The num ber of linearequations, i.e., the order of them atrix A. \(N>=0\).

NRHS (input)
The num ber of right hand sides, ie., the num ber of colum ns of the m atrioes B and X. NRH S >=0.

A (input) C om plex array, dim ension \(\mathbb{N} *(N+1) / 2\) ) The upper or low er triangle of the sym \(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linearamay. The jth colum \(n\) of A is stored in the array A as follow S : if f PLO \(=\) \(U^{\prime}, A(i+(j-1) * j 2)=A(i, 7)\) for \(1<=i<=j\) if UPLO = L', A (i+ (j-1)* (2*n-j) 2 ) =A ( \(i, j\) ) for \(j=i<=n\). See below for furtherdetails.

AF (input/output)
Complex array, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2\) ) If FACT = \(F\) ', then \(A F\) is an inputargum ent and on entry contains the block diagonalm atrix D and the multipliers used to obtain the factor \(U\) orL from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) as com puted by CSPTRF, stored as a packed triangular \(m\) atrix in the sam e storage form atasA.
If \(\mathrm{FACT}=\mathrm{N}\) ', then AF is an output argum ent and on exit contains the block diagonalm atrix D and the m ultipliers used to obtain the factorU or L from the factorization \(A=U * D * U * * T\) or \(A=\) L *D *L**T as com puted by CSPTRF, stored as a packed triangularm atrix in the sam e storage form at as A.

PIVOT (inputoroutput)
Integer array, dim ension (N) IfFACT = F', then \(\mathbb{P} \mathbb{V} O T\) is an inputargum entand on entry contains details of the interchanges and the block structure ofD , as determ ined by CSPTRF. If \(\mathbb{P}\) IV OT (k) \(>0\), then row sand colum nsk and \(\mathbb{P} \mathbb{V} O T(k)\) were interchanged and \(\mathrm{D}(\mathrm{k}, \mathrm{k})\) is a 1 -by -1 diagonal block. IfUPLO = U'and \(\mathbb{P I V O T}(k)=\mathbb{P} \mathbb{V} O T(k-1)\) \(<0\), then row s and colum nsk-1 and - \(\mathbb{P}\) IV OT (k) were interchanged and \(D(k-1 k, k-1 k)\) is a 2 -by-2 diagonal block. If UPLO = L ' and \(\mathbb{P} \mathbb{I V O T}(\mathrm{k})=\) \(\mathbb{P} \mathbb{V} O T(k+1)<0\), then row \(s\) and columns \(k+1\) and \(-\mathbb{P}\) IV O \(T(k)\) w ere interchanged and \(D(k \cdot k+1, k \cdot k+1)\) is a 2 -by-2 diagonalblock.

IfFACT = \(N\) ', then \(\mathbb{P I V O T}\) is an output argum ent and on exitcontains details of the interchanges and the block structure of D, as determ ined by CSPTRF.

B (input) C om plex anay, dim ension (LD B ,NRHS) The N by-NRH S righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the array \(B\). LD B >= max (1,N).

\section*{X (output)}

If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=\mathrm{N}+1\), the N -by-NRH S solution \(m\) atrix \(X\).

LD X (input)
Com plex array, dim ension (LDX,NRHS) The leading dim ension of the aray \(X . \operatorname{LD} X=\max (1, N)\).

\section*{RCOND (output)}

The estim ate of the reciprocal condition num ber of the \(m\) atrix \(A\). IfRCOND is less than them achine precision (in particular, if RCOND \(=0\) ), the \(m\) atrix is singular to working precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO > 0.

\section*{FERR (output)}

C om plex array, dim ension (NRHS) The estim ated forw ard emror bound for each solution vectorX (i) (the \(j\) th colum \(n\) of the solution \(m\) atrix \(X\) ). If XTRUE is the true solution corresponding to \(\mathrm{X}(\mathcal{\nu})\), FERR ( \()\) ) is an estim ated upperbound for the \(m\) agnitude of the largest elem ent in ( \(\mathrm{X}(\mathcal{J})\)-XTRUE) divided by the \(m\) agnitude of the largestelem ent in
X ( \(\mathcal{j}\) ) The estim ate is as reliable as the estim ate forR C OND, and is alm ostalw ays a slightoveresti\(m\) ate of the true error.

\section*{BERR (output)}

Com plex array, dim ension \((\mathbb{N} H\) S) The com ponentw ise relative backw ard error ofeach solution vector \(\mathrm{X}(\mathrm{j})\) (ie., the sm allestrelative change in any elem ent of A orB thatm akes X ( \(\mathcal{I}\) ) an exact solution).

W ORK (w orkspace)
C om plex array, dim ension ( \(2 \star \mathrm{~N}\) )

W ORK 2 (w orkspace)
Integer anray, dim ension (N)
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue
> 0 : if \(\mathbb{N F O}=i\), and \(i\) is
<= N : D (i,i) is exactly zero. The factorization has been completed but the factorD is exactly
singular, so the solution and error bounds could not be com puted. RCOND \(=0\) is retumed. \(=N+1: D\)
is nonsingular, butRCOND is less than m achine precision, \(m\) eaning that the \(m\) atrix is singular to w orking precision. N evertheless, the solution and error bounds are computed because there are a num ber of situations w here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{FURTHER DETAILS}

The packed storage schem e is illustrated by the follow ing exam ple when \(N=4, \mathrm{UPLO}=\mathrm{U}\) ':

Tw o-dim ensionalstorage of the sym m etric m atrix A :
```

a11 a12 a13 a14
a22 a23 a24
a33 a34 (aij= aji)
a44

```

Packed storage of the upper triangle ofA :
\(A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]\)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
csptrf-com pute the factorization of a com plex sym m etric \(m\) atrix A stored in packed form atusing the B unch-K aufm an diagonalpivoting \(m\) ethod

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
COM PLEX AP (*)
\mathbb{NTEGER N, INFO}
INTEGER \mathbb{PIVOT (*)}
SUBROUT\mathbb{NE CSPTRF_64(UPLO,N,AP, PPIVOT,INFO)}
CHARACTER * 1 UPLO
COM PLEX AP (*)
\mathbb{NTEGER*8 N,\mathbb{NFO}}\mathbf{~}+
\mathbb{NTEGER *8 \mathbb{PIVOT (*)}}\mathbf{(})

```

\section*{F95 INTERFACE}
```

SU BROUTINE SPTRF (UPLO ,N,AP, $\mathbb{P} \mathbb{I V O T},[\mathbb{N} F O]$ )
CHARACTER (LEN=1)::UPLO
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::AP
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T$
SU BROUTINE SPTRF_64 (UPLO, N,AP, $\mathbb{P} \mathbb{I V} O T,[\mathbb{N} F O$ ])
CHARACTER (LEN=1) ::UPLO

```

\section*{C INTERFACE}
\#include <sunperfh>
void csptrf(char uple, intn, com plex *ap, int *ipivot, int *info);
void csptrf_64 (charuplo, long n, com plex *ap, long *ipivot, long *info);

\section*{PURPOSE}
csptrf com putes the factorization of a com plex sym \(m\) etric \(m\) atrix A stored in packed form atusing the B unch-K aufm an diagonalpivoting \(m\) ethod:
\[
A=U * D * U * * T \text { or } A=L * D * L * * T
\]
where \(U\) (orL) is a productof perm utation and unit upper (low er) triangular \(m\) atrioes, and \(D\) is sym \(m\) etric and block diagonalw ith 1 -by-1 and 2-by-2 diagonalblocks.

\section*{ARGUMENTS}

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low er triangle ofA is stored.
N (input) The order of the m atrix A. \(\mathrm{N}>=0\).

AP (input/output)
C om plex array, dim ension \(\mathbb{N}^{*}(\mathbb{N}+1) / 2\) ) O n entry, the upper or low er triangle of the sym \(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linear array. The jth colum \(n\) ofA is stored in the array AP as follow \(s\) : ifUPLO \(=U{ }^{\prime}, \mathrm{AP}(i+(j-1) * j 2)=A(i, j)\) for \(1<=\mathrm{i}<=\dot{j}\) if UPLO \(=\mathrm{L}\), 'AP \((i+(j-1) *(2 n-j) / 2)=\) A \((i, 1)\) for \(\dot{j}=i<=n\).

On exit, the block diagonalm atrix \(D\) and the \(m u l\) tipliers used to obtain the factorU orL, stored as a packed triangularm atrix overw riting A (see below for further details).

\section*{IPIVOT (output)}

Integer amay, dim ension \(\mathbb{N}\) ) D etails of the interchanges and the block structure of D. If \(\mathbb{P} \mathbb{I V O T}(\mathrm{k})>0\), then rows and colum ns k and IP IV O T \((k)\) w ere interchanged and \(D(k, k)\) is a 1 -by-1 diagonalblock. IfUPLO \(=\mathrm{U}^{\prime}\) and \(\mathbb{P} \mathbb{I V O T}(k)=\) \(\mathbb{P} \mathbb{I V O T}(k-1)<0\), then row s and colum nsk-1 and - \(\mathbb{P}\) IV O T \((k)\) w ere interchanged and \(D(k-1 * k, k-1 k)\) is a 2 -by-2 diagonal block. If UPLO = L'and \(\mathbb{P} \mathbb{I V} \circ T(k)=\mathbb{P} \mathbb{I} \circ T(k+1)<0\), then row s and colum \(n s\) \(\mathrm{k}+1\) and \(-\mathbb{P} \mathbb{I V O T}(\mathrm{k})\) were interchanged and D \((k: k+1, k: k+1)\) is a 2 -by -2 diagonalblock.

\section*{\(\mathbb{N} F O\) (output)}
= 0: successfiulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the \(i\)-th argum ent had an illegal value
\(>0:\) if \(\mathbb{N F O}=i, D(i, i)\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix D is exactly singular, and division by zero w illoccur if it is used to solve a system of equations.

\section*{FURTHER DETAILS}

5-96 - B ased on m odifications by J. Lew is, B oeing C om puter Services

C om pany

If \(\mathrm{U} P \mathrm{LO}=\mathrm{U}\) ', then \(A=U * D * U\) ', where
\(U=P(n) \star U(n)^{\star} \ldots{ }^{*} P(k) U(k)^{\star} \ldots\),
i.e., \(U\) is a productof term \(s P(k) * U(k)\), where \(k\) decreases from \(n\) to 1 in steps of 1 or 2 , and \(D\) is a block diagonal \(m\) atrix \(w\) ith 1 -by-1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I V} O T(k)\), and \(U(k)\) is a unituppertriangularm atrix, such that if the diagonal
block \(D(k)\) is of orders \((s=1\) or 2 ), then
```

    ( I v 0 ) k-s
    U (k)=(0 I 0 ) s
( 0 0 I ) n-k
k-s s n-k

```

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(1, k-\) \(1, k)\). If \(s=2\), the uppertriangle of \((k)\) overw rites \(A(k-\) \(1, k-1), A(k-1, k)\), and \(A(k, k)\), and \(v\) overw rites \(A(1 k-2, k-\) 1 k).
```

IfU PLO = L', then A = L *D *L',w here
L = P (l)*L (l)* ... *P (k)*L (k)* ...,

```
i.e., \(L\) is a productofterm \(S P(k) * L(k)\), where \(k\) increases from 1 to \(n\) in steps of 1 or2, and \(D\) is ablock diagonal \(m\) atrix \(w\) th 1 -by -1 and 2 -by- 2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I V O T}(k)\), and \(L(k)\) is a unit low ertriangularm atrix, such that if the diagonal block \(D(k)\) is of orders ( \(s=1\) or 2 ), then
```

    ( I 0 0 ) k-1
    L (k)=( 0 I 0 ) s
    ( 0 v I ) n-k-s+1
        k-1 s n-k-s+1
    ```

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites A \((k+1 n, k)\). If \(s=2\), the low er triangle ofD ( \(k\) ) overw rites A \((k, k), A(k+1, k)\), and \(A(k+1, k+1)\), and \(v\) overw rites \(A(k+2 n, k k+1)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
csptri-com pute the inverse of a com plex sym \(m\) etric indefinte \(m\) atrix \(A\) in packed storage using the factorization \(A=\) \(\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by CSPTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CSPTRI(UPLO,N,AP,\mathbb{PIVOT,WORK,INFO)}}\mathbf{N}\mathrm{ (N O}
CHARACTER * 1 UPLO
COM PLEX AP (*),W ORK (*)
INTEGER N,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}

```

```

CHARACTER * 1 UPLO
COM PLEX AP (*),W ORK (*)
INTEGER*8N,\mathbb{NFO}
INTEGER *8 \mathbb{PIVOT (*)}
F95 INTERFACE

```

```

    CHARACTER (LEN=1) ::UPLO
    COMPLEX,DIM ENSION (:) ::AP,W ORK
    \mathbb{NTEGER ::N,\mathbb{NFO}}\mathbf{N}=0
    \mathbb{NTEGER,D IM ENSION (:) ::\mathbb{PIVOT}}\mathbf{T}\mathrm{ (:}
    ```

```

    CHARACTER (LEN=1)::UPLO
    COMPLEX,D IM ENSION (:) ::AP,W ORK
    ```
\(\mathbb{N}\) TEGER (8) ::N, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8),D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V}\) OT

\section*{C INTERFACE}
\#include < sunperfh>
void csptri(charuplo, intn, com plex *ap, int *ipivot, int *info);
void csptri_ 64 (charuplo, long n, com plex *ap, long *ípívot, long *info);

\section*{PURPOSE}
csptri computes the inverse of a complex symm etric indefinite \(m\) atrix \(A\) in packed storage using the factorization \(A=U * D * U * * T\) orA \(=L * D * L * * T\) com puted by CSPTRF .

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) ': U pper triangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\);
= L': Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\).

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).

\section*{AP (input/output)}

C om plex array, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2\) ) O n entry, the block diagonal \(m\) atrix \(D\) and the \(m\) ultipliers used to obtain the factorU orL as com puted by CSPTRF, stored as a packed triangularm atrix.

On exit, if \(\mathbb{N F F O}=0\), the (sym metric) inverse of the originalm atrix, stored as a packed triangular \(m\) atrix. The \(j\) th collm \(n\) of \(\operatorname{inv}(A)\) is stored in the array AP as follows: ifUPLO = U',AP (i+ (j 1) \(* j 2)=\operatorname{inv}(A)(i, \gamma)\) for \(1<=i<=j ; i f U P L O=L^{\prime}\), AP \((i+(j-1) \star(2 n-j / 2)=\operatorname{inv}(A)(i, j)\) for \(j=i<=n\).
\(\mathbb{P I V O T}\) (input)
Integer array, dim ension (N) D etails of the interchanges and the block structure ofD as determ ined by CSPTRF.

W ORK (w orkspace)

C om plex anay, dim ension \(\mathbb{N}\) )
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=i, D(i, i)=0\); the \(m\) atrix is singular and its inverse could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
csptrs-solve a system of linearequations \(A * X=B\) with a com plex sym \(m\) etric \(m\) atrix A stored in packed form atusing the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by CSPTRF

\section*{SYNOPSIS}


CHARACTER * 1 UPLO
COM PLEX AP (*), B (LDB,*)
\(\mathbb{N}\) TEGERN,NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{P} \mathbb{I V O T}\) ( \(\left.{ }^{( }\right)\)
SUBROUTINE CSPTRS_64(UPLO,N,NRHS,AP, \(\mathbb{P} \mathbb{I V O T}, \mathrm{B}, \mathrm{LD} \mathrm{B}, \mathbb{N} F \mathrm{~F}\) )

CHARACTER * 1 UPLO
COM PLEX AP (*), B (LDB,*)
\(\mathbb{N} T E G E R * 8 N, N R H S, L D B, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER*8 \(\mathbb{P} \mathbb{I V O T}{ }^{( }\))

\section*{F95 INTERFACE}

CHARACTER (LEN=1) ::UPLO
COMPLEX,D \(\mathbb{I}\) ENSION (:) ::AP
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::B
\(\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT}\)
SU BROUTINE SPTRS_64 (UPLO ,N, NRHS],AP, \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO
COM PLEX,D \(\mathbb{I}\) ENSION (:) ::AP
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::B
\(\mathbb{N} T E G E R(8):: N, N R H S, L D B, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V} O T\)

\section*{C INTERFACE}
\#include <sunperfh>
void csptrs (charuplo, intn, int nrhs, com plex *ap, int *ịívot, com plex *b, int ldb, int *info);
void csptrs_64 (char uplo, long n, long nrhs, com plex *ap, long *ị̀ívot, com plex *b, long ldb, long *info);

\section*{PURPOSE}
csptrs solves a system of linearequations A *X = B w th a com plex sym \(m\) etric \(m\) atrix A stored in packed form at using the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by CSPTRF.

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) ': U ppertriangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\);
\(=\mathrm{L}\) ': Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\).

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS \(>=0\).

AP (input)
Com plex array, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2\) ) The block diagonal \(m\) atrix \(D\) and the \(m\) ultipliers used to obtain the factorU orL as com puted by CSPTRF, stored as a packed triangularm atrix.

PIVOT (input)
Integer array, dim ension \(\mathbb{N}\) ) D etails of the interchanges and the block structure ofD as determ ined by CSPTRF.

B (input/output)
Com plex array, dim ension (LDB,NRHS) On entry, the
right hand sidem atrix \(B\). On exit, the solution
\(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array \(B . \operatorname{LD} B>=\) \(\max (1, N)\).
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
csrot-A pply a plane rotation.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CSROT N,X,\mathbb{NCX,Y, NNCY,C,S)}}\mathbf{N},\textrm{N}
REALC,S
COM PLEX X (*),Y (*)
INTEGERN,\mathbb{NCX,\mathbb{NCY}}\mathbf{N}=\mp@code{N}

```

```

REALC,S
COM PLEX X (*),Y (*)
INTEGER*8N,\mathbb{NCX,INCY}
F95 INTERFACE

```

```

    REAL ::C,S
    COM PLEX,D IM ENSION (:) ::X,Y
    \mathbb{NTEGER ::N,INCX,INCY}
    ```

```

    REAL::C,S
    COM PLEX,D IM ENSION (:) ::X,Y
    \mathbb{NTEGER (8)::N,\mathbb{NCX,INCY}}\mathbf{N}={
    C INTERFACE
\#include <sunperfh>

```
void csrot(intn, com plex *x, int incx, com plex *y, int incy, floatc, floats);
void csrot_64 (long n, com plex *x, long incx, com plex *y, long incy, float c , float.s);

\section*{PURPOSE}
csrotA pply a plane rotation, w here the cos and sin (c and s) are realand the vectors \(x\) and \(y\) are com plex.

\section*{ARGUMENTS}

N (input)
On entry, N specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.
\(X\) (input/output)
Before entry, the increm ented array X m ust contain the vectorx. Unchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X \mathrm{~m}\) ustnotbe zero. U nchanged on exit.

Y (input/output)
On entry, the increm ented array Y m ust contain the vector \(y\). On exit, \(Y\) is overw rilten by the updated vectory.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of Y. \(\mathbb{N C} C\) m ustnotbe zero. U nchanged on exit.

C (input)
O n entry, the cosine. U nchanged on exit.
\(S\) (input)
On entry, the sin. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

csscal-C om pute y := alpha * y

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CSSCAL N,ALPHA,Y,INCY)}
COM PLEX Y (*)

```

```

REALALPHA
SUBROUT\mathbb{NECSSCAL_64N,ALPHA,Y,INCY)}
COM PLEX Y (*)
\mathbb{NTEGER*8 N,\mathbb{NCY}}\mathbf{}=1
REALALPHA
F95 INTERFACE

```

```

COM PLEX,D IM ENSION (:) ::Y
\mathbb{NTEGER ::N,\mathbb{NCY}}\mathbf{N}=\mp@code{N}
REAL ::ALPHA
SU BROUTINE SCAL_64 (N ],ALPHA,Y,[\mathbb{N CY ])}
COMPLEX,D IM ENSION (:) ::Y
\mathbb{NTEGER (8)::N,\mathbb{NCY}}\mathbf{}\mathrm{ (%)}
REAL ::ALPHA
C INTERFACE
\#include <sunperfh>

```
void csscal(intn, float alpha, com plex *y, int incy);
void csscal_ 64 (long n, floatalpha, com plex *y, long incy);

\section*{PURPOSE}
csscalC om pute y := alpha * y w here alpha is a scalar and y is an \(n\)-vector.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N must be at least one for the subroutine to have any visible effect. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

Y (input/output)
( \(1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C Y)\) ). On entry, the increm ented amay \(Y\) m ust contain the vectory. On exit, \(Y\) is overw rilten by the updated vectory.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of Y. \(\mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
cstedc - com pute alleigenvalues and, optionally, eigenvectors of a sym \(m\) etric tridiagonalm atrix using the divide and conquerm ethod

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CSTEDC (COM PZ,N,D,E,Z,LD Z,W ORK,LW ORK,RW ORK,LRW ORK,}
IN ORK,LIN ORK,INFO)
CHARACTER * 1 COMPZ
COM PLEX Z (LDZ,*),W ORK (*)
\mathbb{NTEGER N,LDZ,LW ORK,LRW ORK,LIN ORK,INFO}
INTEGER IN ORK (*)
REALD (*),E (*),RW ORK (*)
SU BROUTINE CSTED C_64 (COM PZ,N,D,E,Z,LD Z,W ORK,LW ORK,RW ORK,

```

```

CHARACTER * 1 COMPZ
COM PLEX Z (LDZ,*),W ORK (*)
\mathbb{NTEGER*8N,LDZ,LW ORK,LRW ORK,LIN ORK,INFO}
INTEGER*8 IN ORK (*)
REALD (*),E (*),RW ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE STEDC COM PZ, \(\mathbb{N}], D, E, Z,[L D Z],[W O R K],[L W O R K],[R W O R K]\), [LRW ORK], [IW ORK], [LIN ORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::COMPZ
COMPLEX,D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::Z
\(\mathbb{N} T E G E R:: N, L D Z, L W\) ORK,LRW ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
REAL,D IM ENSION (:) ::D ,E,RW ORK
SU BROUTINE STEDC_64 (COM PZ, \(\mathbb{N}], D, E, Z,[L D Z],\left[\begin{array}{l}\text { W ORK ], [LW ORK ], }\end{array}\right.\) \([\mathbb{R W}\) ORK ], [LRW ORK], [ \(\mathbb{W}\) ORK ], [LIN ORK], [ \(\mathbb{N} F O])\)

CHARACTER (LEN=1)::COM PZ
COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:)::Z
\(\mathbb{N} T E G E R(8):: N, L D Z, L W\) ORK,LRW ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL,D \(\mathbb{I M}\) ENSION (:) ::D,E,RW ORK

\section*{C INTERFACE}
\#include < sunperfh>
void cstedc (char com pz, intn, float *d, float *e, com plex
* \(z\), intld \(z\), int *info);
void cstedc_64 (char com pz, long n, float*d, float *e, com plex *z, long ldz, long *info);

\section*{PURPOSE}
cstedc com putes alleigenvalues and, optionally, eigenvectors of a sym \(m\) etric tridiagonalm atrix using the divide and conquerm ethod. The eigenvectors of a fullorband com plex Herm itian matrix can also be found ifCHETRD orCHPTRD or CHBTRD has been used to reduce this \(m\) atrix to tridiagonal form.

This code \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. It w illw ork on m achines w ith a guard digitin add/subtract, or on those binary machines w thout guard digits which subtract like the C ray X \(-\mathrm{M} P, C\) ray \(Y \mathrm{M} P\), C ray \(\mathrm{C}-90\), or C ray-2. It could conœivably fail on hexadecim al or decin al machines w thout guard digits, butw e know of none. Se SLA ED 3 for details.

\section*{ARGUMENTS}

COMPZ (input)
\(=N^{\prime}:\) C om pute eigenvalues only.
= 'I': C om pute eigenvectors of tridiagonalm atrix
also.
\(=\mathrm{V}\) : C om pute eigenvectors of original H erm itian
\(m\) atrix also. On entry, \(Z\) contains the unitary
\(m\) atrix used to reduce the originalm atrix to tridiagonal form .

N (input) The dim ension of the sym \(m\) etric tridiagonalm atrix. \(\mathrm{N}>=0\).

D (input/output)
O n entry, the diagonalelem ents of the tridiagonal m atrix. On exit, if \(\mathbb{N} F O=0\), the eigenvalues in ascending order.

E (input/output)
O n entry, the subdiagonalelem ents of the tridiagonalm atrix. On exit, E has been destroyed.

Z (input) \(O n\) entry, if COMPZ \(=V\) ', then \(Z\) contains the unitary m atrix used in the reduction to tridiagonalform. On exit, if \(\mathbb{N F O}=0\), then if \(C O M P Z=\) V', Z contains the orthonorm aleigenvectors of the original H erm itian m atrix, and if \(\mathrm{COMPZ}=\mathrm{I}\) ', Z contains the orthonorm al eigenvectors of the sym m etric tridiagonalm atrix. If COMPZ \(=\mathrm{N}^{\prime}\) ', then Z is not referenced.

LD Z (input)
The leading din ension of the array Z . LD \(\mathrm{Z}>=1\). If eigenvectors are desired, then LD Z \(>=\mathrm{max}(1, \mathrm{~N})\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, \mathrm{~W}\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array W ORK. IfCOM PZ \(=\mathrm{N}^{\prime}\) or 'I', orN <=1,LW ORK mustbe at least1. If
COM PZ \(=V\) 'and \(N>1\), LW ORK m ust.be at least \(N * N\).

IfLW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
theW ORK anray, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LW ORK is issued by X ERBLA.

RW ORK (w orkspace)
dim ension (LRW ORK) On exit, if \(\mathbb{N} F O=0\), RW ORK (1) retums the optim allRW ORK .

LRW ORK (input)
The dim ension of the array RW ORK. IfCOMPZ \(=N^{\prime}\) or \(\mathrm{N}<=1\), LRW ORK mustbe at least1. IfCOMPZ \(=\)

V'and \(\mathrm{N}>1\),LRW ORK m ustbe atleast1 \(+3 * \mathrm{~N}+\) \(2 * N * \lg +3 * N * * 2\), where \(\lg (N)=s m\) allest integerk such that \(2 \star * \mathrm{k}>=\mathrm{N}\). IfCOM PZ \(=\) I'and \(\mathrm{N}>1\),LRW ORK mustbe at least \(1+4 * N+2 * N * * 2\).

If LRW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the RW ORK array, retums this value as the first entry of the RW ORK amay, and no emorm essage related to LRW ORK is issued by XERBLA.

IV ORK (w orkspace/output)
On exit, if \(\mathbb{N} F O=0, \mathbb{I W}\) ORK (1) retums the optim al LIW ORK.

LIW ORK (input)
The dim ension of the anay \(\mathbb{I W}\) ORK. IfCOMPZ \(=\mathrm{N}^{\prime}\) or \(\mathrm{N}<=1\), LIW ORK m ustbe at least1. IfCOMPZ \(=\) V 'orN > 1, LIW ORK mustbe at least \(6+6 * N+\) \(5 * N * \operatorname{N}\). IfCOMPZ = I'orN \(>1\), LIV ORK must be at least \(3+5 * N\).

IfLIW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the \(\mathbb{I W}\) ORK array, retums this value as the first entry of the \(\mathbb{I W}\) ORK aray, and no errorm essage related to \(L \mathbb{I W} O R K\) is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit.
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue.
\(>0\) : The algorithm failed to com pute an eigenvalue while w orking on the subm atrix lying in row S and colum ns \(\mathbb{N}\) FO \(/ \mathbb{N}+1\) ) through \(m o d(\mathbb{N}\) FO, \(\mathbb{N}+1)\).

\section*{FURTHER DETAILS}

B ased on contributions by JeffR utter, C om puter Science D ivision, U niversity of C alifomia atB erkeley, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
cstegr-Com pute \(T\)-sigm a_i= L_iD_iL_i^T, such that L_i
D_iL_i^T is a relatively robustrepresentation

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CSTEGR (JOBZ,RANGE,N,D,E,VL,VU,\mathbb{I},\mathbb{U},ABSTOL,M,W,}
Z,LDZ,ISUPPZ,W ORK,LW ORK,IN ORK,LIW ORK,\mathbb{NFO)}

```
```

CHARACTER * 1 JOBZ,RANGE
COMPLEX Z (LDZ,*)
\mathbb{NTEGERN,\mathbb{N,}\mathbb{U},M,LDZ,LW ORK,LIN ORK,\mathbb{NFO}}\mathbf{M}\mathrm{ \}
\mathbb{NTEGER ISUPPZ (*), IN ORK (*)}
REALVL,VU,ABSTOL
REALD (*),E (*),W (*),W ORK (*)
SU BROUT\mathbb{NE CSTEGR_64 (OBZ,RANGE,N,D,E,VL,VU,IL,\mathbb{U},ABSTOL,M,}
W,Z,LDZ,ISUPPZ,W ORK,LW ORK,IN ORK,LIN ORK,INFO)

```
CHARACTER * 1 JOBZ,RANGE
COM PLEX Z (LD Z,*)
\(\mathbb{N}\) TEGER*8 \(\mathrm{N}, \mathbb{I}, \mathbb{I}\), M ,LD Z,LW ORK, LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER *8 ISUPPZ (*), \(\mathbb{I N}\) ORK (*)
REALVL,VU,ABSTOL
REALD (*), E (*), W (*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE STEGR (JOBZ,RANGE, \(\mathbb{N}], D, E, V L, V U, \mathbb{L}, \mathbb{U}, A B S T O L, M\), W , Z, [LD Z ], ISUPPZ, [W ORK ], [LW ORK ], [IW ORK ], [LIN ORK], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1):: OBZ,RANGE
COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::Z
\(\mathbb{N} T E G E R:: N, \mathbb{H}, \mathbb{U}, M, L D Z, L W O R K, L \mathbb{N} O R K, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{I M}\) ENSION (:) :: ISUPPZ, \(\mathbb{I}\) ORK
REAL ::VL,VU,ABSTOL
REAL,D \(\mathbb{M}\) ENSION (:) ::D ,E,W ,W ORK

SU BROUTINE STEGR_64 (OBZ,RANGE, \(\mathbb{N}], D, E, V L, V U, \mathbb{L}, \mathbb{U}, A B S T O L\), \(\mathrm{M}, \mathrm{W}, \mathrm{Z},[\mathrm{LD} Z], \operatorname{ISUPPZ},[\mathrm{W} O R K],[L W O R K],[\mathbb{W} O R K],[L \mathbb{W} O R K],[\mathbb{N} F O])\)

CHARACTER (LEN=1): : OBZ,RANGE
COM PLEX, D \(\mathbb{M}\) ENSION (:,:) :: Z
\(\mathbb{N}\) TEGER (8) :: N , \(\mathbb{I}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W\) ORK, LIW ORK, \(\mathbb{N} F O\)
\(\mathbb{I N}\) TEGER (8), D \(\mathbb{I M}\) ENSION (:) :: ISU PPZ , \(\mathbb{I N}\) ORK
REAL ::VL,VU, ABSTOL
REAL,D \(\mathbb{M}\) ENSION (:) ::D ,E,W ,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void cstegr(char jobz, char range, intn, float *d, float
*e, float vl, float vu, intil, intiu, float abstol, int \({ }_{\mathrm{m}} \mathrm{m}\), float \({ }^{*} \mathrm{~W}\), com plex \({ }^{*} \mathrm{z}\), int \(l \mathrm{ld} \mathrm{z}\), int *isuppz, int *info);
void cstegr_64 (char jobz, char range, long n, float *d, float *e, float vl, floatvu, long il, long iu, float abstol, long *m , float *w, com plex *z, long ldz, long *isuppz, long *info);

\section*{PURPOSE}
cstegrb) C om pute the eigenvalues, lam bda_j of L_i D_i L_i^T to high relative accuracy by the dqds algorithm,
(c) If there is a cluster of close eigenvalues, "choose"
sigm a_i
close to the cluster, and go to step (a),
(d) G iven the approxim ate eigenvalue lam boda_jofL_i D _i L_i^T,
com pute the corresponding eigenvectorby form ing a rank-revealing tw isted factorization.
The desired accuracy of the output can be specified by the inputparam eterA BSTOL.

Form ore details, see "A new O ( \(n^{\wedge} 2\) ) algorithm for the sym \(m\) etric tridiagonal eigenvahue/eigenvector problem ", by Inder吕D hillon, C om puterScience D ìvision TechnicalR eport N o. U CB C SD -97-971, U C Berkeley, M ay 1997.

N ote 1 : Cumently CSTEGR is only setup to find ALL the \(n\) eigenvalues and eigenvectors of \(T\) in \(O\left(n^{\wedge} 2\right)\) tim e

N ote 2 : Currently the routine CSTE \(\mathbb{N}\) is called when an appropriate sigm a_i cannot be chosen in step (c) above. CSTE IN invokesm odified G ram -Schm idt when eigenvalues are close.
N ote 3 : C STEGR w orks only on \(m\) achines \(w\) hich follow ieee-754 floating-point standard in their handling of infinities and N aN s. N orm alexecution of CSTEGR m ay create N aN s and infinities and hence \(m\) ay abortdue to a floating pointexception in environm ents w hich do notconform to the ieee standard.

\section*{ARGUMENTS}

JO B Z (input)
\(=\mathrm{N}:\) : Com pute eigenvalues only;
\(=\mathrm{V}^{\prime}:\) C om pute eigenvalues and eigenvectors.
RANGE (input)
= A ': alleigenvalues w illbe found.
\(=\mathrm{V}\) : alleigenvalues in the half-open interval (VL, VU ] w ill be found. = I': the II th through \(\mathbb{I U}\) th eigenvalues w illbe found.

N (input) The order of the m atrix. \(\mathrm{N}>=0\).

D (input/output)
O n entry, the \(n\) diagonalelem ents of the tridiagonalm atrix T.On exit, D is overw rilten.

E (input/output)
O \(n\) entry, the \((n-1)\) subdiagonal elem ents of the tridiagonal \(m\) atrix \(T\) in elem ents 1 to \(N-1\) of \(E\);
\(E(\mathbb{N})\) need notbe set. On exit, \(E\) is overw rilten.

VL (input)
IfRANGE=V', the low er and upper bounds of the interval to be searched foreigenvalues. VL < V U . N ot referenced ifRANGE = A 'or I'.

VU (input)
IfRANGE=V', the low er and upper bounds of the interval to be searched foreigenvalues. \(\mathrm{VL}<\mathrm{VU}\). N ot referenced ifRANGE=A 'or 'I'.

II (input)
IfRA N G E = ' 1 ', the indiges (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{L}<=\mathbb{U}<=N\), if \(N>0\); \(\mathbb{L}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N otreferenced ifRANGE \(=\) A'or V'.

IU (input)
IfRA N G E = I', the indices (in ascending order) of the sm allest and largest eigenvalues to be retumed. \(1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}\), if \(\mathrm{N}>0\); \(\mathbb{L}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N ot referenced ifRANGE= A 'or V'.

ABSTOL (input)
The absolute enror tolerance for the eigenvalues/eigenvectors. \(\mathbb{F} \mathrm{JOBZ}=\mathrm{V}\) ', the eigenvalues and eigenvectors outputhave residual norm s bounded by ABSTOL, and the dotproducts betw een different eigenvectors are bounded by ABSTOL. If ABSTOL is less than N *EPS* \(|\mathrm{F}|\), then \(\mathrm{N} * E P S *|T| w\) illbe used in its place, w here EPS is the \(m\) achine precision and \(F \mid\) is the 1 -norm of the tridiagonalm atrix. The eigenvalues are com puted to an accuracy ofEPS*| \(\mid\) imespective of A BSTOL. If high relative accuracy is im portant, setA BSTO L to DLAM CH (Safem inim um '). See Barlow and Dem m el "C om puting A ccurate Eigensystem s of Scaled D iagonally D om inantM atrices", LA PA CK W orking \(N\) ote \#7 for a discussion of \(w\) hich \(m\) atrioes define their eigenvalues to high relative accuracy.

M (output)
The total num ber ofeigenvalues found. \(0<=\mathrm{M}\) <= N. IfRANGE = A', M = N , and ifRANGE = \(\mathrm{I}^{\prime}, \mathrm{M}=\) \(\mathbb{U}-\mathbb{L}+1\).

W (output)
The firstM elem ents contain the selected eigenvalues in ascending order.

Z (input/output)
If \(J O B Z=V\) ', then if \(\mathbb{N F O}=0\), the first \(M\) colum ns of \(Z\) contain the orthonorm aleigenvectors of them atrix T comesponding to the selected eigenvalues, \(w\) ith the i-th colum \(n\) of \(Z\) holding the eigenvector associated w ith W (i). If \(30 \mathrm{BZ}=\mathrm{N}\) ', then \(Z\) is not referenced. N ote: the userm ust ensure that at leastm ax \((1, M)\) colum ns are supplied in the array \(Z\); ifRANGE = V', the exact value of M is notknow n in advance and an upperbound m ust be used.

LD Z (input)
The leading dim ension of the array \(\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD \(Z>=\mathrm{max}(1, N)\).

ISU PPZ (output)
The support of the eigenvectors in \(Z\), ie., the indices indicating the nonzero elem ents in \(Z\). The \(i\)-th eigenvector is nonzero only in elem ents ISU PPZ (2*i-1 ) through ISU PPZ (2*i).
W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al (andm inim al) LW ORK .

\section*{LW ORK (input)}

The dimension of the array W ORK. LW ORK >= max (1,18*N)

IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IN ORK (w orkspace/output)
On exit, if \(\mathbb{N} F O=0, \mathbb{I N}\) ORK (1) retums the optim al LIN ORK.

LIN ORK (input)
The dim ension of the array \(\mathbb{I N}\) ORK. LIW ORK >= max ( \(1,10 * \mathrm{~N}\) )

If LIV ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the IV ORK array, retums this value as the first entry of the \(\mathbb{I W}\) ORK array, and no errorm essage related to \(L \mathbb{I N} O R K\) is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{I N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
> 0 : if \(\mathbb{N F O}=1\), intemalerror in SLARRE, if \(\mathbb{N} F O=2\), intemalemor in CLARRV .

\section*{FURTHER DETAILS}

B ased on contributions by
Inder"̈تtD hillon, \(\mathbb{B M}\) A \(1 m\) aden, U SA
O sniM arques, LBN L NE ERSC , U SA
K en Stanley, C om puterScience D ívision, U niversity of C alifomia atB erkeley, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cstein - com pute the eigenvectors of a real sym \(m\) etric tridiagonal \(m\) atrix \(T\) comesponding to specified eigenvalues, using inverse iteration

\section*{SYNOPSIS}

```

    \mathbb{FA}|,\mathbb{NNOO}
    COM PLEX Z (LD Z,*)
\mathbb{NTEGERN,M,LDZ,INFO}

```

```

REALD (*),E (*),W (*),W ORK (*)
SUBROUT\mathbb{NECSTE\mathbb{N_64 N,D,E,M,W, BRLOCK,ISPLIT,Z,LD Z,W ORK,}}\mathbf{N},\textrm{N},\textrm{W}
IN ORK,\mathbb{FA}\mathbb{L},\mathbb{N}FO)
COM PLEX Z (LD Z,*)
\mathbb{NTEGER*8N,M,LD Z,INFO}

```

```

REALD (*),E (*),W (*),WORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE STE \(\mathbb{N}\) ( \(\mathbb{N}], D, E, \mathbb{M}], W, \mathbb{B L O C K}, \operatorname{ISPLIT}, \mathrm{Z},[L D Z],[W\) ORK], [ \(\mathbb{I N}\) ORK], \(\mathbb{F A} \mathbb{I},[\mathbb{N} F O]\) )

COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::Z
\(\mathbb{N} T E G E R:: N, M, L D Z, \mathbb{N F O}\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: \(\mathbb{B L O C K}, \operatorname{ISPLIT}, \mathbb{I N} O R K, \mathbb{F A} \mathbb{I}\)
REAL,D \(\mathbb{M}\) ENSION (:) ::D ,E,W ,W ORK
SU BROUTINESTE \(\left.\mathbb{N} \_64(\mathbb{N}], D, E, \mathbb{M}\right], W, \mathbb{B L O C K}, \operatorname{ISPL} \mathbb{T}, \mathrm{Z},[\operatorname{LD} Z]\),
[W ORK], [IW ORK], \(\mathbb{F} A \mathbb{I},[\mathbb{N} F O])\)

COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::Z
\(\mathbb{N} T E G E R(8):: N, M, L D Z, \mathbb{N F O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) :: \(\mathbb{B L O C K}, \operatorname{ISPL} \mathbb{T}, \mathbb{I N} O R K, \mathbb{F A} \mathbb{I}\) REAL,D \(\mathbb{M}\) ENSION (:) ::D, \(\mathrm{E}, \mathrm{W}, \mathrm{W}\) ORK

\section*{C INTERFACE}
\#include <sunperfh>
void cstein (intn, float *d, float *e, intm, float \({ }^{*}\) w, int *iblock, int *isplit, com plex *z, int ldz, int *ifail, int*info);
void cstein_ 64 llong \(n\), float *d, float *e, long \(m\), float * , , long *iblock, long *isplit, com plex *z, long ldz, long *ifail, long *info);

\section*{PURPOSE}
cstein com putes the eigenvectors of a realsym \(m\) etric tridiagonal \(m\) atrix \(T\) corresponding to specified eigenvalues, using inverse iteration.

Them axim um num ber of terations allow ed for each eigenvector is specified by an intemal param eterM A X ITS (currently set to 5).

A though the eigenvectors are real, they are stored in a com plex array, which m ay be passed to CUNM TR orCUPM TR for back transform ation to the eigenvectors of a com plex H erm itian \(m\) atrix w hich \(w\) as reduced to tridiagonal form .

\section*{ARGUMENTS}

N (input) The order of the m atrix. \(\mathrm{N}>=0\).

D (input) The n diagonalelem ents of the tridiagonal \(m\) atrix T.

E (input) The ( \(\mathrm{n}-1\) ) subdiagonalelem ents of the tridiagonal \(m\) atrix \(T\), stored in elem ents 1 to \(N-1 ; E(\mathbb{N})\) need notbe set.

M (input) The num ber of eigenvectors to be found. \(0<=\mathrm{M}<=\) N .

W (input) The firstM elem ents of \(W\) contain the eigenvalues for which eigenvectors are to be com puted. The eigenvalues should be grouped by split-off block and ordered from sm allest to largestw ithin the block. (The output anay \(W\) from SSTEBZ w ith ORDER = B'is expected here.)

IBLOCK (input)
The subm atrix indices associated \(w\) ith the corresponding eigenvalues in W ; \(\mathbb{B L O C K}(i)=1\) if eigenvalueW (i) belongs to the first subm atrix from the top, \(=2\) ifW (i) belongs to the second subm atrix, etc. (The output array \(\mathbb{B L O C K}\) from SSTEBZ is expected here.)

ISPLIT (input)
The splilting points, atw hich \(T\) breaks up into subm atrices. The first subm atrix consists of row s/columns 1 to ISPLIT ( 1 ), the second of row s/colum ns ISPLIT ( 1 ) +1 through ISPLIT (2), etc. (The outputarray ISPLIT from SSTEBZ is expected here.)

\section*{Z (output)}

The com puted eigenvectors. The eigenvector associated w ith the eigenvalue W (i) is stored in the \(i-t h\) colum \(n\) of \(Z\). A ny vectorw hich fails to converge is set to its cument iterate afterM AXITS terations. The im aginary parts of the eigenvectors are set to zero.

\section*{LD Z (input)}

The leading dim ension of the aray \(Z\). LD \(Z \quad>=\) \(\max (1, N)\).

W ORK (w orkspace)
dim ension ( \(5 * \mathrm{~N}\) )
IN ORK (w orkspace)
dim ension (N)

IFA II (output)
On norm alexit, allelem ents of \(\mathbb{F} A \mathbb{I}\) are zero.
If one orm ore eigenvectors fail to converge after
M AX ITS iterations, then their indices are stored
in array \(\mathbb{F A} \mathbb{I}\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an ille-
galvalue
\(>0:\) if \(\mathbb{N F O}=i\), then \(i\) eigenvectors failed to converge in M AXITS terations. Their indioes are stored in array \(\mathbb{F A} \mathbb{I I}\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
csteqr-com pute alleigenvalues and, optionally, eigenvectors of a sym \(m\) etric tridiagonalm atrix using the im plicit \(Q L\) orQ \(R\) m ethod

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CSTEQR (COMPZ,N,D,E,Z,LD Z,W ORK,NNFO)}
CHARACTER * 1 COMPZ
COM PLEX Z (LDZ,*)
\mathbb{NTEGER N,LD Z,INFO}
REALD (*),E (*),WORK (*)
SUBROUT\mathbb{NE CSTEQR_64 (COM PZ,N,D,E,Z,LD Z,W ORK,INFO)}
CHARACTER * 1 COMPZ
COM PLEX Z (LDZ,*)
INTEGER*8N,LD Z,INFO
REALD (*),E (*),W ORK (*)

```
F95 INTERFACE
    SU BROUTINE STEQR COMPZ, \(\mathbb{N}], D, E, Z,[L D Z],[\mathbb{W} O R K],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::COM PZ
    COM PLEX,D \(\mathbb{M}\) ENSION (:r:) ::Z
    \(\mathbb{N} T E G E R:: N, L D Z, \mathbb{N} F O\)
    REAL,D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK
    SU BROUTINE STEQR_64 (COMPZ, \(\mathbb{N}], D, E, Z,[L D Z],[W O R K],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::COM PZ
    COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::Z
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} \mathrm{Z}, \mathbb{N}\) FO
REAL,D \(\mathbb{I}\) ENSION (:) :: D , E,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void csteqr(char com pz, intn, float *d, float *e, com plex
* \(z\), int \(l d z\), int *info);
void csteqr_64 (charcom pz, long n, float *d, float *e, com plex *z, long ldz, long *info);

\section*{PURPOSE}
csteqr computes all eigenvalues and, optionally, eigenvectors of a sym \(m\) etric tridiagonalm atrix using the im plicit Q L orQ R m ethod. The eigenvectors of a full or band com plex H erm Itian \(m\) atrix can also be found ifCHETRD or CHPTRD orCHBTRD has been used to reduce thism atrix to tridiagonalform.

\section*{ARGUMENTS}

COMPZ (input)
\(=\mathrm{N}\) : C om pute eigenvahues only .
\(=\mathrm{V}\) ': Com pute eigenvalues and eigenvectors of the original \(H\) erm itian \(m\) atrix. On entry, \(Z \mathrm{~m}\) ust contain the unitary \(m\) atrix used to reduce the originalm atrix to tridiagonal form . = 'I': C om pute eigenvalues and eigenvectors of the tridiagonal m atrix. Z is initialized to the identity m atrix.

N (input) The order of the m atrix. \(\mathrm{N}>=0\).

D (input/output)
O n entry, the diagonal elem ents of the tridiagonal \(m\) atrix. On exit, if \(\mathbb{N F O}=0\), the eigenvalues in ascending order.

E (input/output)
O \(n\) entry, the ( \(n-1\) ) subdiagonal elem ents of the tridiagonal \(m\) atrix. On exit, E has been destroyed.

Z (input) On entry, if COMPZ \(=V\) ', then \(Z\) contains the unitary \(m\) atrix used in the reduction to tridiagonalform . On exit, if \(\mathbb{N F O}=0\), then if \(\mathrm{COMPZ}=\)

V', Z contains the orthonorm aleigenvectors of the original H erm itian \(m\) atrix, and if \(C O M P Z=' I\) ', \(Z\) contains the orthonorm al eigenvectors of the sym \(m\) etric tridiagonalm atrix. If \(C O M P Z=N\) ', then \(Z\) is not referenced.

LD \(Z\) (input)
The leading dim ension of the array Z . LD \(\mathrm{Z}>=1\), and if eigenvectors are desired, then LD Z >= \(\max (1, N)\).

W ORK (w orkspace)
dim ension (max (1,2*N-2)) IfCOM PZ = \(N\) ', then \(W\) ORK is not referenced.
\(\mathbb{N}\) FO (output)
= 0 : successfulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum enthad an illegalvalue
>0: the algorithm has failed to find all the eigenvalues in a total of \(30 *\) N iterations; if \(\mathbb{N}\) FO \(=i\), then ielem ents of \(E\) have not converged to zero; on exit, \(D\) and \(E\) contain the elem ents of a sym \(m\) etric tridiagonalm atrix which is unitarily sim ilar to the originalm atrix.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cstsv - com pute the solution to a com plex system of linear equations \(A * X=B\) where \(A\) is a \(H\) erm itian tridiagonal \(m\) atrix

\section*{SYNOPSIS}

```

COM PLEX L (*),D (*),SUBL (*),B (LDB,*)
INTEGER N,NRHS,LDB,INFO
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(*)}
SUBROUT\mathbb{NECSTSV_64 N,NRHS,L,D,SUBL,B,LDB,\mathbb{PIV ,NNFO )}}\mathbf{N}\mathrm{ (N,}
COM PLEX L (*),D (*),SUBL (*),B (LDB ,*)
INTEGER*8N,NRHS,LDB, INFO
\mathbb{NTEGER** \mathbb{PIV (*)}}\mathbf{*}\mathrm{ ( }

```

\section*{F95 INTERFACE}

SU BROUTINE STSV ( \(\mathbb{N}], \mathbb{N} R H S], L, D, S U B L, B,[L D B], \mathbb{P} \mathbb{V},[\mathbb{N} F O])\)
COMPLEX,D \(\mathbb{M}\) ENSION (:) ::L,D,SUBL
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::B
\(\mathbb{I N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}\)

SU BROUTINE STSV_64 (N ], \(\mathbb{N} R H S], L, D, S U B L, B,[L D B], \mathbb{P} \mathbb{I V},[\mathbb{N} F O])\)

COMPLEX,D \(\mathbb{I M}\) ENSION (:) ::L,D,SUBL
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::B
\(\mathbb{N}\) TEGER (8) :: N,NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{I} \operatorname{ENS} \operatorname{ION}(:):: \mathbb{P} \mathbb{V}\)

\section*{C INTERFACE}
\#include <sunperfh>
void cstsv (intn, intnrhs, com plex *l, com plex *d, com plex
*subl, com plex *b, int ldb, int *ịì, int *info);
void cstsv_64 (long n, long nrhs, com plex *l, com plex *d, com plex *subl, com plex *b, long ldb, long *ịìì, long *info);

\section*{PURPOSE}
cstsv com putes the solution to a com plex system of linear equations \(\mathrm{A} * \mathrm{X}=\mathrm{B}\) where A is a H erm tian tridiagonal \(m\) atrix.

\section*{ARGUMENTS}

N (input)
The order of them atrix A. \(\mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides in \(B\).
L (input/output)
COM PLEX array, dim ension (N)
O n entry, the \(\mathrm{n}-1\) subdiagonalelem ents of the tridiagonal \(m\) atrix A. On exit, part of the factorization of \(A\).

D (input/output)
REA L array, dim ension \(\mathbb{N}\) )
O n entry, the n diagonalelem ents of the tridiagonalm atrix A. On exit, the \(n\) diagonalelem ents of the diagonalm atrix \(D\) from the factorization of \(A\).

\section*{SU BL (output)}

COM PLEX array, dim ension \(\mathbb{N}\) )
On exit, part of the factorization ofA.
B (input/output)
The collm ns ofB contain the righthand sides.
LD B (input)
The leading dim ension of \(B\) as specified in a type orD \(\mathbb{I M}\) ENSIO N statem ent.
\(\mathbb{P} \mathbb{I} V\) (output)
\(\mathbb{N}\) TEGER array, dim ension \((\mathbb{N})\)
O n exit, the pivot indices of the factorization.
\(\mathbb{I N} F O\) (output)
IN TEGER
= 0: successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-\) i, the i-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{D}(\mathrm{k}, \mathrm{k})\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix D is exactly singular and division
by zero w illoccur if it is used to solve a system of equations.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
csturf-com pute the factorization of a com plex Herm itian tridiagonalm atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CSTTRF N,L,D,SUBL, \mathbb{PIV,NNFO)}}\mathbf{N},\mathbb{N}
COM PLEX L (*),D (*),SUBL (*)
INTEGERN,\mathbb{NFO}
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(*)}

```

```

COM PLEX L (*),D (*),SUBL (*)
INTEGER*8N,\mathbb{NFO}
\mathbb{NTEGER*8 \mathbb{PIV (*)}}\mathbf{*}/

```
F95 INTERFACE
    SU BROUTINE STTRF ( \(\mathbb{N}], L, D, S U B L, \mathbb{P} \mathbb{I}\), \([\mathbb{N} F O]\) )
    COM PLEX,D IM ENSION (:) ::L,D ,SUBL
    \(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
    \(\mathbb{N}\) TEGER,D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V}\)
    SU BROUTINE STTRF_64 ( \(\mathbb{N}\) ],L,D ,SUBL, \(\mathbb{P} \mathbb{I V},[\mathbb{N} F O]\) )
    COM PLEX,D IM ENSION (:) ::L,D,SUBL
    \(\mathbb{N}\) TEGER ( 8 ): : \(\mathrm{N}, \mathbb{N} F \mathrm{O}\)
    \(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION(:) :: \(\mathbb{P} \mathbb{I}\)
C INTERFACE
    \#include <sunperfh>
void csttrf(intn, com plex *l, com plex *d, com plex *subl, int*ipiv, int*info);
void csttrf_64 (long n, com plex *l, com plex *d, com plex *subl, long *ịiv, long *info);

\section*{PURPOSE}
csttrf com putes the \(L * D * L * * H\) factorization of a com plex H er\(m\) itian tridiagonalm atrix A.

\section*{ARGUMENTS}

N (input) \(\mathbb{N}\) TEGER
The order of them atrix \(A . N>=0\).

L (input/output)
COM PLEX aray, dim ension \((\mathbb{N})\)
O n entry, the n-1 subdiagonalelem ents of the tridiagonal m atrix A. On exit, part of the factorization of A .

D (input/output)
REAL array, dim ension \(\mathbb{N}\) )
O n entry, the \(n\) diagonalelem ents of the tridiagonalm atrix A. On exit, the \(n\) diagonalelem ents of the diagonalm atrix \(D\) from the factorization of \(A\).

SUBL (output)
COM PLEX aray, dim ension \((\mathbb{N})\)
O n exit, part of the factorization of .

IP IV (output)
\(\mathbb{N}\) TEGER array, dim ension \((\mathbb{N})\)
O \(n\) exit, the pivot indices of the factorization.
\(\mathbb{I N F O}\) (output)
IN TEGER
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an ille-
galvałue
\(>0:\) if \(\mathbb{N} F O=i, D(k, k)\) is exactly zero. The factorization has been com pleted, but the block
diagonalm atrix D is exactly singular and division
by zero w illoccur if it is used to solve a system
of equations.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
csttrs - com putes the solution to a com plex system of linear equations \(A * X=B\)

\section*{SYNOPSIS}

```

COM PLEX L (*),D (*),SUBL (*),B (LDB,*)
\mathbb{NTEGER N,NRHS,LDB,INFO}
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(*)}

```

```

COM PLEX L (*),D (*),SUBL (*),B (LD B,*)
INTEGER*8N,NRHS,LDB,INFO
\mathbb{NTEGER*8 \mathbb{PIV (*)}}\mathbf{*}/

```

\section*{F95 INTERFACE}

SUBROUTINE STTRS ( \(\mathbb{N}], \mathbb{N} R H S], L, D, S U B L, B,[L D B], \mathbb{P} \mathbb{I},[\mathbb{N} F O])\)
COM PLEX,D \(\mathbb{I M} E N S I O N(:):: L, D, S U B L\)
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::B
\(\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}\) (:) :: \(\mathbb{P} \mathbb{I V}\)
SUBROUTINE STTRS_64 ( \(\mathbb{N}], \mathbb{N} R H S], L, D, S U B L, B,[L D B], \mathbb{P} \mathbb{I},[\mathbb{N} F O])\)
COMPLEX,D \(\mathbb{M} \operatorname{ENSION(:)::L,D,SUBL}\)
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::B
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LD B, \(\mathbb{N}\) FO
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{I} \operatorname{ENS} \mathbb{O} \mathrm{N}(:):: \mathbb{P} \mathbb{I}\)

\section*{C INTERFACE}
\#include <sunperfh>
void csttrs (intn, intnrhs, com plex *l, com plex *d, com plex
*subl, com plex *b, int ldb, int *ipiv, int *info);
void csttrs_64 (long n, long nihs, complex *l, complex *d, com plex *subl, com plex *b, long ldb, long *ipiv, long *info);

\section*{PURPOSE}
csttrs com putes the solution to a com plex system of linear equations \(A * X=B\), where \(A\) is an \(N\) boy \(-N\) sym m etric tridiagonalm atrix and X and B are N -by-N R H S m atrioes.

\section*{ARGUMENTS}

N (input) \(\mathbb{N}\) TEGER
The order of them atrix \(A . N>=0\).

NRHS (input)
\(\mathbb{N}\) TEGER
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B.NRH S >=0.

L (input) COM PLEX array, dim ension \(\mathbb{N}-1\) )
O n entry, the subdiagonalelem ents ofLL and D D.

D (input) COM PLEX aray, dim ension \(\mathbb{N}\) )
O n entry, the diagonalelem ents ofD D .

SUBL (input)
COM PLEX aray, dim ension \((\mathbb{N}-2)\)
O n entry, the second subdiagonalelem ents of LL .

B (input/output)
COM PLEX array, dim ension (LD B , NRHS)
On entry, the N boy-NRHS righthand side m atrix B. On exit, if \(\mathbb{N F O}=0\), the N boy-NRH S solution m atrix X .

LD B (input)
\(\mathbb{N}\) TEGER
The leading dim ension of the aray B. LD B >= \(\max (1, N)\)

IPIV (output)
\(\mathbb{N}\) TEGER array, dim ension \(\mathbb{N}\) )
D etails of the interchanges and block pivot. If \(\mathbb{P} \mathbb{V}(\mathbb{K})>0,1\) by 1 pìvot, and if \(\mathbb{P} \mathbb{V}(\mathbb{K})=K+1\) an interchange done; If \(\mathbb{P} \mathbb{I V}(\mathbb{K})<0,2\) by 2
pivot, no interchange required.
\(\mathbb{I N F O}\) (output)
\(\mathbb{N}\) TEGER
= 0: successfulexit
\(<0\) : if \(\mathbb{N} F O=-k\), the \(k\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

CSw ap -Exchange vectors x and y.

```

\section*{SYNOPSIS}

```

COM PLEX X (*),Y (*)

```

```

SUBROUTINE CSW AP_64 N,X,NNCX,Y,\mathbb{NCY)}
COM PLEX X (*),Y (*)
INTEGER*8N,\mathbb{NCX,INCY}

```

\section*{F95 INTERFACE}

SU BROUTINE SW AP ( \(\mathbb{N}], X,[\mathbb{N} C X], Y,[\mathbb{N} C Y])\)
COM PLEX,D \(\mathbb{I M}\) ENSION (:) :: \(\mathrm{X}, \mathrm{Y}\) \(\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y\)

SU BROUTINE SW AP_64 (N ],X, [ \(\mathbb{N} C X], Y,[\mathbb{N} C Y])\)

COM PLEX,D \(\mathbb{I M}\) ENSION (:) :: \(\mathrm{X}, \mathrm{Y}\)
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathbb{I N C X , ~} \mathbb{N} C Y\)
C INTERFACE
\#include < sunperfh>
void csw ap (intn, com plex *x, int incx, com plex *y, int incy);
void csw ap_64 (long n, com plex *x, long incx, com plex *y,

\section*{PURPOSE}

Csw ap Exchange \(x\) and \(y\) where \(x\) and \(y\) are \(n-v e c t o r s . ~\)

\section*{ARGUMENTS}

N (input)
On entry, \(N\) specifies the num ber of elem ents in the vector. N m ustbe at leastone for the subroutine to have any visible effect. U nchanged on exit.
\(X\) (input/output)
( \(1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C X)\) ). On entry, the
increm ented array \(X\) m ust contain the vector \(x\). On
exit, the \(y\) vector.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X\) m ustnotbe zero. U nchanged on exit.

Y (input/output)
( \(1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C Y)\) ). On entry, the increm ented amay \(Y \mathrm{~m}\) ust contain the vectory. On exit, the x vector.
\(\mathbb{N} C Y\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of Y. \(\mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
csycon -estim ate the reciprocalof the condition num ber (in the 1 -norm ) of a com plex sym m etricm atrix A using the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by CSY TRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NECSYCON(UPLO,N,A,LDA,\mathbb{PIVOT,ANORM,RCOND,W ORK,INFO)}}\mathbf{N}\mathrm{ (N,N}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),W ORK (*)
NNTEGER N,LDA,}\mathbb{N}F
\mathbb{NTEGER \mathbb{PIVOT (*)}}\mathbf{(})
REAL ANORM,RCOND
SUBROUT\mathbb{NECSYCON_64 (UPLO,N,A,LDA, \mathbb{PIVOT,ANORM,RCOND,W ORK,}}\mathbf{N},\textrm{N},\textrm{A}
\mathbb{NFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGER*8N,LDA,}\mathbb{N}FO
\mathbb{NTEGER*8 P\mathbb{IVOT (*)}}\mathbf{(})
REALANORM,RCOND

```

\section*{F95 INTERFACE}
```

SU BROUTINE SYCON (UPLO, $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I} O T, A N O R M, R C O N D,[W O R K]$, [ $\mathbb{N} F O$ ])
CHARACTER (LEN=1)::UPLO
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R:: N, L D A, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T$

```

REAL ::ANORM,RCOND

SU BROUTINE SYCON_64 (UPLO, \(\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I} O T, A N O R M, R C O N D,[W O R K]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) :: UPLO
COMPLEX,D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX , D \(\mathbb{M}\) ENSION (: : : : : A
\(\mathbb{I N}\) TEGER (8) :: N, LD A , \(\mathbb{N F}\) F
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T\)
REAL ::ANORM,RCOND

\section*{C INTERFACE}
\#include <sunperfh>
void csycon (char uple, int n, com plex *a, int lda, int *ịívot, floatanorm, float*roond, int*info);
void csycon_64 (charuplo, long n, com plex *a, long lda, long *ipivot, floatanorm , float *roond, long *info);

\section*{PURPOSE}
csycon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a com plex sym \(m\) etric \(m\) atrix A using the factorization \(A=U * D * U * * T\) orA \(=L * D * L * * T\) com puted by CSY TRF .

A \(n\) estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND \(=1 /\) (ANORM * norm (inv (A))).

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) : : U pper triangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\);
\(=\mathrm{L}^{\prime}:\) Low er triangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\).

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input) The block diagonalm atrix \(D\) and the m ultipliers used to obtain the factorU orL as com puted by CSY TRF.

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, N)\).
\(\mathbb{P} \mathbb{V} O T\) (input)
D etails of the interchanges and the block structure ofD as determ ined by CSY TRF.

\section*{ANORM (input)}

The 1-norm of the originalm atrix A.

\section*{RCOND (output)}

The reciprocal of the condition num ber of the \(m\) atrix \(A\), com puted as RCOND \(=1 /(A N O R M * A \mathbb{N} V N M)\), where \(A \mathbb{N} V N M\) is an estim ate of the 1 -nom of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
csymm -perform one of the \(m\) atrix-m atrix operations \(C:=\) alpha*A *B + beta*C orC := alpha*B *A + beta*C

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CSYMM (S\mathbb{DE,UPLO,M,N,ALPHA,A,LDA,B,LDB,BETA,C,}}\mathbf{M},\textrm{L}
LD C )
CHARACTER * 1SIDE,UPLO
COM PLEX A LPHA,BETA
COM PLEX A (LDA,*),B (LD B,*),C (LD C ,*)
INTEGERM,N,LDA,LDB,LDC
SUBROUTINE CSYMM _64 (SDE,UPLO ,M ,N,ALPHA,A,LDA,B,LDB,BETA,C,
LD C )

```
CHARACTER * 1 SDE,UPLO
COM PLEX ALPHA,BETA
COM PLEX A (LDA , *), B (LD B,*), C (LD C , *)
\(\mathbb{N}\) TEGER*8 M , N,LDA, LD B, LD C

\section*{F95 INTERFACE}

SU BROUTINE SYMM (SDE,UPLO, \(\mathbb{M}\) ], \(\mathbb{N}], A L P H A, A,[L D A], B,[L D B]\), BETA, C, [LDC])

CHARACTER (LEN=1) ::SDE,UPLO
COMPLEX ::ALPHA,BETA
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,B,C
\(\mathbb{N} T E G E R:: M, N, L D A, L D B, L D C\)
SU BROUTINE SYMM_64 (SDE, UPLO, \(\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A], B,[L D B]\), BETA, C, [LDC])

CHARACTER ( \(\llcorner E N=1\) ) : : SDE E , UPLO
COM PLEX ::ALPHA,BETA
COMPLEX, D \(\mathbb{M}\) ENSION (: : : : : A, B, C
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{LD} A, L D B, L D C\)

\section*{C INTERFACE}
\#include <sunperfh>
void csym m (charside, char uplo, int m, int n, com plex
*alpha, com plex *a, int lda, com plex *b, int ldlo, com plex *beta, com plex *c, int ldc);
void csym m_64 (char side, charuplo, long m, long n, com plex *alpha, com plex *a, long lda, com plex *b, long ldlb, com plex *beta, com plex * C , long ldc);

\section*{PURPOSE}
csymm perform sone of the \(m\) atrix-m atrix operations \(C:=\) alpha*A *B + beta*C orC \(:=\) alpha*B *A + beta*C where alpha and beta are scalars, \(A\) is a sym \(m\) etric \(m\) atrix and \(B\) and \(C\) are \(m\) by \(n m\) atrices.

\section*{ARGUMENTS}

SID E (input)
O n entry, SIDE specifiesw hether the sym metric \(m\) atrix A appears on the leftorright in the operation as follow s:
\(S \mathbb{D E}=\) L'or I' \(C:=\) a耳pha*A *B + beta* \(C\),
\(S \mathbb{D} E=R\) 'or \(r^{\prime} \mathrm{C}:=\) alpha*B*A + beta* \(C\),

U nchanged on exit.

\section*{UPLO (input)}

On entry, UPLO specifies whether the upper or lower triangular part of the symm etric \(m\) atrix \(A\) is to be referenced as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or U ' Only the upper triangularpart of the sym \(m\) etric \(m\) atrix is to be referenced.
\(\mathrm{UPLO}=\mathrm{L}\) 'or \(\mathrm{I}^{\prime}\) ' O nly the low ertriangularpart of the sym \(m\) etric \(m\) atrix is to be referenced.

U nchanged on exit.
M (input)
O \(n\) entry, M specifies the num ber of row sof the \(m\) atrix \(C . M\) M \(=0\). U nchanged on exit.

N (input)
O n entry, N specifies the num ber of colum ns of the \(m\) atrix \(C . N>=0\). U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.
A (input)
COM PLEX array ofD \(\mathbb{I M}\) ENSION (LDA,ka), where ka is \(m\) when \(S D E=\mathbb{L}\) 'or \(I^{\prime}\) and is \(n\) other-
w ise.
Before entry with SDDE \(=\mathrm{L}\) 'or 1 ', the \(m\) by \(m\) part of the anay A mustcontain the sym\(m\) etric \(m\) atrix, such thatw hen UPLO \(=U\) 'or 4 ', the leading \(m\) by \(m\) uppertriangularpart of the array A mustcontain the upper triangular part of the symm etric \(m\) atrix and the strictly lower triangularpart of A is not referenced, and \(w\) hen UPLO = L' or \({ }^{1}\) ', the leading \(m\) by \(m\) low er triangularpart of the array A must contain the lower triangular part of the sym \(m\) etric \(m\) atrix and the strictly upper triangular partof A is not referenced.

Before entry with \(S D E=R\) 'or \(r\) ', the \(n\) by \(n\) part of the array \(A\) mustcontain the sym\(m\) etric \(m\) atrix, such thatw hen UPLO \(=U\) 'or L ', the leading \(n\) by \(n\) uppertriangularpart of the aray A mustcontain the upper triangular part of the symm etric \(m\) atrix and the strictly lower triangularpartof \(A\) is not referenced, and when UPLO = L' or 1 ', the leading \(n\) by \(n\) low er triangularpart of the array A must contain the lower triangular part of the sym \(m\) etric \(m\) atrix and the strictly upper triangular part of A is not referenced.

U nchanged on exit.

LDA (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen
\(S \mathbb{D} E=\mathbb{L}\) 'or 1 ' then LD \(A>=m a x(1, m)\), otherw ise LD \(\mathrm{A}>=\max (1, \mathrm{n})\). U nchanged on exit.

B (input)
COM PLEX aray ofD \(\mathbb{I M}\) ENSION (LDB, n ). Before entry, the leading \(m\) by \(n\) part of the array \(B\) \(m\) ustcontain the \(m\) atrix \(B\). U nchanged on exit.

LD B (input)
O n entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program.
LD B \(>=\max (1, m)\). U nchanged on exit.
BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then C need notbe set on input. U nchanged on exit.

C (input/output)
COM PLEX aray ofD \(\mathbb{I M}\) ENSION (LD C, n ). Before entry, the leading \(m\) by \(n\) partof the amay \(C\) \(m\) ustcontain the \(m\) atrix \(C\), exceptw hen beta is zero, in which case \(C\) need notbe seton entry. On exit, the array \(C\) is overw rilten by the \(m\) by \(n\) updated \(m\) atrix.

LD C (input)
O n entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program.
LD C \(>=\max (1, m)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
csyn \(2 k\)-perform one of the sym \(m\) etric rank \(2 k\) operations \(C\) \(:=\) alpha*A *B' + alpha*B*A ' + beta*C orC \(:=\) alpha*A *B + alpha*B *A + beta*C

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CSYR2K (UPLO,TRANSA,N,K,ALPHA,A,LDA,B,LDB,BETA,C,}
LD C)
CHARACTER * 1 UPLO,TRANSA
COM PLEX ALPHA,BETA
COM PLEX A (LDA,*),B (LD B,*),C (LD C,*)
IN TEGER N,K,LDA,LDB,LDC
SUBROUTINE CSYR2K_64 (UPLO,TRANSA,N,K,ALPHA,A,LDA,B,LDB,BETA,
C,LDC)

```
CHARACTER * 1 UPLO, TRANSA
COM PLEX ALPHA,BETA
COM PLEX A (LDA,*), B (LD B,*), C (LD C , *)
\(\mathbb{N}\) TEGER*8N,K,LDA,LDB,LDC

\section*{F95 INTERFACE}

SU BROUTINE SYR2K (UPLO, [TRANSA ], \(\mathbb{N}],[K], A L P H A, A,[L D A], B,[L D B]\), BETA, C, [LDC])

CHARACTER (LEN=1) ::UPLO,TRANSA
COMPLEX ::ALPHA,BETA
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A, B, C
\(\mathbb{N} T E G E R:: N, K, L D A, L D B, L D C\)
SU BROUTINE SYR2K_64 (UPLO, [TRANSA], \(\mathbb{N}],[K], A L P H A, A,[L D A], B\),
[LD B],BETA, C, [LD C ])

CHARACTER (LEN=1) ::UPLO,TRANSA
COMPLEX ::ALPHA,BETA
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,B,C
\(\mathbb{N}\) TEGER (8) ::N , K, LDA ,LDB,LDC

\section*{C INTERFACE}
\#include <sunperfh>
void csyr2k (charuplo, chartransa, intn, int \(k\), com plex *alpha, com plex *a, int lda, com plex *b, int ldb, com plex *beta, com plex *c, int ldc);
void csyr2k_64 (charuplo, chartransa, long n, long k, com plex *alpha, com plex *a, long lda, com plex *b, long ldb, com plex *beta, com plex *c, long ldc);

\section*{PURPOSE}
csyn2k perform s one of the sym \(m\) etric rank 2 k operations \(\mathrm{C}:=\) alpha*A *B'+ alpha*B*A ' + beta*C or C : alpha*A *B + alpha*B *A + beta*C where alpha and beta are scalars, C is an \(n\) by \(n\) symm etric \(m\) atrix and \(A\) and \(B\) are \(n\) by \(k\) \(m\) atrices in the first case and \(k\) by \(n m\) atrices in the second case.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the upper
or lower triangular part of the array \(C\) is
to be referenced as follow s:

UPLO = U'or L' Only the upper triangular partof \(C\) is to be referenced.

UPLO = L'or I' Only the lower triangular partof C is to be referenced.

U nchanged on exit.
TRANSA (input)
O \(n\) entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} \quad C:=a l p h a * A * B^{\prime}+\) alpha*B*A '+ beta*C .

TRANSA \(=\) T' or t' \(C=\) alpha*A *B + alpha*B *A + beta*C.

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

N (input)
O n entry, \(N\) specifies the order of the m atrix C. N m ustibe at least zero. U nchanged on exit.

K (input)
On entry w ith TRANSA = N 'or h', K specifies the num ber of colum ns of the \(m\) atrioes \(A\) and \(B\), and on entry w th TRANSA = T' or \(t^{\prime}, \mathrm{K}\) specifies the num ber of row s of the \(m\) atrices \(A\) and B. K m ustibe at least zero. U nchanged on ex止.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
COM PLEX aray ofD \(\mathbb{M} E N S I O N\) (LDA, ka ),
where ka isk when TRANSA = N 'or h', and is
n otherw ise. Before entry w ith TRANSA \(=\mathrm{N}^{\prime}\) or
\(h\) ', the leading \(n\) by k partof the array \(A\) \(m\) ustcontain the \(m\) atrix \(A\), otherw ise the leading k by n partof the array A mustcontain the \(m\) atrix A. U nchanged on exit.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program.
W hen TRANSA \(=N\) 'or \(h\) 'then LDA must be at least \(\max (1, n)\), otherw ise LD A m ustbe at least \(\max (1, k)\). U nchanged on exit.

B (input)
COM PLEX aray ofD \(\mathbb{M} E N S I O N\) (LDB, kb ),
where kb isk when TRANSA = N 'or h', and is
n otherw ise. Before entry w ith TRANSA \(=\mathrm{N}^{\prime}\) or
\(h\) ', the leading \(n\) by k part of the array \(B\) m ustcontain the \(m\) atrix \(B\), otherw ise the leading k by n partof the aray \(B \mathrm{~m}\) ustcontain the \(m\) atrix \(B\). U nchanged on exit.

LD B (input)

On entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program. W hen TRANSA = N 'or h'then LDB must be at least \(\max (1, n)\), otherw ise LD \(B\) ustbe at least \(\max (1, k)\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. U nchanged on exit.
C (input/output)
COM PLEX array ofD \(\mathbb{M}\) ENSION (LDC,n).
Before entry w ith UPLO = U 'or L', the leading \(n\) by \(n\) upper triangularpart of the array \(C\) \(m\) ustcontain the upper triangular part of the sym metric \(m\) atrix and the strictly low er triangularpartofC is not referenced. On exit, the upper triangularpart of the array \(C\) is overw ritten by the upper triangularpart of the updated \(m\) atrix.

Before entry w ith UPLO = L'or 1', the leading \(n\) by \(n\) low er triangularpart of the array \(C\) m ustcontain the low er triangular part of the sym \(m\) etric \(m\) atrix and the strictly uppertriangularpartofC is not referenced. On exit, the low er triangularpart of the array \(C\) is overw ritten by the low er triangularpart of the updated \(m\) atrix.

LD C (input)
O n entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program.
LD C must be at leastmax (1,n). Unchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
csyrfs - im prove the com puted solution to a system of linear equations when the coefficientm atrix is sym \(m\) etric indefinte, and provides emorbounds and backw ard error estim ates for the solution

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CSYRFS (UPLO,N,NRHS,A,LDA,AF,LDAF, \mathbb{PIVOT,B,LDB,X,}}\mathbf{N},\textrm{L},\textrm{L}
LD X,FERR,BERR,W ORK,W ORK 2, INFO)
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),AF (LDAF,*),B (LDB,*),X (LDX,*),W ORK (*)
\mathbb{NTEGERN,NRHS,LDA,LDAF,LDB,LDX,}\mathbb{N}FO
\mathbb{NTEGER \mathbb{PIVOT (*)}}\mathbf{(})
REAL FERR(*),BERR(*),WORK2 (*)
SU BROUTINE CSYRFS_64 (UPLO,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,B,LDB,}
X,LDX,FERR,BERR,W ORK,W ORK 2,INFO)
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),AF (LDAF,*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGER*8N,NRHS,LDA,LDAF,LDB,LDX,}\mathbb{N}F
\mathbb{NTEGER*8 P\mathbb{IVOT (*)}}\mathbf{(})
REAL FERR (*),BERR (*),W ORK2 (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE SYRFS (UPLO, $\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F], \mathbb{P} \mathbb{I V O T}, \mathrm{B}$, [LDB], $X,[L D X], F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N F O}])$
CHARACTER (LEN=1) ::UPLO
COM PLEX,D IM ENSION (:) ::W ORK
COMPLEX,D $\mathbb{M}$ ENSION (:,:): :A,AF,B,X

```
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL,D \(\mathbb{M} E N S I O N(:):: F E R R, B E R R, W\) ORK 2

SU BROUTINE SYRFS_64 (UPLO, \(\mathbb{N}], \mathbb{N R H S}], A,[L D A], A F,[L D A F], \mathbb{P} \mathbb{I} O T\), \(B,[L D B], X,[L D X], F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO
COM PLEX , D \(\mathbb{M} E N S I O N(:):: W\) ORK
COM PLEX, D \(\mathbb{M}\) ENSION (: : : : : A, AF, B, X
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{V} O T\)
REAL,D \(\mathbb{M} E N S I O N(:):: F E R R, B E R R, W\) ORK 2
C INTERFACE
\#include <sunperfh>
void csyrfs (charuplo, intn, intnihs, com plex *a, int lda, com plex *af, int ldaf, int*ipivot, com plex *b, int ldlo, com plex *x, int ldx, float *ferr, float *berr, int *info);
void csyrfs_64 (charuplo, long n, long nrhs, com plex *a, long lda, com plex *af, long ldaf, long *ipivot, com plex *b, long ldb, com plex *x, long ldx, float * ferrr, float *berr, long *info);

\section*{PURPOSE}
csyrfs im proves the com puted solution to a system of linear equations \(w\) hen the coefficientm atrix is sym \(m\) etric indefintie, and provides errorbounds and backw ard error estim ates for the solution.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': Upper triangle ofA is stored;
= IL ': Low er triangle ofA is stored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrices B and X. NRHS \(>=0\).

A (input) The symm etricm atrix A . If \(\mathrm{UPLO}=\mathrm{U}\) ', the leading N -by- N uppertriangularpartofA contains the
upper triangularpart of the \(m\) atrix \(A\), and the strictly low ertriangularpartofA is not referenced. IfU PLO = L', the leading N -by-N lower triangularpart ofA contains the low er triangular part of the \(m\) atrix A, and the strictly upper triangularpartofA is not referenced.

LD A (input)
The leading dim ension of the anay A. LDA >= \(\max (1, N)\).

\section*{AF (input)}

The factored form of them atrix A. AF contains the block diagonal \(m\) atrix \(D\) and the \(m\) ultipliers used to obtain the factor \(U\) orL from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) as com puted by CSY TRF.

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, N)\).
\(\mathbb{P I V O T}\) (input)
D etails of the interchanges and the block structure ofD as determ ined by CSY TRF.
\(B\) (input) The righthand side \(m\) atrix \(B\).
LD B (input)
The leading dim ension of the amay \(B\). LD B >= \(\max (1, \mathbb{N})\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by CSY TRS. On exit, the im proved solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the anay X . LD X >= \(\max (1, N)\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X()\) ) the \(j\) th column of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{H}), \operatorname{FERR}(\mathcal{)})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{\nu})-X\) TRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(j)\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector \(X\) ( \(\mathcal{j}\) ) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{J}\) ) an exactsolution).
W ORK (w orkspace)
dim ension \((2 * N)\)
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO = -i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
csyık -perform one of the sym \(m\) etric rank \(k\) operations \(C\) : alpha*A *A ' beta*C orC : alpha*A *A + beta*C

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CSYRK (UPLO,TRANSA,N,K,ALPHA,A,LDA,BETA,C,LDC)}
CHARACTER * 1 UPLO,TRANSA
COM PLEX ALPHA,BETA
COM PLEX A (LDA,*),C (LDC ,*)
INTEGER N,K,LDA,LDC
SU BROUTINE CSYRK_64 (UPLO,TRANSA ,N,K,A LPHA,A,LDA,BETA,C,LDC)
CHARACTER * 1 UPLO,TRANSA
COM PLEX ALPHA,BETA
COM PLEX A (LDA,*),C (LDC ,*)
INTEGER*8N,K,LDA,LDC

```

\section*{F95 INTERFACE}

SU BROUTINE SYRK (UPLO, [TRANSA], \(\mathbb{N}],[K], A L P H A, A,[L D A], B E T A, C\), [LD C ])

CHARACTER (LEN=1) ::UPLO,TRANSA
COMPLEX ::ALPHA,BETA
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : ::A, C
\(\mathbb{N} T E G E R:: N, K, L D A, L D C\)
SU BROUTINE SYRK_64 (UPLO, [TRANSA], \(\mathbb{N}],[K], A L P H A, A,[L D A], B E T A\), C, [LD C ])

CHARACTER (LEN=1) ::UPLO,TRANSA

COM PLEX ::ALPHA,BETA
COM PLEX,D \(\operatorname{IM}\) ENSION (:,:) ::A,C
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathrm{LD} \mathrm{C}\)

\section*{C INTERFACE}
\#include <sunperfh>
void csyrk (charuplo, chartransa, int \(n\), int \(k\), com plex *alpha, com plex *a, intlda, com plex *beta, com plex *c, int ldc);
void csyık_64 (char uplo, char transa, long n, long k, com plex *alpha, com plex *a, long lda, com plex *beta, com plex \({ }^{*}\) c, long ldc);

\section*{PURPOSE}
csyik perform s one of the sym \(m\) etric rank \(k\) operations \(C:=\) alpha*A *A '+ beta*C orC := alpha*A *A + beta*C where alpha and beta are scalars, \(C\) is an \(n\) by \(n\) sym \(m\) etric \(m\) atrix and \(A\) is an \(n\) by \(k m\) atrix in the first case and a \(k\) by \(n\) \(m\) atrix in the second case.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the upper
or low er triangular part of the amay \(C\) is
to be referenced as follow s:

UPLO = U'or \(\mathrm{U}^{\prime}\) Only the upper triangular part of \(C\) is to be referenced.

UPLO = L'or I' Only the lower triangular part of \(C\) is to be referenced.

U nchanged on exit.

TRANSA (input)
O n entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} \mathrm{C}:=\) alpha*A *A ' + beta* C.

TRANSA \(=\) T'ort' \(\mathrm{C}:=\) alpha*A *A + beta*C.

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF95 \(\mathbb{I N T E R F A C E .}\)
N (input)
On entry, \(N\) specifies the order of the \(m\) atrix \(C\).
N m ustbe at least zero. U nchanged on exit.
\(K\) (input)
On entry with TRANSA \(=N\) 'or \(h\) ', \(K\) specifies the number of columns of the matrix \(A\), and on entry \(w\) th TRANSA \(=T^{\prime}\) or \(t^{\prime}, \mathrm{K}\) specifies the num ber of row sof the m atrix A. K m ust.be at least zero. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha.
U nchanged on exit.

A (input)
COM PLEX aray ofD \(\mathbb{M}\) ENSION (LDA, ka ),
where ka isk when TRANSA = N 'or h', and is
n otherw ise. Before entry w ith TRANSA \(=\mathrm{N}^{\prime}\) or
h ', the leading n by k part of the array A
m ustcontain the \(m\) atrix \(A\), otherw ise the leading k by n partof the amay A mustcontain the \(m\) atrix A. U nchanged on exit.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program.
W hen TRANSA \(=\mathrm{N}\) 'or h 'then LDA must be at
least \(\max (1, \mathrm{n})\), otherw ise LDA m ust.be at least \(\max (1, k)\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. U nchanged on exit.

C (input/output)
COMPLEX aray ofD \(\mathbb{M} E N S \mathbb{N} N(L D C, n)\).
Before entry w ith UPLO = U 'or L', the leading \(n\) by \(n\) upper triangularpart of the array \(C\) \(m\) ustcontain the upper triangular part of the sym \(m\) etric \(m\) atrix and the strictly low ertriangularpartofC is not referenced. On exit, the upper triangularpart of the array \(C\) is overw ritten by the upper triangularpart of the updated \(m\) atrix.

Before entry with UPLO = L'or 1', the leading \(n\) by \(n\) low er triangularpart of the anray \(C\) \(m\) ustcontain the low er triangular part of the sym \(m\) etric \(m\) atrix and the strictly upper triangularpartof \(C\) is not referenced. On exit, the low er triangularpart of the array \(C\) is overw ritten by the low er triangularpart of the updated \(m\) atrix.

LD C (input)
On entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program. LD C must be at leastmax(1,n). Unchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
csysv - com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CSYSV (UPLO,N,NRHS,A,LDA, \mathbb{PIV ,B,LDB,W ORK,LW ORK,}}\mathbf{N},\textrm{L}
\mathbb{NFO)}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LD B,*),W ORK (*)
\mathbb{NTEGERN,NRHS,LDA,LDB,LW ORK,INFO}
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(})
SU BROUT\mathbb{NE CSYSV_64 (UPLO,N,NRHS,A,LDA,\mathbb{PIV ,B,LDB,W ORK,LW ORK,}}\mathbf{~}\mathrm{ , N,}
INFO)
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LD B,*),W ORK (*)
\mathbb{NTEGER*8N,NRHS,LDA,LDB,LW ORK,INFO}
\mathbb{NTEGER*8 \mathbb{PIV (*)}}\mathbf{*})

```

\section*{F95 INTERFACE}

SU BROUTINE SYSV (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I}, B,[L D B],[W\) ORK], [LW ORK], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::UPLO
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,B
\(\mathbb{N}\) TEGER ::N,NRHS,LDA,LDB,LW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}\)
SU BROUTINE SYSV_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I}, B,[L D B],[W\) ORK],
\[
[L W \text { ORK ], [ } \mathbb{N} F O] \text { ) }
\]

CHARACTER (LEN=1) ::UPLO
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::A, B
\(\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D B, L W O R K, \mathbb{N F O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I}\)

\section*{C INTERFACE}
\#include <sunperfh>
void csysv (charuplo, intn, intnrhs, com plex *a, int lda, int *ipiv, com plex *b, int ldb, int *info);
void csysv_64 (charuplo, long n, long nrhs, com plex *a, long lda, long *ịív, complex *b, long ldb, long *info);

\section*{PURPOSE}
csysv com putes the solution to a com plex system of linear equations
\(A * X=B\), where \(A\) is an \(N\) boy \(-N\) sym m etric \(m\) atrix and \(X\) and \(B\) are \(N\)-by \(-N\) RH S \(m\) atrices.

The diagonalpivoting \(m\) ethod is used to factorA as
\(A=U * D * U * T\), if \(U P L O=U\) ', or
\(A=L * D * L * *\), if \(U P L O=L '\),
where \(U\) (orL) is a productof perm utation and unit upper (low er) triangular \(m\) atrices, and \(D\) is sym \(m\) etric and block diagonalw ith 1 -by -1 and 2 -by -2 diagonalblocks. The factored form of \(A\) is then used to solve the system of equations \(\mathrm{A} * \mathrm{X}=\mathrm{B}\).

\section*{ARGUMENTS}

\section*{UPLO (input)}
= U : U pper triangle ofA is stored;
= L': Low ertriangle ofA is stored.
N (input) The num ber of linear equations, i.e., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS \(>=0\).

A (input/output)

On entry, the symm etric \(m\) atrix \(A\). If \(\mathrm{PLO}=\mathrm{U}\) ', the leading N -by -N upper triangularpart of A contains the upper triangular part of the \(m\) atrix \(A\), and the strictly low er triangularpart of A is not referenced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading N -by -N low er triangularpart of A contains the low ertriangularpart of the \(m\) atrix \(A\), and the strictly upper triangular partofA is not referenced.

On exit, if \(\mathbb{N F O}=0\), the block diagonalm atrix \(D\) and the multipliers used to obtain the factor \(U\) or L from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) as com puted by CSY TRF.
LDA (input)
The leading dim ension of the array A. LDA >= \(\max (1, \mathbb{N})\).

IPIV (output)
D etails of the interchanges and the block structure ofD, as determ ined by CSY TRF. If \(\mathbb{P} \mathbb{I V}(k)>\) 0 , then row \(s\) and colum ns \(k\) and \(\mathbb{P} \mathbb{I V}(k)\) w ere interchanged, and \(\mathrm{D}(\mathrm{k}, \mathrm{k})\) is a 1-by-1 diagonalblock. If \(U P L O=U\) 'and \(\mathbb{P} \mathbb{I V}(k)=\mathbb{P} \mathbb{I}(k-1)<0\), then rows and colum ns \(k-1\) and \(-\mathbb{P} I V(k)\) were interchanged and \(D(k-1 k, k-1 k)\) is a \(2-b y-2\) diagonal block. IfU PLO = L'and \(\mathbb{P} \mathbb{I V}(k)=\mathbb{P} \mathbb{I V}(k+1)<0\), then row \(s\) and colum ns \(k+1\) and - \(\mathbb{P}\) IV ( \(k\) ) w ere interchanged and \(D(k: k+1, k \cdot k+1)\) is a \(2-b y-2\) diagonal block.

B (input/output)
On entry, the N -by-NRH S righthand side m atrix B . On exit, if \(\mathbb{N} F O=0\), the \(N\)-by \(-N\) RH \(S\) solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray B. LD B \(>=\) \(\max (1, N)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F}\) F \(=0, \mathrm{~W}\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The length ofW ORK. LW ORK >=1, and forbestperform ance LW ORK \(>=N * N B\), where \(N B\) is the optim al blocksize forCSYTRF.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of
the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i, D(i, i)\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, so the solution could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
csysvx - use the diagonalpivoting factorization to com pute the solution to a com plex system of linearequations A * \(\mathrm{X}=\) B,

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE CSYSVX (FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,B,}}\mathbf{N},\textrm{N},\textrm{N}
LDB,X,LDX,RCOND,FERR,BERR,W ORK,LDW ORK,W ORK2,INFO)
CHARACTER * 1FACT,UPLO
COM PLEX A (LDA,*),AF (LDAF,*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDA,LDAF,LDB,LDX,LDW ORK,INFO
INTEGER \mathbb{PIVOT (*)}
REAL RCOND
REAL FERR(*),BERR(*),WORK2 (*)
SU BROUT\mathbb{NE CSYSVX_64(FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,}}\mathbf{N},\textrm{N},\textrm{N},
B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,LDW ORK,W ORK 2,\mathbb{NFO)}

```

CHARACTER * 1 FACT,UPLO
COM PLEX A (LDA,*),AF (LDAF,*), B (LDB,*), X (LDX,*), W ORK (*)
\(\mathbb{N} T E G E R * 8 N, N R H S, L D A, L D A F, L D B, L D X, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER * \(8 \mathbb{P} \mathbb{I V O T}\) ( \({ }^{*}\) )
REAL RCOND
REAL FERR ( \({ }^{*}\) ), BERR (*), \(\mathrm{W} O \operatorname{OR} 2\left({ }^{*}\right)\)

\section*{F95 INTERFACE}

SU BROUTINE SYSVX \(\mathbb{F} A C T, U P L O, \mathbb{N}], \mathbb{N R H S}], A,[L D A], A F,[L D A F]\), \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LD} \mathrm{B}], \mathrm{X},[\operatorname{LD} \mathrm{X}], R C O N D, F E R R, B E R R,[\mathbb{O}\) ORK], [LDW ORK], [W ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1)::FACT,UPLO

COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,AF,B,X
\(\mathbb{N}\) TEGER ::N,NRHS,LDA,LDAF,LDB,LDX,LDW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{V} O T\)
REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

SU BROUTINE SYSVX_64 (FACT,UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F]\), IPIVOT,B,[LDB],X, [LDX],RCOND,FERR,BERR, [W ORK], [LDW ORK], [W ORK2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::FACT,UPLO
COMPLEX,D \(\mathbb{I M}\) ENSION (:) ::W ORK
COMPLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,AF,B,X
\(\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D A F, L D B, L D X, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V} O T\)
REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void csysvx (char fact, char uplo, intn, int nrhs, com plex
*a, int lda, com plex *af, int ldaf, int *ipivot, com plex *b, int ldb, com plex *x, int ldx, float *rcond, float * ferr, float *berr, int *info);
void csysvx_64 (char fact, charuplo, long n, long nrhs, com plex *a, long lda, com plex *af, long ldaf, long *ịíivot, com plex *b, long ldb, com plex *x, long ldx, float * rcond, float * ferr, float *berr, long *info);

\section*{PURPOSE}
csysvx uses the diagonal pivoting factorization to com pute the solution to a com plex system of linear equations A * X = \(B\), where \(A\) is an \(N\) by \(-N\) sym \(m\) etric \(m\) atrix and \(X\) and \(B\) are \(N-\) by-N RH S m atrices.

Errorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT \(=\mathrm{N}\) ', the diagonalpivoting m ethod is used to factorA.
The form of the factorization is
\[
A=U * D * U * * T \text {, if } U P L O=U \text { ', or }
\]
\[
A=L * D * L \star * T \text {, if } U P L O=L '
\]
where \(U\) (orL) is a productofperm utation and unitupper (low er)
triangularm atrioes, and D is sym \(m\) etric and block diagonalw ith

1-by-1 and 2-by-2 diagonalblocks.
2. If som eD \((i, i)=0\), so thatD is exactly singular, then the routine
retums w ith \(\mathbb{N N F O}=\) i. O therw ise, the factored form of \(A\) is used
to estim ate the condition num ber of the \(m\) atrix A. If the reciprocal of the condition num ber is less than \(m\) achine precision,
\(\mathbb{N} F O=N+1\) is retumed as a \(w\) aming, but the routine stillgoes on
to solve for \(X\) and com pute error bounds as described below.
3.The system ofequations is solved forX using the factored form of A.
4. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for it.

\section*{ARGUMENTS}

\section*{FACT (input)}

Specifies w hether ornot the factored form of \(A\) has been supplied on entry.\(=F\) ': On entry, A F and \(\mathbb{P} \mathbb{I V O T}\) contain the factored form of A. A, AF and \(\mathbb{P} \mathbb{I V O T} w\) ill not be modified. \(=\mathrm{N}\) ': The m atrix A w illibe copied to AF and factored.

UPLO (input)
\(=\mathrm{U}\) : U ppertriangle of A is stored;
\(=\mathbb{L}\) ': Low er triangle ofA is stored.

N (input) The num ber of linearequations, i.e., the order of them atrix A. \(N>=0\).

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of the m atrioes B and X. NRH S >=0.

A (input) The symm etricm atrix \(A\). If \(U P L O=U\) ', the leading \(N\)-by -N upper triangularpart of \(A\) contains the upper triangularpart of the \(m\) atrix \(A\), and the strictly low ertriangularpartofA is not referenced. If UPLO = L', the leading N -by-N lower triangularpartofA contains the low er triangular partof the \(m\) atrix \(A\), and the strictly upper triangularpart ofA is not referenced.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

AF (input/output)
If \(F A C T=F '\), then \(A F\) is an input argum ent and on entry contains the block diagonalm atrix D and the m ultipliers used to obtain the factor \(U\) orL from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) as com puted by C SY TRF .

If FA C T = N ', then AF is an output argum ent and on exit retums the block diagonalm atrix D and them ultipliers used to obtain the factorU or L from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\) L *D *L**T.

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, N)\).

IPIVOT (inputoroutput)
If \(F A C T=F '\), then \(\mathbb{P I V O T}\) is an input argum ent and on entry contains details of the interchanges and the block structure of \(D\), as determ ined by CSY TRF. If \(\mathbb{P}\) IV OT ( \(k\) ) > 0, then row sand colum nsk and \(\mathbb{P} \mathbb{I V O T}(k)\) w ere interchanged and \(D(k, k)\) is a 1 -by-1 diagonal block. If UPLO \(=U^{\prime}\) and \(\mathbb{P} \mathbb{I V} O T(k)=\mathbb{P} \mathbb{I V} O T(k-1)<0\), then row sand colum ns \(\mathrm{k}-1\) and \(-\mathbb{P} \mathbb{I V}\) O (k) were interchanged and D ( \(\mathrm{k}-\) \(1 \mathrm{k}, \mathrm{k}-1 \mathrm{k})\) is a \(2-\) by-2 diagonalblock. IfU PLO \(=\) L 'and \(\mathbb{P} \mathbb{I V}\) OT \((k)=\mathbb{P} \mathbb{I V}\) OT \((k+1)<0\), then row \(s\) and colum nsk+1 and - \(\mathbb{P}\) IV OT (k) were interchanged and D \((k: k+1, k k+1)\) is a \(2-b y-2\) diagonalblock.

IfFACT = \(N\) ', then \(\mathbb{P} \mathbb{I V O T}\) is an output argum ent and on exit contains details of the interchanges and the block structure of D, as determ ined by CSY TRF.
\(B\) (input) The \(N\)-by-N RH S righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=N+1\), the \(N\) boy \(-N\) RH S solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the array \(\mathrm{X} . \mathrm{LDX}>=\) \(\max (1, N)\).

\section*{RCOND (output)}

The estim ate of the reciprocal condition num ber of
the matrix A. IfRCOND is less than the m achine precision (in particular, if RCOND \(=0\) ), the \(m\) atrix is singular to \(w\) orking precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO > 0 .

FERR (output)
The estim ated forw ard errorbound for each solution vectorX ( 1 ) the \(j\) th colum \(n\) of the solution \(m\) atrix \(X)\). If \(X T R U E\) is the true solution comesponding to \(\mathrm{X}(\mathcal{1}), \mathrm{FERR}(\mathcal{)}\) is an estim ated upperbound forthe \(m\) agnitude of the largest ele\(m\) entin ( \(X(\mathcal{)})-X\) TRU \(E\) ) divided by the \(m\) agnitude of the largestelem entin X ( 7 ) . The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slightoverestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vectorX (i) (ie., the sm allest relative change in any elem entof \(A\) orB thatm akes \(X\) ( 7 ) an exactsolution).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LDW ORK.

LD W ORK (input)
The length of W ORK. LDW ORK \(>=2 \star N\), and for best perform ance LDW ORK \(>=N * N B\), where \(N B\) is the optim alblocksize forC SY TRF .

If LD W ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK aray, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA .

W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=i\), and \(i\) is
<= N : D (i,i) is exactly zero. The factorization has been completed but the factorD is exactly singular, so the solution and error bounds could not be com puted. RCOND \(=0\) is retumed. \(=N+1\) : D is nonsingular, butRCOND is less than machine precision, \(m\) eaning that the \(m\) atrix is singular to w orking precision. N evertheless, the solution and enror bounds are com puted because there are a num ber of situationsw here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
csytf2 -com pute the factorization of a com plex sym m etric \(m\) atrix A using the Bunch \(-K\) aufm an diagonalpivoting \(m\) ethod

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
COM PLEX A (LDA,*)
\mathbb{NTEGERN,LDA,}\mathbb{NNFO}
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(*)}

```

```

CHARACTER * 1 UPLO
COM PLEX A (LDA,*)
INTEGER*8N,LDA,INFO
\mathbb{NTEGER*8 \mathbb{PIV (*)}}\mathbf{*}\mathrm{ ( }

```

\section*{F95 INTERFACE}
```

SU BROUTINE SY TF2 (UPLO, $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I}$, $[\mathbb{N} F O])$
CHARACTER (LEN=1) ::UPLO
COM PLEX,D $\mathbb{I M}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER ::N,LDA, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}$
SU BROUTINE SY TF2_64 (UPLO, $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V},[\mathbb{N} F O])$
CHARACTER (LEN=1)::UPLO
COM PLEX,D $\mathbb{I}$ ENSION (:,:) ::A

```
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} A, \mathbb{N}\) FO
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I}\)

\section*{C INTERFACE}
\#include <sunperfh>
void csytf2 (charuplo, int n, complex *a, int lda, int
*ípİ, int *info);
void csytf2_64 (charuplo, long n, com plex *a, long lda, long *ịiv, long *info);

\section*{PURPOSE}
csytf2 com putes the factorization of a com plex symm etric \(m\) atrix A using the Bunch \(K\) aufm an diagonalpivoting \(m\) ethod:
\[
A=U * D * U{ }^{\prime} \text { or } A=L * D * L^{\prime}
\]
where U (orL) is a productofperm utation and unit upper (low er) triangularm atriges, U 'is the transpose of U , and D is sym \(m\) etric and block diagonalw ith 1 -by-1 and 2 -by-2 diagonalblocks.

This is the unblocked version of the algorithm, calling Level2 BLAS.

\section*{ARGUMENTS}

\section*{UPLO (input)}

Specifies w hether the upper or low er triangular part of the sym m etricm atrix A is stored:
\(=\mathrm{U}\) ': Upper triangular
\(=\mathrm{L}\) ': Low ertriangular

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the sym m etric m atrix A. If \(\mathrm{ULO}=\mathrm{U}\) ', the leading \(n-b y-n\) uppertriangularpartofA contains the upper triangular part of the \(m\) atrix \(A\), and the strictly low er triangularpart of A is not referenced. If \(\mathrm{UPLO}=\mathbb{L}\) ', the leading \(\mathrm{n}-\mathrm{boy}-\mathrm{n}\) low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of A is not referenced.

On exit, the block diagonalm atrix D and the mul
tipliers used to obtain the factorU orl (see below for further details).

LDA (input)
The leading dim ension of the anray A. LD A >= \(\max (1, \mathbb{N})\).

IPIV (output)
D etails of the interchanges and the block structure ofD. If \(\mathbb{P} \mathbb{I V}(k)>0\), then row \(s\) and colum ns \(k\) and \(\mathbb{P} \mathbb{I V}(k)\) were interchanged and \(D(k, k)\) is a 1 -by-1 diagonalblock. IfUPLO \(=U\) 'and \(\mathbb{P} \mathbb{I V}(k)\) \(=\mathbb{P} \mathbb{V}(k-1)<0\), then row \(s\) and colmm ns \(k-1\) and \(-\mathbb{P} \mathbb{V}(k)\) w ere interchanged and \(D(k-1 * k, k-1 *)\) is a 2 -by-2 diagonalblock. IfUPLO \(=\mathbb{L}\) 'and \(\mathbb{P} \mathbb{I V}(k)\)
\(=\mathbb{P} \mathbb{I}(k+1)<0\), then row sand colum ns \(k+1\) and \(-\mathbb{P} \mathbb{V}(k)\) w ere interchanged and \(D(k, k+1, k \mathrm{k}+1)\) is a 2-by-2 diagonalblock.
\(\mathbb{N F O}\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\mathrm{k}\), the k -th argum enthad an illegalvalue \(>0:\) if \(\mathbb{N} F O=k, D(k, k)\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, and division by zero w illoccur if it is used to solve a system of equations.

\section*{FURTHER DETAILS}

\section*{1-96 -B ased on m odifications by J.Lew is, Boeing Com puter}

\section*{Services}

Com pany
If U PLO \(=\mathrm{U}\) ', then \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}\) ', where
\(U=P(n) \star U(n) * \ldots * P(k) U(k) * \ldots\),
i.e., \(U\) is a productof term \(S P(k) * U(k)\), where \(k\) decreases from \(n\) to 1 in steps of 1 or2, and \(D\) is ablock diagonal \(m\) atrix \(w\) ith 1 -by -1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{V}(k)\), and \(U(k)\) is a unituppertriangularm atrix, such that if the diagonal block \(D(k)\) is of orders ( \(s=1\) or 2 ), then
\[
\begin{aligned}
& \text { ( I v 0 ) k-s } \\
& \mathrm{U}(\mathrm{k})=(0 \mathrm{I} 0) \mathrm{s} \\
& \text { ( } 0 \text { O I ) } \mathrm{n}-\mathrm{k} \\
& \mathrm{k}-\mathrm{s} \mathrm{~s} \mathrm{n}-\mathrm{k}
\end{aligned}
\]

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(1 k-\)
\(1, k)\). If \(s=2\), the upper triangle ofD ( \(k\) ) overw rites \(A(k-\) \(1, k-1)\), A \((k-1, k)\), and \(A(k, k)\), and \(v\) overw rites A ( 1 k- \(2, k-\) \(1 \mathrm{k})\).

If \(\operatorname{PLO}=\mathrm{L}\) ', then \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L}\) ', where
\(\mathrm{L}=\mathrm{P}(1) \star \mathrm{L}(1){ }^{*} \ldots * \mathrm{P}(k) \star \mathrm{L}(k)^{*} \ldots\),
i.e., \(L\) is a product of term \(S P(k) * L(k)\), where \(k\) increases from 1 to \(n\) in steps of 1 or 2, and \(D\) is a block diagonal \(m\) atrix \(w\) ith 1 -by -1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{V}(k)\), and \(L(k)\) is a unit low ertriangularm atrix, such that if the diagonal
block D (k) is of orders ( \(s=1\) or2), then
\[
\begin{aligned}
& \text { ( } \left.\begin{array}{llll}
\mathrm{I} & 0 & 0
\end{array}\right) \mathrm{k}-1 \\
& L(k)=\left(\begin{array}{lll}
0 & I & 0
\end{array}\right) s \\
& \text { ( } 0 \text { v I ) } n-k-s+1 \\
& \text { k-1 s n-k-s+1 }
\end{aligned}
\]

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites
\(A(k+1 n, k)\). If \(s=2\), the low er triangle ofD ( \(k\) ) overw rites A \((k, k), A(k+1, k)\), and \(A(k+1, k+1)\), and \(v\) overw rites A \((k+2 m, k k+1)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
csytrf-com pute the factorization of a com plex sym m etric \(m\) atrix A using the Bunch \(-K\) aufm an diagonalpivoting \(m\) ethod

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
COM PLEX A (LDA,\star),W ORK (*)
INTEGERN,LDA,LDWORK,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}

```

```

CHARACTER * 1 UPLO
COM PLEX A (LDA,*),W ORK (*)
INTEGER*8N,LDA,LDW ORK, INFO
INTEGER*8 \mathbb{PIVOT (*)}

```

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E \operatorname{SY} \operatorname{TRF}(\mathrm{UPLO}, \mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V O T}, \mathbb{W}\) ORK ], [LDW ORK ], [ \(\mathbb{N F O}])\)
CHARACTER (LEN=1) ::UPLO
COMPLEX,D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : ::A
\(\mathbb{N} T E G E R:: N, L D A, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{V} O T\)
SU BROUTINE SYTRF_64 (UPLO, \(\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V O T},\left[\begin{array}{l}\text { O ORK }],[L D W ~ O R K], ~\end{array}\right.\) [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO
COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R(8):: N, L D A, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T\)

\section*{C INTERFACE}
\#include <sunperfh>
void csytrf(charuplo, int \(n\), com plex *a, int lda, int *ịívot, int*info);
void csytrf_64 (charuplo, long n, com plex *a, long lda, long *ịíivot, long *info);

\section*{PURPOSE}
csytrf com putes the factorization of a com plex symm etric \(m\) atrix A using the B unch-K aufm an diagonalpivoting \(m\) ethod. The form of the factorization is
\[
\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T} \text { or } \mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}
\]
where \(U\) (orL) is a productof perm utation and unit upper (low er) triangular \(m\) atrices, and \(D\) is sym \(m\) etric and block diagonalw ith w th 1 -by-1 and 2 -by- 2 diagonalblocks.

This is the blocked version of the algorithm, calling Level 3 BLAS.

\section*{ARGUMENTS}

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input/output)
O n entry, the sym m etric m atrix A. If UPLO = U', the leading N -by -N uppertriangularpartofA contains the upper triangular part of the \(m\) atrix \(A\), and the strictly low ertriangularpart of A is not referenced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading N -by -N low er triangularpart ofA contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpartofA is not referenced.

On exit, the block diagonalm atrix D and the multipliers used to obtain the factorU orL (see below for further details).

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

\section*{IPIVOT (output)}

D etails of the interchanges and the block structure of D. If \(\mathbb{P I V O T}(k)>0\), then row sand columnsk and \(\mathbb{P I V O T}(k)\) were interchanged and \(D(k, k)\) is a \(1-b y-1\) diagonalblock. If \(U P L O=U^{\prime}\) and \(\mathbb{P} \mathbb{V} O T(k)=\mathbb{P} \mathbb{V} O T(k-1)<0\), then row \(s\) and colum ns \(k-1\) and - \(\mathbb{P I V O T}(k)\) were interchanged and D ( \(k-1 * k, k-1 k)\) is a \(2-b y-2\) diagonal block. If UPLO \(=\mathrm{L}\) 'and \(\mathbb{P I V O T}(k)=\mathbb{P} \mathbb{I V O T}(k+1)<0\), then row sand colum ns \(k+1\) and \(-\mathbb{P} \mathbb{V} O T(k)\) were interchanged and \(D(k k+1, k k+1)\) is a \(2-b y-2\) diagonal block.

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, \mathrm{~W}\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length ofW ORK. LDW ORK >=1. Forbestperfor\(m\) ance LDW ORK >=N *NB, where NB is the block size retumed by \(\amalg A E N V\).

IfLDW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfinlexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{D}(\mathrm{i}, \mathrm{i})\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, and division by zero w illoccur if it is used to solve a system of equations.

\section*{FURTHER DETAILS}

If \(\mathrm{ULO}=\mathrm{U}\) ', then \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}\) ', where
\(U=P(n) \star U(n)^{\star} \ldots{ }^{\star} P(k) U(k)^{\star} \ldots\),
ie., \(U\) is a product ofterm \(\operatorname{sP}(k) * U(k)\), where \(k\) decreases from \(n\) to 1 in steps of 1 or 2 , and \(D\) is a block diagonal \(m\) atrix w ith 1 -by-1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I V O T}(\mathrm{k})\), and \(\mathrm{U}(\mathrm{k})\) is a unituppertriangularm atrix, such that if the diagonal block D (k) is of orders ( \(s=1\) or 2 ), then
\[
\begin{aligned}
& \text { ( I v 0 ) k-s } \\
& U(k)=\left(\begin{array}{lll}
0 & I
\end{array}\right) s \\
& \text { ( } 000 \text { I ) n-k } \\
& \mathrm{k}-\mathrm{s} \text { s n-k }
\end{aligned}
\]

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(1 k-\) \(1, k\) ). If \(s=2\), the upper triangle ofD \((k)\) overw rites \(A(k-\) \(1, k-1), A(k-1, k)\), and \(A(k, k)\), and \(V\) overw rites \(A(1 k-2, k-\) \(1 \mathrm{k})\).

If \(\mathrm{UPLO}=\mathrm{L}\) ', then \(A=\mathrm{L} * \mathrm{D} * \mathrm{~L}\) ', where
\(L=P(1) \star L(1)^{\star} \ldots * P(k) \star L(k) * \ldots\)
ie., \(L\) is a productofterm \(s P(k) * L(k)\), where \(k\) increases
from 1 to n in steps of 1 or 2 , and D is a block diagonal \(m\) atrix \(w\) th 1 -by -1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I V O T}(k)\), and \(L(k)\) is a unitlow ertriangularm atrix, such that if the diagonal
block \(D(k)\) is of orders ( \(s=1\) or 2 ), then
\[
\begin{aligned}
& \text { ( } \left.\begin{array}{llll}
I & 0 & 0
\end{array}\right) k-1 \\
& L(k)=\left(\begin{array}{lll}
0 & I & 0
\end{array}\right) s \\
& \text { ( } 0 \mathrm{~V} \text { I ) } \mathrm{n}-\mathrm{k}-\mathrm{s}+1 \\
& \mathrm{k}-1 \text { s } \mathrm{n}-\mathrm{k}-\mathrm{s}+1
\end{aligned}
\]

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(k+1 m, k)\). If \(s=2\), the low ertriangle ofD \((k)\) overw rites \(A(k, k), A(k+1, k)\), and \(A(k+1, k+1)\), and \(v\) overw rites A \((k+2 m, k: k+1)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
csytri-com pute the inverse of a com plex sym \(m\) etric indefinte \(m\) atrix \(A\) using the factorization \(A=U * D * U * * T\) orA \(=\) L*D *L**T com puted by CSY TRF

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
COM PLEX A (LDA,*),W ORK (*)
NNTEGER N,LDA,}\mathbb{N}F
INTEGER \mathbb{PIVOT(*)}

```

```

CHARACTER * 1 UPLO
COM PLEX A (LDA,*),W ORK (*)
INTEGER*8N,LDA,INFO
INTEGER*8 \mathbb{PIVOT (*)}

```
F95 INTERFACE
    SU BROUTINE SYTRI(UPLO, \(\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V O T},[\mathbb{N} O R K],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO
    COM PLEX,D IM ENSION (:) ::W ORK
    COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N}\) TEGER ::N,LDA, \(\mathbb{N} F O\)
    \(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
    SU BROUTINE SYTRI_64 (UPLO, N ],A, [LDA ], \(\mathbb{P} \mathbb{I V O T},[\mathbb{W} O R K],[\mathbb{N F O}])\)
    CHARACTER (LEN=1)::UPLO

COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{I}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER (8) :: N, LDA, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{V} O T\)

\section*{C INTERFACE}
\#include <sunperfh>
void csytri(charuplo, int n, com plex *a, int lda, int *ipivot, int*info);
void csytri_ 64 (charuple, long n, com plex *a, long lda, long *ịívot, long *info);

\section*{PURPOSE}
csytricom putes the inverse of a com plex sym m etric indefinite \(m\) atrix A using the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\) L*D *L**T com puted by CSY TRF .

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) ': U pper triangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\);
= L': Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\).
N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the block diagonalm atrix D and the m ultipliers used to obtain the factorU orL as com puted by CSY TRF.

On exit, if \(\mathbb{N F F O}=0\), the (sym metric) inverse of the original m atrix. If \(\mathrm{UPLO}=\mathrm{U}\) ', the upper triangularpart of the inverse is form ed and the partofA below the diagonal is not referenced; if UPLO = 'L' the lower triangular part of the inverse is form ed and the partofA above the diagonal is not referenced.

LD A (input)
The leading din ension of the aray A. LDA >= \(\max (1, N)\).
\(\mathbb{P I V O T}\) (input)
D etails of the interchanges and the block structure ofD as determ ined by CSY TRF.

W ORK (w orkspace)
dim ension ( 2 * N )
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i, D(i, i)=0\); the \(m\) atrix is singular and its inverse could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
csytre - solve a system of linearequations \(A * X=B\) with a complex symmetric \(m\) atrix \(A\) using the factorization \(A=\) \(\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by CSY TRF

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LD B,*)
INTEGERN,NRHS,LDA,LDB,INFO
INTEGER \mathbb{PIVOT (*)}
SU BROUT\mathbb{NE CSYTRS_64 (UPLO,N,NRHS,A,LDA, IPIVOT,B,LDB,INFO )}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),B (LD B,*)
INTEGER*8N,NRHS,LDA,LDB,INFO
INTEGER *8 \mathbb{PIVOT (*)}

```
F95 INTERFACE
    SU BROUTINE SYTRS (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I V} O T, B,[L D B],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO
    COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : A, B
    \(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, \mathbb{N} F O\)
    \(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION(:)::\mathbb {P}\mathbb {O}OT}\)
    SU BROUTINE SYTRS_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I V O T}, B,[L D B]\),
        [ \(\mathbb{N} F O\) ])
    CHARACTER (LEN=1)::UPLO

COM PLEX , D \(\mathbb{M}\) ENSION (: : : : : A , B
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDA,LDB, \(\mathbb{N}\) FO
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)

\section*{C INTERFACE}
\#include <sunperfh>
void csytrs (charuplo, intn, intnrhs, com plex *a, intlda, int *ipivot, com plex *b, int ldb, int *info);
void csytus_64 (charuplo, long n, long nrhs, com plex *a, long lda, long *ípívot, com plex *b, long lolo, long *info);

\section*{PURPOSE}
csytrs solves a system of linearequations \(A * X=B\) w ith \(a\) complex symm etric \(m\) atrix \(A\) using the factorization \(A=\) \(\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by CSY TRF.

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) : : U ppertriangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\);
\(=\mathrm{L}^{\prime}:\) Low er triangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\).

N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS \(>=0\).

A (input) The block diagonalm atrix D and the m ultipliers used to obtain the factorU orL as com puted by CSYTRF.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

IPIVOT (input)
D etails of the interchanges and the block structure ofD as determ ined by CSY TRF.

B (input/output)

O \(n\) entry, the right hand side \(m\) atrix \(B\). On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctbcon -estim ate the reciprocal of the condition num ber of a triangular band \(m\) atrix \(A\), in etherthe 1 -norm orthe infinity-norm

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NECTBCON NORM,UPLO,D IAG,N,KD,A,LDA,RCOND,W ORK,}
W ORK2,\mathbb{NFO)}
CHARACTER * 1 NORM,UPLO,DIAG
COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGERN,KD,LDA,}\mathbb{NFO}
REAL RCOND
REAL W ORK2 (*)
SUBROUT\mathbb{NECTBCON_64 NORM,UPLO,D IAG,N,KD,A,LDA,RCOND,W ORK,}
WORK2, \mathbb{NFO)}
CHARACTER * 1NORM,UPLO,DIAG
COM PLEX A (LDA,*),W ORK (*)
INTEGER*8N,KD,LDA,INFO
REALRCOND
REAL W ORK 2 (*)

```

\section*{F95 INTERFACE}
```

SU BROUT $\mathbb{N} E T B C O N \mathbb{N} O R M, U P L O, D \mathbb{I} G, \mathbb{N}], K D, A,[L D A], R C O N D,[W O R K]$, [ W ORK 2], [ $\mathbb{N} F \mathrm{~F}$ ])
CHARACTER (LEN=1) ::NORM,UPLO,D IAG
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (: : : : : : A
$\mathbb{N} T E G E R:: N, K D, L D A, \mathbb{N} F O$

```

REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::W ORK2
SU BROUTINE TBCON_64 NORM, UPLO,D \(\mathbb{A} G, \mathbb{N}], K D, A,[L D A], R C O N D\), [W ORK], [W ORK2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::NORM,UPLO,DIAG
COMPLEX,D IM ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R(8):: N, K D, L D A, \mathbb{N} F O\)
REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void ctbcon (charnorm , charuplo, chardiag, intn, int kd, com plex *a, int lda, float *rcond, int *info);
void ctbcon_64 (charnorm , charuplo, chardiag, long n, long kd, com plex *a, long lda, float *rcond, long *info);

\section*{PURPOSE}
ctbcon estim ates the reciprocal of the condition num ber of a triangular band matrix \(A\), in either the 1 -norm or the infinity-norm.

The norm ofA is com puted and an estim ate is obtained for norm (inv (A )), then the reciprocal of the condition num ber is com puted as
```

RCOND = 1/(norm (A)* norm (inv(A))).

```

\section*{ARGUMENTS}
```

NORM (input)
Specifies w hether the 1-nom condition num ber or
the infinity-norm condition num ber is required:
= I'or O ': 1-nom;
= I': Infinity-nom .
UPLO (input)
= U ': A is uppertriangular;
= LL':A is low ertriangular.

```
D IA G (input)
\(=\mathrm{N}\) ': A is non-unit triangular;
\(=\mathrm{U}\) ': A is unit triangular.
N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).
KD (input)
The num berof superdiagonals or subdiagonals of the triangularband \(m\) atrix A. KD \(>=0\).

A (input) The upper or low er triangular band \(m\) atrix A, stored in the firstkd+1 row sof the amay. The \(j\) th column ofA is stored in the \(j\) th column of the anay A as follow s: if UPLO = U',A (kd+1+i\(j, j)=A(i, 7)\) for \(\max (1, j k d)<=i<=j\); f UPLO \(=\) L', A \((1+i-j\rangle)=A(i, 7)\) for \(j<=i<=m\) in \((n, j+k d)\). IfD IA G = U', the diagonalelem ents of A are not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the anay A. LDA >= K D +1 .

RCOND (output)
The reciprocal of the condition number of the \(m\) atrix \(A\), computed as RCOND \(=1 /(\) norm (A) * norm (inv (A))).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctom \(v\)-perform one of the \(m\) atrix-vectoroperations \(x:=\) \(A{ }^{*} x\), or \(x: A{ }^{*} x\), or \(x:=\operatorname{con} g\left(A^{\prime}\right)^{*} x\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CTBMV (UPLO,TRANSA,D IAG,N,K,A,LDA,Y, INCY)}
CHARACTER * 1 UPLO,TRANSA,DIAG
COM PLEX A (LDA,*),Y (*)
\mathbb{NTEGERN,K,LDA,INCY}
SU BROUT\mathbb{NECTBM V_64 (UPLO,TRANSA,D IAG,N,K,A,LDA,Y,INCY)}
CHARACTER * 1UPLO,TRANSA,DIAG
COM PLEX A (LDA,*),Y (*)
\mathbb{NTEGER*8N,K,LDA, NNCY}

```
F95 INTERFACE
    SU BROUTINE TBMV (UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{N}], K, A,[L D A], Y,[\mathbb{N C Y}])\)
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
    COMPLEX,D \(\mathbb{M}\) ENSION (:) ::Y
    COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N} T E G E R:: N, K, L D A, \mathbb{N C Y}\)
    SU BROUTINE TBM V_64 (UPLO, [TRANSA ],D \(\mathbb{A} G, \mathbb{N}], K, A,[L D A], Y\),
        [ \(\mathbb{N} C Y\) ])
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
    COMPLEX,D \(\mathbb{I M} E N S I O N(:):: Y\)
    COM PLEX,D \(\mathbb{M}\) ENSION (:r:) ::A
    \(\mathbb{N} T E G E R(8):: N, K, L D A, \mathbb{N} C Y\)

\section*{C INTERFACE}
\#include < sunperfh>
void ctbm v (charuplo, chartransa, chardiag, intn, int k, com plex *a, int lda, com plex *y, int incy);
void ctbm v_64 (charuplo, chartransa, char diag, long n, long k, com plex *a, long lda, com plex *y, long incy);

\section*{PURPOSE}
ctom \(v\) perform s one of the \(m\) atrix-vectoroperations \(x:=A * x\), or \(x:=A{ }^{*} x\), or \(x:=\infty \quad \dot{g}\left(A^{\prime}\right){ }^{*} x\) where \(x\) is an \(n\) elem ent vectorand \(A\) is an \(n\) by \(n\) unit, ornon-unit, upper or low er triangularband \(m\) atrix, \(w\) ith ( \(k+1\) ) diagonals.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies w hether the \(m\) atrix is an upper or low er triangularm atrix as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or \(\mathrm{L}^{\prime} \mathrm{A}\) is an upper triangular \(m\) atrix.

UPLO = L' or I' A is a lower triangular \(m\) atrix.

U nchanged on exit.
TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) 'or \(h^{\prime} \mathrm{x}:=\mathrm{A}{ }^{*} \mathrm{x}\).
TRANSA \(=\) T'ort' \(x:=A * x\).

TRANSA \(=\) C'ort \(^{\prime} \mathrm{x}:=\operatorname{conj} \mathrm{g}^{\prime}\left(\mathrm{A}^{\prime}\right){ }^{\star} \mathrm{x}\).
U nchanged on exit.
TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.

O n entry, D IA G specifies w hether ornotA is unit triangular as follow \(s\) :

D \(\mathbb{A G G}=\mathrm{U}\) 'or \(\mathrm{L}^{\prime} \mathrm{A}\) is assum ed to be unit triangular.
\(D \mathbb{A} G=N\) 'or \(h\) ' \(A\) is notassumed to be unit triangular.

U nchanged on exit.

N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.
\(K\) (input)
On entry w th UPLO \(=U\) 'or L ', K specifies the num ber of super-diagonals of them atrix \(A\). On entry w ith UPLO = L' or I', K specifies the num ber of sub-diagonals of the \(m\) atrix \(A . K>=0\). U nchanged on exit.

A (input)
Before entry w th UPLO = U 'or G ', the leading ( \(k+1\) ) by \(n\) part of the array A m ust contain the upper triangularband part of the \(m\) atrix of coefficients, supplied colum \(n\) by colum \(n\), w ith the leading diagonal of the \(m\) atrix in row ( \(k+1\) ) of the anay, the firstsuper-diagonal starting at position 2 in row \(k\), and so on. The top leftk by \(k\) triangle of the amay A is not referenced. The follow ing program segm entw ill transfer an upper triangular band \(m\) atrix from conventional full \(m\) atrix storage to band storage:
\[
\begin{aligned}
& \text { DO } 20, J=1, N \\
& \text { M }=\mathrm{K}+1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{M} A X(1, \mathrm{~J}-\mathrm{K}), \mathrm{J} \\
& \text { A }(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \text { CONTINE } \\
& 20 \text { CONTINUE }
\end{aligned}
\]

Before entry w ith UPLO = L 'or I', the leading ( \(k+1\) ) by \(n\) part of the amay A \(m\) ust contain the low er triangularband part of the \(m\) atrix of coefficients, supplied colum n by colum n, w th the leading diagonal of the \(m\) atrix in row 1 of the array, the firstsub-diagonal starting atposition 1 in row 2 , and so on. The bottom right \(k\) by \(k\) triangle of the amay A is not referenced. The follow ing program segm entw ill transfer a low er
triangular band \(m\) atrix from conventional full \(m\) atrix storage to band storage:
\[
\begin{aligned}
& \text { DO } 20, J=1, N \\
& \mathrm{M}=1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{J}, \mathrm{M} \mathbb{N}(\mathrm{~N}, \mathrm{~J}+\mathrm{K}) \\
& \mathrm{A}(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \operatorname{CONTINUE} \\
& 20 \mathrm{CONTINUE}
\end{aligned}
\]
\(N\) ote thatw hen D \(\mathbb{A} G=U\) 'or L 'the elem ents of the array A comesponding to the diagonalelem ents of the \(m\) atrix are not referenced, but are assum ed to be unity. U nchanged on exit.
LDA (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD A \(>=\) ( \(\mathrm{k}+1\) ). U nchanged on exit.

Y (input/output)
\((1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) must contain the \(n\) elem ent vectorx. On exit, \(Y\) is overw rilten \(w\) th the tranform ed vector \(x\).
\(\mathbb{N C Y}\) (input)
O n entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of. \(\mathbb{N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}

> ctbrfs -provide errorbounds and backw ard error estim ates for the solution to a system of linear equations w ith a tri-angularband coefficientm atrix

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CTBRFS (UPLO,TRANSA,D IAG,N,KD,NRHS,A,LDA,B,LDB,}
X,LDX,FERR,BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1UPLO,TRANSA,DIAG
COM PLEX A (LDA,*),B (LDB,*),X (LDX ,*),W ORK (*)
INTEGERN,KD,NRHS,LDA,LDB,LDX,}\mathbb{N}F
REAL FERR (*),BERR (*),W ORK2 (*)
SUBROUTINE CTBRFS_64 (UPLO,TRANSA,D IAG,N,KD,NRHS,A,LDA,B,
LDB,X,LDX,FERR,BERR,W ORK,W ORK2, INFO)
CHARACTER * 1UPLO,TRANSA,DIAG
COM PLEX A (LDA,*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGER*8N,KD,NRHS,LDA,LDB,LDX,INFO
REAL FERR (*),BERR (*),W ORK2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE TBRFS (UPLO, [TRANSA],D IAG, \(\mathbb{N}], K D,[N R H S], A,[L D A]\), B, [LDB], \(X,[L D X], F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
COMPLEX,D IM ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A, B, X
\(\mathbb{N} T E G E R:: N, K D, N R H S, L D A, L D B, L D X, \mathbb{N} F O\)
REAL,D IM ENSION (:) ::FERR,BERR,W ORK2

SU BROUTINE TBRFS_64 (UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{N}], K D, \mathbb{N R H S ] , A , [ L D A ] , ~}\) B, [LDB], \(X,[\operatorname{LDX}], F E R R, B E R R,[\mathbb{W} O R K],[W O R K 2],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO, TRANSA,D IA G
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : :: A, B, X
\(\mathbb{N} T E G E R(8):: N, K D, N R H S, L D A, L D B, L D X, \mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include < sunperfh>
void ctbrfs (char uplo, char transa, chardiag, int n, int kd, intnrhs, com plex *a, int lda, com plex *b, int ldb, com plex *x, int ldx, float *ferr, float *berr, int*info);
void ctbrfs_64 (charuplo, chartransa, char diag, long n, long kd, long nrhs, com plex *a, long lda, com plex *b, long ldb, com plex *x, long ldx, float *ferr, float *berr, long *info);

\section*{PURPOSE}
ctbrfs provides errorbounds and backw ard error estim ates forthe solution to a system of linearequations \(w\) th a triangularband coefficientm atrix.

The solution \(m\) atrix \(X\) m ustibe com puted by CTBTRS or some other \(m\) eans before entering this routine. CTBRFS does not do iterative refinem entbecause doing so cannotim prove the backw ard error.

\section*{ARGUMENTS}

\section*{UPLO (input)}
\(=\mathrm{U}\) : A is uppertriangular;
= L' ': A is low er triangular.
TRANSA (input)
Specifies the form of the system of equations:
\(=N\) : A * \(\mathrm{X}=\mathrm{B} \quad\) N \(\circ\) transpose)
\(=T: A * * T * X=B \quad(T\) ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose)
TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
\(=\mathrm{N}\) : A is non-unit triangular;
\(=\mathrm{U}:\) : A is unit triangular.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
\(K D\) (input)
The num berof superdiagonals or subdiagonals of the triangularband \(m\) atrix \(A . K D>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the m atrices B and X. NRHS \(>=0\).
A (input) The upper or low er triangular band \(m\) atrix \(A\), stored in the firstkd+1 row sof the amay. The \(j\) th column ofA is stored in the \(j\) th column of the anay A as follow s: if UPLO = U',A (kd+1+i\(j, 7)=A(i, j)\) for \(\max (1, j \mathrm{kd})<=i<=j\) if UPLO \(=\) L', A \((1+i-j\rangle)=A(i, 7)\) for \(j<=i<=m\) in \((n, j+k d)\). IfD \(\mathbb{I A}=U\) ', the diagonalelem ents of A are not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the anay A. LDA >= K D +1.
\(B\) (input) The righthand side m atrix \(B\).

LD B (input)
The leading dim ension of the array B . LD B >= \(\max (1, N)\).
\(X\) (input) The solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the anay X . LD X >= \(\max (1, N)\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X()\) (the \(j\) th column of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{H}), \operatorname{FERR}(1)\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{D})-X\) TRU \(E\) ) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

BERR (output)

The com ponentw ise relative backw ard error of each solution vector \(X(\mathcal{)}\) (ie., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{(}\) ) an exactsolution).

W ORK (w orkspace) dim ension \(\left(2{ }^{*} \mathrm{~N}\right.\) )
W ORK 2 (w orkspace) dim ension (N)
\(\mathbb{N F O}\) (output)
= 0 : successfiulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctbsv -solve one of the system sofequations \(A{ }^{*} x=b\), or
\(A^{*} \mathrm{x}=\mathrm{b}\), orcong \(\left(\mathrm{A}^{\prime}\right)^{*} \mathrm{x}=\mathrm{b}\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NECTBSV (UPLO,TRANSA,D IAG,N,K,A,LDA,Y, INCY)}
CHARACTER * 1 UPLO,TRANSA,DIAG
COM PLEX A (LDA,*),Y (*)
\mathbb{NTEGER N,K,LDA,INCY}

```

```

CHARACTER * 1UPLO,TRANSA,DIAG
COM PLEX A (LDA,*),Y (*)
INTEGER*8N,K,LDA, NNCY

```

\section*{F95 INTERFACE}

SU BROUTINE TBSV (UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{N}], K, A,[L D A], Y,[\mathbb{N C Y}])\)
CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
COMPLEX,D \(\mathbb{I}\) ENSION (:) ::Y
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: N, K, L D A, \mathbb{N C Y}\)
SUBROUTINE TBSV_64 (UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{N}], K, A,[L D A], Y\), [ \(\mathbb{N} C Y\) ])

CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
COMPLEX,D \(\mathbb{I M} E N S I O N(:):: Y\)
COM PLEX,D \(\mathbb{M}\) ENSION (:r:) ::A
\(\mathbb{N} T E G E R(8):: N, K, L D A, \mathbb{N} C Y\)

\section*{C INTERFACE}
\#include < sunperfh>
void ctbsv (char uplo, chartransa, chardiag, intn, int k, com plex *a, int lda, com plex *y, int incy);
void ctbsv_64 (charuplo, chartransa, char diag, long n, long k, complex *a, long lda, com plex *y, long incy);

\section*{PURPOSE}
ctbsv solves one of the system sofequations \(A * x=b\), or \(A^{*} \mathrm{x}=\mathrm{b}\), or cong ( \(\mathrm{A}^{\prime}\) ')* \(\mathrm{x}=\mathrm{b}\) where b and x are n elem ent vectors and \(A\) is an \(n\) by \(n\) unit, ornon-unit, upperor low er triangularband \(m\) atrix, \(w\) ith \((k+1)\) diagonals.

No test for singularity ornear-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the \(m\) atrix is an upper or low er triangularm atrix as follow \(s\) :
\(\mathrm{UPLO}=\mathrm{U}\) 'or \(\mathrm{L}^{\prime} \mathrm{A}\) is an upper triangular \(m\) atrix.

UPLO = L' or 1 ' A is a lower triangular \(m\) atrix.

U nchanged on exit.
TRANSA (input)
On entry, TRANSA specifies the equations to be solved as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} A * x=b\).
TRANSA \(=T^{\prime}\) or \(t^{\prime} A^{*} x=b\).
TRANSA \(=C^{\prime}\) ort' cong \(\left(A^{\prime}\right) \star \mathrm{x}=\mathrm{b}\).
U nchanged on exit.

TRANSA is defaulted to N 'forF95 \(\mathbb{I N}\) TERFACE.

D IA G (input)
On entry, D IA G specifies w hether ornotA is unit triangular as follow s:

D \(\mathbb{I}\) G \(=U\) 'or \(\mathrm{l}^{\prime} A\) is assum ed to be unit triangular.

D IA G = N 'or h' A is notassumed to be unit triangular.

U nchanged on exit.

N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.

K (input)
On entry with UPLO \(=U\) 'or U ', \(K\) specifies the num ber of super-diagonals of them atrix \(A\). On entry w th UPLO = L' or I', K specifies the num ber of sub-diagonals of the \(m\) atrix \(A . K>=0\). U nchanged on exit.

A (input)
Before entry w ith UPLO = U 'or L ', the leading ( \(k+1\) ) by \(n\) part of the array A m ust contain the upper triangularband part of the \(m\) atrix of coefficients, supplied colum \(n\) by colum \(n\), w ith the leading diagonal of the \(m\) atrix in row ( \(k+1\) ) of the aray, the firstsuper-diagonalstarting at position 2 in row \(k\), and so on. The top leftk by \(k\) triangle of the array A is not referenced. The follow ing program segm entw illtransfer an upper triangular band \(m\) atrix from conventional full \(m\) atrix storage to band storage:

D O 20, J=1, N
\(\mathrm{M}=\mathrm{K}+1-\mathrm{J}\)
DO \(10, I=M A X(1, J-K), J\)
\(A(M+I, J)=m \operatorname{atrix}(I, J)\)
10 CONTINUE
20 CONTINUE
Before entry w ith UPLO = L 'or 1', the leading ( \(k+1\) ) by \(n\) part of the array A m ustcontain the low er triangularband part of the \(m\) atrix of coefficients, supplied column by colum n, w th the
leading diagonal of the m atrix in row 1 of the array, the firstsub-diagonalstarting atposition
1 in row 2 , and so on. The bottom right \(k\) by \(k\) triangle of the array \(A\) is not referenced. The follow ing program segm entw ill transfer a low er triangular band \(m\) atrix from conventional full \(m\) atrix storage to band storage:
\[
\text { D O } 20, \mathrm{~J}=1, \mathrm{~N}
\]
\[
M=1-J
\]
\[
\text { DO } 10, I=J, M \mathbb{N}(N, J+K)
\]
\[
A(M+I, J)=m \operatorname{atrix}(I, J)
\]

10 CONTINUE
20 CONTINUE
\(N\) ote thatw hen D IA G = U 'or L'the elem ents of the amay A comesponding to the diagonalelem ents of the \(m\) atrix are not referenced, but are assum ed to be unity. U nchanged on exit.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD A >= ( \(\mathrm{k}+1\) ). U nchanged on exit.

Y (input/output)
\((1+(n-1) \star \operatorname{abs}(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) must contain the \(n\) elem ent right-hand side vectorb. On exit, \(Y\) is overw ritten \(w\) th the solution vectorx.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents ofY. \(\mathbb{N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctbtrs-solve a triangularsystem of the form \(A * X=B\), \(\mathrm{A} * * \mathrm{~T} * \mathrm{X}=\mathrm{B}\),orA \({ }^{*}{ }^{*} \mathrm{H} * \mathrm{X}=\mathrm{B}\),

\section*{SYNOPSIS}
```

SUBROUTINE CTBTRS (UPLO,TRANSA,DIAG,N,KD,NRHS,A,LDA,B,LDB,
\mathbb{NFO)}
CHARACTER * 1 UPLO,TRANSA,DIAG
COM PLEX A (LDA,*),B (LDB,*)
\mathbb{NTEGER N,KD,NRHS,LDA,LDB,INFO}
SU BROUTINE CTBTRS_64 (UPLO,TRANSA,D IAG ,N,KD,NRHS,A,LDA,B,
LD B, INFO)
CHARACTER * 1UPLO,TRANSA,DIAG
COM PLEX A (LDA,*),B (LD B,*)
INTEGER*8N,KD,NRHS,LDA,LDB,INFO

```

\section*{F95 INTERFACE}

SU BROUTINE TBTRS (UPLO,TRANSA,D IAG, \(\mathbb{N}], K D, \mathbb{N} R H S], A,[L D A], B\), [LDB], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,B
\(\mathbb{N} T E G E R:: N, K D, N R H S, L D A, L D B, \mathbb{N} F O\)
SU BROUTINE TBTRS_64 (UPLO,TRANSA, D IA G, \(\mathbb{N}], K D, \mathbb{N R H S}], A,[L D A]\), \(\mathrm{B},[\mathrm{LDB}],[\mathbb{N} \mathrm{FO}])\)

CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
COM PLEX,D \(\mathbb{I}\) ENSION (:,:) ::A, B
\(\mathbb{N}\) TEGER (8) ::N,KD,NRHS,LDA, LD B, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void ctbtrs (charuplo, char transa, chardiag, int n, int kd , intnins, com plex *a, int lda, com plex *b, int ldb, int *info);
void ctbtrs_64 (charuplo, chartransa, char diag, long n, long kd, long nihs, com plex *a, long lda, com plex *b, long ldb, long *info);

\section*{PURPOSE}
ctbtres solves a triangular system of the form
\(w\) here \(A\) is a triangularband \(m\) atrix of order \(N\), and \(B\) is an N -by-NRHS matrix. A check ism ade to verify thatA is nonsingular.

\section*{ARGUMENTS}

UPLO (input)
= U ': A is upper triangular;
= LI': A is low er triangular.

TRANSA (input)
Specifies the form of the system of equations:
\(=N\) : A * \(\mathrm{X}=\mathrm{B} \quad\) N o transpose)
\(=T: A * * T * X=B \quad(T\) ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose)

D IA G (input)
= \(N\) ': A is non-unit triangular;
\(=\mathrm{U}\) : A is unit triangular.

N (input) The order of the matrix A. \(\mathrm{N}>=0\).
KD (input)
The num berof superdiagonals or subdiagonals of the triangularband \(m\) atrix A. KD \(>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS \(>=0\).

A (input) The upper or low er triangular band \(m\) atrix A, stored in the first kd+1 row sofA. The jth column ofA is stored in the \(j\) th column of the array A as follow s: if UPLO = U',A (kd+1+i-j)
\(=A(i, j)\) form ax \((1, j \mathrm{jkd})<=\mathrm{i}<=\dot{j}\) if UPLO \(=\mathrm{L}^{\prime}\), A \((1+i-j-j)=A(i, j)\) for \(j=i<=m\) in \((n, j+k d)\). If
\(D \mathbb{I A}=U\) ', the diagonalelem ents of \(A\) are not referenced and are assum ed to be 1 .

LDA (input)
The leading dim ension of the aray A. LDA >= K D +1 .

B (input/output)
On entry, the righthand side m atrix B. On exit, if \(\mathbb{N} F O=0\), the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array \(B\). LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the \(i\)-th diagonalelem ent of \(A\)
is zero, indicating that the \(m\) atrix is singular and the solutions X have notbeen com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctgevc - com pute som e orallof the right and/or left generalized eigenvectors of a pair of com plex upper triangular \(m\) atrices ( \(A, B\) )

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NECTGEVC (SDE,HOWMNY,SELECT,N,A,LDA,B,LDB,VL,LDVL,}
VR,LDVR,MM,M,W ORK,RW ORK,INFO)
CHARACTER * 1SDEE,HOW MNY
COM PLEX A (LDA,*),B (LDB,*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
INTEGERN,LDA,LDB,LDVL,LDVR,MM,M,INFO
LOG ICAL SELECT (*)
REAL RW ORK (*)
SUBROUT\mathbb{NECTGEVC_64 (SDDE,HOW MNY,SELECT,N,A,LDA,B,LDB,VL,}
LDVL,VR,LDVR,MM,M,W ORK,RW ORK, INFO)
CHARACTER * 1SDEE,HOWMNY
COM PLEX A (LDA,*),B (LD B,*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
INTEGER*8N,LDA,LDB,LDVL,LDVR,MM,M,\mathbb{NFO}
LOGICAL*8 SELECT (*)
REAL RW ORK (*)

```

\section*{F95 INTERFACE}
```

SUBROUTINE TGEVC (SDE,HOW MNY,SELECT, $\mathbb{N}], A,[L D A], B,[L D B], V L$, [LDVL],VR, [LDVR], MM,M, [W ORK], RW ORK], [ $\mathbb{N F O}])$
CHARACTER (LEN=1) ::SDE,HOW MNY
COM PLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{I M}$ ENSION (: : : : : A, B,VL,VR
$\mathbb{N} T E G E R:: N, L D A, L D B, L D V L, L D V R, M M, M, \mathbb{N} F O$

```

LOG ICAL,D \(\mathbb{I M}\) ENSION (:) ::SELECT
REAL,D \(\mathbb{M}\) ENSION (:) ::RW ORK
SU BROUTINE TGEVC_64 (SDE,HOW M NY, SELECT, \(\mathbb{N}], A,[L D A], B,[L D B]\), VL, [LDVL], VR, [LDVR], MM, M, [W ORK], RW ORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::SDE,HOW MNY
COMPLEX,D IM ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::A,B,VL,VR
\(\mathbb{N}\) TEGER (8) :: N, LDA,LDB,LDVL,LDVR,MM,M, \(\mathbb{N} F O\)
LOG ICAL (8), D IM ENSION (:) ::SELECT
REAL,D \(\mathbb{M}\) ENSION (:) ::RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void ctgevc (char side, char how m ny, int *select, intn, com -
plex *a, int lda, com plex *b, int ldb, com plex
*vl, int ldvl, com plex *vr, int ldvr, intm \(m\), int
*m,int*info);
void ctgevc_64 (char side, charhow m ny, long *select, long n, com plex *a, long lda, com plex *b, long ldb, com plex *vl, long ldvl, com plex *vr, long ldvr, long m m, long *m, long *info);

\section*{PURPOSE}
ctgevc com putes som e orallof the right and/or left generalized eigenvectors of a pair of com plex upper triangular \(m\) atrices \((A, B)\).

The right generalized eigenvectorx and the left generalized eigenvectory of \((A, B)\) corresponding to a generalized eigenvaluew are defined by:
\[
(A-w B) * x=0 \text { and } y^{\star *} H *(A-w B)=0
\]
\(w h e r e ~ y * * H\) denotes the conjugate tranpose of \(y\).
If an eigenvalue w is determ ined by zero diagonal elem ents of both A and B, a unit vector is retumed as the comesponding eigenvector.

If alleigenvectors are requested, the routine \(m\) ay either retum the \(m\) atrices \(X\) and/or \(Y\) of rightor lefteigenvectors of \((A, B)\), or the products \(Z * X\) and/or \(Q\) * \(Y\), where \(Z\) and \(Q\) are input unitary \(m\) atrioes. If ( \(A, B\) ) was obtained from the generalized Schur factorization of an original pair ofm atrices
( \(0, B 0\) ) \(=Q * A * Z * * H, Q * B * Z * * H)\),
then \(Z * X\) and \(Q * Y\) are the \(m\) atrices of right or lefteigenvectors ofA.

\section*{ARGUMENTS}

STDE (input)
\(=\mathrm{R}\) ': com pute righteigenvectors only;
\(=\mathrm{L}\) ': com pute lefteigenvectors only;
= \(\mathrm{B}^{\prime}:\) com pute both right and lefteigenvectors.

HOW M NY (input)
= A ': com pute all right and/or lefteigenvectors;
= B ': com pute all right and/or lefteigenvectors, and backtransform them using the inputm atrices supplied in VR and/orV L; = S': com pute selected right and/or lefteigenvectors, specified by the logicalarray SELECT.

SELECT (input)
If HOW M NY = S', SELECT specifies the eigenvectors to be computed. IfHOW M NY=A 'or B', SELECT is not referenced. To select the eigenvector corresponding to the jth eigenvalue, SELECT (i) m ust.be set to .TRUE ..
\(N\) (input) The order of the \(m\) atrices \(A\) and \(B . N>=0\).

A (input) The upper triangularm atrix A .
LD A (input)
The leading dim ension of array A. LDA >= \(\max (1, \mathbb{N})\).

B (input) The upper triangularm atrix B . B m ust have real diagonalelem ents.

LD B (input)
The leading dimension of array B. LDB >= \(\max (1, \mathbb{N})\).

VL (input/output)
On entry, if \(S D E=L\) 'or \(B\) 'and HOW MNY = \(B\) ', VL must contain an \(N\)-by N m atrix \(Q\) (usually the unitary \(m\) atrix \(Q\) of leftSchurvectors retumed by CHGEQZ). On exit, if \(S \mathbb{D} E=L\) 'or \(B^{\prime}, V L\) contains: if HOW MNY = A', the matrix Y of left eigenvectors of \((A, B)\); if HOW M NY = \(B\) ', the matrix \(Q * Y\); ifHOW M NY \(=S^{\prime}\), the left eigenvectors of
(A , B ) specified by SELEC T, stored consecutively in the colum ns of \(\mathrm{V} L\), in the same order as their eigenvalues. If \(S \mathbb{D} E=R \prime V L\) is not referenced.

LDVL (input)
The leading dimension of array \(V L . L D V L>=\) \(\max (1, \mathbb{N})\) if \(S \mathbb{D} E=L\) 'or \(B ' ; L D V L>=1\) otherwise.

VR (input/output)
On entry, if \(S \mathbb{D} E=R\) 'or \(B\) 'and HOW MNY = \(B^{\prime}\) ', VR m ust contain an N -by- N m atrix Q (usually the unitary m atrix \(Z\) of rightSchur vectors retumed by CHGEQZ). On exit, if \(S \mathbb{D} E=R\) 'or \(B^{\prime}, V R\) contains: if HOW M NY = A', them atrix \(X\) of right eigenvectors of \((A, B)\); if HOW M NY = B', them atrix \(Z * X\); if HOW \(M N Y=S\) ', the right eigenvectors of (A , B ) specified by SELEC T, stored consecutively in the colum ns ofVR, in the same order as their eigenvalues. If \(S \mathbb{D} E=L\) ', \(V R\) is not referenced.

LDVR (input)
The leading dim ension of the array VR. LDVR >= \(\max (1, N)\) if \(S \mathbb{D} E=R\) 'or \(B^{\prime} ; L D V R>=1\) otherwise.

M M (input)
The num ber of colum ns in the arrays VL and/or VR. \(\mathrm{M} M>=\mathrm{M}\) 。

M (output)
The num ber of colum ns in the arrays \(V \mathrm{~L}\) and/or VR actually used to store the eigenvectors. If HOW M NY = A 'or B', M is set to N. Each selected eigenvector occupies one colum \(n\).

W ORK (w orkspace)
dim ension \((2 \star \mathrm{~N})\)

RW ORK (w orkspace)
dim ension \((2 * N)\)

INFO (output)
= 0: successfulexit.
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i-\) th argum ent had an illegalvalue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ctgexc - reorder the generalized Schur decom position of a com plex \(m\) atrix pair \((A, B)\), using an unitary equivalence transform ation \((A, B):=Q\) * \((A, B)\) * \(Z\) ', so that the diagonalblock of ( \(A, B\) ) w ith row index \(\mathbb{F S T}\) ism oved to row \(\mathbb{L S T}\)

\section*{SYNOPSIS}
```

SUBROUTINECTGEXC (W ANTQ,W ANTZ,N,A,LDA,B,LDB,Q,LDQ,Z,LD Z,
FST,|ST,\mathbb{NFO)}

```
COM PLEX A (LDA, *), B (LDB,*), Q (LDQ,\(\left.^{\star}\right), \mathrm{Z}(\mathrm{LD} Z, \star)\)
\(\mathbb{N}\) TEGER N,LDA,LDB,LDQ,LD Z, \(\mathbb{F S T}, \mathbb{L} S T, \mathbb{N} F O\)
LOG ICALW ANTQ,W ANTZ
SU BROUTINE CTGEXC_64 (NANTQ,W ANTZ,N,A,LDA,B,LDB,Q,LDQ,Z,LDZ,
    \(\mathbb{F S T}, \mathbb{L} S T, \mathbb{N} F O\) )
COM PLEX A (LDA,*), B (LDB,*), Q (LDQ , *), Z (LD Z,*)
\(\mathbb{N}\) TEGER* \(8 \mathrm{~N}, \mathrm{LD} A, L D B, L D Q, L D Z, \mathbb{F S T}, \mathbb{L S T}, \mathbb{N} F O\)
LOGICAL*8W ANTQ,WANTZ

\section*{F95 INTERFACE}

SU BROUTINE TGEXC \(\mathbb{N}\) ANTQ,W ANTZ, \(\mathbb{N}], A,[L D A], B,[L D B], Q,[L D Q], Z\), \([\mathrm{LD} Z], \mathbb{F S T}, \mathbb{L} \mathrm{ST},[\mathbb{N F O}])\)

COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A, \(B, Q, Z\)
\(\mathbb{N} T E G E R:: N, L D A, L D B, L D Q, L D Z, \mathbb{F S T}, \mathbb{L} T, \mathbb{N} F O\)
LOGICAL ::WANTQ,WANTZ
SU BROUTINE TGEXC_64 (NANTQ,W ANTZ, \(\mathbb{N}], A,[L D A], B,[L D B], Q,[L D Q]\), Z, [LD Z], \(\mathbb{F S T}, \mathbb{L} S T,[\mathbb{N F O}])\)

COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A, \(\mathrm{B}, \mathrm{Q}, \mathrm{Z}\)
\(\mathbb{N}\) TEGER (8) ::N,LDA,LDB,LDQ,LD Z, \(\mathbb{F} S T, \mathbb{L} T, \mathbb{N} F O\) LOG ICAL (8) ::W ANTQ,W ANTZ

\section*{C INTERFACE}
\#include < sunperfh>
void ctgexc (intw antg, intw antz, int \(n\), com plex *a, int lda, com plex *b, int ldb, com plex *q, int ldq, com plex *z, int ldz, int *ifst, int *ilst, int *info);
void ctgexc_64 (long w antq, long wantz, long n, com plex *a, long lda, com plex *b, long ldb, com plex *q, long ldq, com plex *z, long ldz, long *ifst, long *ilst, long *info);

\section*{PURPOSE}
ctgexc reorders the generalized Schur decom position of a com plex \(m\) atrix pair ( \(A, B\) ), using an unitary equivalence transform ation \((A, B):=Q\) * \((A, B)\) * \(Z\) ', so that the diagonal block of ( \(A, B\) ) w th row index \(\mathbb{F} S T\) is m oved to row山ST.
(A , B) m ustbe in generalized Schur canonical form, that is, \(A\) and \(B\) are both upper triangular.

O ptionally, the matrices \(Q\) and \(Z\) of generalized Schur vectors are updated.
\(Q\) (in) *A (in) * \(Z\) (in) \({ }^{\prime}=Q\) (out) *A (out) * \(Z\) (out)'
\(Q\) (in) * \(B(\) in \() * Z(\) in \() '=Q\) (out) * \(B\) (out) * \(Z\) (out) '

\section*{ARGUMENTS}

W ANTQ (input)

W ANTZ (input)
N (input) The order of the m atrices A and \(\mathrm{B} . \mathrm{N}>=0\).

A (input/output)
On entry, the upper triangular \(m\) atrix \(A\) in the pair (A, B). O \(n\) exit, the updated \(m\) atrix A.

LD A (input)
The leading dim ension of the aray A. LD A >= \(\max (1, N)\).

B (input/output)
O \(n\) entry, the upper triangular \(m\) atrix \(B\) in the pair ( \(A, B\) ). On exit, the updated \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

Q (input/output)
On entry, ifW ANTQ = TRUE , the unitary \(m\) atrix \(Q\).
On exit, the updated matrix \(Q\). If \(W\) ANTQ \(=\) FALSE., Q is not referenced.

LD Q (input)
The leading dim ension of the array \(Q\). LD \(Q>=1\); If \(\mathrm{W} A N T Q=. \operatorname{RUE}, \operatorname{LDQ}>=\mathrm{N}\).

Z (input/output)
On entry, ifW ANTZ = TRUE., the unitary \(m\) atrix \(Z\).
On exit, the updated matrix \(Z\). If \(W\) ANTZ \(=\) FALSE., Z is not referenced.

LD Z (input)
The leading dim ension of the array Z . LD \(\mathrm{Z}>=1\); If
W ANTZ = .TRUE.,LD Z >= N.
FST (input/output)
Specify the reordering of the diagonal blocks of ( \(A, B\) ). The block w th row index \(\mathbb{F} S T\) ism oved to row ILST, by a sequence of Sw apping betw een adjcentblocks.

ISST (input/output)
See the description of IFST .
\(\mathbb{N} F \mathrm{O}\) (output)
=0: Successfulexit.
<0: if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvahue.
=1: The transform ed \(m\) atrix pair ( \(A, B\) ) w ould be too far from generalized Schur form ; the problem is ill-conditioned. (A, B) m ay have been partially reordered, and ILST points to the first row of the current position of the block being \(m\) oved.

\section*{FURTHER DETAILS}

B ased on contributions by
Bo K agstrom and PeterPorom aa, D epartm ent of Com puting Science,
Um ea U niversity, S-901 87 Um ea, Sw eden.
[1] B . K agstrom ; A D irectM ethod forReordering Eigenvalues in the G eneralized RealSchurForm of a RegularM atrix Pair (A, B), in

M S.M oonen etal (eds), LinearA lgebra for Large Scale and

Real-T in e A pplications, K luw erA cadem ic Publ. 1993, pp 195-218.
[2] B . K agstrom and P.Porom aa; Com puting Eigenspaces w ith Specified

Eigenvalues of a RegularM atrix Pair (A, B) and Condition

Estim ation: Theory, A lgorithm s and Softw are, Report
UM \(\mathbb{N} F-94.04\), Departm ent of Com puting Science, Umea U níversity,

S-901 87 Um ea, Sw eden, 1994.A lso as LA PA CK W orking N ote 87.

To appear in N um erical A lgorithm s, 1996.
[3] B . K agstrom and P. Porom aa, LA PA C K -Style A lgorithm s and Softw are
for Solving the Generalized Sylvester Equation and Estim ating the

Separation betw een RegularM atrix Pairs, ReportUM IN F 9323,

D epartm entofC om puting Science, Um ea U niversity, S-901 87 Um ea,

Sw eden, Decem ber 1993, Revised A pril 1994, A lso as LAPACK w orking

N ote 75.To appear in A CM Trans. on \(M\) ath. Softw are, \(V\) ol \(22, N \circ 1\),
1996.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ctgsen - reorder the generalized Schur decom position of a complex matrix pair ( \(A, B\) ) (in term \(s\) of an unitary equivalence trans-form ation \(Q\) '* \((A, B)\) * \(Z\) ), so that a selected cluster of eigenvalues appears in the leading diagonalblocks of the pair ( \(A, B\) )

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CTGSEN (IJOB,W ANTQ,W ANTZ,SELECT,N,A,LDA,B,LDB,}
ALPHA,BETA,Q,LDQ,Z,LDZ,M,PL,PR,D \mathbb{F,W ORK,LW ORK,IN ORK,}
L\mathbb{IN ORK, \mathbb{NFO)}}\mathbf{}\mathrm{ )}

```

Z (LD Z, \({ }^{\star}\) ), WORK ( \({ }^{\star}\) )
\(\mathbb{N}\) TEGER LJB \(, N, L D A, L D B, L D Q, L D Z, M, L W O R K, L \mathbb{N} O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{I N}\) ORK (*)
LOG ICALW ANTQ, W ANTZ
LOG ICAL SELECT (*)
REAL PL, PR
REALD \(\left.\mathbb{F}^{( }{ }^{\star}\right)\)
SU BROUTINE CTGSEN_64 (LJOB,W ANTQ,W ANTZ,SELECT,N,A,LDA,B,LDB,
    ALPHA,BETA, Q,LDQ,Z,LDZ,M,PL,PR,D \(\mathbb{F}, W\) ORK,LW ORK, IW ORK,
    \(\mathrm{L} \mathbb{I} \mathrm{O} O R \mathrm{~K}, \mathbb{N} F \mathrm{O})\)
COM PLEX A (LDA ,*), B (LDB,*), ALPHA (*), BETA (*), Q (LDQ,\(\left.^{\star}\right)\),
Z (LD Z, \({ }^{\star}\) ), W ORK ( \({ }^{\star}\) )
\(\mathbb{N}\) TEGER*8 IJOB,N,LDA,LDB,LDQ, LD Z, M, LW ORK, LIN ORK,
\(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER*8 \(\mathbb{I N}\) ORK (*)
LOGICAL*8W ANTQ,WANTZ

\section*{F95 INTERFACE}

SU BROUTINE TG SEN (IMOB,W ANTQ,W ANTZ, SELECT, \(\mathbb{N}], A,[L D A], B,[L D B]\), \(A L P H A, B E T A, Q,[L D Q], Z,[L D Z], M, P L, P R, D \mathbb{F},[W O R K],[L W O R K]\), \([\mathbb{I}\) ORK \(],[\mathbb{L} \mathbb{W}\) ORK \(],[\mathbb{N} F O])\)

COM PLEX,D \(\mathbb{I}\) ENSION (:) :: ALPHA,BETA,WORK
COM PLEX, D \(\mathbb{M} E N S I O N(:,:):: A, B, Q, Z\)
\(\mathbb{N}\) TEGER : \(: \operatorname{IJOB}, N, L D A, L D B, L D Q, L D Z, M, L W O R K, L \mathbb{N} O R K\),
\(\mathbb{I N} \mathrm{FO}\)
\(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{I W}\) ORK
LOGICAL ::W ANTQ, W ANTZ
LOGICAL,D \(\mathbb{I M} E N S I O N(:):: S E L E C T\)
REAL ::PL,PR
REAL,D \(\mathbb{M}\) ENSION (:) ::D \(\mathbb{F}\)

\([L D B], A L P H A, B E T A, Q,[L D Q], Z,[L D Z], M, P L, P R, D \mathbb{F},\left[\begin{array}{l}W\end{array}\right.\)
\([\) LW ORK \(],[\mathbb{W}\) ORK \(],[\) IW ORK \(],[\mathbb{N} F O])\)

COM PLEX,D \(\mathbb{I M} E N S I O N\) (:) ::ALPHA,BETA,W ORK
COM PLEX , D \(\mathbb{M}\) ENSION (:r:) ::A, \(B, Q, Z\)
\(\mathbb{N} \operatorname{TEGER}\) (8) :: IJOB , N,LDA,LDB,LDQ,LDZ,M,LWORK,LIWORK,
\(\mathbb{N} \mathrm{FO}\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I W} O R K\)
LOGICAL (8) ::W ANTQ,W ANTZ
LOGICAL (8), D \(\mathbb{M}\) ENSION (:) :: SELECT
REAL ::PL,PR
REAL,D \(\mathbb{M}\) ENSION (:) ::D \(\mathbb{F}\)

\section*{C INTERFACE}
\#include <sunperfh>
void ctgsen (int ijob, intw antq, intw antz, int * select, int n, com plex *a, int lda, com plex *b, int ldb, com plex *alpha, com plex *beta, com plex *q, int ldq, com plex *z, int ldz, int *m, float *pl, float *pr, float*dif, int*info);
void ctgsen_64 (long ij̀b, long w anta, long w antz, long *select, long n, com plex *a, long lda, com plex *b, long ldb, com plex *ałpha, com plex *beta, com plex
*q, long ldq, com plex *z, long ldz, long *m, float *pl, float *pr, float *dif, long *info);

\section*{PURPOSE}
ctgsen reorders the generalized Schur decom position of a complex \(m\) atrix pair ( \(A, B\) ) (in term \(s\) of an unitary equivalence trans-form ation \(Q\) '* \((A, B)\) * \(Z\) ), so that a selected chuster of eigenvalues appears in the leading diagonalblocks of the pair ( \(A, B\) ).The leading colum ns of \(Q\) and \(Z\) form unitary bases of the corresponding left and right eigenspaces (deflating subspaces). (A , B) m ust be in generalized Schur canonical form, that is, A and B are both upper triangular.

C TG SEN also com putes the generalized eigenvalues
w \((\mathcal{j})=A L P H A(\mathcal{j}) / B E T A(\mathcal{O})\)
of the reordered \(m\) atrix pair ( \(A, B\) ).
Optionally, the routine com putes estim ates of reciprocal condition num bers foreigenvalues and eigenspaces. These are \(D\) ifin [ \(A 11, B 11\) ), (A 22, B22)] and D ifl[ \((A 11, B 11)\), (A 22, B22)], i.e. the separation (s) betw een the \(m\) atrix pairs (A 11, B11) and (A22,B22) that comespond to the selected cluster and the eigenvalues outside the cluster, resp., and norm sof "pro jections" onto left and right eigenspaces w r.t. the selected cluster in the \((1,1)\)-block .

\section*{ARGUMENTS}
```

INO B (input)
Specifies w hether condition num bers are required
for the cluster ofeigenvalues (PL and PR) or the
deflating subspaces (D ifiu and D ifl):
=0:Only reorderw r.t.SELEC T .N o extras.
=1:Reciprocal of norm s of "pro jections" onto left
and righteigenspaces w r.t. the selected cluster
(PL and PR). = 2:U pperbounds on D ifu and D ifl.
F-norm -based estim ate
(D F (1:2)).
=3:Estim ate ofD ifu and D ifl.1-norm -based esti-
m}\mathrm{ ate
(D IF (1:2)). A bout5 tim es as expensive as IJO B =
2. =4: Com pute PL,PR and D FF (i.e.0,1 and 2
above): Econom ic version to getitall. =5: C om -
pute PL,PR and D FF (i.e.0,1 and 3 above)

```
W ANTQ (input)

\section*{SELECT (input)}

SELEC T specifies the eigenvalues in the selected cluster. To selectan eigenvalue w ( \()\), SELECT ( \()\) m ustbe set to

N (input) The order of the m atrioes A and \(\mathrm{B} . \mathrm{N}>=0\).

A (input/output)
On entry, the uppertriangularm atrix \(A\), in generalized Schur canonical form. On exit, A is overw rilten by the reordered \(m\) atrix A.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).
B (input/output)
On entry, the upper triangularm atrix \(B\), in generalized Schur canonical form. On exit, \(B\) is overw ritten by the reordered \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the aray \(\mathrm{B} . \operatorname{LD} \mathrm{B}>=\) \(\max (1, N)\).

\section*{A LPHA (output)}

The diagonalelem ents of \(A\) and \(B\), respectively, when the pair \((A, B)\) has been reduced to generalized Schurform. A LPHA (i) BETA (i) \(i=1, \ldots, \mathrm{~N}\) are the generalized eigenvalues.

BETA (output)
See the description of A LPH A .
Q (input/output)
On entry, if WANTQ = TRUE., \(Q\) is an \(N\)-by \(-N\) \(m\) atrix. On exit, \(Q\) has been postm ultiplied by the leftunitary transform ation \(m\) atrix which reorder ( \(\mathrm{A}, \mathrm{B}\) ); The leading \(M\) colum ns ofe form orthonor\(m\) albases for the specified pair of left eigenspaces (deflating subspaces). IfW ANTQ = FALSE., \(Q\) is not referenced.

LD Q (input)
The leading dim ension of the array \(\mathrm{Q} . \mathrm{LD} \mathrm{Q}>=1\). IfW \(A N T Q=\) TRUE., LDQ \(>=N\).

Z (input/output)
On entry, if W ANTZ = .TRUE., Z is an N -by -N \(m\) atrix. On exit, \(Z\) has been postm ultiplied by the leftunitary transform ation \(m\) atrix which reorder
( \(\mathrm{A}, \mathrm{B}\) ); The leading \(M\) colum ns of \(Z\) form orthonorm albases for the specified pair of left eigenspaces (deflating subspaces). IfW ANTZ = FALSE., Z is not referenced.

LD \(Z\) (input)
The leading dim ension of the array \(\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1\). IfW ANTZ \(=\).TRUE, \(\mathrm{LD} Z>=\mathrm{N}\).
M (output)
The dim ension of the specified pair of left and righteigenspaces, (deflating subspaces) \(0<=\mathrm{M}<=\) N .

PL (output)
IF \(\mathrm{IJOB}=1,4\), or \(5, \mathrm{PL}, \mathrm{PR}\) are low er bounds on the reciprocal of the norm of "projections" onto leftand right eigenspace \(w\) ith respect to the selected cluster.
\(0<\mathrm{PL}, \mathrm{PR}<=1\). If \(\mathrm{M}=0\) orM \(=\mathrm{N}, \mathrm{PL}=\mathrm{PR}=1\). If \(\mathrm{IJO} \mathrm{B}=0,2\), or \(3 \mathrm{PL}, \mathrm{PR}\) are not referenced.

PR (output)
See the description ofPL .

D IF (output)
If IOO B >= 2, D \(\mathbb{F}(1: 2)\) store the estim ates ofD ifu and \(D\) ifl.
If \(\mathrm{IOO} B=2\) or \(4, \mathrm{D} \mathbb{F}(1: 2)\) are F -nom -based upper bounds on
\(D\) ifu and \(D\) ifl. If \(I \mathcal{L D} B=3\) or \(5, D \mathbb{F}(1: 2)\) are 1norm -based estim ates of \(D\) ifu and \(D\) ifl, com puted using reversed com \(m\) unication \(w\) ith CLACON. IfM \(=\) 0 orN,\(D \operatorname{FF}(1: 2)=F\) norm ( \([A, B]\) ). If \(I J B=0\) or \(1, \mathrm{D} \mathbb{F}\) is not referenced.

W ORK (w orkspace)
If \(I O B=0, W\) ORK is not referenced. O therw ise, on exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK.LW ORK >= 1 If \(\mathrm{IJOB}=1,2\) or \(4, L W\) ORK \(>=2 \star M *(N+M)\) If \(\mathrm{IJOB}=3\) or \(5, \operatorname{LW}\) ORK >= \(4 * M *(\mathbb{N}+\mathrm{M})\)

If LW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IW ORK (w orkspace/output)
If \(I O B=0, \mathbb{I W} O R K\) is notreferenced. O therw ise, on exit, if \(\mathbb{N F}\) FO \(=0, \mathbb{I} W\) ORK (1) retums the optim al LIW ORK.

LIW ORK (input)
The dim ension of the aray \(\mathbb{I W}\) ORK.LIW ORK \(>=1\). If \(\mathrm{IJOB}=1,2\) or \(4, \mathrm{LIW}\) ORK \(>=\mathrm{N}+2\); If \(\mathrm{IWOB}=3\) or \(5, L \mathbb{I} W\) ORK \(>=M A X \mathbb{N}+2,2 * M * \mathbb{N}+M)\) );

IfLIW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the \(\mathbb{I W}\) ORK array, retums this value as the first entry of the \(\mathbb{I W}\) ORK array, and no errorm essage related to \(L \mathbb{I W}\) ORK is issued by X ERBLA.
\(\mathbb{I N F O}\) (output)
=0: Successfulexit.
\(<0:\) If \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegal
value.
=1: Reordering of ( \(A\), B) failed because the transform ed \(m\) atrix pair ( \(A, B\) ) w ould be too far from generalized Schur form ; the problem is very ill-conditioned. (A, B) m ay have been partially reordered. If requested, 0 is retumed in D \(\mathbb{F}(*)\), \(P L\) and \(P R\).

\section*{FURTHER DETAILS}

CTG SEN first collects the selected eigenvalues by com puting unitary \(U\) and \(W\) thatm ove them to the top leftcomer of ( \(A\), B ). In otherw ords, the selected eigenvalues are the eigenvalues of (A11, B11) in
\[
\begin{gathered}
U *(A, B) * W=(A 11 A 12)(B 11 B 12) n 1 \\
(0 \text { A 22) (0 B 22) n2 } \\
n 1 \mathrm{n} 2 \mathrm{n} 1 \mathrm{n} 2
\end{gathered}
\]
where \(N=n 1+n 2\) and \(U\) ' \(m\) eans the conjugate transpose of \(U\). The first n1 colum ns of \(U\) and \(W\) span the specified pair of leftand righteigenspaces (deflating subspaces) of (A, B).

If \((A, B)\) has been obtained from the generalized real Schur decom position of a \(m\) atrix pair \((C, D)=Q\) * \((A, B) * Z\) ', then the reordered generalized Schur form of ( \(C, D\) ) is given by
\[
(\mathrm{C}, \mathrm{D})=(\mathrm{Q} * \mathrm{U}) \star(\mathrm{U} *(\mathrm{~A}, \mathrm{~B}) * \mathrm{~W}) \star(\mathrm{Z} * \mathrm{~W})^{\prime}
\]
and the firstn 1 colum ns of \(Q\) * \(U\) and \(Z * W\) span the correspond-
ing deflating subspaces of ( \(\mathrm{C}, \mathrm{D}\) ) Q and Z store Q * U and Z *W, resp.).

N ote that if the selected eigenvalue is sufficiently illconditioned, then its valuem ay differ significantly from its value before reordering.

The reciprocalcondition num bers of the left and right eigenspaces spanned by the firstn1 colum nsofU and \(W\) (or \(Q * U\) and \(Z * W) m\) ay be retumed in \(D \operatorname{FF}(1: 2)\), corresponding to \(D\) ifu and \(D\) ifl, resp.

The \(D\) ifu and \(D\) iflare defined as:
ifu \([\) A 11, B11), (A 22, B22) \(]=\operatorname{sigm} \operatorname{arm}\) in ( Zu )
and
where sigm a-m in \((Z u)\) is the sm allest singular value of the ( \(2 *_{n} 1 *_{n} 2\) )-by-( \(2 *_{n} 1 *_{n} 2\) ) m atrix
\(\mathrm{u}=[\mathrm{kron}(\mathrm{In} 2, \mathrm{~A} 11)-\mathrm{kron}(\mathrm{A} 22\) ', In1) \(]\)
[kron(In2, B11) kron (B22', In1)].
H ere, \(I n x\) is the identily \(m\) atrix of size \(n x\) and \(A 22\) 'is the transpose of A \(22 . \operatorname{kron}(X, Y)\) is the \(K\) roneckerproduct betw een the \(m\) atrices \(X\) and \(Y\).

W hen D IF (2) is sm all, sm all changes in (A, B) can cause large changes in the deflating subspace. A \(n\) approxim ate (asym ptotic) bound on the \(m\) axim um angularemor in the com puted deflating subspaces is PS * norm ( \((\mathrm{A}, \mathrm{B})\) )/D \(\mathbb{F}(2)\),
where EPS is the \(m\) achine precision.
The reciprocal norm of the pro jectors on the left and right eigenspaces associated w ith (A 11, B 11) m ay be retumed in PL and PR. They are com puted as follow s. First we com pute L and \(R\) so that \(P\) * \((A, B) * Q\) is block diagonal, w here \(=(I-4) n 1 \quad \mathrm{Q}=(\mathrm{IR}) \mathrm{n} 1\)
( 0 I ) n2 and ( 0 I ) n2
n1 n2 n1 n2
and ( \((,, R)\) is the solution to the generalized Sylvester equation \(11 * \mathrm{R}-\mathrm{L} *\) A \(22=-\mathrm{A} 12\)

Then PL \(=(\mathbb{E} \text {-norm }(L) * * 2+1)^{* *}(-1 / 2)\) and \(P R=(F-\) norm \(\left.(\mathbb{R})^{* *} 2+1\right)^{* *}(-1 / 2)\). A \(n\) approxim ate (asym ptotic) bound on the average absolute error of the selected eigenvalues is EPS * norm ( \(A, B)\) )/PL.

There are also globalemorbounds which valid forperturbations up to a certain restriction: A low erbound \((x)\) on the
sm allest \(F\)-norm ( \(E, F\) ) forw hich an eigenvalue of (A 11, B 11) \(m\) ay \(m\) ove and coalesce \(w\) th an eigenvalue of (A 22,B22) under perturbation \((\mathbb{E})\), (i.e. \((A+E, B+F)\), is

X

\section*{\(=\)}
\(m\) in \((\mathbb{D}\) ifu, D ifl)/( \((1 / \mathbb{P L} * \mathrm{PL})+1 /(\mathrm{PR} * \mathrm{PR})) * *(1 / 2)+2 * \mathrm{~m}\) ax \((1 / P L, 1 / P R))\).
A \(n\) approxim ate bound on \(x\) can be com puted from \(D \mathbb{F}(1: 2)\), PL and PL.

If \(y=(F-\) norm \((E, F) / x)<=1\), the angles betw een the perturbed ( \((1, R)\) and unperturbed ( \(L, R\) ) left and right deflating subspaces associated \(w\) ith the selected chuster in the \((1,1)\)-blocks can be bounded as
max-angle \((L, L)<=\arctan (y * P L /(1-y *(1-P L *\) PL)** (1/2))
\(\max \operatorname{angle}(R, R)<=\arctan (y * P R /(1-y *(1-P R *\) PR)** (1/2))

See LA PA CK U sers G uide section 4.11 or the follow ing references form ore inform ation.
\(N\) ote that if the default \(m\) ethod for com puting the Frobenius-norm - based estim ate D \(\mathbb{F}\) is not wanted (see CLATDF), then the param eter \(\mathbb{D} \mathbb{F}\) JB (see below) should be changed from 3 to 4 (routine CLATDF (IJOB \(=2\) w ill be used)). See C TG SY L form ore details.

B ased on contributions by
Bo K agstrom and PeterPorom aa, D epartm ent of Com puting Science,
U m ea U niversity, S-901 87 Um ea, Sw eden.

References
\(========\)
[1] B . K agstrom ; A D irectM ethod forReordering Eigenvahues in the

G eneralized RealSchurForm of a RegularM atrix Pair (A, B), in

M S .M oonen etal (eds), LinearA lgebra for Large Scale and

Real-T im e A pplications, K luw erA cadem ic Publ. 1993, pp 195-218.
[2] B . K agstrom and P .Porom aa; C om puting Eigenspaces w ith Specified

Eigenvalues of a RegularM atrix Pair ( \(A\), B ) and C ondition

Estim ation: Theory, A lgorithm s and Softw are, Report

UM \(\mathbb{N} F-94.04\), Departm ent of Computing Science, Um ea U niversity, S-901 87 U m ea, Sw eden, 1994. A lso as LA PA CK W orking N ote 87.

To appear in N um ericalA lgorithm S, 1996.
[3] B . K agstrom and P . Porom aa, LA PA C K -Style A lgorithm s and Softw are
for Solving the Generalized Sylvester Equation and Estim ating the Separation betw een RegularM atrix Pairs, ReportUM \(\mathbb{N} F\) 93 23,

D epartm entofC om puting Science, U m ea U niversity, S-901 87 Um ea, Sw eden, D ecem ber 1993, Revised A pril 1994, A lso as LA PACK working

N ote 75.To appear in A CM Trans. on M ath. Softw are, V ol \(22, N \circ 1\), 1996.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctgsja -com pute the generalized singular value decom position (G SVD ) of tw o com plex upper triangular (or trapezoidal) \(m\) atrices \(A\) and \(B\)

\section*{SYNOPSIS}
```

SUBROUTINECTGSJA (JO BU,NOBV,NOBQ,M,P,N,K,L,A,LDA,B,LDB,
TOLA,TOLB,ALPHA,BETA,U,LDU,V,LDV,Q,LDQ,W ORK,NCYCLE,
\mathbb{NFO)}
CHARACTER * 1 JOBU,NOBV,NOBQ
COM PLEX A (LDA,*),B (LD B,*), U (LDU ,*), V (LDV ,*), Q (LDQ ,*),
W ORK (*)
INTEGER M,P,N,K,L,LDA,LDB,LDU,LDV,LDQ NCY CLE, IN FO
REALTOLA,TOLB
REAL ALPHA (*),BETA (*)
SUBROUT\mathbb{NECTGSJA_64(JOBU,JOBV,JOBQ,M,P,N,K,L,A,LDA,B,LD B,}
TOLA,TOLB,ALPHA,BETA,U,LDU,V,LDV,Q,LDQ,W ORK,NCYCLE,
\mathbb{NFO)}
CHARACTER * 1 JOBU,NOBV,NOBQ
COM PLEX A (LDA,*),B (LD B,*), U (LDU ,*), V (LDV ,*), Q (LDQ ,*),
W ORK (*)
\mathbb{NTEGER*8M,P,N,K,L,LDA,LDB,LDU, LDV,LDQ,NCYCLE,}
\mathbb{NFO}
REAL TOLA,TOLB
REAL ALPHA (*),BETA (*)

```

F95 INTERFACE
SU BROUTINE TGSJA (JOBU, JOBV, JOBQ, \(\mathbb{M}],[\mathbb{P}], \mathbb{N}], K, L, A,[L D A], B\), \([L D B], T O L A, T O L B, A L P H A, B E T A, U,[L D U], V,[L D V], Q,[L D Q]\),
[W ORK],NCYCLE, [ \(\mathbb{N F O}\) ])

CHARACTER ( \(\llcorner E N=1):: 10 B U, J O B V, J B B Q\)
COM PLEX ,D \(\mathbb{M}\) ENSION (:) ::W ORK
COMPLEX, D \(\mathbb{M} E N S I O N(:,:): A, B, U, V, Q\)
\(\mathbb{N}\) TEGER :: M , P, N , K , L, LDA \(, L D B, L D U, L D V, L D Q, N C Y C L E\), \(\mathbb{N}\) FO
REAL ::TOLA,TOLB
REAL, D \(\mathbb{M} E N S I O N(:):: A L P H A, B E T A\)

SU BROUTINE TGSJA_64 (JOBU, OBC, OBQ, M ], \(\mathbb{P}], \mathbb{N}], K, L, A,[L D A]\), \(B,[L D B], T O L A, T O L B, A L P H A, B E T A, U,[L D U], V,[L D V], Q,[L D Q]\), [W ORK],NCYCLE, [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) :: OBB , OBC, OBB
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::W ORK
COMPLEX,D \(\mathbb{M} E N S I O N(:,:): A, B, U, V, Q\)
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{P}, \mathrm{N}, \mathrm{K}, \mathrm{L}, \mathrm{LD} A, L D B, L D U, L D V, L D Q, N C Y-\)
\(C L E, \mathbb{N F O}\)
REAL ::TOLA,TOLB
REAL,D \(\mathbb{I M} E N S I O N(:):: A L P H A, B E T A\)

\section*{C INTERFACE}
\#include <sunperfh>
void ctgsja (char jobu, char jंbv, char joboq, intm, int p, int \(n\), intk, intl, com plex *a, int lda, com plex *b, int ldb, float tola, float tolb, float *alpha, float *beta, com plex *u, int ldu, com plex *v, int ldv, com plex *q, int ldq, int *ncycle, int *info);
void ctgs’á_64 (char j’bu, char j’bv, char j’.bq, long m, long
p, long \(n\), long \(k\), long l, com plex *a, long lda, com plex *b, long ldlo, float tola, float tolb, float *alpha, float *beta, com plex *u, long ldu, com plex *v, long ldv, com plex *q, long ldq, long *ncycle, long *info);

\section*{PURPOSE}
ctgsja com putes the generalized singular value decom position (G SVD ) of two com plex upper triangular (ortrapezoidal) \(m\) atrices \(A\) and \(B\).

On entry, it is assum ed thatm atrices A and B have the follow ing form \(s\), which \(m\) ay be obtained by the preprocessing subroutine CG G SV P from a generalM boy \(-\mathrm{N} m\) atrix \(A\) and \(P-b y-N\) m atrix B :
```

        NK工 K L
    A = K (0 A12 A 13) ifM K L >=0;
L(0 0 A 23)
M K-工(0 0 0 )
NK工K L
A = K (0 A12 A13) ifM K L < 0;
M K (0 0 A 23)
N-K\dashv K L
B=L(0 0 B13}
P-工(0 0 0 )

```
where the K －by -K m atrix A 12 and \(L\)－by -m atrix B 13 are non－ singularuppertriangular；A 23 is L－by－L uppertriangular if


On exit，
U＊A＊Q＝D 1＊（0R），V＊B＊Q＝D 2＊（0R），
where \(U, V\) and \(Q\) are unitary \(m\) atrices，\(Z\)＇denotes the conju－ gate transpose of \(Z, R\) is a nonsingular upper triangular \(m\) atrix，and D 1 and D 2 are＇＇diagonal＂m atriges，which are of the follow ing structures：

IfM \(K-\perp=0\) ，
```

            K L
    DI= K (I O)
L (0 C )
M K 工(0 0)
K L
D 2 = L (0 S )
P-L (0 0)

```
            N K K K L
(0R) \(\quad=\mathrm{K}\) ( 0 R11 R12) \(K\)
    L ( \(0 \quad 0 \quad \mathrm{R} 22\) ) L
where
\(C=\operatorname{diag}(A L P H A(K+1), \ldots, A L P H A(K+L))\) ， \(S=\operatorname{diag}(\operatorname{BETA}(K+1), \ldots, B E T A(K+L))\) ， \(C * * 2+S * * 2=I\) 。
\(R\) is stored in \(A(1: K+L, N-K-工+1 \mathbb{N})\) on exit．

IfM K 工＜0，
```

            K M K K +L M
    D1=K(IO O )
M K (OC O )

```
```

            K M K K +L-M
    D2 = MK (0 S 0 )
K+L-M (0 0 I )
P-(0 0 0 )

```
            \(N \dashv \_\quad \mathrm{K} \quad \mathrm{M}\) K \(\mathrm{K}+\mathrm{L} \mathrm{M}\)
        MK (0 O R22 R23)
    \(\mathrm{K}+\mathrm{L} \mathrm{M}\) (0 0 0 R33)
w here
\(C=\operatorname{diag}(A L P H A(K+1), \ldots, A L P H A(M))\), \(S=\operatorname{diag}(B E T A(K+1), \ldots, B E T A(M))\), \(C * * 2+S * * 2=I\).
\(R=(R 11 R 12 R 13)\) is stored in \(A(1 M, N-K-1 \mathbb{N})\) and \(R 33\) is stored
( 0 R22R23)
in \(B(M-K+1: L+M-K-\rfloor+1 \mathbb{N})\) on exit.

The com putation of the unitary transform ation \(m\) atrices \(U, V\) or \(Q\) is optional. These \(m\) atrioes \(m\) ay eitherbe form ed explicitly, or they \(m\) ay be postm ultiplied into input \(m\) atrices \(\mathrm{U} 1, \mathrm{~V} 1\), or Q 1 .

CTGSJA essentially uses a variantof K ogbetliantz algorithm to reduce \(m\) in ( \(\llcorner, M \mathrm{~K}\) ) -by L triangular (or trapezoidal) m atrix A 23 and L -by -m atrix B 13 to the form :
\(\mathrm{U} 1{ }^{*} \mathrm{~A} 13^{*} \mathrm{Q} 1=\mathrm{C} 1 * \mathrm{R} 1\); V 1 * \(\mathrm{B} 13^{*} \mathrm{Q} 1=\mathrm{S} 1 * \mathrm{R} 1\),
where \(\mathrm{U} 1, \mathrm{~V} 1\) and Q 1 are unitary \(m\) atrix, and \(Z\) 'is the conjugate transpose of Z . C1 and S1 are diagonalm atrioes satisfying
\(\mathrm{C} 1 * * 2+\mathrm{S} 1 * * 2=\mathrm{I}\),
and R1 is an L-oy- L nonsingularuppertriangularm atrix.

\section*{ARGUMENTS}

JOBU (input)
\(=\mathrm{U}\) ': U m ustcontain a unitary matrix U 1 on entry, and the productU \(1 * \mathrm{U}\) is retumed; = I ': U is initialized to the unitm atrix, and the unitary m atrix U is retumed; \(=\mathrm{N}: \mathrm{U}\) is not com puted.

JOBV (input)
\(=\mathrm{V}\) ': V m ustcontain a unitary matrix V1 on
entry, and the product \(1 * V\) is retumed; = \(I\) ': V is initialized to the unitm atrix, and the unitary m atrix V is retumed; \(=\mathrm{N}: \mathrm{V}\) is not com puted.

JOBQ (input)
\(=\mathrm{Q}\) ': Q m ustcontain a unitary matrix Q 1 on entry, and the product \(\mathrm{Q} 1^{*} \mathrm{Q}\) is retumed; = I ': Q is initialized to the unitm atrix, and the unitary m atrix Q is retumed; \(=\mathrm{N}\) : Q is notcom puted.

M (input) The num ber of row s of them atrix \(A . M>=0\).
\(P\) (input) The num ber of row s of the \(m\) atrix \(B . P>=0\).

N (input) The num ber of colum ns of the \(m\) atrioes A and B. N \(>=0\).
\(K\) (input) \(K\) and \(L\) specify the subblocks in the input \(m\) atrices \(A\) and \(B\) : \(A 23=A(K+1 \mathbb{M} \mathbb{N}(\mathbb{K}+L, M), N-1+1 \mathbb{N})\) and \(B 13=\) \(B(1: L, N- \pm+1 \mathbb{N})\) of \(A\) and \(B\), whose G SVD is going to be com puted by CTGSJA. See the Further D etails section below .

L (input) See the description of K.

A (input/output)
On entry, the \(M\) boy \(-N\) m atrix \(A\). On exit, \(A \mathbb{N}-\) \(K+1 \mathbb{N}, 1 \mathbb{M} \mathbb{N}(\mathbb{K}+\mathrm{L}, \mathrm{M})\) ) contains the triangular \(m\) atrix R orpart of \(R\). See Punpose for details.

LD A (input)
The leading dim ension of the array A.LDA >= \(\max (1, M)\).

B (input/output)
On entry, the \(\mathrm{P}-\mathrm{by}-\mathrm{N} m\) atrix B. On exit, if neces sary, \(B(M-K+1: N+M-K-1 \mathbb{N})\) contains a partof \(R\). See Purpose for details.

LD B (input)
The leading dim ension of the aray \(\mathrm{B} \cdot \mathrm{LD} \mathrm{B}>=\) \(\max (1, P)\).

TOLA (input)
TO LA and TO LB are the convergence criteria for the Jacobi- K ogbetliantz iteration procedure. G enerally, they are the sam e as used in the prepro-
 TOLB \(=M A X(\mathbb{P} N)^{\star}\) norm \((B) \star M A C H E P S\).

\section*{TOLB (input)}

See the description of TO LA .

ALPHA (output)
On exit, A LPHA and BETA contain the generalized singularvalue pairs of \(A\) and \(B\); \(A \operatorname{LPH} A(1: K)=1\), \(\operatorname{BETA}(1: K)=0\), and ifM \(K->=0\), ALPHA \((\mathbb{K}+1 \mathbb{K}+L)\)
\(=\operatorname{diag}(\mathrm{C})\),
BETA \((\mathbb{K}+1 \mathbb{K}+L)=\operatorname{diag}(S)\), or if \(M K- \pm<0\), ALPHA \((\mathbb{K}+1 \mathrm{M})=C, A L P H A(M+1 K+L)=0\)
BETA \((\mathbb{K}+1 \mathbb{M})=S, B E T A(M+1: K+L)=1\). Furtherm ore, if \(\mathrm{K}+\mathrm{L}<\mathrm{N}, \mathrm{A} L P H A(K+\mathrm{L}+1 \mathbb{N})=0\)
BETA \((\mathbb{K}+\mathrm{L}+1 \mathbb{N})=0\) 。

BETA (output)
See the description of A LPH A.

U (input) On entry, if \(\mathrm{JOBU}=\mathrm{U}\) ', U m ustcontain a matrix U 1 (usually the unitary matrix retumed by CGGSVP). On exit, if \(J O B U=\) I', \(U\) contains the unitary matrix U ; if \(\mathrm{JOBU}=\mathrm{U}\) ', U contains the product I * U . If \(\mathrm{JOBU}=\mathrm{N}\) ', U is notreferenced.

LDU (input)
The leading dim ension of the aray U.LDU >= \(m\) ax (1, M) if OB BU \(=\mathrm{U}\) '; LD \(\mathrm{U}>=1\) otherw ise.

V (input) On entry, if JOBV \(=\mathrm{V}^{\prime}, \mathrm{V}\) m ustcontain a matrix V 1 (usually the unitary \(m\) atrix retumed by CGGSVP). On ex正, if \(\mathrm{JOBV}=\mathrm{I}, \mathrm{V}\) contains the unitary \(m\) atrix \(V\); if \(J O B V=V ' V\) contains the productV 1*V. If JO BV = N',V is notreferenced.

LDV (input)
The leading dim ension of the array \(V\). LD V >= \(m\) ax ( \(1, \mathrm{P}\) ) if \(\mathrm{OB} \mathrm{BV}=\mathrm{V}\); LD \(V>=1\) otherw ise.
\(Q\) (input) \(O n\) entry, if \(J O B Q=Q\) ', \(Q\) m ustcontain a matrix Q 1 (usually the unitary \(m\) atrix retumed by CGGSVP). On exit, if \(J O B Q=\) I', Q contains the unitary \(m\) atrix \(Q\); if \(J B Q=Q ', Q\) contains the productQ \(1 *\) Q. If \(J O B Q=N\) ', Q is notreferenced.

LDQ (input)
The leading dim ension of the array \(Q . \operatorname{LDQ}>=\) \(m a x(1, N)\) if \(\mathrm{OBQ}=\mathrm{Q}^{\prime} ; \mathrm{LDQ}>=1\) otherw ise.

W ORK (w orkspace)
dim ension ( \(2 * N\) )

NCYCLE (output)
The num ber of cycles required for convergence.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue.
= 1 : the procedure does not converge after M A X IT cycles.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ctgsna -estim ate reciprocalcondition num bers for specified
eigenvalues and/oreigenvectors of a \(m\) atrix pair ( \(A, B\) )

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CTGSNA (JOB,HOW MNT,SELECT,N,A,LDA,B,LDB,VL,LDVL,}
VR,LDVR,S,D\mathbb{F},MM,M,WORK,LWORK,I\mathbb{ORK,INFO)}
CHARACTER * 1 JOB,HOW MNT
COM PLEX A (LDA,*),B (LDB,*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
INTEGERN,LDA,LDB,LDVL,LDVR,MM,M,LW ORK,\mathbb{NFO}
INTEGER IV ORK (*)
LOG ICAL SELECT (*)
REALS (*),D F (*)
SUBROUT\mathbb{NECTGSNA_64(JOB,HOW MNT,SELECT,N,A,LDA,B,LDB,VL,}
LDVL,VR,LDVR,S,D F,MM,M,W ORK,LW ORK,INORK,INFO)
CHARACTER * 1 JOB,HOWMNT
COM PLEX A (LDA,*),B (LDB,*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
\mathbb{NTEGER*8N,LDA,LD B,LDVL,LDVR,MM ,M ,LW ORK, INFO}
INTEGER*8 IN ORK (*)
LOGICAL*8 SELECT (*)
REALS (*),D FF (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE TGSNA (JOB,HOW MNT,SELECT, $\mathbb{N}], A,[L D A], B,[L D B], V L$, [LDVL],VR, [LDVR],S,D $\mathbb{F}, M M, M,\left[\begin{array}{l}\text { M ORK ], [LW ORK ], [W ORK ], }\end{array}\right.$ [ $\mathbb{N} F O$ ])

```

COMPLEX，D \(\mathbb{M}\) ENSION（：）：：W ORK
COM PLEX，D \(\mathbb{I M}\) ENSION（：，：）：：A，B，VL，VR
\(\mathbb{N} T E G E R:: N, L D A, L D B, L D V L, L D V R, M M, M, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION（：）：： \(\mathbb{I N}\) ORK
LOG ICAL，D IM ENSION（：）：：SELECT
REAL，D \(\mathbb{I M} E N S I O N(:):: S, D \mathbb{F}\)

SU BROUTINE TGSNA＿64（0⿴囗十，HOW M NT，SELECT， \(\mathbb{N}], A,[L D A], B,[L D B], V L\), ［LDVL］，VR，［LDVR］，S，D \(\mathbb{F}, M M, M,[W O R K],[L W O R K],[\mathbb{W} O R K]\), ［ \(\mathbb{N} F O\) ］）

CHARACTER（LEN＝1）：：JOB，HOW M NT
COMPLEX，D \(\mathbb{I M}\) ENSION（：）：：W ORK
COMPLEX，D \(\mathbb{M}\) ENSION（：：：：：A，B，VL，VR
\(\mathbb{N}\) TEGER（8）：：N ，LDA，LDB，LDVL，LDVR，MM，M，LW ORK， \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
LOG ICAL（8），D IM ENSION（：）：：SELECT
REAL，D \(\mathbb{M}\) ENSION（：）：：S，D \(\mathbb{F}\)

\section*{C INTERFACE}
\＃include＜sunperfh＞
void ctgsna（char job，charhow mnt，int＊select，intn，com－ plex＊a，int lda，com plex＊b，intldb，com plex ＊vl，int ldvl，com plex＊vr，int ldvr，float＊s， float＊dif，intm \(m\) ，int＊m，int＊info）；
void ctgsna＿64（char job，charhow mnt，long＊select，long n， com plex＊a，long lda，com plex＊b，long ldb，com－ plex＊vl，long ldvl，com plex＊vr，long ldvr，float ＊s，float＊dif，long m m ，long＊m，long＊info）；

\section*{PURPOSE}
ctgsna estim ates reciprocal condition num bers for specified eigenvalues and／oreigenvectors of a \(m\) atrix pair（ \(A, B\) ）．
（ \(A, B\) ）\(m\) ustbe in generalized Schurcanonical form ，that is， \(A\) and \(B\) are both upper triangular．

\section*{ARGUMENTS}

JOB（input）
Specifies w hether condition num bers are required
foreigenvalues（S）oreigenvectors（D \(\mathbb{F}\) ）：
＝E＇：foreigenvalues only（S）；
＝V＇：foreigenvectors only（D \(\mathbb{F}\) ）；
＝ B ：forboth eigenvalues and eigenvectors（ S
and \(D \mathbb{F}\) ).

HOW MNT (input)
= A ': com pute condition num bers for all eigen-
pairs;
= \(\mathrm{S}^{\prime}\) : com pute condition num bers for selected eigenpairs specified by the array SELEC T .

\section*{SELECT (input)}

If HOW M NT = S',SELECT specifies the eigenpairs
for which condition num bers are required. To
select condition num bers for the corresponding \(j\)
th eigenvalue and/oreigenvector, SELECT ( ) \() \mathrm{m}\) ust
be setto TRUE.. IfHOWMNT=A',SELECT is not referenced.

N (input) The order of the square \(m\) atrix pair \((A, B) . N \quad>=\) 0.
\(A\) (input) The uppertriangularm atrix \(A\) in the pair \((A, B)\).
LD A (input)
The leading dim ension of the aray A. LD A >= \(\max (1, \mathbb{N})\).
\(B\) (input) The uppertriangularm atrix \(B\) in the pair ( \(A, B\) ).

\section*{LD B (input)}

The leading dim ension of the array \(B . L D B>=\) \(\max (1, \mathbb{N})\).

VL (input)
If \(\mathrm{JOB}=\mathrm{E}\) 'or B ', VL m ust contain left eigenvectors of \((A, B)\), comesponding to the eigenpairs specified by H OW M NT and SELECT. The eigenvectors m ust be stored in consecutive colum ns of V , as retumed by CTGEVC. If JOB \(=V\) ', \(V L\) is not referenced.

LDVL (input)
The leading dim ension of the array VL. .LD VL \(>=1\); and If \(J 0 B=E\) 'or \(B\) ', LDVL \(>=N\).

VR (input)
If \(\mathrm{JO}=\mathrm{E}\) ' or B ', VR m ust contain right eigenvectors of \((A, B)\), comesponding to the eigenpairs specified by HOW M NT and SELECT. The eigenvectors \(m\) ust be stored in consecutive colum ns of V , as retumed by CTGEVC. If JOB \(=V^{\prime}, V R\) is not referenced.

LDVR (input)
The leading dim ension of the array VR.LD VR >= 1; If \(\mathrm{JOB}=\mathrm{E}\) 'or \(\mathrm{B}^{\prime}, \mathrm{LDVR}>=\mathrm{N}\).

S (output)
If \(J O B=E\) ' or \(B\) ', the reciprocal condition num bers of the selected eigenvalues, stored in consecutive elem ents of the array. If \(\mathrm{JOB}=\mathrm{V}\) ', \(S\) is not referenced.

D \(\mathbb{F}\) (output)
If \(\mathrm{JOB}=\mathrm{V}\) 'or B ', the estim ated reciprocalcondition numbers of the selected eigenvectors, stored in consecutive elem ents of the array. If the eigenvalues cannot be reordered to com pute D \(\mathbb{F}(\mathcal{J}), \mathrm{D} \mathbb{F}(\mathcal{)})\) is setto 0 ; this can only occur when the true value w ould be very sm allanyw ay. Foreach eigenvalue/vector specified by SELECT, D \(\mathbb{F}\) stores a Frobenius norm -based estim ate of \(D\) ifl. If \(J O B=E ', D \mathbb{F}\) is not referenced.

M M (input)
The num berofelem ents in the arays \(S\) and \(D \mathbb{F} . M\) M \(>=M\).

M (output)
The num berof elem ents of the amays \(S\) and D \(\mathbb{F}\) used to store the specified condition num bers; for each selected eigenvalue one elem ent is used. If HOWMNT=A', M is set to N.

\section*{W ORK (w orkspace)}

If \(J \mathrm{OB}=\mathrm{E}, \mathrm{W} \mathrm{ORK}\) is not referenced. O therw ise, on exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the aray \(W\) ORK. LW ORK >= 1. If JOB = V'or B', LW ORK >= 2*N *N.

IV ORK (w orkspace)
dim ension \((\mathbb{N}+2)\) If \(J O B=E ', \mathbb{I N} O R K\) is not referenced.
\(\mathbb{N} F O\) (output)
= 0 : Successfulexit
\(<0:\) If \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

The reciprocal of the condition num ber of the i-th generalized eigenvalue \(\mathrm{w}=(\mathrm{a}, \mathrm{b})\) is defined as
\[
S(I)=\left(\left|V^{A} u\right| * * 2+|v B u| * * 2\right)^{\star *}(1 / 2) /
\]
(norm (u)*norm (v))
\(w\) here \(u\) and \(v\) are the right and lefteigenvectors of ( \(A, B\) ) comesponding to w ; \(|z|\) denotes the absolute value of the com plex num ber, and norm (u) denotes the 2-norm of the vector \(u\). The pair ( \(a, b\) ) comesponds to an eigenvalue \(w=a \neq\) \(v A u / v B u\) ) of the \(m\) atrix pair ( \(A, B\) ). If both \(a\) and \(b\) equal zero, then \((A, B)\) is singular and \(S(I)=-1\) is retumed.

A n approxim ate errorbound on the chordal distance betw een the i-th computed generalized eigenvalue \(w\) and the comesponding exacteigenvalue lam bda is
\[
\text { chord }(w, \operatorname{lam} \text { bda) }<=\text { EPS * norm }(A, B) / S(I),
\]
where EPS is the m achine precision.
The reciprocal of the condition num ber of the right eigenvector \(u\) and lefteigenvectorv corresponding to the generalized eigenvalue w is defined as follow s. Suppose
\[
\begin{array}{rl}
(\mathrm{A}, \mathrm{~B}) & =(\mathrm{a} *)(\mathrm{b} *) 1 \\
(0 & \mathrm{A} 22),(0 \mathrm{~B} 22) \mathrm{n}-1 \\
1 & \mathrm{n}-1 \\
1 \mathrm{n}-1
\end{array}
\]

Then the reciprocalcondition num berD \(\mathbb{F}(\mathbb{I})\) is
\(D\) ifl \([(a, b),(A 22, B 22)]=\operatorname{sigm} \operatorname{arm}\) in ( \(Z 1\) )
where sigm a-m in (2l) denotes the sm allest singular value of
```

Zl= [kron (a, In-1)-kron(1,A 22)]
[kron (b, In-1) -kron (1,B22)].

```

H ere In - 1 is the identity m atrix of size \(\mathrm{n}-1\) and X ' is the conjugate transpose of \(\mathrm{X} . \mathrm{kron}(\mathrm{X}, \mathrm{Y})\) is the K roneckerproduct.betw een the \(m\) atrices \(X\) and \(Y\).

W e approxim ate the sm allest singularvalue of \(\mathrm{Zl} w\) ith an upperbound. This is done by C LA TD F.

A \(n\) approxim ate errorbound for a com puted eigenvector \(V \mathrm{~L}\) (i) orV R (i) is given by

EPS * norm ( \(\mathrm{A}, \mathrm{B}\) )/D \(\mathbb{F}\) (i).

See ref. [2-3] form ore details and further references.

B ased on contributions by
Bo K agstrom and PeterPorom aa, D epartm ent of Com puting Science,

Um ea U niversity, S-901 87 Um ea, Sw eden.

\section*{References}
==========
[1] B . K agstrom ; A D irectM ethod forReordering E igenvalues in the

Generalized RealSchurForm of a RegularM atrix Pair \(\mathbb{A}\), B), in

M S.M oonen etal (eds), LinearA lgebra for Large Scale and

Real-T im e A pplications, K luw erA cadem ic Publ. 1993, pp 195-218.
[2] B . K agstrom and P.Porom aa; C om puting Eigenspaces with Specified

E igenvalues of a RegularM atrix Pair ( \(\mathrm{A}, \mathrm{B}\) ) and Condition

Estim ation: Theory, A lgorithm s and Softw are, Report
UM INF - 94.04 , D epartm ent of Computing Science, Um ea U niversity,

S-901 87 U m ea, Sw eden, 1994. A lso as LA PA C K W orking N ote 87.

To appear in N um ericalA lgorithm s, 1996.
[3] B . K agstrom and P . Porom aa, LA PA C K -Style A lgorithm s and Softw are
for Solving the Generalized Sylvester Equation and Estim ating the

Separation betw een RegularM atrix Pairs, ReportUM IN F 93 23,

D epartm ent ofC om puting Science, Um ea U niversity, S-901 87 Um ea,

Sw eden, D ecem ber 1993, Revised A pril 1994, A lso as LAPACK W orking

N ote 75.
To appear in A CM Trans. on M ath. Softw are, V ol22, N o 1, 1996.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ctgsyl-solve the generalized Sylvesterequation

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NECTGSYL (TRANS,IDOB,M,N,A,LDA,B,LDB,C,LDC,D,LDD,}
E,LDE,F,LDF,SCALE,D \mathbb{F,W ORK,LW ORK,IN ORK,INFO)}
CHARACTER * 1 TRANS
COM PLEX A (LDA,*),B (LD B ,*), C (LD C ,*), D (LDD,*), E (LDE,*),
F (LDF,*),W ORK (*)
\mathbb{NTEGER LDOB,M ,N,LDA,LDB,LDC, LDD, LDE, LDF, LW ORK,}
\mathbb{NFO}

```

```

REAL SCALE,D F
SU BROUT\mathbb{NE CTGSYL_64 (TRANS,IWOB,M,N,A,LDA,B,LDB,C,LDC,D,}

```

CHARACTER * 1 TRANS
COM PLEX A (LDA \(\left.{ }^{\star}\right), \mathrm{B}(\mathrm{LDB}, \star), \mathrm{C}(\mathrm{LDC}, \star), \mathrm{D}(\mathrm{LDD}, \star), \mathrm{E}(\mathrm{LDE}, \star)\),
F (LDF, \({ }^{\star}\) ), W ORK ( \({ }^{\star}\) )
\(\mathbb{N}\) TEGER*8 IJOB,M,N,LDA,LDB,LDC,LDD,LDE, LDF, LW ORK,
\(\mathbb{N}\) FO
\(\mathbb{N} T E G E R * 8 \mathbb{I N}\) ORK (*)
REAL SCALE,D \(\mathbb{F}\)

\section*{F95 INTERFACE}

SU BROUTINE TGSYL (TRANS, IJBB, \(\mathbb{M}], \mathbb{N}], A,[L D A], B,[L D B], C,[L D C]\), D, [LDD ], E, [LDE],F, [LDF],SCALE,D \(\mathbb{F},\left[\begin{array}{l}\text { W ORK ], [LW ORK ], [IW ORK ], }\end{array}\right.\) [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::TRANS
COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX,D IM ENSION (:,:)::A,B,C,D,E,F
\(\mathbb{N} T E G E R:: I J O B, M, N, L D A, L D B, L D C, L D D, L D E, L D F, L W O R K\),
\(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
REAL ::SCALE,D \(\mathbb{F}\)

SU BROUTINE TGSY L_64 (TRANS, IJOB, M ], \(\mathbb{N}], A,[L D A], B,[L D B], C\), [LDC],D, [LDD ], E, [LDE],F, [LDF],SCALE,D \(\mathbb{F},\left[\begin{array}{l}\text { ORK], [LW ORK], }\end{array}\right.\) [ \(\mathbb{I N}\) ORK], \([\mathbb{N F O}])\)

CHARACTER (LEN=1)::TRANS
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D IM ENSION (:,:) ::A,B,C,D,E,F

LW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{I N}\) ORK
REAL ::SCALE,D \(\mathbb{F}\)

\section*{C INTERFACE}
\#include <sunperfh>
void ctgsyl(chartrans, int ijob, intm, intn, com plex *a, int Ida, com plex *b, int ldb, com plex *c, int ldc, com plex *d, int ldd, com plex *e, int lde, com plex *f, int ldff, float *scale, float *dif, int *info);
void ctgsyl 64 (chartrans, long ijob, long m, long n, com plex *a, long lda, com plex *b, long ldb, com plex *c, long ldc, com plex *d, long ldd, com plex *e, long lde, com plex *f, long ldf, float*scale, float *dif, long *info);

\section*{PURPOSE}
ctgsylsolves the generalized Sylvesterequation:
\(A * R-L * B=\) scale * C
\(D * R-L * E=\) scale \(* F\)
(1)
where \(R\) and \(L\) are unknow \(n m\)-by-n matrices, ( \(A\), D ), ( \(B\), E) and ( \(C, F\) ) are given \(m\) atrix pairs of size \(m-b y-m, n-b y-n\) and \(m\)-by-n, respectively, with com plex entries. A, B, D and E are upper triangular (ie., \((A, D)\) and \((B, E)\) in generalized Schur form ).

The solution ( \(\mathbb{R}, \mathrm{L}\) ) overw rites ( \(\mathrm{C}, \mathrm{F}\) ). \(0<=\) SCALE \(<=1\) is an output scaling factor chosen to avoid overflow .

In \(m\) atrix notation (1) is equivalent to solve \(\mathrm{Zx}=\) scale*b, where \(Z\) is defined as
```

Z = [kron(In,A ) -kron(B', Im )] (2)
[kron(In,D) -kron(E',Im)],

```

Here Ix is the identity m atrix of size x and X 'is the conjugate transpose of X . K ron \((X, Y)\) is the K roneckerproduct betw een the \(m\) atrices \(X\) and \(Y\).

If TRANS = C',y in the conjugate transposed system Z "y = scale*b is solved for, which is equivalent to solve for \(R\) and \(L\) in
\[
\begin{align*}
& A^{\prime} * R+D^{\prime} * L=\text { scale * C }  \tag{3}\\
& R^{*} B^{\prime}+L^{*} E^{\prime}=\text { scale } *
\end{align*}
\]

This case (TRANS = C ) is used to com pute an one-norm -based estim ate of \(D\) if \([(A, D),(B, E)]\), the separation betw een the \(m\) atrix pairs \((A, D)\) and \((B, E)\), using C LA CON .

If IJO \(B>=1, C T G S Y L\) com putes a \(F\) robenius norm -based esti\(m\) ate ofD if \([(A, D),(B, E)]\). That is, the reciprocalofa low er bound on the reciprocal of the sm allest singular value of \(Z\).

This is a level-3 B LA S algorithm .

\section*{ARGUMENTS}

TRANS (input)
= N ': solve the generalized sylvester equation
(1).
= C':solve the "conjugate transposed" system
(3).

IO B (input)
Specifies whatkind of functionality to be perform ed. \(=0\) : solve (1) only .
\(=1\) : The functionality of 0 and 3 .
\(=2\) :The functionality of 0 and 4 .
\(=3: 0\) nly an estim ate ofD if \([(A, D),(B, E)]\) is com-
puted. (look ahead strategy is used). \(=4: 0\) nly an estim ate of \(D\) if \([(A, D),(B, E)]\) is com puted.
(CGECON on sub-system s is used). N ot referenced ifTRANS = C \({ }^{\prime}\).
\(M\) (input) The order of the \(m\) atrices \(A\) and \(D\), and the row
dim ension of the \(m\) atrioes \(C, F, R\) and \(L\).

N (input) The order of the \(m\) atrioes \(B\) and E , and the \(c o l m m\) dim ension of the \(m\) atrices \(C, F, R\) and \(L\).

A (input) The upper triangularm atrix A .
LD A (input)
The leading dim ension of the aray A. LDA >= \(m a x(1, M)\).
\(B\) (input) The upper triangularm atrix \(B\).
LD B (input)
The leading dim ension of the array B. LD B >= max (1,N).

C (input/output)
On entry, \(C\) contains the right-hand-side of the first \(m\) atrix equation in (1) or (3). On exit, if IJO \(B=0,1\) or \(2, C\) has been overw ritten by the solution R. If \(I \mathcal{O D}=3\) or 4 and TRANS \(=N^{\prime}\) ', C holds R, the solution achieved during the com putation of the \(D\) if-estim ate.

LD C (input)
The leading dim ension of the array C. LD C >= max (1, M).

D (input) The uppertriangularm atrix D .

LD D (input)
The leading din ension of the anay D. LDD >= \(m a x(1, M)\).

E (input) The upper triangularm atrix E .

LDE (input)
The leading dim ension of the array E. LDE >= \(m a x(1, N)\).

\section*{F (input/output)}

O n entry, F contains the right-hand-side of the second \(m\) atrix equation in (1) or (3). On exit, if IJO B \(=0,1\) or \(2, \mathrm{~F}\) has been overw ritten by the solution L. If \(\mathrm{IO} B=3\) or 4 and TRANS \(=N^{\prime}, \mathrm{F}\) holds \(L\), the solution achieved during the com putation of the D if-estim ate.

LD F (input)

The leading dim ension of the array F. LDF >= max (1, M).

D \(\mathbb{F}\) (output)
On exitSCA LE is the reciprocalof a low er bound of the reciprocal of the \(D\) if-function, i.e. SCA LE is an upperbound ofD if \([(A, D),(B, E)]=\) sigm a\(m\) in ( \(Z\) ), where \(Z\) as in (2). If \(I O B=0\) orTRAN \(S=\) \(C^{\prime}\),SCA LE is not referenced.

SCALE (output)
On exitSCA LE is the reciprocalofa lower bound of the reciprocal of the \(D\) if-function, ie. SCA LE is an upperbound ofD if \([(A, D),(B, E)]=\operatorname{sigm}\) a\(m\) in ( \(Z\) ), where \(Z\) as in (2). If \(I O B=0\) orTRAN \(S=\) C',SCALE is notreferenced.

W ORK (w orkspace)
If \(\mathrm{IJOB}=0, \mathrm{~W} O R K\) is not referenced. O therw ise, on exit, if \(\mathbb{N F O}=0\) then \(W\) ORK (1) retums the optim allW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK.LW ORK \(>=1\). If \(\mathrm{IJOB}=1\) or 2 and TRANS \(=\mathrm{N}^{\prime}\), LW ORK >= \(2 * \mathrm{M} * \mathrm{~N}\) 。

If LW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

IW ORK (w orkspace)
If \(I O B=0, \mathbb{I N} O R K\) is not referenced.
\(\mathbb{I N F O}\) (output)
\(=0\) : successfulexit
\(<0:\) If \(\mathbb{N}\) FO \(=-\) - , the \(i\)-th argum ent had an illegal
value.
>0: \((A, D)\) and \((B, E)\) have com \(m\) on or very close eigenvalues.

\section*{FURTHER DETAILS}

B ased on contributions by
Bo K agstrom and PeterPorom aa, D epartm ent of C om puting
Science,
Um ea U niversity, S-901 87 Um ea, Sw eden.
[1] B. K agstrom and P. Porom aa, LA PA C K -Style A lgorithm \(s\) and

\section*{Softw are}
for Solving the Generalized Sylvester Equation and Estim ating the

Separation betw een RegularM atrix Pairs, ReportUM \(\mathbb{N} F\) 9323,

D epartm entofC om puting Science, Um ea U niversity, S-901 87 Um ea,

Sw eden, D ecem ber 1993, Revised A pril 1994, A lso as LAPACK W orking

N ote 75. To appear in A CM Trans. on \(M\) ath. Softw are, \(V\) ol 22,

No1,1996.
[2] B .K agstrom , A Perturbation A nalysis of the G eneralized Sylvester

Equation (AR -LB,DR -LE ) \(=(C, F)\), SIAM J. M atrix A nal.

A ppl, 15 (4):1045-1060, 1994.
[3] B . K agstrom and L. W estin, G eneralized Schur M ethods w ith

C ondition Estim ators for Solving the G eneralized Sylvester

Equation, \(\mathbb{E E E}\) Transactions on A utom atic C ontrol, V ol. 34, No.7, July 1989, pp 745-751.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctpcon - estim ate the reciprocal of the condition num ber of a packed triangular \(m\) atrix \(A\), in either the 1 -nom or the infinity-norm

\section*{SYNOPSIS}

```

CHARACTER * 1NORM,UPLO,DIAG
COM PLEX A (*),W ORK (*)
\mathbb{NTEGERN,\mathbb{NFO}}\mathbf{N}=0
REALRCOND
REALW ORK2 (*)
SUBROUT\mathbb{NECTPCON_64 NORM,UPLO,DIAG,N,A,RCOND,W ORK,W ORK2,}
INFO)
CHARACTER * 1NORM,UPLO,DIAG
COM PLEX A (*),W ORK (*)
INTEGER*8 N,INFO
REAL RCOND
REALW ORK2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE TPCON \(\mathbb{N} O R M, U P L O, D \mathbb{I} G, N, A, R C O N D,[\mathbb{W} O R K],[W O R K 2]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1)::NORM,UPLO,DIAG
COMPLEX,D \(\mathbb{I}\) ENSION (:) ::A,W ORK
\(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::W ORK2

SU BROUTINE TPCON_64 \(\mathbb{N} O R M, U P L O, D \mathbb{I} G, N, A, R C O N D,[\mathbb{O} O R K],[W O R K 2]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1)::NORM,UPLO,D IAG
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::A,W ORK
\(\mathbb{N}\) TEGER (8) ::N, \(\mathbb{N} F O\)
REAL ::RCOND
REAL,D \(\mathbb{I M} E N S I O N(:):: W\) ORK 2

\section*{C INTERFACE}
\#include < sunperfh>
void ctpcon (charnorm, charuplo, chardiag, intn, com plex *a, float * rcond, int *info);
void ctpcon_64 (charnorm , charuplo, chardiag, long n, com plex *a, float *rcond, long *info);

\section*{PURPOSE}
ctpcon estim ates the reciprocal of the condition num ber of a packed triangular matrix A, in eitherthe 1-norm orthe infinity-norm .

The norm ofA is com puted and an estim ate is obtained for norm (inv (A )), then the reciprocal of the condition num ber is com puted as
\(\operatorname{RCOND}=1 /(\operatorname{nom}(A) *\) norm (inv (A)) ).

\section*{ARGUMENTS}

NORM (input)
Specifies w hether the 1-nom condition num ber or the infinity-norm condition num ber is required:
= 1'or \(^{0}\) ': 1-nom ;
= I': Infinity-norm .

UPLO (input)
\(=\mathrm{U}\) : A is uppertriangular;
\(=\mathrm{L}\) ': A is low ertriangular.
D IA G (input)
\(=\mathrm{N}: \mathrm{A}\) is non-unit triangular;
\(=\mathrm{U}\) ': A is unit triangular.
\(N\) (input) The order of the m atrix \(A . N>=0\).

A (input) The upper or low er triangular m atrix A, packed colum nw ise in a linear array. The jth colum n of A is stored in the aray A as follow s: if UPLO = \(U^{\prime}, A(i+(j-1) * j 2)=A(i, j)\) for \(1<=i<=j\); if \(\mathrm{UPLO}=\mathrm{L}, \mathrm{A}(i+(j-1) *(2 n-j) / 2)=A(i, j)\) for \(j=i<=n\). IfD \(\mathbb{A} G=U\) ', the diagonalelem ents of \(A\) are not referenced and are assum ed to be 1 .

RCOND (output)
The reciprocal of the condition num ber of the \(m\) atrix \(A\), computed as RCOND \(=1 /(\) noim (A) * norm (inv (A))).
W ORK (w orkspace)
dim ension \((2 * N)\)

W ORK 2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctpm \(v\)-perform one of the \(m\) atrix-vectoroperations \(x\) := \(A{ }^{*} x\), or \(x: A{ }^{*} x\), orx \(:=\operatorname{cong}\left(A^{\prime}\right){ }^{*} x\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NECTPMV (UPLO,TRANSA,D IAG,N,A,Y, INCY)}
CHARACTER * 1UPLO,TRANSA,DIAG
COM PLEX A (*),Y (*)

```


```

CHARACTER * 1 UPLO,TRANSA,D IAG
COM PLEX A (*),Y (*)
INTEGER*8 N,\mathbb{NCY}

```
F95 INTERFACE
    SU BROUTINE TPM V (UPLO, [TRANSA ], D \(\mathbb{I A} G, \mathbb{N}], A, Y,[\mathbb{N C Y}])\)
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
    COM PLEX,D \(\mathbb{I}\) ENSION (:) ::A, Y
    \(\mathbb{N} T E G E R:: N, \mathbb{N C Y}\)
    SU BROUTINE TPM V_64 (UPLO, [TRANSA ],D \(\mathbb{I A G}, \mathbb{N}], A, Y,[\mathbb{N} C Y])\)
    CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
    COMPLEX,D \(\mathbb{I}\) ENSION (:) ::A,Y
    \(\mathbb{N}\) TEGER ( 8 ) :: \(\mathrm{N}, \mathbb{N} C Y\)
C INTERFACE
    \#include <sunperfh>
void ctpm v (charuplo, chartransa, chardiag, intn, com plex
*a, com plex *y, intincy);
void ctpm v_64 (charuplo, chartransa, char diag, long n, com plex *a, com plex *y, long incy);

\section*{PURPOSE}
ctpm v perform s one of the \(m\) atrix-vector operations \(x:=A * x\), or \(x:=A{ }^{*} x\), or \(x:=c o n \dot{g}\left(A^{\prime}\right)^{\star} x w h e r e x\) is an \(n\) elem ent vector and \(A\) is an \(n\) by \(n\) unit, ornon-unit, upper or low er triangularm atrix, supplied in packed form.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies w hether the \(m\) atrix is an upper or low er triangularm atrix as follow s:

UPLO = U'or 4 ' A is an upper triangular \(m\) atrix.

UPLO = L' or 1 ' A is a lower triangular m atrix.

U nchanged on exit.
TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) 'or \(h^{\prime} \mathrm{x}:=\mathrm{A} \mathrm{*}_{\mathrm{x}}\).

TRANSA \(=\) T'ort' \(x:=A * x\).

TRANSA \(=\) C'or \(\mathrm{E}^{\prime} \mathrm{x}:=\operatorname{conjg}\left(\mathrm{A}^{\prime}\right) \star \mathrm{x}\).
U nchanged on exit.

TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
O n entry, D IA G specifies w hether ornotA is unit triangular as follow s:

D \(\mathbb{I}\) G \(=\) U 'or L ' \(A\) is assum ed to be unit tri-
angular.
\(D \mathbb{A G}=N\) 'or \(h\) ' \(A\) is notassum ed to be unit triangular.

U nchanged on exit.

N (input)
O n entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.

A (input)
( \((n *(n+1)) / 2)\). Before entry with UPLO = \(U\) ' or \(L\) ', the amay A m ustcontain the upper triangularm atrix packed sequentially, colum \(n\) by colum \(n\), so thatA (1) contains a (1, 1), A (2) and \(A(3)\) contain \(a(1,2)\) and \(a(2,2)\) respectively, and so on. Before entry w ith UPLO = L' or I', the anray A m ust contain the low er triangular m atrix packed sequentially, colum \(n\) by colum \(n\), so thatA (1) contains a (1,1),A(2) and \(A(3)\) contain \(a(2,1)\) and \(a(3,1)\) respec tively, and so on. N ote thatw hen D IA G \(=U U^{\prime}\) or \(G\) ', the diagonal elem ents of A are notreferenced, butare assum ed to be unity. U nchanged on exit.

Y (input/output)
\((1+(n-1) * a b s(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) must contain the \(n\) elem ent vectorx. O n exit, \(Y\) is overw ritten \(w\) ith the tranform ed vector \(x\).
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(. \mathbb{N} C Y<>0\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctprfs -provide enrorbounds and backw ard enror estim ates for the solution to a system of linearequations \(w\) th a triangular packed coefficientm atrix

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CTPRFS (UPLO,TRANSA,D IAG,N,NRHS,A,B,LDB,X,LDX,}
FERR,BERR,W ORK,W ORK2,INFO)
CHARACTER * 1 UPLO,TRANSA,D IAG
COM PLEX A (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
REAL FERR (*),BERR (*),WORK2 (*)
SU BROUT\mathbb{NECTPRFS_64 (UPLO,TRANSA,D IAG,N,NRHS,A,B,LDB,X,LDX,}
FERR,BERR,W ORK,W ORK2, INFO)
CHARACTER * 1UPLO,TRANSA,DIAG
COM PLEX A (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGER*8N,NRHS,LDB,LDX,\mathbb{NFO}
REAL FERR (*),BERR (*),WORK2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE TPRFS (UPLO, [TRANSA],D IAG,N, NRHS],A,B,[LDB],X, [LDX],FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1)::UPLO,TRANSA,D IA G
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::A,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) :: B, X
\(\mathbb{N}\) TEGER :: N,NRHS,LDB,LDX, \(\mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (: : : FERR,BERR,W ORK 2

SU BROUTINE TPRFS_64 (UPLO, [TRANSA],D \(\mathbb{I} G, N, \mathbb{N} R H S], A, B,[L D B], X\), [LDX],FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::UPLO,TRANSA,DIAG
COM PLEX,D IM ENSION (:) ::A,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : : B , X
\(\mathbb{N} T E G E R(8):: N, N R H S, L D B, L D X, \mathbb{N} F O\)
REAL,D IM ENSION (:) ::FERR,BERR,WORK2

\section*{C INTERFACE}
\#include < sunperfh>
void ctprfs (char uplo, char transa, chardiag, int n, int nuhs, com plex *a, com plex *b, int ldb, com plex *x, int ldx, float * ferrs, float *berr, int *info);
void ctprfs_64 (charuplo, chartransa, char diag, long n, long nihs, com plex *a, com plex *b, long ldb, com plex *x, long ldx, float *ferr, float *berr, long *info);

\section*{PURPOSE}
ctpris provides errorbounds and backw ard error estim ates for the solution to a system of linear equations \(w\) th a triangular packed coefficientm atrix.

The solution \(m\) atrix \(X\) m ustbe com puted by CTPTRS or some other \(m\) eans before entering this routine. C TPRFS does not do iterative refinem entbecause doing so cannotim prove the backw ard error.

\section*{ARGUMENTS}

UPLO (input)
= U ': A is uppertriangular;
\(=\mathrm{L}\) ': A is low ertriangular.

TRANSA (input)
Specifies the form of the system of equations:
\(=N^{\prime}: A * X=B \quad\) N \(\circ\) transpose)
\(=T\) ': A ** \(T\) * \(\mathrm{X}=\mathrm{B}\) ( T ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose)
TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.
D IA G (input)
\(=N^{\prime}\) ': A is non-unit triangular;
\(=\mathrm{U}: \mathrm{A}\) is unit triangular.
\(N\) (input) The order of the matrix A. \(N>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the m atrices \(B\) and X. NRH \(S>=0\).

A (input) The upper or low er triangular m atrix A, packed colum nw ise in a linear anray. The jth colum n of A is stored in the amay A as follow s: if UPLO = \(U^{\prime}, A(i+(j-1) * j 2)=A(i, j)\) for \(1<=i<=j\) if UPLO = L', A ( \(\left.i+(j-1)^{*}(2 n-j) / 2\right)=A(i, 7)\) for \(j=i<=n\). If \(\mathbb{A} G=U\) ', the diagonalelem ents of A are not referenced and are assum ed to be 1 .
\(B\) (input) The righthand side \(m\) atrix \(B\).
LD B (input)
The leading dim ension of the aray B. LD B \(>=\) \(\max (1, \mathbb{N})\).
\(X\) (input) The solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the amay X . LD X >= \(\max (1, \mathbb{N})\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X()\) ) the \(j\) th colum n of the solution \(m\) atrix X). If XTRUE is the true solution comesponding to \(X(\mathcal{O})\), FERR ( \()\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{D})-X\) TRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each
solution vectorX ( \(\mathcal{j}\) ) (i.e., the sm allest relative change in any elem entofA orB thatm akes X (i) an exactsolution).

W ORK (w orkspace)
dim ension \((2 * N)\)
W ORK 2 (w orkspace)
dim ension ( N )
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctpsv - solve one of the system sofequations \(A * x=b\), or \(A^{*} \mathrm{x}=\mathrm{b}\), orcong \(\left(\mathrm{A}^{\prime}\right)^{*} \mathrm{x}=\mathrm{b}\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CTPSV (UPLO,TRANSA,DIAG,N,A,Y, INCY)}

```
CHARACTER * 1 UPLO, TRANSA, D IA G
COM PLEX A (*), Y (*)
\(\mathbb{N} T E G E R N, \mathbb{N} C Y\)
SUBROUTINECTPSV_64 (UPLO, TRANSA, D \(\mathbb{A} G, N, A, Y, \mathbb{N} C Y\) )
CHARACTER * 1 UPLO, TRANSA,D IA G
COMPLEXA (*), Y (*)
\(\mathbb{N} T E G E R * 8 N, \mathbb{N C Y}\)

\section*{F95 INTERFACE}

SU BROUTINE TPSV (UPLO, [TRANSA ], D \(\mathbb{A} G, \mathbb{N}], A, Y,[\mathbb{N C Y}])\)
CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
COM PLEX,D IM ENSION (:) ::A, Y
\(\mathbb{N} T E G E R:: N, \mathbb{N C Y}\)
SU BROUTINE TPSV_64 (UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{N}], A, Y,[\mathbb{N} C Y])\)
CHARACTER (LEN=1)::UPLO,TRANSA,D IA G COMPLEX,D \(\mathbb{I}\) ENSION (:) ::A,Y
\(\mathbb{N} T E G E R(8):: N, \mathbb{N} C Y\)

\section*{C INTERFACE}
\#include <sunperfh>
void ctpss (char uplo, chartransa, chardiag, intn, com plex *a, com plex *y, intincy);
void ctpsv_64 (charuple, chartransa, char diag, long n, com plex *a, com plex *y, long incy);

\section*{PURPOSE}
ctpsv solves one of the system s of equations \(A * x=b\), or \(A{ }^{*} x=b\), or con \(\dot{g}\left(A^{\prime}\right)^{\star} x=b\) where \(b\) and \(x\) are \(n\) elem ent vectors and \(A\) is an \(n\) by \(n\) unit, ornon-unit, upper or low er triangularm atrix, supplied in packed form.

N o testforsingularity or near-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.

\section*{ARGUMENTS}

UPLO (input)
O n entry, UPLO specifies w hether the \(m\) atrix is an upper or low er triangularm atrix as follow \(s\) :
\(\mathrm{UPLO}=\mathrm{U}\) 'or \(\mathrm{U}^{\prime} \mathrm{A}\) is an upper triangular m atrix .
\(\mathrm{UPLO}=\mathrm{L}\) ' or \(\mathrm{I}^{\prime}\) A is a lower triangular \(m\) atrix.

U nchanged on exit.

TRANSA (input)
On entry, TRANSA specifies the equations to be solved as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} A{ }^{*} \mathrm{x}=\mathrm{b}\).

TRANSA \(=T\) 'ort' \(A{ }^{*} \mathrm{x}=\mathrm{b}\).

TRANSA \(=C^{\prime}\) or \(t^{\prime} \operatorname{cong} \dot{g}\left(A^{\prime}\right) \star x=b\).

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)

On entry, D IA G specifies whether ornotA is unit triangular as follow s:

D \(\mathbb{A G G}=\mathrm{U}\) 'or \(\mathrm{L}^{\prime} \mathrm{A}\) is assum ed to be unit triangular.
\(D \mathbb{A} G=N\) 'or \(h\) ' \(A\) is notassumed to be unit triangular.

U nchanged on exit.

N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.
A (input)
\(\left(\left(\mathrm{n}^{*}(\mathrm{n}+1)\right) / 2\right)\). Before entry w th \(\mathrm{UPLO}=\) \(U\) ' or U ', the array A m ustcontain the upper triangularm atrix packed sequentially, colum \(n\) by colum n , so that A (1) contains a ( 1,1 ), A (2) and \(A(3)\) contain \(a(1,2)\) and \(a(2,2)\) respectively, and so on. Before entry w ith UPLO = \(\mathrm{L}^{\prime}\) or 1 ', the amay A m ust contain the low er triangular \(m\) atrix packed sequentially, colum \(n\) by colum n, so thatA (1) contains a (1,1), A (2) and \(A\) ( 3 ) contain a \((2,1)\) and a \((3,1)\) respectively, and so on. N ote thatw hen D IA G \(=\) U' or G ', the diagonal elem ents of A are notreferenced, but are assum ed to be unity. Unchanged on exit.

Y (input/output)
\((1+(n-1) * a b s(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y \mathrm{~m}\) ust contain the n elem ent righthand side vectorb. O \(n\) exit, \(Y\) is overw ritten \(w\) ith the solution vector \(x\).
\(\mathbb{N} C Y\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of. \(\mathbb{I N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ctptri-com pute the inverse of a com plex upper or low er triangularm atrix A stored in packed form at

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CTPTRI(UPLO,D IAG,N,A,\mathbb{N FO)}}\mathbf{N}\mathrm{ (N)}
CHARACTER * 1 UPLO,DIAG
COMPLEX A (*)
INTEGERN,\mathbb{NFO}
SU BROUT\mathbb{NE CTPTRI_64 (UPLO,D IA G,N ,A , NNFO )}
CHARACTER * 1 UPLO,DIAG
COM PLEX A (*)
INTEGER*8 N, INFO
F95 INTERFACE
SUBROUT\mathbb{NE TPTRI(UPLO,D IAG,N,A,[\mathbb{NFO ])}}\mathbf{(})=
CHARACTER (LEN=1) ::UPLO,D IAG
COMPLEX,D IM ENSION (:) ::A
\mathbb{NTEGER ::N,\mathbb{NFO}}\mathbf{N}=
SU BROUTINE TPTRI_64 (UPLO,D IA G ,N,A, [\mathbb{NFO ])}
CHARACTER (LEN=1) ::UPLO,D IAG
COMPLEX,D IM ENSION (:) ::A
\mathbb{NTEGER (8)::N,\mathbb{NFO}}\mathbf{~}\mathrm{ ( }

```
void ctptri(char uple, chardiag, int n, com plex *a, int *info);
void ctptri_64 (charuplo, chardiag, long n, complex *a, long *info);

\section*{PURPOSE}
ctptri com putes the inverse of a com plex upper or low er triangularm atrix A stored in packed form at.

\section*{ARGUMENTS}
```

UPLO (input)
= U ':A is uppertriangular;
= IL':A is low ertriangular.

```

D IA G (input)
\(=\mathrm{N}: A\) is non-unittriangular;
\(=\mathrm{U}: \mathrm{A}\) is unit triangular.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the upper or low er triangularm atrix A, stored colum nw ise in a linearamay. The jth colum \(n\) of \(A\) is stored in the array \(A\) as follow \(s\) : if UPLO \(=U ', A(i+(j-1) * j 2)=A(i, j)\) for \(1<=i<=j\) ifUPLO \(=\mathbb{L}, A(i+(j 1) \star((2 * n-j) / 2)=\) \(A(i, j)\) for \(j<=i<=n\). See below for further details. On exit, the (triangular) inverse of the original \(m\) atrix, in the sam e packed storage format.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=i\), \(A(i, i)\) is exactly zero. The triangular \(m\) atrix is singular and its inverse can notbe com puted.

\section*{FURTHER DETAILS}

A triangularm atrix A can be transferred to packed storage using one of the follow ing program segm ents:
\(\mathrm{UPLO}=\mathrm{U} ': \quad \mathrm{UPLO}=\mathrm{L}^{\prime}:\)
J \(=1\)
DO \(2 \mathrm{~J}=1\), N
\(\pi=1\)
DO \(2 \mathrm{~J}=1\), N
D○ \(1 \mathrm{I}=1\), J
DO \(1 \mathrm{I}=\mathrm{J}, \mathrm{N}\)
\(A(J C+I-1)=A(I, J) \quad A(J C+I-J)=\)
A (I, N)
\(\begin{array}{ll}1 & \text { CONTINUE } \\ \mathbb{C}=\mathrm{C}+\mathrm{J} & \mathbb{C O N T I N U E} \\ & \mathbb{C}=\mathrm{J}+\mathrm{N}-\mathrm{J}+\end{array}\)
1
2 CONTINUE \(\quad 2\) CONTINUE

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctptrs-solve a triangular system of the form \(A * X=B\), \(\mathrm{A} *{ }^{*} \mathrm{~T} * \mathrm{X}=\mathrm{B}\),orA \({ }^{*}{ }^{*} \mathrm{H} * \mathrm{X}=\mathrm{B}\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CTPTRS(UPLO,TRANSA,D IAG,N,NRHS,A,B,LDB, INFO)}
CHARACTER * 1 UPLO,TRANSA,DIAG
COM PLEX A (*),B (LDB,*)
INTEGER N,NRHS,LDB,INFO
SUBROUT\mathbb{NECTPTRS_64(UPLO,TRANSA,D IAG,N,NRHS,A,B,LDB,INFO)}
CHARACTER * 1 UPLO,TRANSA,D IAG
COM PLEX A (*),B (LD B,*)
INTEGER*8N,NRHS,LDB,INFO

```

\section*{F95 INTERFACE}

SU BROUTINE TPTRS (UPLO, TRANSA,D \(\mathbb{I A G}, N, \mathbb{N} R H S], A, B,[L D B],[\mathbb{N} F O])\)
CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
COMPLEX,D \(\mathbb{I M} E N S I O N(:):: A\)
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : : B
\(\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O\)
SU BROUTINE TPTRS_64 (UPLO,TRANSA,D \(\mathbb{I A} G, N, \mathbb{N} R H S], A, B,[L D B],[\mathbb{N} F O])\)
CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
COMPLEX,D IM ENSION (:) ::A
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::B
\(\mathbb{N}\) TEGER (8) :: N,NRHS,LD B, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void ctptrs (charuple, chartransa, chardiag, int \(n\), int nihs, com plex *a, com plex *b, int ldb, int *info);
void ctptrs_64 (charuplo, chartransa, char diag, long n, long nihs, com plex *a, com plex *b, long ldb, long *info);

\section*{PURPOSE}
ctpters solves a triangular system of the form
where \(A\) is a triangularm atrix of orderN stored in packed form at, and B is an N boy-NRH S m atrix. A check ism ade to verify thatA is nonsingular.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}:\) A is upper triangular;
\(=\mathbb{L}\) ': A is low ertriangular.

TRANSA (input)
Specifies the form of the system of equations:
\(=\mathrm{N}: A * \mathrm{X}=\mathrm{B} \quad\) (Notranspose)
\(=T\) ': \(A * * T * X=B \quad\) ( ranspose)
\(=C: A * * H * X=B \quad\) (C onjugate transpose)

D IA G (input)
\(=\mathrm{N}: A\) is non-unit triangular;
\(=U\) : A is unittriangular.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS \(>=0\).

A (input) The upper or low er triangular matrix A , packed colum nw ise in a linear array. The jth colum n of A is stored in the amay A as follow s: if UPLO = \(U ', A(i+(j-1) * j 2)=A(i, j)\) for \(1<=i<=j\) if \(\mathrm{UPLO}=\mathrm{L}, \mathrm{A}(i+(j 1) \star(2 \star \mathrm{n}-\mathrm{j} / 2)=A(i, j)\) for \(j=i<=n\).

B (input/output)
On entry, the righthand side \(m\) atrix B. On exit, if \(\mathbb{N} F O=0\), the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array \(B\). LD B >= \(\max (1, N)\).
\(\mathbb{I N} F O\) (output)
= 0 : successfinlexit
\(<0\) : if \(\mathbb{N F O}=-i\), the \(i\)-th argum ent had an illegal value
> \(0:\) if \(\mathbb{N} F O=i\), the \(i\)-th diagonalelem ent of \(A\)
is zero, indicating that the \(m\) atrix is singular and the solutions \(X\) have notbeen com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

ctrans - transpose and scale sourcem atrix

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CTRANS(PLACE,SCALE,SOURCE,M ,N,DEST)}
CHARACTER * 1 PLACE
COM PLEX SCALE
COM PLEX SOURCE (*),DEST (*)
\mathbb{NTEGERM ,N}
SUBROUTINECTRANS_64(PLACE,SCALE,SOURCE,M,N,DEST)
CHARACTER * 1 PLACE
COM PLEX SCALE
COM PLEX SOURCE (*),DEST (*)
INTEGER*8M ,N

```
F95 INTERFACE
    SUBROUTINE TRANS ([PLACE],SCALE,SOURCE,M,N, DEST])
    CHARACTER (LEN=1) ::PLACE
    COM PLEX ::SCALE
    COMPLEX,D \(\mathbb{I M}\) ENSION (:) ::SOURCE,DEST
    \(\mathbb{N} T E G E R:: M, N\)
    SU BROUTINE TRANS_64 (PLACE],SCALE,SOURCE,M,N, DEST])
    CHARACTER (LEN=1) ::PLACE
    COMPLEX ::SCALE
    COMPLEX,D \(\mathbb{I M} E N S I O N(:):: S O U R C E, D E S T\)
    \(\mathbb{N} T E G E R(8):: M, N\)

\section*{C INTERFACE}
\#include <sunperfh>
void ctrans (charplace, com plex *scale, com plex *source, int \(m\), intn, com plex *dest);
void ctrans_64 (charplace, com plex *scale, com plex *source, long \(m\), long \(n\), com plex *dest);

\section*{PURPOSE}
ctrans scales and transposes the source m atrix. The N \(2 \times \mathrm{N} 1\) result is w ritten into SO U RCE when PLACE = I'or 'i', and DEST when PLACE = 0 'or b'.
PLACE = 'I'or \({ }^{1}\) ': SOURCE = SCALE * SOURCE'
PLACE = O'orb':DEST = SCALE * SOURCE'

\section*{ARGUMENTS}

PLACE (input)
Type of transpose. 'I'or i'for in-place, \(0^{\prime}\) or \(b\) 'for out-of-place. ' I ' is default.

SCALE (input)
Scale factor on the SO U RCE m atrix.
SOURCE (input/output)
on input. A may of ( \(\mathrm{N}, \mathrm{M}\) ) on output if in-place
transpose.
\(M\) (input)
N um ber of row \(s\) in the SO U RCE m atrix on input.
\(N\) (input)
\(N\) um ber of colum ns in the SO U RCE m atrix on input.
DEST (output)
Scaled and transposed SOURCE m atrix if out-ofplace transpose. N ot referenced if in-place transpose.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctrcon -estim ate the reciprocal of the condition num ber of a triangular \(m\) atrix \(A\), in either the 1 -norm or the infinity-norm

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NECTRCON NORM,UPLO,D IAG,N,A,LDA,RCOND,W ORK,W ORK 2,}
\mathbb{NFO)}
CHARACTER * 1 NORM,UPLO,DIAG
COM PLEX A (LDA,*),W ORK (*)
INTEGERN,LDA, INFO
REAL RCOND
REAL W ORK2 (*)
SUBROUT\mathbb{NECTRCON_64 NORM,UPLO,DIAG,N,A,LDA,RCOND,W ORK,W ORK2,}
\mathbb{NFO)}
CHARACTER * 1NORM,UPLO,DIAG
COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGER*8N,LDA,}\mathbb{NNO}
REALRCOND
REAL W ORK 2 (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE TRCON $\mathbb{N} O R M, U P L O, D \mathbb{I A G}, \mathbb{N}], A,[L D A], R C O N D,[W$ ORK ], [W ORK 2], [ $\mathbb{N} F O$ ])
CHARACTER (LEN=1) ::NORM,UPLO,D IAG
COMPLEX,D $\mathbb{I M}$ ENSION (:) ::W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R:: N, L D A, \mathbb{N} F O$

```

SU BROUTINE TRCON_64 \(\mathbb{N} O R M, U P L O, D \mathbb{A} G, \mathbb{N}], A,[L D A], R C O N D,[W O R K]\), [W ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::NORM,UPLO,DIAG
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER ( 8 ) :: \(\mathrm{N}, \mathrm{LD} \mathrm{A}, \mathbb{N} F \mathrm{O}\)
REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void ctroon (charnoim , charuplo, chardiag, intn, com plex *a, int lda, float * rcond, int *info);
void ctroon_64 (charnorm , charuplo, chardiag, long n, com plex *a, long lda, float * roond, long *info);

\section*{PURPOSE}
ctrcon estim ates the reciprocal of the condition num ber of a triangular \(m\) atrix \(A\), in either the 1 -norm orthe infinitynorm .

The norm ofA is com puted and an estim ate is obtained for norm (inv (A )), then the reciprocal of the condition num ber is com puted as
\(\operatorname{RCOND}=1 /(\) norm (A) * norm (inv (A))).

\section*{ARGUMENTS}

NORM (input)
Specifies w hether the 1-nom condition num ber or the infinity-norm condition num ber is required:
= 1 'or \(\mathrm{O}^{\prime}\) : 1 -nom ;
= I': Infinity-norm .
UPLO (input)
\(=\mathrm{U}:\) : A is uppertriangular;
= L ': A is low er triangular.
D IA G (input)
\(=\mathrm{N}: \mathrm{A}\) is non-unit triangular;
\(=\mathrm{U}\) ': A is unit triangular.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input) The triangularm atrix \(A\). If \(\mathrm{UPLO}=\mathrm{U}\) ', the leading N -by N upper triangularpart of the array A contains the upper triangular matrix, and the strictly low ertriangularpartofA is not referenced. If UPLO = L', the leading N -by N lower triangular part of the array A contains the low er triangularm atrix, and the strictly uppertriangular part ofA is not referenced. IfD \(\mathbb{I A} G=U\) ', the diagonalelem ents ofA are also not referenced and are assum ed to be 1 .
LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, \mathbb{N})\).

RCOND (output)
The reciprocal of the condition number of the \(m\) atrix \(A\), com puted as RCOND \(=1 /(\) norm (A) * norm (inv (A))).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{I N}\) FO = -i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ctrevc - com pute som e or all of the right and/or lefteigen-
vectors of a com plex upper triangularm atrix \(T\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CTREVC (SDE,HOW M NY,SELECT,N,T,LDT,VL,LDVL,VR,}
LDVR,MM,M,W ORK,RWORK,\mathbb{NFO)}

```
CHARACTER * 1 SDE D ,HOW MNY

\(\mathbb{N}\) TEGER \(N, L D T, L D V L, L D V R, M M, M, \mathbb{N} F O\)
LO G ICAL SELECT (*)
REAL RW ORK ( \({ }^{*}\) )
SU BROUTINE CTREVC_64 (SDE,HOW MNY,SELECT,N,T,LDT,VL,LDVL,VR,
    LDVR,MM,M,W ORK,RWORK, \(\mathbb{N} F O)\)
CHARACTER * 1 SDE EHOW M NY
COM PLEX \(T(L D T, \star), V L(L D V L, \star), V R(L D V R, \star), W O R K(*)\)
\(\mathbb{N} T E G E R * 8 N, L D T, L D V L, L D V R, M M, M, \mathbb{N F O}\)
LO G ICAL*8 SELECT (*)
REAL RW ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE TREVC (SDE,HOW M NY, SELECT, \(\mathbb{N}], T,[L D T], V L,[L D V L], V R\), [LDVR],MM,M,[WORK], RW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SDE,HOW M NY
COMPLEX,D \(\mathbb{M}\) ENSION (:) ::W ORK
COMPLEX,D \(\mathbb{M}\) ENSION (:,:) ::T,VL,VR
\(\mathbb{N}\) TEGER ::N,LDT,LDVL,LDVR,MM,M, \(\mathbb{N} F O\)

LOG ICAL,D \(\mathbb{I M}\) ENSION (:) ::SELECT
REAL,D \(\mathbb{I M} E N S I O N(:):: R W\) ORK
SU BROUTINE TREVC_64 (SDE,HOW MNY,SELECT, \(\mathbb{N}], T,[L D T], V L,[L D V L]\), VR, [LDVR], MM, M, [W ORK], RW ORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::SDE,HOW MNY
COMPLEX,D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::T,VL,VR
\(\mathbb{N} T E G E R(8):: N, L D T, L D V L, L D V R, M M, M, \mathbb{N F O}\)
LOG ICAL (8), D IM ENSION (:) ::SELECT
REAL,D \(\mathbb{M}\) ENSION (:) ::RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void ctrevc (char side, char how m ny, int *select, intn, com plex *t, intldt, com plex *vl, int ldvl, com plex
*vr, int ldvr, intm \(m\), int *m, int *info);
void ctrevc_64 (charside, charhow m ny, long *select, long n, com plex *t, long ldt, com plex *vl, long ldvl, com plex *vr, long ldvr, long mm, long *m, long *info);

\section*{PURPOSE}
ctrevc com putes som e orallof the rightand/or left eigenvectors of a com plex upper triangularm atrix T .

The righteigenvector \(x\) and the left eigenvector \(y\) of \(T\) comesponding to an eigenvalue w are defined by:
\[
\mathrm{T}^{*} \mathrm{x}=\mathrm{w}^{*} \mathrm{x}, \quad \mathrm{Y} \mathrm{y}^{*} \mathrm{~T}=\mathrm{w}^{*} \mathrm{y}^{\prime}
\]
where \(y\) 'denotes the conjugate transpose of the vectory.
If alleigenvectors are requested, the routine \(m\) ay either retum the \(m\) atrices \(X\) and/or \(Y\) of rightor lefteigenvectors of \(T\), or the products \(Q * X\) and/or \(Q * Y\), where \(Q\) is an input unitary
\(m\) atrix. If \(T\) w as obtained from the Schur factorization of an original \(m\) atrix \(A=Q * T * Q\) ', then \(Q * X\) and \(Q * Y\) are the \(m\) atrices of right or lefteigenvectors of \(A\).

\section*{ARGUMENTS}
\(=\mathrm{R}^{\prime}\) : com pute righteigenvectors only;
= L ': com pute lefteigenvectors only;
= B ': com pute both right and lefteigenvectors.

HOW M NY (input)
= A ': com pute all right and/or left eigenvectors;
= B ': com pute all right and/or left eigenvectors, and backtransform them using the input m atrices supplied in VR and/orV L; = S ': com pute selected right and/or lefteigenvectors, specified by the logicalanay SELECT .

\section*{SELECT (input/output)}

If H OW M NY = S', SELEC T specifies the eigenvectors to be com puted. IfHOW M NY = A 'or B', SELECT is not referenced. To select the eigenvector comesponding to the \(j\) th eigenvalue, SELECT ( 1 ) m ustbe set to .TRUE..

N (input) The order of the matrix \(\mathrm{T} . \mathrm{N}>=0\).
T (input/output)
The upper triangularm atrix T. T ism odified, but restored on exit.

LD T (input)
The leading dim ension of the array T. LD T >= \(\max (1, \mathbb{N})\).

VL (input/output)
On entry, if \(S D E=L\) 'or \(B\) 'and HOWMNY = \(B^{\prime}\) ', VL must contain an \(N\) by N m atrix Q (usually the unitary \(m\) atrix \(Q\) of Schur vectors retumed by CHSEQR). On exit, if \(S \mathbb{D} E=\) L'or \(B^{\prime}, V L\) contains: if HOW MNY = A', the matrix Y of left eigenvectors of ; VL is low ertriangular. The ith column VL (i) of VL is the eigenvector corresponding to \(\mathrm{T}(i, i)\). if HOW MNY = B', the \(m\) atrix \(Q * Y\); if HOW M NY = \(S^{\prime}\), the lefteigenvectors of \(T\) specified by SELEC \(T\), stored consecutively in the colum ns ofVL, in the same order as their eigenvalues. If \(S \mathbb{D} E=R\) ', \(V L\) is notreferenced.

LD V L (input)
The leading dim ension of the array VL. LDVL >= \(\max (1, N)\) if \(S \mathbb{D} E=L\) 'or \(B^{\prime}\) ';LDVL \(>=1\) otherw ise.

VR (input/output)

On entry, if \(S D E=R\) 'or \(B\) 'and \(H O W M N Y=B\) ', VR m ust contain an \(N\)-by-N \(m\) atrix \(Q\) (usually the unitary \(m\) atrix \(Q\) of Schur vectors retumed by CHSEQR). On exit, if \(S \mathbb{D} E=R\) 'or \(B \prime, V R\) contains: ifHOW MNY = A', the \(m\) atrix \(X\) of right eigenvectors of \(T\); \(V R\) is uppertriangular. The ith column VR (i) of VR is the eigenvector comesponding to \(T(i, i)\). if HOW MNY \(=\mathrm{B}^{\prime}\), the \(m\) atrix \(Q * X\); if HOW MNY \(=S^{\prime}\), the right eigenvectors of \(T\) specified by SELECT, stored consecutively in the colum ns of \(V R\), in the sam e order as their eigenvalues. If \(S \mathbb{D} E=L^{\prime}, V R\) is not referenced.
LDVR (input)
The leading dim ension of the amay VR. LDVR >= \(\max (1, N)\) if \(S \mathbb{D} E=R\) 'or \(B\) '; LDVR \(>=1\) other w ise.

\section*{M M (input)}

The num berof \(\infty\) lum ns in the anrays \(V L\) and/or VR. \(M M>=M\).

M (output)
The num berof colum ns in the amays VL and/or VR actually used to store the eigenvectors. If HOW M NY = A 'or B', M is set to N. Each selected eigenvector occupies one colum \(n\).

W ORK (w orkspace)
dim ension \(\left(2{ }^{*} \mathrm{~N}\right)\)

RW ORK (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{N N}\) FO \(=-\) i, the i-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

The algorithm used in this program is basically backw ard (forw ard) substitution, w ith scaling to \(m\) ake the the code robustagainst possible overflow .

E ach eigenvector is nom alized so that the elem ent of largest \(m\) agnitude has \(m\) agnitude 1 ; here the \(m\) agnitude of a com plex num ber \((x, y)\) is taken to be \(|x|+|y|\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctrexc - reorder the Schur factorization of a com plex \(m\) atrix \(\mathrm{A}=\mathrm{Q} * \mathrm{~T} * \mathrm{Q} * * \mathrm{H}\), so that the diagonalelem entof T w th row index \(\mathbb{F S T}\) ism oved to row \(\mathbb{L S T}\)

\section*{SYNOPSIS}

```

CHARACTER * 1 COMPQ
COM PLEX T (LDT,*),Q (LDQ ,*)
INTEGERN,LDT,LDQ,\mathbb{FST},\mathbb{LST,}\mathbb{NFO}
SU BROUT\mathbb{NE CTREXC_64(COMPQ,N,T,LDT,Q,LDQ, FST, \#ST,INFO)}
CHARACTER * 1 COMPQ
COM PLEX T (LDT,*),Q (LDQ,*)

```


\section*{F95 INTERFACE}

SU BROUTINE TREXC (COM PQ, \(\mathbb{N}], T,[L D T], Q,[L D Q], \mathbb{F} S T, \mathbb{L} S T,[\mathbb{N} F O])\)
CHARACTER (LEN=1) ::COMPQ
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::T,Q
\(\mathbb{N}\) TEGER ::N,LDT,LDQ, \(\mathbb{F S T}, \mathbb{L S T}, \mathbb{N} F O\)
SU BROUTINE TREXC_64 (COMPQ, \(\mathbb{N}], T,[L D T], Q,[L D Q], \mathbb{F S T}, \mathbb{L} S T,[\mathbb{N F O}])\)
CHARACTER (LEN=1) ::COMPQ
COM PLEX,D \(\mathbb{I}\) ENSION (: : : : : T, Q
\(\mathbb{N}\) TEGER (8) ::N , LD T,LDQ, \(\mathbb{F S T}, \mathbb{L} S T, \mathbb{N} F O\)
\#include < sunperfh>
void ctrexc (char com pq, intn, com plex *t, int ldt, com plex * \(q\), int ldq, int ifst, int ilst, int *info);
void ctrexc_64 (charcom pq, long n, com plex *t, long ldt, com plex *q, long ldq, long ifst, long ilst, long *info);

\section*{PURPOSE}
ctrexc reorders the Schur factorization of a com plex \(m\) atrix \(A=Q * T * Q * * H\), so that the diagonalelem entof \(T\) with row index \(\mathbb{F S T}\) ism oved to row \(\mathbb{H} S T\).
The Schur form \(T\) is reordered by a unitary sim ilarity transform ation \(\mathrm{Z} * *{ }_{\mathrm{H}}{ }^{*} \mathrm{~T} * \mathrm{Z}\), and optionally the m atrix Q of Schurvectors is updated by postm ultplying itw ith Z .

\section*{ARGUMENTS}

COMPQ (input)
\(=\mathrm{V}\) ': update the m atrix Q of Schurvectors;
\(=N^{\prime}\) : do notupdate Q .
N (input) The order of the m atrix \(\mathrm{T} . \mathrm{N}>=0\).
T (input/output)
On entry, the uppertriangularm atrix T. On exit, the reordered upper triangularm atrix.

LD T (input)
The leading dim ension of the array T. LD T >= \(\max (1, \mathbb{N})\).

Q (input) \(O n\) entry, ifCOM \(P Q=V\) ', them atrix \(Q\) of Schur vectors. On exit, if \(C O M P Q=V\) ', \(Q\) has been postm ultiplied by the unitary transform ation \(m\) atrix \(Z\) which reorders \(T\). If \(C O M P Q=N\) ', \(Q\) is not referenced.

LD Q (input)
The leading dim ension of the array \(Q\). LDQ >= \(\max (1, N)\).

FST (input)
Specify the reordering of the diagonalelem ents of
\(T\) : The elem ent w ith row index \(\mathbb{F} S T\) ism oved to
row ILST by a sequence of transpositions betw een adjacent elem ents. \(1<=\mathbb{F S T}<=\mathrm{N} ; 1<=\mathbb{L} S T<=\) N .

UST (input)
See the description of IFST.
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N F O}=-\) i, the i-th argum ent had an illegal value

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctrm \(m\)-perform one of the \(m\) atrix \(m\) atrix operations \(B:=\) alpha*op ( A ) *B, orB : alpha*B *op (A) where alpha is a scalar, \(B\) is an \(m\) by \(n m\) atrix, \(A\) is a unit, or non-unit, upper or low er triangularm atrix and op (A) is one of op (


\section*{SYNOPSIS}
```

SUBROUTINE CTRMM (S\mathbb{DE,UPLO,TRANSA,D IAG,M,N,ALPHA,A,LDA,B,}
LD B )
CHARACTER * 1SDE,UPLO,TRANSA,D IAG
COM PLEX ALPHA
COM PLEX A (LDA,*),B (LDB,*)
INTEGERM,N,LDA,LDB
SU BROUTINE CTRMM _64(S\mathbb{DE,UPLO,TRAN SA,D IA G,M ,N ,A LPHA,A,LDA,B,}
LD B)

```
CHARACTER * 1 SDE , UPLO, TRANSA, D IA G
COM PLEX ALPHA
COM PLEX A (LDA,*), B (LDB,*)
\(\mathbb{N}\) TEGER*8 M , N , LDA , LD B

\section*{F95 INTERFACE}

SU BROUTINE TRMM (SDDE,UPLO, [TRANSA ],D IA G, \(\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A]\), B, [LD B])

CHARACTER (LEN=1)::SDE,UPLO,TRANSA,D IA G
COMPLEX ::ALPHA
COM PLEX,D \(\mathbb{I}\) ENSION (:,:) ::A,B
\(\mathbb{N} T E G E R:: M, N, L D A, L D B\)

SU BROUTINE TRM M _64 (S \(\mathbb{D} E, U P L O,[T R A N S A], D \mathbb{I} G, \mathbb{M}], \mathbb{N}], A L P H A, A\), [LD A ], B, [LD B])

CHARACTER (LEN=1)::SDE,UPLO,TRANSA,D IAG
COMPLEX ::ALPHA
COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) :: A, B
\(\mathbb{N}\) TEGER (8) ::M , N,LDA,LDB

\section*{C INTERFACE}
\#include <sunperfh>
void ctrm \(m\) (char side, char uplo, char transa, chardiag, int \(m\), int \(n\), com plex *alpha, com plex *a, int lda, com plex *b, int ldb);
void ctrm m _64 (char side, charuplo, char transa, char diag, long m, long n, com plex *alipha, com plex *a, long lda, com plex *b, long ldb);

\section*{PURPOSE}
ctrm \(m\) perform sone of the \(m\) atrix-m atrix operations \(B:=\) alpha*op ( A ) B , or \(\mathrm{B}:=\) alpha* B *op ( A ) where alpha is a scalar, \(B\) is an \(m\) by \(n m\) atrix, \(A\) is a unit, or non-unit, upper or low er triangularm atrix and op (A ) is one of op ( \(\mathrm{A})=\mathrm{A} \operatorname{orop}(\mathrm{A})=\mathrm{A}^{\prime} \operatorname{orop}(\mathrm{A})=\operatorname{conjg}\left(\mathrm{A}^{\prime}\right)\)

\section*{ARGUMENTS}

STDE (input)
On entry, SDE specifies whether op (A) m ultiplies \(B\) from the leftor rightas follow \(s\) :
\(S \mathbb{D} E=\) L'or I' B : alpha*op (A )*B.

U nchanged on exit.

UPLO (input)
On entry, UPLO specifies whether them atrix A is an upper or low er triangularm atrix as follow \(s\) :

UPLO = U'or 4 ' \(A\) is an upper triangular \(m\) atrix.

UPLO = L' or I' A is a lower triangular
m atrix.

U nchanged on exit.

TRANSA (input)
On entry, TRAN SA specifies the form of op (A ) to be used in the \(m\) atrix \(m\) ultiplication as follow \(s\) :

TRANSA \(=N^{\prime}\) 'or \(h^{\prime}\) op (A) \()=A\).

TRANSA \(=\) T'ort'op \((A)=A\).

TRANSA \(=C^{\prime}\) or \(C^{\prime} o p(A)=c o n \dot{g}\left(A^{\prime}\right)\).

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.
D IA G (input)
O n entry, D IA G specifies w hether ornotA is unit triangular as follow s:
\(D \mathbb{A G}=U\) 'or \(U^{\prime} A\) is assum ed to be unit triangular.

D IA G \(=N\) 'or \(h^{\prime} A\) is notassum ed to be unit triangular.

U nchanged on exit.

M (input)
O \(n\) entry, \(M\) specifies the num ber of row \(s\) of \(B . M\) \(>=0\). U nchanged on exit.

N (input)
O n entry, \(N\) specifies the num ber of colum ns of \(B\). \(\mathrm{N}>=0\). U nchanged on exit.

ALPHA (input)
On entry, A LPH A specifies the scalar alpha.W hen alpha is zero then \(A\) is notreferenced and \(B\) need notbe setbefore entry. U nchanged on exit.

A (input)
COM PLEX aray ofD \(\mathbb{I M} E N S I O N\) (LDA,k), wherek is m when \(S \mathbb{D} E=\mathbb{L}\) 'or \(\mathrm{I}^{\prime}\) and is n when \(S \mathbb{D} E=\) R 'or 'r'.

Before entry w th UPLO = U'or L', the leading \(k\) by \(k\) upper triangularpart of the array \(A\) \(m\) ustcontain the upper triangularm atrix and the
strictly low ertriangularpartofA is not referenced.

Before entry with UPLO = L'or 1', the leading \(k\) by \(k\) low er triangularpart of the array A m ust contain the low er triangularm atrix and the strictly uppertriangularpartofA is not referenced.

N ote thatw hen \(\mathrm{D} \mathbb{I A} G=\mathrm{U}\) ' or U ', the diagonal elem ents of A are not referenced either, butare assum ed to be unity.

U nchanged on exit.
LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen SDE \(=\) L'or 1 'then LD \(A>=\max (1, M)\), when \(S \mathbb{D} E\) \(=R^{\prime}\) or 'r'then LD A \(>=\max (1, N)\). U nchanged on exit.

B (input/output)
COM PLEX aray ofD \(\mathbb{I M}\) ENSION (LD B, n ). Before entry, the leading \(M\) by \(N\) part of the array \(B\) m ust contain the \(m\) atrix \(B\), and on exit is overw rilten by the transform ed \(m\) atrix.

LD B (input)
O n entry, LD B specifies the firstdim ension of B as declared in the calling subprogram. LD B m ust be at leastm ax ( \(1, M\) ). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctrm \(v\)-penform one of the \(m\) atrix-vectoroperations \(x:=\)


\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CTRMV (UPLO,TRANSA,D IAG,N,A,LDA,Y, INCY)}
CHARACTER * 1 UPLO,TRANSA,DIAG
COM PLEX A (LDA,*),Y (*)
\mathbb{NTEGERN,LDA,}\mathbb{N}CY
SU BROUT\mathbb{NECTRM V_64(UPLO,TRANSA,D IAG,N,A,LDA,Y,INCY)}
CHARACTER * 1 UPLO,TRANSA,D IAG
COM PLEX A (LDA,*),Y (*)
INTEGER*8N,LDA,}\mathbb{NCY

```
F95 INTERFACE
    SU BROUTINE TRMV (UPLO, [TRANSA],D \(\mathbb{I A G}, \mathbb{N}], A,[L D A], Y,[\mathbb{N} C Y])\)
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
    COMPLEX,D \(\mathbb{M}\) ENSION (:) ::Y
    COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N} T E G E R:: N, L D A, \mathbb{N} C Y\)
    SU BROUTINE TRM V_64 (UPLO, [TRANSA ], D \(\mathbb{A} G, \mathbb{N}], A,[L D A], Y,[\mathbb{N C Y}])\)
    CHARACTER (LEN=1)::UPLO,TRANSA,D IA G
    COMPLEX,D \(\mathbb{I M}\) ENSION (:) ::Y
    COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::A
    \(\mathbb{N}\) TEGER (8) ::N,LDA, \(\mathbb{N} C Y\)

\section*{C INTERFACE}
\#include <sunperfh>
void ctrm v (charuplo, chartransa, chardiag, intn, com plex
*a, intlda, com plex *y, intincy);
void ctrm v_64 (charuplo, chartransa, char diag, long n, com plex *a, long lda, com plex *y, long incy);

\section*{PURPOSE}
ctrm \(v\) perform s one of the \(m\) atrix-vector operations \(x:=A{ }^{*} x\), or \(\mathrm{x}:=A{ }^{*} \mathrm{x}\), or \(\mathrm{x}:=\operatorname{con} \dot{g}\left(\mathrm{~A}^{\prime}\right){ }^{\star} \mathrm{x}\) where x is an n elem ent vectorand \(A\) is an \(n\) by \(n\) unit, ornon-unit, upper or low er triangularm atrix.

\section*{ARGUMENTS}

\section*{UPLO (input)}

O n entry, UPLO specifies w hether the \(m\) atrix is an upper or low er triangularm atrix as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or \(\mathrm{U}^{\prime} A\) is an upper triangular \(m\) atrix.
\(\mathrm{UPLO}=\mathrm{L}\) ' or I' A is a lower triangular \(m\) atrix.

U nchanged on exit.

TRANSA (input)
O n entry, TRANSA specifies the operation to be perform ed as follow \(s\) :

TRANSA \(=N^{\prime}\) or \(h^{\prime} x:=A * x\).

TRANSA = T'ort'x:=A*x.


U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
O n entry, D IA G specifies w hether or notA is unit triangular as follow s:

D \(\mathbb{A} G=U\) 'or \(u\) ' \(A\) is assum ed to be unit triangular.

D \(\mathbb{A} G=N\) 'or \(h\) ' A is notassum ed to be unit triangular.

U nchanged on exit.
N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix \(A\). \(\mathrm{N}>=0\). U nchanged on exit.

A (input)
Before entry w ith UPLO = U 'or L', the leading \(n\) by \(n\) upper triangular part of the array A \(m\) ust contain the upper triangular \(m\) atrix and the strictly low er triangular part of A is not referenced. Before entry with UPLO = L 'or I', the leading \(n\) by \(n\) low er triangularpart of the array A m ust contain the low er triangular matrix and the strictly uppertriangularpartofA is not referenced. N ote thatw hen D IA G = U' or L', the diagonal elem ents of A are not referenced either, but are assum ed to be unity. U nchanged on exit.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A \(>=\) \(m a x(1, n)\). U nchanged on exit.

Y (input/output)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) must contain the \(n\) elem ent vectorx. On exit, \(Y\) is overw rilten \(w\) ith the tranform ed vector \(x\).
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of. \(\mathbb{N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctrnfs - provide emorbounds and backw ard enror estim ates for the solution to a system of linear equationsw ith a triangular coefficientm atrix

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE CTRRFS (UPLO,TRANSA,D IAG,N,NRHS,A,LDA,B,LDB,X,}
LD X,FERR,BERR,W ORK,W ORK 2,INFO)
CHARACTER * 1 UPLO,TRANSA,DIAG
COM PLEX A (LDA,*),B (LD B,*),X (LDX,*),W ORK (*)
\mathbb{NTEGERN,NRHS,LDA,LDB,LDX,}\mathbb{N}FO
REAL FERR (*),BERR (*),W ORK2 (*)
SU BROUT\mathbb{NE CTRRFS_64 (UPLO,TRANSA,D IAG,N,NRHS,A,LDA,B,LDB,X,}
LDX,FERR,BERR,W ORK,W ORK2,INFO)

```
CHARACTER * 1 UPLO, TRANSA, D IA G

\(\mathbb{N}\) TEGER*8N,NRHS,LDA,LDB,LDX, \(\mathbb{N} F O\)
REAL FERR (*), BERR ( \({ }^{*}\) ), \(\mathrm{W} O \operatorname{OR} 2\) ( \({ }^{*}\) )

\section*{F95 INTERFACE}

SU BROUTINE TRRFS (UPLO, [TRANSA],D \(\mathbb{I} G, \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B]\), X, [LD X],FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::UPLO ,TRANSA,D IA G
COMPLEX,D IM ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,B,X
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, L D X, \mathbb{N} F O\)
REAL,D IM ENSION (:) ::FERR,BERR,W ORK 2

SU BROUTINE TRRFS_64 (UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{N}], \mathbb{N} R H S], A,[L D A], B\), [LDB],X, [LDX],FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::UPLO, TRANSA,D IA G
COM PLEX,D IM ENSION (:) ::W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A, B, X
\(\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D B, L D X, \mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK2

\section*{C INTERFACE}
\#include < sunperfh>
void ctrmf (char uplo, char transa, chardiag, int n, int nihs, com plex *a, int lda, com plex *b, int ldb, com plex *x, int ldx, float *ferr, float *berr, int *info);
void ctmfs_64 (charuplo, chartransa, char diag, long n, long nrhs, com plex *a, long lda, com plex *b, long ldb, com plex *x, long ldx, float *ferr, float *berr, long *info);

\section*{PURPOSE}
ctrifs provides errorbounds and backw ard error estim ates for the solution to a system of linear equations \(w\) th a triangular coefficientm atrix.

The solution \(m\) atrix \(X\) m ustbe com puted by CTRTRS or some other \(m\) eans before entering this routine. CTRRFS does not do iterative refinem entbecause doing so cannotim prove the backw ard error.

\section*{ARGUMENTS}

\section*{UPLO (input)}
\(=\mathrm{U}\) ': A is uppertriangular;
= L' ': A is low er triangular.
TRANSA (input)
Specifies the form of the system of equations:
\(=\mathrm{N}: \mathrm{A} * \mathrm{X}=\mathrm{B} \quad\) N 0 transpose)
\(=T: A * * T * X=B \quad(T\) ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose)
TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
\(=\mathrm{N}: A\) is non-unit triangular;
\(=U\) ': A is unit triangular.

N (input) The order of the matrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of right hand sides, ie., the num ber of colum ns of the m atrices B and X . NRH S >=0.

A (input) The triangularm atrix A. If \(\mathrm{U} P L O=\mathrm{U}\) ', the leading N -by- N upper triangularpart of the aray A contains the upper triangular \(m\) atrix, and the strictly low er triangular part of \(A\) is not referenced. IfUPLO = L ', the leading N -by- N low er triangular part of the amray A contains the low ertriangularm atrix, and the strictly upper triangularpart ofA is not referenced. IfD IA G = U', the diagonal elem ents of A are also not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).
\(B\) (input) The righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).
\(X\) (input) The solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the array X. LD X >= \(\max (1, N)\).

FERR (output)
The estim ated forw ard emorbound for each solution vectorX (i) the \(j\) th colum \(n\) of the solution \(m\) atrix \(X)\). If \(X T R U E\) is the true solution comesponding to \(X(\mathcal{i}), \operatorname{FERR}(\mathcal{)}\) is an estim ated upperbound forthe \(m\) agnitude of the largest ele\(m\) entin (X ( \()\)-X TRUE) divided by the magninude of the largestelem entin X ( 7 ) . The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slightoverestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each
solution vectorX (i) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{(})\) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0: successfulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ctrsen - reorder the Schur factorization of a com plex m atrix
\(A=Q * T * Q * * H\), so that a selected clusterofeigenvalues appears in the leading positions on the diagonal of the upper triangularm atrix \(T\), and the leading colum ns of form an orthonorm albasis of the corresponding right invariant subspace

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE CTRSEN (JOB,COMPQ,SELECT,N,T,LDT,Q,LDQ,W,M,S,}
SEP,W ORK,LW ORK,INFO)
CHARACTER * 1 JOB,COMPQ
COM PLEX T (LDT,*),Q (LDQ ,*),W (*),W ORK (*)
\mathbb{NTEGER N,LDT,LDQ,M,LW ORK,INFO}
LOG ICAL SELECT (*)
REALS,SEP
SU BROUTINE CTRSEN_64(OB B,COM PQ,SELECT,N,T,LD T,Q,LDQ,W ,M ,S,
SEP,W ORK,LW ORK,INFO)
CHARACTER * 1 JOB,COMPQ
COM PLEX T (LDT,*),Q (LDQ,*),W (*),W ORK (*)
INTEGER*8N,LDT,LDQ,M,LW ORK, INFO
LOG ICAL*8 SELECT (*)
REALS,SEP

```

F95 INTERFACE
SU BROUTINE TRSEN (JOB,COMPQ,SELECT, \(\mathbb{N}], T,[L D T], Q,[L D Q], W, M\), S, SEP, [W ORK ], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) :: JOB,COMPQ
COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::W ,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : \(\mathrm{T}, \mathrm{Q}\)
\(\mathbb{N} T E G E R:: N, L D T, L D Q, M, L W O R K, \mathbb{N} F O\)
LO G ICAL,D IM ENSION (:) ::SELECT
REAL ::S,SEP

SU BROUTINE TRSEN_64 (JO B,COM PQ, SELECT, \(\mathbb{N}], T,[L D T], Q,[L D Q], W\), M ,S,SEP, [W ORK ], [LW ORK ], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1)::JOB,COMPQ
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::T,Q
\(\mathbb{N} T E G E R(8):: N, L D T, L D Q, M, L W O R K, \mathbb{N} F O\)
LO G ICAL (8), D IM ENSIO N (:) ::SELECT
REAL ::S,SEP

\section*{C INTERFACE}
\#include <sunperfh>
void ctrsen (char job, char com pq, int *select, int n, com plex *t, int ldt, com plex *q, int ldq, com plex *w, int *m, float *s, float*sep, int*info);
void ctrsen_64 (char job, char com pq, long *select, long n, com plex *t, long ldt, com plex *q, long ldq, com plex *w, long *m, float *s, float *sep, long *info);

\section*{PURPOSE}
ctrsen reorders the Schur factorization of a com plex m atrix \(\mathrm{A}=\mathrm{Q} * \mathrm{~T} * \mathrm{Q} * * \mathrm{H}\), so that a selected clusterofeigenvalues appears in the leading positions on the diagonal of the upper triangularm atrix \(T\), and the leading colum ns ofQ form an orthonom albasis of the comesponding right invariant subspace.

Optionally the routine com putes the reciprocal condition num bers of the cluster ofeigenvalues and/or the invariant subspace.

\section*{ARGUMENTS}

JO B (input)
Specifies w hether condition num bers are required for the cluster ofeigenvalues (S) or the invariantsubspace (SEP):
= N ': none;
= E': foreigenvalues only (S);
= V ': for invariant subspace only (SEP);
= B ': forboth eigenvalues and invariant subspace
( S and SEP).
\(C O M P Q\) (input)
\(=\mathrm{V}\) ': update the m atrix Q of Schurvectors;
\(=N\) ': do notupdate \(Q\).

\section*{SELECT (input)}

SELEC T specifies the eigenvalues in the selected cluster. To select the \(j\) th eigenvahue, SELEC T ( 7 ) m ust.be set to .TRUE..

N (input) The order of the m atrix \(\mathrm{T} . \mathrm{N}>=0\).
T (input/output)
On entry, the upper triangularm atrix T. On exit, T is overw ritten by the reordered m atrix T , w th the selected eigenvalues as the leading diagonal elem ents.

LD T (input)
The leading dim ension of the aray T. LD T >= \(\max (1, N)\).

Q (input) \(O n\) entry, if \(C O M P Q=V\) ', them atrix \(Q\) of Schur vectors. On exit, if \(C O M P Q=V\) ', Q hasbeen postm ultiplied by the unitary transform ation \(m\) atrix which reorders \(T\); the leading \(M\) colum ns of \(Q\) form an orthonorm al basis for the specified invariant subspace. If \(C O M P Q=N^{\prime}, \mathrm{Q}\) is not referenced.

LD Q (input)
The leading \(d i m\) ension of the anray \(Q . L D Q>=1\); and if \(C O M P Q=V\) ', LD \(Q>=N\).

W (output)
The reordered eigenvalues of \(T\), in the sam e order as they appear on the diagonal of \(T\).

M (output)
The dim ension of the specified invariant subspace. \(0<=\mathrm{M}<=\mathrm{N}\).

S (output)
If \(J 0 B=E\) 'or \(B ', S\) is a lower bound on the reciprocal condition num ber for the selected clusterofeigenvalues. S cannot underestim ate the
true reciprocal condition num berby \(m\) ore than a factorof sqit \(\mathbb{N}\) ). If \(M=0\) or \(N, S=1\). If \(J 0 B=\) N 'or \(V\) ', S is not referenced.

SEP (output)
If \(\mathrm{OB}=\mathrm{V}\) 'or B ', SEP is the estim ated reciprocal condition num ber of the specified invariant subspace. IfM \(=0\) or \(N\), \(\mathrm{SEP}=\) norm ( I ). If \(\mathrm{JOB}=\) N 'or E',SEP is not referenced.
W ORK (w orkspace)
If \(\mathrm{JO} \mathrm{B}=\mathrm{N}\) ', W ORK is not referenced. O therw ise, on exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the anay \(W\) ORK. If \(J 0 B=N\) ', LW ORK >=1; if \(J O B=E\) ', LW ORK \(=M *(N+M)\); if \(J O B\) \(=V\) 'or \(B^{\prime}\), LW ORK \(>=2 * M *(N+M)\).

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N}\) FO (output)
= 0 : successfulexit
< \(0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

C TRSEN first collects the selected eigenvalues by com puting a unitary transform ation \(Z\) to \(m\) ove them to the top left comerofT. In otherw ords, the selected eigenvalues are the eigenvalues of T11 in:
\[
\begin{gathered}
\mathrm{Z} \text { *T*Z }=(\mathrm{T} 11 \mathrm{~T} 12) \mathrm{n} 1 \\
(0 \mathrm{~T} 22) \mathrm{n} 2 \\
\mathrm{n} 1 \mathrm{n} 2
\end{gathered}
\]
\(w\) here \(N=n 1+n 2\) and \(Z\) ' \(m\) eans the conjugate transpose of \(Z\). The firstn1 colum ns of \(Z\) span the specified invariant subspace of \(T\).

IfT has been obtained from the Schur factorization of a \(m\) atrix \(A=Q * T * Q '\), then the reordered Schur factorization of \(A\) is given by \(\left.A=(Q *) *\left(Z \quad{ }^{*} T Z\right) * Q * Z\right)\) ', and the first \(n 1\) colum ns of Q Z span the comesponding invariant subspace of A.

The reciprocal condition num ber of the average of the eigenvalues of T11 m ay be retumed in S.S lies betw een 0 (very badly conditioned) and 1 (very w ellconditioned). It is com puted as follow s. Firstw e com pute R so that
\[
\begin{gathered}
P=\left(\begin{array}{l}
\text { I R }) ~ n 1 ~ \\
(00) n 2 \\
n 1 n 2
\end{array}\right.
\end{gathered}
\]
is the pro jector on the invariant subspace associated with T11. R is the solution of the Sylvesterequation:
\[
\mathrm{T} 11 * \mathrm{R}-\mathrm{R} * \mathrm{~T} 22=\mathrm{T} 12 .
\]

LetF-norm M) denote the Frobenius-norm of M and 2 -norm (M) denote the tw o-norm of . Then \(S\) is com puted as the low er bound
\[
(1+F-\operatorname{nom}(\mathbb{R}) * * 2)^{\star *}(-1 / 2)
\]
on the reciprocal of 2 -norm ( P ), the true reciprocal condition num ber. \(S\) cannotunderestim ate \(1 / 2\)-norm (P) by \(m\) ore than a factorof sqred ).

A n approxim ate errorbound for the com puted average of the eigenvalues of T11 is
EPS * norm (T) /S
where EPS is the \(m\) achine precision.
The reciprocal condition num ber of the right invariant subspace spanned by the firstn1 colum nsof \(Z\) (orofQ*Z) is retumed in SEP. SEP is defined as the separation of T11 and T 22 :
\[
\operatorname{sep}(T 11, T 22)=\text { sigm a-m in (C ) }
\]
where sigm a-m in (C) is the sm allest singularvalue of the \(n 1 * n 2-b y-n 1 * n 2 m\) atrix
\[
C=\operatorname{kprod}(I(n 2), T 11)-\operatorname{kprod}(\text { transpose }(T 22), I(n 1))
\]

I( \(m\) ) is an \(m\) by \(m\) identity \(m\) atrix, and kprod denotes the \(K\) ronecker product. W e estim ate sigm a-m in (C) by the reciprocalofan estim ate of the 1 -norm of inverse (C). The true reciprocal 1-norm of inverse ( \(C\) ) cannot differ from sigm a\(m\) in (C ) by \(m\) ore than a factor of squt (n1*n2).

W hen SEP is sm all, sm all changes in T can cause large
changes in the invariant subspace. A \(n\) approxim ate bound on the \(m\) axim um angularerror in the com puted right invariant subspace is

EPS * norm (T) /SEP

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctrsm - solve one of the \(m\) atrix equations op (A )*X = alpha*B \(\operatorname{orX}\) *op ( A ) \(=\) alpha*B

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NECTRSM(SDE,UPLO,TRANSA,D IAG,M,N,ALPHA,A,LDA,B,}
LD B )
CHARACTER * 1SDE,UPLO,TRANSA,D IAG
COM PLEX ALPHA
COM PLEX A (LDA,*),B (LD B,*)
INTEGERM,N,LDA,LDB

```

```

    LD B)
    ```
CHARACTER * 1 SDE,UPLO,TRANSA,D IA G
COM PLEX ALPHA
COM PLEX A (LDA, \(\left.)^{\prime}\right)\), (LDB,*)
\(\mathbb{N}\) TEGER*8M,N,LDA,LDB

\section*{F95 INTERFACE}

SU BROUTINE TRSM (SDE, UPLO, [TRANSA ],D \(\mathbb{I A G}, \mathbb{M}], \mathbb{N}], A L P H A, A,[L D A]\), B, [LD B])

CHARACTER (LEN=1)::SDE,UPLO,TRANSA,D IA G
COMPLEX ::ALPHA
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,B
\(\mathbb{N} T E G E R:: M, N, L D A, L D B\)
SU BROUTINE TRSM_64 (SDE,UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{M}], \mathbb{N}], A L P H A, A\), [LDA], B, [LDB])

CHARACTER (LEN=1)::SDE,UPLO,TRANSA,D IA G
COMPLEX ::ALPHA
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,B
\(\mathbb{N}\) TEGER (8) ::M , N ,LDA,LDB

\section*{C INTERFACE}
\#include <sunperfh>
void ctrsm (char side, charuplo, char transa, chardiag, int \(m\), int \(n\), com plex *alpha, com plex *a, int lda, com plex *b, int ldb);
void ctrsm _64 (char side, charuplo, chartransa, char diag, long m, long n, com plex *alpha, com plex *a, long lda, com plex *b, long ldb);

\section*{PURPOSE}
ctrsm solves one of the \(m\) atrix equations op (A )*X = alpha*B, or X *op (A ) = alpha*B where alpha is a scalar, X and \(B\) arem by \(n m\) atrioes, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix and op (A) is one of
\(\mathrm{op}(\mathrm{A})=\mathrm{A}\) or \(\mathrm{op}(\mathrm{A})=\mathrm{A}^{\prime}\) or \(\mathrm{op}(\mathrm{A})=\operatorname{con} \overline{\mathrm{g}}(\) A').

Them atrix \(X\) is overw rilten on \(B\).

\section*{ARGUMENTS}

STDE (input)
On entry, SDD E specifies w hetherop (A ) appears on the leftor rightofX as follow s:

SDE = L'or I' op (A )*X = alpha*B .
\(S \mathbb{D} E=R\) 'or 'r' \(X\) *op (A ) = alpha*B.

U nchanged on exit.

UPLO (input)
On entry, UPLO specifies whether the matrix A is an upper or low er triangularm atrix as follow \(s\) :

UPLO = U'or \(\mathrm{L}^{\prime} A\) is an upper triangular \(m\) atrix.

UPLO = L' or I' A is a lower triangular \(m\) atrix.

U nchanged on exit.
TRANSA (input)
On entry, TRAN SA specifies the form ofop (A ) to be used in the \(m\) atrix \(m\) ultiplication as follow \(s\) :

TRANSA \(=N^{\prime}\) 'or \(h^{\prime}\) op ( \(\left.A\right)=A\).

TRANSA = T'ort'op(A) = A'.

TRANSA \(=\) C'or \(\mathrm{E}^{\prime} \mathrm{op}(\mathrm{A})=\operatorname{conjg}\left(\mathrm{A}^{\prime}\right)\).
U nchanged on exit.
TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE .

D IA G (input)
On entry, D IA G specifies w hether ornotA is unit triangular as follow s:

D \(\mathbb{A} G=U\) 'or \(l^{\prime} A\) is assum ed to be unit triangular.
\(D \mathbb{A} G=N\) 'or \(h\) ' \(A\) is notassumed to be unit triangular.

U nchanged on exit.
M (input)
O \(n\) entry, M specifies the num ber of row sof B.M \(>=0\). Unchanged on exit.

N (input)
O \(n\) entry, \(N\) specifies the num ber of colum ns of \(B\).
\(\mathrm{N}>=0\). U nchanged on exit.
ALPHA (input)
O n entry, A LPH A specifies the scalar alpha.W hen alpha is zero then \(A\) is not referenced and \(B\) need notbe setbefore entry. U nchanged on exit.

A (input)
COM PLEX aray ofD \(\mathbb{M}\) ENSION (LDA, k),
where \(k\) is \(m\) when \(S \mathbb{D} E=\mathbb{L}\) 'or \(\mathbb{I}\) and is \(n\)
when \(S \mathbb{D E}=\) R'or 'r'.
Before entry \(w\) th UPLO \(=U\) 'or \(u\) ', the lead-
ing \(k\) by \(k\) upper triangularpart of the array \(A\) \(m\) ust contain the upper triangularm atrix and the strictly low ertriangularpartofA is not referenced.

Before entry w th UPLO = L'or 1', the leading \(k\) by \(k\) low er triangularpart of the array \(A\) m ustcontain the low er triangularm atrix and the strictly uppertriangularpart ofA is not referenced.

N ote thatw hen D \(\mathbb{A}\) G = U ' or U ', the diagonal elem ents of A are not referenced either, butare assum ed to be unity. U nchanged on exit.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen \(S \mathbb{D} E=\mathbb{L}\) 'or \(I^{\prime}\) then LD \(A>=m a x(1, M)\), when \(S \mathbb{D} E\) \(=R^{\prime}\) or ' \(r^{\prime}\) then LD \(A>=m\) ax \((1, N)\). U nchanged on exit.

B (input/output)
COM PLEX array ofD \(\mathbb{I M} E N S \mathbb{I} N(L D B, n)\).
Before entry, the leading \(M\) by \(N\) part of the array
\(B m\) ust contain the righthand side \(m\) atrix \(B\), and on exit is overw rilten by the solution \(m\) atrix \(X\).

LD B (input)
O n entry, LD B specifies the firstdim ension of B as declared in the calling subprogram. LD B >= max ( \(1, M\) ). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ctrsna - estim ate reciprocalcondition num bers for specified eigenvalues and/or right eigenvectors of a com plex upper triangularm atrix T (orofany m atrix \(\mathrm{Q} * \mathrm{~T} * \mathrm{Q} * * \mathrm{H}\) w ith Q unitary)

\section*{SYNOPSIS}
```

SUBROUTINE CTRSNA (JOB,HOW MNY,SELECT,N,T,LDT,VL,LDVL,VR,LDVR,
S,SEP,MM ,M ,W ORK,LDW ORK,W ORK1, NNFO)
CHARACTER * 1 JOB,HOW MNY
COM PLEX T (LDT,*),VL (LDVL,*),VR (LDVR,*),W ORK (LDW ORK,*)
\mathbb{NTEGERN,LDT,LDVL,LDVR,MM,M,LDW ORK,INFO}
LOG ICAL SELECT (*)
REAL S (*),SEP (*),W ORK 1 (*)
SUBROUT\mathbb{NECTRSNA_64(JOB,HOW MNY,SELECT,N,T,LDT,VL,LDVL,VR,}
LDVR,S,SEP,MM ,M,W ORK,LDW ORK,W ORK 1, INFO)
CHARACTER * 1 JOB,HOW MNY
COM PLEX T (LDT,*),VL (LDVL,*),VR (LDVR,*),W ORK (LDW ORK,*)
NNTEGER*8N,LDT,LDVL,LDVR,MM ,M ,LDW ORK,INFO
LOG ICAL*\& SELECT (*)
REALS (*),SEP (*),W ORK1 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE TRSNA (JOB,HOW M NY, SELECT, \(\mathbb{N}], T,[L D T], V L,[L D V L], V R\), [LDVR],S,SEP,MM,M,[WORK], [LDW ORK], [W ORK1], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1):: JOB,HOW MNY
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::T,VL,VR,W ORK
\(\mathbb{N} T E G E R:: N, L D T, L D V L, L D V R, M M, M, L D W O R K, \mathbb{N} F O\)
LOGICAL, D \(\mathbb{I M} E N S I O N(:):: S E L E C T\)
REAL,D \(\mathbb{I M} E N S I O N\) (:) :: S, SEP,W ORK1

SU BROUTINE TRSNA_64 (OB B, HOW M NY, SELECT, \(\mathbb{N}], T,[L D T], V L,[L D V L]\), VR, [LDVR], S, SEP, M M , M , [W ORK ], [LDW ORK ], [W ORK1], [ \(\mathbb{N} F O\) ])

CHARACTER ( \(L E N=1\) ) : : JOB , HOW M NY
COM PLEX , D \(\mathbb{M}\) ENSION (:,:) ::T,VL,VR,W ORK
\(\mathbb{N}\) TEGER (8) :: N , LD T, LDVL, LDVR, M M , M , LDW ORK , \(\mathbb{N} F O\)
LOGICAL (8), D \(\mathbb{M}\) ENSION (:) :: SELECT
REAL,D \(\mathbb{I M} E N S I O N\) (:) :: S, SEP,W ORK1

\section*{C INTERFACE}
\#include <sunperfh>
void ctrsna (char job, char how m ny, int *select, intn, com plex *t, intldt, com plex *vl, int ldvl, com plex
*Vr, int ldvr, float*s, float*sep, int mm , int
*m, int ldw ork, int*info);
void ctrsna_64 (char j̄b, charhow m ny, long *select, long n, com plex *t, long ldt, com plex *vl, long ldvl, com plex *vr, long ldvr, float*s, float *sep, long m m , long *m, long ldw ork, long *info);

PURPOSE
ctrsna estim ates reciprocalcondition num bers for specified eigenvahues and/or right eigenvectors of a com plex upper triangularm atrix T (orofany \(m\) atrix \(Q * T * Q * * H\) with \(Q\) unitary).

\section*{ARGUMENTS}

JOB (input)
Specifies w hethercondition num bers are required
foreigenvalues (S) oreigenvectors (SEP):
\(=\mathrm{E}\) ': foreigenvalues only ( S );
\(=\mathrm{V}\) : foreigenvectors only (SEP);
\(=B^{\prime}:\) forboth eigenvalues and eigenvectors ( \(S\) and SEP).

H OW M NY (input)
= A ': com pute condition num bers for all eigenpairs;
\(=S\) : com pute condition num bers for selected
eigenpairs specified by the anray SELECT .

SELECT (input)
IfHOW M NY = S',SELECT specifies the eigenpairs for which condition num bers are required. To selectcondition num bers for the \(j\) th eigenpair,
SELECT ( \(\ddagger\) ) mustbe set to TRUE .. IfHOW M NY = A ',
SELECT is notreferenced.

N (input) The order of the m atrix \(\mathrm{T} \cdot \mathrm{N}>=0\).

T (input) The upper triangularm atrix T .

LD T (input)
The leading dim ension of the array \(\mathrm{T} . \mathrm{LD} \mathrm{T}>=\) \(\max (1, \mathbb{N})\).

VL (input)
If \(\mathrm{JOB}=\mathrm{E}\) 'or B ', VL mustcontain left eigenvectors of \(T\) (or of any \(\mathrm{Q} * \mathrm{~T} * \mathrm{Q} * * \mathrm{H}\) w ith Q unitary), corresponding to the eigenpairs specified by H OW M NY and SELECT.The eigenvectors m ustbe stored in consecutive colum ns of \(V L\), as retumed by \(C H S E \mathbb{I}\) orCTREVC. If \(J B=V\) ', \(V L\) is notreferenced.

LDVL (input)
The leading dim ension of the array V L. LD V L >=1; and if \(J 0 B=E\) 'or \(B ', L D V L>=N\).

VR (input)
If \(\mathrm{JOB}=\mathrm{E}\) 'or B ', VR m ust contain right eigenvectors of \(T\) (or of any \(Q * T * Q * * H\) with Q unitary), corresponding to the eigenpairs specified by H OW M NY and SELEC T. The eigenvectors m ustbe stored in consecutive columns of VR, as retumed by \(C H S E \mathbb{N}\) orCTREVC. If \(O B=V\) ',VR is notreferenced.

LDVR (input)
The leading din ension of the array VR. LD V R >=1; and if \(J 0 B=E\) 'or \(B ', L D V R>=N\).

S (output)
If \(\mathrm{OOB}=\mathrm{E}\) ' or \(\mathrm{B}^{\prime}\), the reciprocal condition num bers of the selected eigenvalues, stored in consecutive elem ents of the array. Thus \(S(\mathcal{)}\), \(\operatorname{SEP}(\mathcal{j})\), and the \(j\) th colum ns of VL and VR all comespond to the sam e eigenpair (butnotin general the \(j\) th eigenpair, unless alleigenpairs are selected). If \(\mathrm{JOB}=\mathrm{V}^{\prime}\) ', S is not referenced.

SEP (output)
If \(\mathrm{JOB}=\mathrm{V}\) ' or B ', the estim ated reciprocal condition numbers of the selected eigenvectors, stored in consecutive elem ents of the aray. If \(\mathrm{JOB}=\mathrm{E}\) ', SEP is not referenced.

M M (input)
The num berofelem ents in the arrays \(S\) (if \(J O B=\) E' or B') and/orSEP (if \(J O B=V\) 'or B ).MM \(>=M\).

M (output)
The num berof elem ents of the arrays \(S\) and/or SEP actually used to store the estim ated condition num bers. If HOW M NY = A', M is set to \(N\).

W ORK (w orkspace)
dim ension (LDW ORK, N+1) If \(\mathrm{OOB}=\mathrm{E}\) ', W ORK is not referenced.

LDW ORK (input)
The leading dim ension of the array W ORK. LDW ORK \(>=1\); and if \(J O B=V\) 'or \(B\) ', LDW ORK \(>=N\).

W ORK 1 (w orkspace)
dim ension \((\mathbb{N})\) If \(\mathrm{JOB}=E\) ', W ORK 1 is not referenced.
\(\mathbb{N F O}\) (output)
\(=0\) : successfulexit
< 0 : if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{FURTHER DETAILS}

The reciprocal of the condition num ber of an eigenvalue lam boda is defined as
\[
S(\operatorname{lam} \text { bda })=\left|N^{*} u\right| /(\text { norm }(u) * \text { norm }(v))
\]
\(w\) here \(u\) and \(v\) are the right and left eigenvectors of \(T\) comesponding to lam bda; v 'denotes the conjugate transpose of \(v\), and norm (u) denotes the Euclidean norm. These reciprocal condition num bers alw ays lie betw een zero (very badly conditioned) and one (very well conditioned). If \(\mathrm{n}=1\), S (lam bda) is defined to be 1 .

A \(n\) approxim ate errorbound for a com puted eigenvalue \(W\) (i) is given by
```

EPS * norm (T) /S (i)

```
where EPS is the \(m\) achine precision.
The reciprocal of the condition num ber of the right eigenvector \(u\) corresponding to lam bda is defined as follow s. Suppose
\[
\begin{gathered}
\mathrm{T}=(\operatorname{lam} \text { bda } \mathrm{c}) \\
\left(\begin{array}{cc}
\mathrm{T} 22
\end{array}\right)
\end{gathered}
\]

Then the reciprocalcondtion num ber is
SEP (lam bda, T22 \()=\) sigm a-m in (T22 -lam bda*I \()\)
where sigm a-m in denotes the sm allest singular value. \(W\) e approxim ate the sm allest singularvalue by the reciprocal of an estim ate of the one-norm of the inverse of T22 lam bda*I. If \(n=1, \operatorname{SEP}(1)\) is defined to be \(\operatorname{abs}(T(1,1))\).

A \(n\) approxim ate errorbound for a com puted right eigenvector VR (i) is given by
EPS * norm (I) /SEP (i)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctrsv - solve one of the system sofequations \(A * x=b\), or \(A^{*} \mathrm{x}=\mathrm{b}\), orcong \(\left(\mathrm{A}^{\prime}\right)^{*} \mathrm{x}=\mathrm{b}\)

\section*{SYNOPSIS}

SUBROUTINECTRSV (UPLO,TRANSA,D IAG,N,A,LDA,Y, \(\mathbb{N} C Y\) )
CHARACTER * 1 UPLO, TRANSA, D IA G
COM PLEXA (LDA,*), Y (*)
\(\mathbb{N}\) TEGERN,LDA, \(\mathbb{N} C Y\)
SU BROUTINECTRSV_64 (UPLO, TRANSA, D \(\mathbb{I A} G, N, A, L D A, Y, \mathbb{N} C Y)\)
CHARACTER * 1 UPLO, TRANSA, D IAG
COM PLEX A (LDA,*), Y (*)
\(\mathbb{N} T E G E R * 8 N, L D A, \mathbb{N C Y}\)

\section*{F95 INTERFACE}

SUBROUTINE TRSV (UPLO, [TRANSA],D \(\mathbb{I A G}, \mathbb{N}], A,[L D A], Y,[\mathbb{N C Y}])\)
CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
COMPLEX,D \(\mathbb{I M} E N S I O N(:):: Y\)
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : ::A
\(\mathbb{N} T E G E R:: N, L D A, \mathbb{N} C Y\)
SU BROUTINE TRSV_64 (UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{N}], A,[L D A], Y,[\mathbb{N} C Y])\)
CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
COMPLEX,D \(\mathbb{I M}\) ENSION (:) ::Y
COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R(8):: N, L D A, \mathbb{N} C Y\)

\section*{C INTERFACE}
\#include <sunperfh>
void ctrsv (charuplo, chartransa, chardiag, intn, com plex
*a, intlda, com plex *y, intincy);
void ctrsv_64 (charuplo, chartransa, char diag, long n, com plex *a, long lda, com plex *y, long incy);

\section*{PURPOSE}
ctrsv solves one of the system s ofequations \(A *_{x}=b\), or \(A{ }^{*} x=b\), orcon \(j g\left(A^{\prime}\right){ }^{\star} x=b\) where \(b\) and \(x\) are \(n\) elem ent vectors and \(A\) is an \(n\) by \(n\) unit, ornon-unit, upper or low er triangularm atrix.
N o testforsingularity ornear-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.

\section*{ARGUMENTS}

UPLO (input)
O n entry, UPLO specifies w hether the \(m\) atrix is an upper or low er triangularm atrix as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or \(\mathrm{U}^{\prime} A\) is an upper triangular m atrix .
\(\mathrm{UPLO}=\mathrm{L}\) ' or \(\mathrm{I}^{\prime}\) A is a lower triangular \(m\) atrix .

U nchanged on exit.

TRANSA (input)
On entry, TRAN SA specifies the equations to be solved as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} A{ }^{*} \mathrm{x}=\mathrm{b}\).

TRANSA = T'ort' \(A{ }^{*} \mathrm{x}=\mathrm{b}\).

TRANSA \(=C^{\prime}\) or \(\mathrm{C}^{\prime} \operatorname{con} \dot{\rho}\left(\mathrm{A}^{\prime}\right) \star \mathrm{x}=\mathrm{b}\).

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
O n entry, D IA G specifies w hether ornotA is unit
triangular as follow s:
\(D \mathbb{A G}=U\) 'or \(U^{\prime} A\) is assum ed to be unit triangular.
\(D \mathbb{A G}=N^{\prime}\) 'or \(h^{\prime} A\) is notassum ed to be unit triangular.

U nchanged on exit.

N (input)
O n entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.
A (input)
Before entry w ith UPLO = U 'or L ', the leading \(n\) by \(n\) upper triangularpart of the anray A m ust contain the upper triangular \(m\) atrix and the strictly low ertriangularpart of \(A\) is notreferenced. Before entry w ith UPLO = 'L 'or '1', the leading \(n\) by \(n\) low er triangularpart of the array A m ustcontain the low ertriangularm atrix and the strictly uppertriangularpart of \(A\) is notreferenced. N ote thatw hen DIAG \(=U^{\prime}\) or \(L^{\prime}\) ', the diagonal elem ents ofA are notreferenced either, butare assum ed to be unity. U nchanged on exit.

LD A (input)
O n entry, LD A specifies the first dim ension of A as declared in the calling (sub) program .LD A >= \(\max (1, n)\). U nchanged on exit.

Y (input/output)
\((1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) m ust contain the \(n\) elem ent right-hand side vectorb. On exit, \(Y\) is overw ritten \(w\) th the solution vectorx.
\(\mathbb{N} C Y\) (input)
O n entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(. \mathbb{N} C Y<>0\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctrsyl-solve the com plex Sylvesterm atrix equation

\section*{SYNOPSIS}
```

SUBROUTINE CTRSYL(TRANA,TRANB,ISGN,M,N,A,LDA,B,LDB,C,LDC,
SCALE,\mathbb{NFO)}

```
CHARACTER * 1 TRANA, TRANB
COM PLEX A (LDA, *), B (LD B,*), C (LDC, *)
\(\mathbb{N}\) TEGER ISGN,M,N,LDA,LDB,LDC, \(\mathbb{N}\) FO
REAL SCALE
SU BROUTINE CTRSYL_64 (TRANA, TRANB, ISGN,M,N,A,LDA,B,LDB,C,
    LD C, SCALE, \(\mathbb{N} F O\) )
CHARACTER * 1 TRANA, TRANB
COM PLEX A (LDA,*), B (LDB,*), C (LD C, *)
\(\mathbb{N} T E G E R * 8 \mathbb{I S G N}, \mathrm{M}, \mathrm{N}, \mathrm{LDA}, \mathrm{LD} B, L D C, \mathbb{N} F O\)
REAL SCALE

\section*{F95 INTERFACE}

SU BROUTINE TRSYL (IRANA, TRANB, \(\operatorname{ISGN}, \mathbb{M}], \mathbb{N}], A,[L D A], B,[L D B], C\), [LDC],SCALE, [ \(\mathbb{N F F O}\) ])

CHARACTER (LEN=1) ::TRANA, TRANB
COM PLEX,D \(\mathbb{I M}\) ENSION (: : : : ::A, B , C
\(\mathbb{N} T E G E R:: \mathbb{I S G N}, \mathrm{M}, \mathrm{N}, L D A, L D B, L D C, \mathbb{N} F O\)
REAL ::SCALE

SU BROUTINE TRSY L_64 (TRANA,TRANB, ISGN, \(\mathbb{M}], \mathbb{N}], A,[L D A], B,[L D B]\), C, [LDC],SCALE, [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::TRANA,TRANB
COM PLEX , D \(\mathbb{M}\) ENSION (:,:) ::A , B , C
\(\mathbb{N}\) TEGER (8) :: ISGN,M,N,LDA,LDB,LDC, \(\mathbb{N} F O\)
REAL :: SCALE

\section*{C INTERFACE}
\#include <sunperfh>
void ctrsyl(chartrana, char tranb, int isgn, intm, int n, com plex *a, int lda, com plex *b, int ldb, com plex \({ }^{*}\) c, int ldc, float *scale, int *info);
void ctrsyl 64 (chartrana, chartranb, long isgn, long m, long \(n\), com plex *a, long lda, com plex *b, long ldb, com plex *C, long ldc, float *scale, long *info);

\section*{PURPOSE}
ctrsylsolves the com plex Sylvesterm atrix equation:
op (A ) *X \(+X\) *op \((B)=\) scale \({ }^{\star} C\) or \(o p(A) * X-X * o p(B)=\) scale \({ }^{*} C\),
where op \((A)=A\) or \(A * * H\), and \(A\) and \(B\) are both upper triangular. \(A\) is \(M\) boy -M and B is N foy -N ; the righthand side C and the solution X are M boy N ; and scale is an output.scale factor, set<= 1 to avoid overflow in \(X\).

\section*{ARGUMENTS}

TRANA (input)
Specifies the option op (A):
\(=N^{\prime}: \operatorname{op}(A)=A \quad(N \circ\) transpose)
\(=C\) ':op \((A)=A * * H \quad\) (C onjugate transpose)

TRANB (input)
Specifies the option op (B):
\(=N\) : op \((B)=B \quad(N \circ\) transpose)
\(=C\) ':op \((B)=B * * H \quad\) (Conjugate transpose)
ISGN (input)
Specifies the sign in the equation:
\(=+1\) : solve op (A ) *X + X *op (B) = scale*C
\(=-1\) : solve op \((A) * X-X\) *op \((B)=\) scale \({ }^{*} C\)
\(M\) (input) The order of the \(m\) atrix \(A\), and the num ber of row \(s\)
in the m atrioes X and \(\mathrm{C} . \mathrm{M}>=0\).

N (input) The order of the \(m\) atrix \(B\), and the num ber of collm ns in the m atrices X and \(\mathrm{C} . \mathrm{N}>=0\).

A (input) The upper triangularm atrix A .
LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, \mathrm{M})\).
\(B\) (input) The upper triangularm atrix \(B\).
LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

C (input/output)
On entry, the \(M-b y-N\) righthand side \(m\) atrix C. On exit, \(C\) is overw rilten by the solution \(m\) atrix \(X\).

LD C (input)
The leading dim ension of the aray C. LD C >= \(\max (1, \mathrm{M})\)

SCALE (output)
The scale factor, scale, set <= 1 to avoid overflow in \(X\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthad an illegalvalue
\(=1\) : A and B have com m on or very close eigenvalues; perturbed values w ere used to solve the equation (but the \(m\) atrices \(A\) and \(B\) are unchanged).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctrti2 -com pute the inverse of a com plex upper or low er triangularm atrix

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NECTRTI2 (UPLO,DIAG,N,A,LDA, INFO)}
CHARACTER * 1 UPLO,DIAG
COM PLEX A (LDA,*)
\mathbb{NTEGER N,LDA,INFO}
SUBROUTINECTRTI2_64(UPLO,D IAG,N,A,LDA, INFO)
CHARACTER * 1UPLO,DIAG
COM PLEX A (LDA,*)
INTEGER*8N,LDA,INFO
F95 INTERFACE
SUBROUT\mathbb{NE TRTI2 (UPLO,D IAG, N ],A ,[LDA],[NFO])}
CHARACTER (LEN=1) ::UPLO,D IAG
COM PLEX,D IM ENSION (:,:) ::A
INTEGER ::N,LDA,}\mathbb{NFO
SUBROUT\mathbb{NE TRTI2_64 (UPLO,D IAG, N ],A,[LDA ], [NNFO])}
CHARACTER (LEN=1)::UPLO,D IA G
COM PLEX,D IM ENSION (:,:)::A
\mathbb{NTEGER (8)::N,LDA,}\mathbb{NFO}

```
C INTERFACE
    \#include <sunperfh>
void ctrti2 (char uple, chardiag, int \(n\), complex *a, int lda, int*info);
void ctrti2_64 (charuplo, chardiag, long n, com plex *a, long lda, long *info);

\section*{PURPOSE}
ctrti2 com putes the inverse of a com plex upper or low er triangularm atrix.

This is the Level2 B LAS version of the algorithm .

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the m atrix A is upper or low er
triangular. = U ': U pper triangular
= L': Low ertriangular

D IA G (input)
Specifies w hether ornot the m atrix A is unittriangular. \(=\mathrm{N}\) ': N on-unittriangular
\(=\mathrm{U}\) ': Unittriangular

N (input) The order of them atrix A. \(\mathrm{N}>=0\).

A (input/output)
O n entry, the triangularm atrix A. If UPLO = \(\mathrm{U}^{\prime}\), the leading \(n\) by \(n\) uppertriangularpart of the array A contains the uppertriangularm atrix, and the strictly low er triangular part of A is not referenced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading n by n low er triangularpart of the array A contains the low ertriangularm atrix, and the strictly upper triangularpart ofA is not referenced. IfD IA G \(=\) U',the diagonal elem ents of A are also not referenced and are assum ed to be 1 .

On exit, the (triangular) inverse of the original \(m\) atrix, in the sam e storage form at.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

IN FO (output)
= 0: successfulexit
< 0: if \(\mathbb{N N}\) FO \(=-\mathrm{k}\), the k -th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctrtri-com pute the inverse of complex upper or low er triangularm atrix A

\section*{SYNOPSIS}
```

SU BROUTINE CTRTRI(UPLO,DIAG,N,A,LDA, NNFO)
CHARACTER * 1 UPLO,D IAG
COM PLEX A (LDA,*)
INTEGER N,LDA,INFO
SU BROUT\mathbb{NE CTRTRI_64(UPLO,D IAG,N,A,LDA, INFO)}
CHARACTER * 1 UPLO,D IAG
COM PLEX A (LDA,*)
INTEGER*8N,LDA,INFO

```
F95 INTERFACE
    SU BROUTINE TRTRI(UPLO, D \(\mathbb{I A} G, \mathbb{N}], A,[L D A],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) :: UPLO, D IAG
    COMPLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N}\) TEGER ::N,LDA, \(\mathbb{N}\) FO
    SU BROUTINE TRTRI_64 (UPLO, D \(\mathbb{I A G}, \mathbb{N}], A,[L D A],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO,D IA G
    COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::A
    \(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} A, \mathbb{N}\) FO
C INTERFACE
    \#include <sunperfh>
void ctrtri(charuplo, chardiag, int n, com plex *a, int lda, int*info);
void ctrtri_64 (charuplo, chardiag, long n, com plex *a, long lda, long *info);

\section*{PURPOSE}
ctrtricom putes the inverse of a com plex upper or low er triangularm atrix A.

This is the Level3 B LAS version of the algorithm .

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}:\) A is upper triangular;
\(=\mathbb{L}\) ': A is low er triangular.

D IA G (input)
\(=\mathrm{N}^{\prime}: A\) is non-unittriangular;
\(=U\) : A is unittriangular.

N (input) The order of them atrix A. \(\mathrm{N}>=0\).

A (input/output)
O n entry, the triangularm atrix A . If \(\mathrm{UPLO}=\mathrm{U}\) ', the leading N -by- N uppertriangularpart of the anay A contains the uppertriangularm atrix, and the strictly low er triangular part of A is not referenced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading N boy -N low er triangularpart of the array A contains the low er triangularm atrix, and the strictly upper triangularpart ofA is not referenced. IfD IA G = U', the diagonal elem ents of \(A\) are also not referenced and are assum ed to be 1 . O n exit, the (triangular) inverse of the original \(m\) atrix, in the sam e storage form at.

LDA (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an ille-
galvalue
\(>0:\) if \(\mathbb{N} F O=i, A(i, i)\) is exactly zero. The triangular \(m\) atrix is singular and its inverse can notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ctrtes - solve a triangular system of the form \(A * X=B\), \(\mathrm{A} * * \mathrm{~T} * \mathrm{X}=\mathrm{B}\),orA \({ }^{* *} \mathrm{H}_{\mathrm{H}} * \mathrm{X}=\mathrm{B}\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CTRTRS(UPLO,TRANSA,D IAG,N,NRHS,A,LDA,B,LDB, INFO)}
CHARACTER * 1UPLO,TRANSA,DIAG
COM PLEX A (LDA,*),B (LDB,*)
INTEGERN,NRHS,LDA,LDB,}\mathbb{N}F
SU BROUT\mathbb{NECTRTRS_64 UPLO,TRANSA,DIAG,N,NRHS,A,LDA,B,LDB,}
\mathbb{NFO)}
CHARACTER * 1UPLO,TRANSA,DIAG
COM PLEXA (LDA,*),B (LDB,*)
\mathbb{NTEGER*8N,NRHS,LDA,LDB,INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE TRTRS (UPLO, [TRANSA],D IAG, \(\mathbb{N}], \mathbb{N R H S}], A,[L D A], B,[L D B]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,B
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, \mathbb{N} F O\)

SU BROUTINE TRTRS_64 (UPLO, [TRANSA],D IAG, \(\mathbb{N}],[\mathbb{N} R H S], A,[L D A], B\), [LDB], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : ::A, B
\(\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D B, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void ctrtrs (charuplo, chartransa, chardiag, int \(n\), int nrhs, com plex *a, int lda, com plex *b, int ldb, int*info);
void ctrtrs_64 (charuplo, chartransa, char diag, long n, long nirhs, com plex *a, long lda, com plex *b, long ldb, long *info);

\section*{PURPOSE}
ctutrs solves a triangular system of the form where \(A\) is a triangularm atrix of order \(N\), and \(B\) is an \(N\) -by-NRHS m atrix. A check ism ade to verify that A is nonsingular.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}: \mathrm{A}\) is uppertriangular;
= L': A is low ertriangular.
TRANSA (input)
Specifies the form of the system of equations:
\(=\mathrm{N}\) ': A * \(\mathrm{X}=\mathrm{B} \quad\) N o transpose)
\(=T: A * * T * X=B \quad(T\) ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose)
TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
\(=N^{\prime}: A\) is non-unit triangular;
\(=\mathrm{U}\) ': A is unit triangular.
N (input) The order of the \(m\) atrix \(\mathrm{A} . \mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the m atrix B. NRHS \(>=0\).

A (input) The triangularm atrix A. If \(\mathrm{PLO}=\mathrm{U}\) ', the lead-
ing N -by N upper triangularpart of the array A
contains the upper triangular \(m\) atrix, and the strictly low ertriangularpartofA is not refer-
enced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading N -by N lower triangular part of the array A contains the low er triangularm atrix, and the strictly upper triangular part ofA is not referenced. IfD \(\mathbb{I A} G=U\) ', the diagonalelem ents of \(A\) are also not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, N)\).

B (input/output)
On entry, the right hand side m atrix B. On exit, if \(\mathbb{N} F O=0\), the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array \(B\). LD B \(>=\) \(\max (1, N)\).
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the \(i\)-th diagonalelem ent of \(A\) is zero, indicating that the \(m\) atrix is singular and the solutions \(X\) have notbeen com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ctzrqf-routine is deprecated and has been replaced by routine C TZRZF

\section*{SYNOPSIS}
```

SUBROUTINECTZRQFM,N,A,LDA,TAU,\mathbb{NFO)}
COM PLEX A (LDA,*),TAU (*)

```

```

SUBROUT\mathbb{NECTZRQF_64M,N,A,LDA,TAU, INFO)}
COM PLEX A (LDA,*),TAU(*)
INTEGER*8M,N,LDA,INFO
F95 INTERFACE
SUBROUT\mathbb{NE TZRQF (M ], N ],A, [LDA],TAU, [NFO])}
COM PLEX,D IM ENSION (:) ::TAU
COMPLEX,D IM ENSION (:,:) ::A
\mathbb{NTEGER ::M ,N,LDA,NNFO}

```

```

    COM PLEX,D IM ENSION (:) ::TAU
    COM PLEX,D IM ENSION (:,:)::A
    \mathbb{NTEGER (8) ::M,N,LDA,INFO}
    ```
C INTERFACE
    \#include <sunperfh>
void ctzrqf(intm, intn, com plex *a, int lda, com plex *tau, int*info);
void ctzrqf_64 (long m , long n, com plex *a, long lda, com plex
*tau, long *info);

\section*{PURPOSE}
ctzrqf routine is deprecated and has been replaced by routine CTZRZF.

C TZRQF reduces the M -by -N ( \(\mathrm{M}<=\mathrm{N}\) ) com plex upper trapezoidal \(m\) atrix \(A\) to upper triangular form by meansofunitary transform ations.

The upper trapezoidalm atrix A is factored as
\[
A=\left(\begin{array}{ll}
R & 0
\end{array}\right) * Z,
\]
w here Z is an N -by -N unitary m atrix and R is an M -by -M upper triangularm atrix.

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of colum ns of the matrix \(\mathrm{A} . \mathrm{N}>=\mathrm{M}\).

A (input/output)
O \(n\) entry, the leading \(\mathrm{M}-b y-\mathrm{N}\) upper trapezoidal part of the amay A m ust contain them atrix to be factorized. On exit, the leading \(M\)-by -M upper triangularpart ofA contains the upper triangular \(m\) atrix \(R\), and elem ents \(M+1\) to \(N\) of the first \(M\) row s of \(A, w\) ith the array \(T A U\), represent the unitary \(m\) atrix \(Z\) as a productof \(M\) elem entary reflectors.

LD A (input)
The leading dim ension of the array A. LDA >= max (1,M).

TAU (output)
The scalar factors of the elem entary reflectors.
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
< 0 : if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an ille-

\section*{FURTHER DETAILS}

The factorization is obtained by H ouseholder'sm ethod. The k th transform ation \(m\) atrix, Z ( \(k\) ), w hose conjugate transpose is used to introduce zeros into the ( \(m-k+1\) ) th row of \(A\), is given in the form
\[
\left.\begin{array}{c}
\mathrm{Z}(\mathrm{k})=\left(\begin{array}{ll}
\mathrm{I} & 0
\end{array}\right), \\
\left(\begin{array}{l}
\mathrm{O}
\end{array} \mathrm{~T}(\mathrm{k})\right.
\end{array}\right),
\]
where
\[
\begin{gathered}
T(k)=I-\tan * u(k) * u(k))^{\prime} \quad u(k)=\left(\begin{array}{ll}
1
\end{array}\right), \\
\left(\begin{array}{l}
0
\end{array}\right) \\
(z(k))
\end{gathered}
\]
tau is a scalar and \(z(k)\) is an ( \(n-m\) ) elem ent vector. tau and \(\mathrm{z}(\mathrm{k})\) are chosen to annihilate the elem ents of the kth row of .

The scalartau is retumed in the \(k\) th elem entofTA \(U\) and the vectoru ( \(k\) ) in the \(k\) th row of \(A\), such that the elem ents of \(z(k)\) are in \(a(k, m+1), \ldots, a(k, n)\). The elem ents ofR are retumed in the upper triangularpartofA.
\(Z\) is given by
\[
Z=Z(1) * Z(2) * \ldots * Z(m) .
\]

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ctzrzf-reduce the M -by -N ( \(\mathrm{M}<=\mathrm{N}\) ) complex upper trapezoidal \(m\) atrix A to upper triangular form by \(m\) eans of unitary transform ations

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CTZRZFM,N,A,LDA,TAU,W ORK,LW ORK,INFO)}
COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGERM,N,LDA,LWORK,INFO

```

```

COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGER*8M,N,LDA,LW ORK,INFO
F95 INTERFACE
SU BROUT\mathbb{NE TZRZF (\mathbb{M ], N ],A, [LDA ],TAU, [W ORK ], [LW ORK ], [NFO ])}}\mathbf{[N})
COM PLEX,DIM ENSION (:) ::TAU,W ORK
COM PLEX,D IM ENSION (:,:) ::A
\mathbb{NTEGER ::M ,N,LDA,LW ORK,INFO}

```

```

    COMPLEX,DIM ENSION (:) ::TAU,W ORK
    COM PLEX,D IM ENSION (:,:) ::A
    \mathbb{NTEGER (8) ::M ,N,LDA,LW ORK,INFO}
    C INTERFACE
\#include <sunperfh>

```
void ctzrzf(intm, intn, com plex *a, int lda, com plex *tau, int*info);
void ctzrzf_64 (long m, long n, com plex *a, long lda, com plex *tau, long *info);

\section*{PURPOSE}
ctzrzf reduces the M boy \(-\mathrm{N} \quad(\mathrm{M}<=\mathrm{N})\) com plex uppertrapezoidal \(m\) atrix \(A\) to upper triangular form by \(m\) eans of unitary transform ations.

The uppertrapezoidalm atrix \(A\) is factored as
\[
A=\left(\begin{array}{ll}
R & 0
\end{array}\right) * Z,
\]
\(w\) here \(Z\) is an \(N\) boy \(-N\) unitary \(m\) atrix and \(R\) is an \(M\) boy \(-M\) upper triangularm atrix.

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input/output)
On entry, the leading M -by -N upper trapezoidal part of the array A m ust contain the m atrix to be factorized. On exit, the leading M -by -M upper triangularpart ofA contains the upper triangular \(m\) atrix \(R\), and elem ents \(M+1\) to \(N\) of the first \(M\) row s of \(A\), w ith the array TA \(U\), represent the unitary \(m\) atrix \(Z\) as a productof \(M\) elem entary reflectors.

LD A (input)
The leading dim ension of the array A. LDA >= max (1, M).

TAU (output)
The scalar factors of the elem entary reflectors.
W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

The dimension of the array \(W\) ORK. LW ORK >= \(m\) ax \((1, M)\). Foroptim um perform anœ \(L W O R K>=M * N B\), w here NB is the optim alblocksize.

If LW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{N F O}=-\) i, the \(i\)-th argum ent had an illegal value

\section*{FURTHER DETAILS}

B ased on contributions by
A. Petitet, C om puterScience D ept., U niv . of Tenn., K noxville, U SA

The factorization is obtained by H ouseholdersm ethod. The \(k\) th transform ation \(m\) atrix, \(Z(k)\), which is used to introduce zeros into the ( \(m-k+1\) )th row ofA, is given in the form
\[
\begin{gathered}
\mathrm{Z}(\mathrm{k})=\left(\begin{array}{ll}
\mathrm{I} & 0
\end{array}\right), \\
\binom{\mathrm{O}}{\mathrm{~T}(\mathrm{k})}
\end{gathered}
\]
where
\[
\begin{gathered}
\left.T(k)=I-\tan { }^{*} u(k) * u(k)\right), \quad u(k)=\left(\begin{array}{ll}
1
\end{array}\right), \\
\left(\begin{array}{l}
0
\end{array}\right) \\
(z(k))
\end{gathered}
\]
tau is a scalar and \(z(k)\) is an ( \(n-m\) ) elem ent vector. tau and \(\mathrm{z}(\mathrm{k})\) are chosen to annihilate the elem ents of the kth row of .

The scalartau is retumed in the kth elem entofTAU and the vectoru ( \(k\) ) in the kth row of A, such that the elem ents of \(z(k)\) are in \(a(k, m+1), \ldots, a(k, n)\). The elem ents ofR are retumed in the upper triangularpartofA.
\(Z\) is given by
\[
Z=Z(1) * Z(2) * \ldots * Z(m) .
\]

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cung2l-generate an \(m\) by \(n\) com plex \(m\) atrix \(Q\) w th orthonormalcolum ns,

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CUNG2L M,N,K,A,LDA,TAU,W ORK,INFO)}
COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGERM,N,K,LDA,INFO
SU BROUT\mathbb{NE CUNG 2L_64M,N,K,A,LDA,TAU,W ORK, INFO)}
COM PLEX A (LDA,*),TAU (*),W ORK (*)
\mathbb{NTEGER*8M,N,K,LDA, INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNG 2L \(\mathbb{M}, \mathbb{N}], \mathbb{K}], A,[L D A], T A U,[W O R K],[\mathbb{N} F O])\)
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : A
\(\mathbb{N} T E G E R:: M, N, K, L D A, \mathbb{N} F O\)
SU BROUTINE UNG2L_64 \(\mathbb{M}, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{W} O R K],[\mathbb{N} F O])\)
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : A
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LD} A, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include < sunperfh>
void cung2l(intm, intn, intk, com plex *a, int lda, com -
void cung2l 64 (long m , long n, long k, com plex *a, long lda, com plex *tau, long *info);

\section*{PURPOSE}
cung2lL generates an \(m\) by \(n\) com plex \(m\) atrix \(Q w\) th orthonorm al colum ns, which is defined as the lastn colum ns of a product ofk elem entary reflectors of orderm
\[
Q=H(k) \ldots H(2) H(1)
\]
as retumed by CGEQ LF.

\section*{ARGUMENTS}

M (input) The num ber of row sof the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).
N (input) The num ber of Colum ns of the m atrix \(\mathrm{Q} . \mathrm{M}>=\mathrm{N}>=\) 0.
\(K\) (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q . N>=K>=0\).

A (input/output)
On entry, the \((n-k+i)\)-th colum nm ust contain the vector which defines the elem entary reflector H (i), for \(i=1,2, \ldots, k\), as retumed by CGEQ LF in the last \(k\) colum ns of its anray argum entA. On exit, the \(m\)-by-n m atrix \(Q\).

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CG EQ LF .
```

W ORK (w orkspace)

```
dim ension \((\mathbb{N})\)
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthas an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cung2r-generate an \(m\) by \(n\) com plex \(m\) atrix \(Q\) w th orthonorm alcolum ns,

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE CUNG2R M,N,K,A,LDA,TAU,W ORK, NNFO)}
COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGERM,N,K,LDA, INFO
SUBROUT\mathbb{NE CUNG2R_64M,N,K,A,LDA,TAU,W ORK,INFO)}
COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGER*8 M ,N,K,LDA, IN FO

```

\section*{F95 INTERFACE}

SU BROUTINE UNG 2R \(\mathbb{M}, \mathbb{N}],[K], A,[L D A], T A U,[W O R K],[\mathbb{N} F O])\)
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : : A
\(\mathbb{N} T E G E R:: M, N, K, L D A, \mathbb{N} F O\)

SU BROUTINE UNG 2R_64 \(\mathbb{M}, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{W} O R K],[\mathbb{N F O}])\)
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : A
\(\mathbb{N}\) TEGER (8) ::M , N , K,LDA, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include < sunperfh>
void cung \(2 r\) (intm , intn, int , com plex *a, int lda, com -
void cung2r_64 (long m, long n, long k, com plex *a, long lda, com plex *tau, long *info);

\section*{PURPOSE}
cung2rR generates an \(m\) by \(n\) com plex \(m\) atrix \(Q w\) th orthonorm al colum ns, which is defined as the firstn colum ns of a product ofk elem entary reflectors of orderm
\[
Q=H(1) H(2) \ldots H(k)
\]
as retumed by CGEQRF.

\section*{ARGUMENTS}

M (input) The num ber of row sof the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).
N (input) The num ber of Colum ns of the m atrix \(\mathrm{Q} . \mathrm{M}>=\mathrm{N}>=\) 0.
\(K\) (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q . N>=K>=0\).

A (input/output)
On entry, the \(i\)-th columnm ustcontain the vector which defines the elem entary reflectorH (i), for i \(=1,2, \ldots, k\), as retumed by CG EQRF in the first k colum ns of its array argum entA. On exit, the \(m\) by \(n m\) atrix \(Q\).

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CGEQRF.

W ORK (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthas an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cungbr-generate one of the com plex unitary \(m\) atrices \(Q\) or
\(P * * H\) determ ined by CGEBRD when reducing a com plex \(m\) atrix A
to bidiagonal form

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CUNGBR NECT,M ,N,K,A,LDA,TAU,W ORK,LW ORK,INFO)}
CHARACTER * 1VECT
COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGER M,N,K,LDA,LW ORK,\mathbb{NFO}

```

```

CHARACTER * 1VECT
COM PLEX A (LDA,*),TAU (*),W ORK (*)
\mathbb{N}TEGER*8M,N,K,LDA,LW ORK,\mathbb{NFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNGBR (NECT, M, \(\mathbb{N}], K, A,[L D A], T A U,[W O R K],[L W ~ O R K]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::VECT
COM PLEX,D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : : A
\(\mathbb{N} T E G E R:: M, N, K, L D A, L W O R K, \mathbb{N} F O\)
SU BROUTINE UNGBR_64 \(N E C T, M, \mathbb{N}], K, A,[L D A], T A U,[W O R K],[L W O R K]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::VECT
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void cungbr(charvect, intm, intn, intk, com plex *a, int lda, com plex *tau, int *info);
void cungbr_64 (charvect, long m, long \(n\), long \(k\), com plex
*a, long lda, com plex *tau, long *info);

\section*{PURPOSE}
cungbrgenerates one of the com plex untary matrioes Q or \(\mathrm{P} * * \mathrm{H}\) determ ined by CGEBRD when reducing a com plex \(m\) atrix \(A\) to bidiagonal form : \(A=Q * B * P * * H . Q\) and \(P * * H\) are defined as products ofelem entary reflectors \(H\) (i) orG (i) respectively .

IfVECT \(=Q\) ', \(A\) is assum ed to have been an \(M\) boy \(K m\) atrix, and \(Q\) is of orderM :
ifm \(>=k, Q=H(1) H(2) \ldots H(k)\) and CUNGBR retams the firstn colum \(n s\) of \(Q\), where \(m>=n>=k\);
ifm \(<k, Q=H(1) H(2) \ldots H(m-1)\) and CUNGBR retums \(Q\) as an \(M\) łoy \(-M\) matrix.

IfVECT \(=P\) ', A is assum ed to have been a K -by -N m atrix, and \(P * * H\) is oforderN :
if \(k<n, P * * H=G(k) \ldots G(2) G(1)\) and \(C U N G B R\) retums the firstm row sof \({ }^{* *} H\), where \(n>=m>=k\);
if \(k>=n, P * * H=G(n-1) \ldots G(2) G(1)\) and CUNGBR retums \(\mathrm{P} * * \mathrm{H}\) as an N boy -N m atrix.

\section*{ARGUMENTS}

VECT (input)
Specifies w hether the m atrix Q orthem atrix \(P * * H\)
is required, as defined in the transform ation
applied by CGEBRD :
= Q ': generate Q ;
\(=P^{\prime}:\) generate \(\mathrm{P}^{* *} \mathrm{H}\).

M (input) The num ber of row s of the \(m\) atrix \(Q\) orP** \(H\) to be retumed. \(\mathrm{M}>=0\).
\(N\) (input) The num ber of colum ns of the \(m\) atrix \(Q\) or \(P * * H\) to
be retumed. \(\mathrm{N}>=0\). IfVECT \(=\) Q', \(\mathrm{M}>=\mathrm{N}>=\) \(m\) in \((M, K) ;\) ifVECT \(=P^{\prime}, N>=M>=m\) in \((\mathbb{N}, K)\).

K (input) IfVECT = Q', the num ber of colum ns in the original \(M\) by \(K\) matrix reduced by CGEBRD. IfVECT = \(P\) ', the num ber of row \(s\) in the original \(K\) boy \(N\) \(m\) atrix reduced by \(C G E B R D . K>=0\).

A (input/output)
O n entry, the vectors w hich define the elem entary reflectors, as retumed by CGEBRD. On exit, the M boy -N m atrix Q orP**H.

LDA (input)
The leading dim ension of the aray A.LDA >=M.

TAU (input)
\((m\) in \(M, K))\) ifVECT \(=Q^{\prime}(m\) in \((\mathbb{N}, K))\) ifVECT \(=P^{\prime}\)
TAU (i) m ustcontain the scalar factor of the ele\(m\) entary reflectorH (i) orG (i), which determ ines Q or \(P^{* *} H\), as retumed by \(C\) G EBRD in its array argu\(m\) entTAUQ orTAUP.

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the anay W ORK. LW ORK >= \(m\) ax \((1, m\) in \(M, N))\). Foroptim um perform ance LW ORK \(>=\) \(m\) in \(M, N) \star N B, w\) here \(N B\) is the optim alblocksize.

IfLW O RK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cunghr-generate a complex unitary \(m\) atrix \(Q\) which is defined as the productof \(\mathbb{H}\) I-HO elem entary reflectors of order \(N\), as retumed by CGEH RD

\section*{SYNOPSIS}

```

COM PLEX A (LDA,*),TAU (*),W ORK (*)

```


```

COM PLEX A (LDA,*),TAU (*),W ORK (*)

```

F95 INTERFACE
SUBROUTINE UNGHR ( \(\mathbb{N}], \mathbb{I} O, \mathbb{H} I, A,[L D A], T A U,[\mathbb{W}\) ORK ], [LW ORK ], [ \(\mathbb{N} F O]\) )
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COMPLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: N, \mathbb{L} O, \mathbb{H} I, L D A, L W O R K, \mathbb{N} F O\)
SU BROUTINE UNGHR_64 ( \(\mathbb{N}], \mathbb{H}, \mathbb{H} I, A,[L D A], T A U,[W O R K],[L W\) ORK ],
    [ \(\mathbb{N}\) FO ])
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : ::A
\(\mathbb{N} T E G E R(8):: N, \mathbb{L} O, \mathbb{H} \mathrm{I}, \mathrm{LD} A, L W O R K, \mathbb{N} F O\)
C INTERFACE
    \#include <sunperfh>
void cunghr(intn, intilo, int ini, com plex *a, int lda, com plex *tau, int *info);
void cunghr_64 (long n, long 1 , long ini, com plex *a, long lda, com plex *tau, long *info);

\section*{PURPOSE}
cunghrgenerates a com plex unitary \(m\) atrix \(Q\) which is defined as the product of IH I-HO elem entary reflectors of orderN, as retumed by CGEHRD :
\(\mathrm{Q}=\mathrm{H}\) ( i ) H ( \(\mathrm{i} \mathrm{O}+1\) ) . . . H (ihi-1).

\section*{ARGUMENTS}

N (input) The order of the m atrix \(\mathrm{Q} . \mathrm{N}>=0\).
IIO (input)
IO and \(\mathbb{H}\) Im usthave the sam e values as in the previous call of CGEHRD.Q is equal to the unit \(m\) atrix except in the subm atrix
Q (ilo+1: ihi,ilo+1: :hin). \(1<=\mathbb{H O}<=\mathbb{H} \mathrm{I}<=\mathrm{N}\), if \(\mathrm{N}>0\); \(\mathbb{H} \mathrm{O}=1\) and \(\mathbb{H} \mathrm{I}=0\), if \(\mathrm{N}=0\).

IH I (input)
See the description of IH I.

A (input/output)
O \(n\) entry, the vectors w hich define the elem entary reflectors, as retumed by CGEHRD. On exit, the N by -N unitary m atrix Q .

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

TAU (input)
TAU (i) m ust contain the scalar factor of the elem entary reflectort (i), as retumed by CG EH RD .

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK ( 1 ) retums the optim al LW ORK.

The dim ension of the array \(W\) ORK. LW ORK >= \(\mathbb{H} I-\mathbb{H O}\). For optim um perform ance LW ORK \(>=(\mathbb{H} I-\mathbb{H} O)^{*} N B\), where NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim al size of theW ORK ancay, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
<0: if \(\mathbb{N F O}=-\) i, the \(i\)-th argum ent had an illegal value

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cung12 - generate an \(m\)-by-n com plex \(m\) atrix \(Q\) w th orthonorm alrow s,

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE CUNGL2M,N,K,A,LDA,TAU,W ORK,INFO)}
COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGERM,N,K,LDA,INFO
SU BROUT\mathbb{NE CUNGL2_64 M ,N ,K,A,LDA ,TAU,W ORK, NNFO)}
COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGER*8 M ,N,K,LDA,\mathbb{NFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNGL2 ( \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{N} O R K],[\mathbb{N} F O])\)
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : : A
\(\mathbb{N} T E G E R:: M, N, K, L D A, \mathbb{N} F O\)
SU BROUTINE UNGL2_64 ( \(\mathbb{M}], \mathbb{N}],[\mathbb{K}], A,[L D A], T A U,[\mathbb{W} O R K],[\mathbb{N} F O])\)
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : A
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LD} A, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include < sunperfh>
void cungl2 (intm , intn, intk, com plex *a, int lda, com -
void cungl2_64 (long m , long n, long k, com plex *a, long lda, com plex *tau, long *info);

\section*{PURPOSE}
cungl2 generates an \(m\)-by-n com plex \(m\) atrix \(Q\) with orthonorm al row S , w hich is defined as the firstm row s of a productofk elem entary reflectors of ordern
\[
Q=H(k) ' \ldots H(2) \cdot H(1) '
\]
as retumed by CGELQF.

\section*{ARGUMENTS}

M (input) The num ber of row sof the \(m\) atrix \(Q . M>=0\).
N (input) The num ber of C lum ns of the m atrix \(\mathrm{Q} . \mathrm{N}>=\mathrm{M}\).
\(K\) (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q . M>=K>=0\).

A (input/output)
On entry, the \(i-\) th row must contain the vector which defines the elem entary reflectort (i), for i
\(=1,2, \ldots, k\), as retumed by CG ELQ F in the first k row sof its amay argum entA. On exit, them by \(n\) \(m\) atrix \(Q\).

LD A (input)
The first dim ension of the array A. LDA >= \(\max (1, M)\).

TAU (input)
TAU (i) mustcontain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CG ELQ F.

W ORK (w orkspace)
dim ension M)
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0\) : if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthas an illegalvahue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cunglq - generate an M -by -N com plex m atrix Q w ith orthonormalrow s,

\section*{SYNOPSIS}

```

COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGERM,N,K,LDA,LW ORK,\mathbb{NFO}
SUBROUTINE CUNGLQ_64 M ,N,K,A,LDA,TAU,W ORK,LW ORK,NNFO )
COM PLEX A (LDA,*),TAU (*),W ORK (*)

```


\section*{F95 INTERFACE}

SU BROUTINE UNGLQ \(M, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{N} O R K],[L W ~ O R K],[\mathbb{N} F O])\)
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, K, L D A, L W\) ORK, \(\mathbb{N} F O\)
SU BROUTINE UNGLQ_64 \(M, \mathbb{N}],[K], A,[L D A], T A U,[W O R K],[L W O R K]\), [ \(\mathbb{N} F O\) ])

COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{I M}\) ENSION (:,: : ::A
\(\mathbb{N} T E G E R(8):: M, N, K, L D A, L W O R K, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include < sunperfh>
void cunglq (intm, intn, intk, com plex *a, int lda, com plex *tau, int*info);
void cunglq_64 (long m , long n, long k, com plex *a, long lda, com plex *tau, long *info);

\section*{PURPOSE}
cunglq generates an M -by N com plex m atrix Q w ith orthonorm al row \(S\), which is defined as the firstM row \(s\) of a product of \(K\) elem entary reflectors of orderN
\[
Q=H(k)^{\prime} \ldots . . \text { H (2)'H (1)' }
\]
as retumed by CGELQ F .

\section*{ARGUMENTS}

M (input) The num ber of row sof the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{Q} \cdot \mathrm{N}>=\mathrm{M}\).
\(K\) (input) The num ber of elem entary reflectors w hose product
defines the \(m\) atrix \(Q . M>=K>=0\).
A (input/output)
On entry, the \(i\)-th row must contain the vector which defines the elem entary reflectorH (i), for i \(=1,2, \ldots, k\), as retumed by CGELQF in the first \(k\) row sof its amay argum entA. On exit, the \(M\) boy N \(m\) atrix \(Q\).

LDA (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectorH (i), as retumed by CGELQ F.

W ORK (w orkspace)
On exit, if \(\mathbb{N F} F=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the array W ORK. LW ORK >= \(m a x(1, M)\). Foroptim um perform anœ LW ORK \(>=M\) *NB,
w here NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0 : successfiulexit;
\(<0:\) if \(\mathbb{I N F O}=-i\), the \(i\) th argum enthas an illegalvałue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cungql-generate an M -by N com plex m atrix Q w th orthonormalcolum ns,

\section*{SYNOPSIS}

```

COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGERM,N,K,LDA,LW ORK,INFO
SUBROUTINE CUNGQL_64 M ,N,K,A,LDA,TAU,W ORK,LW ORK,NNFO )
COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGER*8M,N,K,LDA,LW ORK,\mathbb{NFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNGQL M, \(\mathbb{N}],[K], A,[L D A], T A U,[\mathbb{N} O R K],[L W ~ O R K],[\mathbb{N} F O])\)
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, K, L D A, L W O R K, \mathbb{N} F O\)
SUBROUTINE UNGQL_64 \(\mathbb{M}, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{W}\) ORK ], [LW ORK], [ \(\mathbb{N} F O\) ])

COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{I M}\) ENSION (:,: : ::A
\(\mathbb{N} T E G E R(8):: M, N, K, L D A, L W O R K, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include < sunperfh>
void cungql(intm, intn, intk, com plex *a, int lda, com plex *tau, int*info);
void cungql 64 (long m , long n, long k, com plex *a, long lda, com plex *tau, long *info);

\section*{PURPOSE}
cungqlgenerates an M -by -N com plex m atrix Q w ith orthonorm al colum ns, which is defined as the lastN colum ns of a product of K elem entary reflectors of orderM
\[
Q=H(k) \ldots H(2) H(1)
\]
as retumed by CGEQ LF.

\section*{ARGUMENTS}

M (input) The num ber of row sof the \(m\) atrix \(Q . M>=0\).
N (input) The num ber of colum ns of the matrix Q.M \(>=\mathrm{N} \quad>=\) 0.

K (input) The num ber of elem entary reflectors \(w\) hose product defines the m atrix \(\mathrm{Q} . \mathrm{N}>=\mathrm{K}>=0\).

A (input/output)
On entry, the \((n-k+i)\)-th colum nm ust contain the vector which defines the elem entary reflector H (i), for \(i=1,2, \ldots, k\), as retumed by CGEQ LF in the last k colum ns of its anay argum entA. On exit, the \(M\)-by-N \(m\) atrix \(Q\).

LD A (input)
The first dim ension of the array A. LDA >= \(m a x(1, M)\).

TAU (input)
TAU (i) must contain the scalar factorof the ele\(m\) entary reflectorH (i), as retumed by CG EQ LF.

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, \mathrm{~W}\) ORK (1) retums the optim al
LW ORK.
LW ORK (input)
The dimension of the array W ORK. LW ORK >=
\(\max (1, N)\). Foroptim um penform ance LW ORK \(>=N * N B\), where NB is the optim alblocksize.

If LW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent has an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cungqr-generate an M -by N com plex m atrix Q w th orthonormalcolumns,

\section*{SYNOPSIS}

```

COM PLEX A (LDA,*),TAU (*),W ORK\mathbb{N (*)}
INTEGERM,N,K,LDA,LW ORK IN, INFO

```

```

COM PLEX A (LDA,*),TAU (*),W ORK IN (*)

```


\section*{F95 INTERFACE}

SU BROUTINE UNGQR \(M, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{O} O R K \mathbb{N}],[L W O R K \mathbb{N}]\), [ \(\mathbb{N} F O\) ])

COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK \(\mathbb{N}\)
COM PLEX,D \(\mathbb{M}\) ENSION (:,:)::A
\(\mathbb{N}\) TEGER :: M , N, K,LDA, LW ORK \(\mathbb{N}, \mathbb{N} F O\)
SU BROUTINE UNGQR_64 \(\mathbb{M}, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{W} O R K \mathbb{N}],[L W O R K \mathbb{N}]\), [ \(\mathbb{N}\) FO ])

COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK \(\mathbb{N}\)
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{K}\), LDA, LW ORK \(\mathbb{N}, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include < sunperfh>
void cungqr(intm, intn, intk, com plex *a, int lda, com plex *tau, int*info);
void cungqr_64 (long m, long \(n\), long k, com plex *a, long lda, com plex *tau, long *info);

\section*{PURPOSE}
cungqrgenerates an \(M\) boy -N com plex \(m\) atrix \(Q\) w ith orthonorm al colum ns, which is defined as the firstN colum ns of a product of K elem entary reflectors of orderM
\(\mathrm{Q}=\mathrm{H}(1) \mathrm{H}(2) \ldots \mathrm{H}(\mathrm{k})\)
as retumed by CGEQRF.

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{Q} . \mathrm{M}>=\mathrm{N}>=\) 0 .

K (input) The num ber of elem entary reflectors w hose product defines the m atrix \(\mathrm{Q} \cdot \mathrm{N}>=\mathrm{K}>=0\).

A (input/output)
On entry, the i-th colum \(n m\) ustcontain the vector which defines the elem entary reflectorH (i), for \(i\)
\(=1,2, \ldots, k\), as retumed by CGEQRF in the first k colum ns of its array argum entA. On exit, the \(M-\) by \(-\mathrm{N} m\) atrix Q .

LDA (input)
The first dimension of the array A. LDA >= \(\max (1, M)\).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflector \(H\) (i), as retumed by CGEQRF.

W ORK \(\mathbb{N}\) (w orkspace)
On exit, if \(\mathbb{N F O}=0, \mathrm{~W} O R K \mathbb{N}(1)\) retums the optim alLW ORK \(\mathbb{N}\).

LW ORK \(\mathbb{N}\) (input)
The dim ension of the array \(W\) ORK \(\mathbb{N}\). LW ORK \(\mathbb{N}>=\)
\(\max (1, N)\). For optim um perform ance LW ORK \(\mathbb{N}>=\) \(\mathrm{N} * \mathrm{NB}\), where NB is the optim alblocksize.

If LW ORK \(\mathbb{N}=-1\), then a workspace query is assum ed; the routine only calculates the optim al size of the \(W\) ORK \(\mathbb{N}\) array, retums this value as the firstentry of the \(W\) ORK \(\mathbb{N}\) array, and no emor \(m\) essage related to LW ORK \(\mathbb{N}\) is issued by XERBLA.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthas an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cungr2 - generate an \(m\) by \(n\) com plex \(m\) atrix \(Q\) with orthonorm alrow s,

\section*{SYNOPSIS}
```

SU BROUTINE CUNGR2M,N,K,A,LDA,TAU,W ORK, NNFO)
COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGERM,N,K,LDA,INFO
SU BROUT\mathbb{NE CUNGR2_64M,N,K,A,LDA,TAU,W ORK,INFO)}
COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGER*8 M ,N,K,LDA,INFO

```

\section*{F95 INTERFACE}

SU BROUTINE UNGR2 ( \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{N} O R K],[\mathbb{N} F O])\)
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : : A
\(\mathbb{N} T E G E R:: M, N, K, L D A, \mathbb{N} F O\)
SU BROUTINE UNGR2_64 ( \(\mathbb{M}], \mathbb{N}],[\mathbb{K}], A,[L D A], T A U,[\mathbb{W} O R K],[\mathbb{N} F O])\)
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : A
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LD} A, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include < sunperfh>
void cungr2 (intm , intn, intk, com plex *a, int lda, com -
void cungr2_64 (long m, long n, long k, com plex *a, long lda, com plex *tau, long *info);

\section*{PURPOSE}
cungr2 generates an \(m\) by \(n\) com plex \(m\) atrix \(Q\) w th orthonorm al row \(s\), which is defined as the lastm row s of a product ofk elem entary reflectors of ordern
\[
Q=H(1) \cdot H(2)^{\prime} \ldots H(k)^{\prime}
\]
as retumed by CGERQF.

\section*{ARGUMENTS}

M (input) The num ber of row sof the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).
N (input) The num ber of C lum ns of the m atrix \(\mathrm{Q} . \mathrm{N}>=\mathrm{M}\).
\(K\) (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q . M>=K>=0\).

A (input/output)
O \(n\) entry, the ( \(m-k+i\) )-th row \(m\) ustcontain the vector which defines the elem entary reflector H (i), fori \(=1,2, \ldots, k\), as retumed by CGERQF in the lastk row sof its array argum entA. O n exit, the \(m-b y-n m\) atrix \(Q\).

LD A (input)
The first dim ension of the array A. LDA >= \(\max (1, M)\).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CG ERQF.

W ORK (w orkspace)
dim ension M)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{N N F O}=-\) i, the \(i\)-th argum enthas an illegalvahue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cungrq - generate an M -by N com plex m atrix Q w th orthonormalrow s,

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CUNGRQ M,N,K,A,LDA,TAU,W ORK,LW ORK,INFO)}
COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGERM,N,K,LDA,LW ORK,\mathbb{NFO}
SU BROUTINE CUNGRQ_64 M ,N,K,A,LDA,TAU,W ORK,LW ORK, NNFO)
COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGER*8M,N,K,LDA,LW ORK,\mathbb{NFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNGRQ \(M, \mathbb{N}],[K], A,[L D A], T A U,[W O R K],[L W O R K],[\mathbb{N} F O])\)
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, K, L D A, L W\) ORK, \(\mathbb{N} F O\)

SU BROUTINE UNGRQ_64 \(M, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{W} O R K],[L W O R K]\), [ \(\mathbb{N} F O\) ])

COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{I M}\) ENSION (:,: : ::A
\(\mathbb{N} T E G E R(8):: M, N, K, L D A, L W O R K, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void cungrq (intm, intn, int \(k\), com plex *a, int lda, com plex *tau, int*info);
void cungrq_64 (long m, long n, long k, com plex *a, long lda, com plex *tau, long *info);

\section*{PURPOSE}
cungrq generates an M -by N com plex m atrix Q w ith orthonorm al row \(S\), which is defined as the lastM row s of a product of \(K\) elem entary reflectors of orderN
\[
Q=H(1) \cdot H(2)^{\prime} \ldots H(k) '
\]
as retumed by CGERQF.

\section*{ARGUMENTS}

M (input) The num ber of row sof the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{Q} \cdot \mathrm{N}>=\mathrm{M}\).
\(K\) (input) The num ber of elem entary reflectors \(w\) hose product defines the \(m\) atrix \(Q . M>=K>=0\).

A (input/output)
On entry, the ( \(m-k+i\) )-th row \(m\) ust contain the vector which defines the elem entary reflectorH (i), for \(i=1,2, \ldots, k\), as retumed by CGERQF in the lastk row sof its amay argum entA. O n exit, the M Hy -N m atrix Q .

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CGERQF.

W ORK (w orkspace)
On exit, if \(\mathbb{N F} F=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the anay \(W\) ORK. LW ORK >= max (1,M). Foroptim um perform anœ LW ORK \(>=M\) *NB,
w here NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{I N F O}=-i\), the i-th argum enthas an illegalvałue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cungtr-generate a complex unitary matrix \(Q\) which is defined as the productofn-1 elem entary reflectors of order N , as retumed by CH ETRD

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CUNGTR (UPLO,N,A,LDA,TAU,W ORK,LW ORK,INFO )}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),TAU (*),W ORK (*)
\mathbb{NTEGER N,LDA,LW ORK, INFO}
SUBROUT\mathbb{NE CUNGTR_64(UPLO,N,A,LDA,TAU,W ORK,LW ORK,INFO )}
CHARACTER * 1 UPLO
COM PLEX A (LDA,*),TAU (*),W ORK (*)
\mathbb{N}TEGER*8N,LDA,LW ORK,\mathbb{NFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNGTR (UPLO, \(\mathbb{N}], A,[L D A], T A U,[W O R K],[L W O R K],[\mathbb{N} F O])\)
CHARACTER (LEN=1) ::UPLO
COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: N, L D A, L W\) ORK, \(\mathbb{N} F O\)

SU BROUTINE UNGTR_64 (UPLO, \(\mathbb{N}\) ],A, [LDA ],TAU, [W ORK ], [LW ORK ], [ \(\mathbb{N F O}\) ])
CHARACTER (LEN=1) ::UPLO
COM PLEX,D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : A
\(\mathbb{N}\) TEGER (8) :: N, LDA,LW ORK, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void cungtr(charuplo, intn, com plex *a, int lda, com plex
*tau, int *info);
void cungtr_64 (char uplo, long n, com plex *a, long lda, com plex *tau, long *info);

\section*{PURPOSE}
cungtrgenerates a com plex unitary \(m\) atrix \(Q\) which is defined as the productofn-1 elem entary reflectors of orderN, as retumed by CHETRD :
if \(U P L O=U ', Q=H(n-1) \ldots\) (2) \(H(1)\),


\section*{ARGUMENTS}

UPLO (input)
= U ': U ppertriangle of A contains elem entary
reflectors from CHETRD; = L':Low ertriangle of A
contains elem entary reflectors from CHETRD.
N (input) The order of the \(m\) atrix \(\mathrm{Q} . \mathrm{N}>=0\).
A (input/output)
O \(n\) entry, the vectors which define the elem entary reflectors, as retumed by CHETRD. On exit, the N by N unitary m atrix Q .

LD A (input)
The leading dim ension of the array A .LD A \(>=\mathrm{N}\).
TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CHETRD .

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, \mathrm{~W}\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. LW ORK \(>=N-1\).
For optim um perform ance LW ORK \(>=(\mathbb{N}-1) * N B\), where

NB is the optim alblocksize.

IfLW O RK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cunm br-VECT = Q',CUNM BR overw rites the general com plex \(M\) by \(-\mathrm{N} m\) atrix \(C\) with \(S \mathbb{D} E=L ' S \mathbb{D} E=R\) 'TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUTINE CUNM BR NECT,S\mathbb{DE,TRANS,M,N,K,A,LDA,TAU,C,LDC,}
W ORK,LW ORK,INFO)
CHARACTER * 1VECT,SIDE,TRANS
COM PLEX A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
INTEGERM,N,K,LDA,LDC,LW ORK,\mathbb{NFO}
SUBROUTINE CUNMBR_64 NECT,SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,
W ORK,LW ORK,\mathbb{NFO)}
CHARACTER * 1 VECT,SDE,TRANS
COM PLEX A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
\mathbb{NTEGER*8M,N,K,LDA,LDC,LW ORK, INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNM BR NECT,SIDE, [TRANS], \(\mathbb{M}], \mathbb{N}], K, A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::VECT,SDE,TRANS
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:): ::A, C
\(\mathbb{N}\) TEGER :: M , N, K,LDA,LDC,LW ORK, \(\mathbb{N} F O\)
SU BROUTINE UNMBR_64 (NECT,SDE, [TRANS], \(\mathbb{M}], \mathbb{N}], K, A,[L D A], T A U\), C, [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::VECT,SDE,TRANS

\section*{C INTERFACE}
\#include <sunperfh>
void cunm br(charvect, char side, char trans, intm, int \(n\), int \(k\), com plex *a, int lda, com plex *tau, com plex * c , int ldc, int *info);
void cunm br_64 (charvect, charside, char trans, long m, long \(n\), long \(k\), com plex *a, long lda, com plex *tau, com plex *c, long ldc, long *info);

\section*{PURPOSE}
cunm brVECT = Q ', CUNM BR overw rites the general com plex M by -N m atrix C w ith
```

        SDEE= L' SDDE = R'TRANS = N':
    Q * C C *Q TRANS=C': Q**H * C C *

```
Q**H

IfVECT = P', CUNM BR overw rites the general com plex M -by -N \(m\) atrix \(C\) w ith
\[
S \mathbb{D} E=L^{\prime} \quad S \mathbb{D} E=R^{\prime}
\]

TRANS = N: \(\quad \mathrm{P} * \mathrm{C} \quad \mathrm{C} * \mathrm{P}\)
TRANS = C : \(\quad \mathrm{P} * * H * C \quad C * P * * H\)

Here \(Q\) and \(P * * H\) are the unitary \(m\) atrices determ ined by CGEBRD when reducing a com plex m atrix A to bidiagonal form : \(A=Q * B * P * * H . Q\) and \(P * * H\) are defined as products of ele\(m\) entary reflectors \(H\) (i) and \(G\) (i) respectively.

Letnq \(=m\) if \(S \mathbb{D} E=L\) 'and \(n q=n\) if \(S \mathbb{D} E=R\) '. Thus \(n q\) is the order of the unitary \(m\) atrix \(Q\) or \(P * * H\) that is applied.

IfVECT = Q',A is assum ed to have been an NQ -by K m atrix:
if \(n q>=k, Q=H(1) H(2) \ldots H(k)\);
ifnq \(<k, Q=H(1) H(2) \ldots H(n q-1)\).
IfVECT \(=P^{\prime}, A\) is assum ed to have been a \(K-b y-N Q\) matrix:
ifk < nq, P = G (1) G (2) .. . G (k);
if \(k>=n q, P=G(1) G(2) \ldots G(n q-1)\).

\section*{ARGUMENTS}

VECT (input)
\(=Q\) ': apply \(Q\) or \(Q * * H\);
\(=P\) ': apply P orP**H .

SID E (input)
\(=L\) ': apply \(Q, Q^{* *} H, P\) orP**H from the Left;
\(=R\) ': apply \(Q, Q{ }^{* *} H, P\) or \({ }^{* *}{ }^{*} H\) from the \(R\) ight.

TRANS (input)
\(=\mathrm{N}\) ': N o transpose, apply Q orP;
\(=C\) ': C onjugate transpose, apply \(\mathrm{Q} * * H\) or \({ }^{* * * H}\).

TRANS is defaulted to \(N\) 'forF \(95 \mathbb{N}\) TERFACE.

M (input) The num ber of row s of the m atrix \(\mathrm{C} . \mathrm{M}>=0\).
N (input) The num ber of colum ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\).

K (input) IfVECT = Q', the num ber of colum ns in the original \(m\) atrix reduced by CGEBRD. IfVECT \(=P\) ', the num ber of row \(S\) in the originalm atrix reduced by CGEBRD. \(K>=0\) 。

A (input) (LDA,min (nq,K)) ifVECT = Q' (LDA,nq) if
\(\mathrm{VECT}=\mathrm{P}\) 'The vectors w hich define the elem entary reflectors \(H\) (i) and G (i) , w hose products determ ine the \(m\) atrioes \(Q\) and \(P\), as retumed by CGEBRD.

LD A (input)
The leading dim ension of the array A. If VECT = Q', LDA \(>=\max (1, n q)\); if \(V E C T=P^{\prime}, L D A>=\) \(m \operatorname{ax}(1, m\) in \((n q, K))\).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectorH (i) orG (i) which determ ines Q or P , as retumed by CGEBRD in the array argum ent TAUQ orTAUP.

C (input/output)
On entry, the \(M\) boy \(-\mathrm{N} m\) atrix C . On exit, C is overw ritten by \(\mathrm{Q} * \mathrm{C}\) or \(\mathrm{Q} * * \mathrm{H} * \mathrm{C}\) or \(\mathrm{C} \mathrm{Q}^{* *} \mathrm{H}\) or \(\mathrm{C} \mathrm{Q}^{\mathrm{Q}}\) or \(P{ }^{*} \mathrm{C}\) or \(\mathrm{P}^{* *} \mathrm{H} * \mathrm{C}\) or C * or \(\mathrm{C}^{*} \mathrm{P} * * \mathrm{H}\).

LD C (input)
The leading dim ension of the aray C. LD C >= \(\max (1, M)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al

LW ORK .

LW ORK (input)
The dim ension of the array \(W\) ORK. IfSDE \(E L\) ',
LW ORK >= max (1,N); if \(S \mathbb{D} E=R '\) LW ORK >=
max ( \(1, \mathrm{M}\) ). Foroptim um perform ance LW ORK >= N *NB
if \(S \mathbb{D} E=L^{\prime}\), and LW \(O R K>=M * N B\) if \(S \mathbb{D} E=R\) ',
where NB is the optim alblocksize.
IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LW ORK is issued by X ERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cunm hr-overw rite the general com plex M -by -N m atrix C w ith \(S \mathbb{D E}=\mathrm{L}^{\prime} \mathrm{SD} \mathbb{D}=\mathrm{R}^{\prime}\) TRANS \(=\mathrm{N}^{\prime}\)

\section*{SYNOPSIS}

```

    W ORK,LW ORK,INFO)
    CHARACTER * 1SIDE,TRANS
COM PLEX A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
INTEGER M ,N,\mathbb{LO,\mathbb{H}I,LDA,LDC,LW ORK, INFO}

```

```

    LDC,W ORK,LW ORK,INFO)
    CHARACTER * 1SIDE,TRANS
COM PLEX A (LDA,*),TAU (*),C (LDC,*),W ORK (*)

```


\section*{F95 INTERFACE}

SU BROUTINE UNM HR (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}], \mathbb{Z}, \mathbb{H} I, A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [NFO])

CHARACTER ( \(\lfloor E N=1):: S \mathbb{D} E, T R A N S\)
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : :: A, C
\(\mathbb{N} T E G E R:: M, N, \mathbb{L O}, \mathbb{H} I, L D A, L D C, L W O R K, \mathbb{N} F O\)
SUBROUTINE UNMHR_64 (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}], \mathbb{L} O, \mathbb{H} I, A,[L D A], T A U\), C, [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SIDE,TRANS

COM PLEX,D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:): ::A, C
\(\mathbb{N}\) TEGER (8) ::M , N, \(\mathrm{LO}, \mathbb{H} \mathrm{I}, \mathrm{LD} A, L D C, L W O R K, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void cunm hr(charside, chartrans, intm, int \(n\), int ilo, int ini, com plex *a, int lda, com plex *tau, com plex *c, int ldc, int *info);
void cunm hr_64 (char side, char trans, long m, long n, long ilo, long ihi, com plex *a, long lda, com plex *tau, com plex *c, long ldc, long *info);

\section*{PURPOSE}
cunm hroverw rites the general com plex M -by-N m atrix C w ith
TRANS = C : \(\quad \mathrm{Q} * * \mathrm{H} * \mathrm{C} \quad \mathrm{C} * \mathrm{Q} * * \mathrm{H}\)
\(w\) here \(Q\) is a com plex unitary \(m\) atrix of ordernq, \(w\) th \(n q=m\) if \(S \mathbb{D} E=L\) 'and \(n q=n\) if \(S \mathbb{D} E=R\) '. \(Q\) is defined as the product of IH I-LIO elem entary reflectors, as retumed by CGEHRD :
\(\mathrm{Q}=\mathrm{H}\) (ilo) H (ilo+1) . . H (ihi-1).

\section*{ARGUMENTS}

STDE (input)
= L ': apply Q orQ **H from the Left;
\(=R\) ': apply Q or Q **H from the R ight.

TRANS (input)
= N': apply Q \(\mathrm{N} \circ\) otranspose)
\(=\mathrm{C}\) : apply \(\mathrm{Q} * * \mathrm{H}\) (C onjugate transpose)

TRANS is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFA CE .

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{C} . \mathrm{M}>=0\).
N (input) The num ber of Colm l ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\).

ШO (input)
HO and \(\mathbb{H}\) Im usthave the sam evalues as in the previous call of CGEHRD.Q is equal to the unit
\(m\) atrix except in the subm atrix

Q (ilo+1:ihi,ilo+1: : ihi). IfS \(\mathbb{D} E=4\) ', then \(1<=\)
\(\mathbb{H O}<=\mathbb{H} I<=M\), if \(M>0\), and \(\mathbb{H O}=1\) and \(\mathbb{H} I=\)
0 , if \(M=0\); if \(S \mathbb{D} E=R\) ', then \(1<=\mathbb{H O}<=\mathbb{H} I\)
\(<=\mathrm{N}\), if \(\mathrm{N}>0\), and \(\mathbb{H} \mathrm{O}=1\) and \(\mathbb{H} \mathrm{I}=0\), if \(\mathrm{N}=0\).

IH I (input)
See the description of IIO .

A (input) (LDA, \(M\) ) if \(S \mathbb{D} E=L^{\prime}(\mathbb{L D A}, N)\) if \(S \mathbb{D} E=R^{\prime} T\) The vectors which define the elem entary reflectors, as retumed by CGEHRD.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, M)\) ifSDE \(=L ; \operatorname{LDA}>=\max (1, N)\) if \(S \mathbb{D} E=\) R'.

TAU (input)
\((M-1)\) if \(S \mathbb{D} E=L^{\prime}(\mathbb{N}-1)\) if \(S \mathbb{D E}=R^{\prime} T A U(i)\) \(m\) ust contain the scalar factor of the elem entary reflectorH (i), as retumed by CGEHRD.

C (input/output)
On entry, the \(M\) by \(-N\) matrix \(C\). On exit, \(C\) is overw rilten by Q * C or \(\mathrm{Q} * \mathrm{H}_{\mathrm{H}}\) *C orC *Q **H orC *Q.

LD C (input)
The leading dim ension of the aray C. LD C >= max (1, M).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al
LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. IfSDE \(=\mathrm{L}\) ', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LW ORK \(>=\) \(\max (1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=L\) ', and LW ORK \(>=M * N B\) if \(S \mathbb{D} E=R\) ', w here NB is the optim alblocksize.

If LW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an ille-
galvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cunm 12 -overw rite the general com plex \(m\)-by-n matrix C with
\(Q\) * \(C\) if \(S \mathbb{D} E=\mathbb{L}\) 'and TRANS \(=N\) ', or \(Q * C\) if \(S \mathbb{D} E=\) L'and TRANS = C', or \(C * Q\) if \(S \mathbb{D} E=R\) 'and TRANS = \(N\) ', or \(C\) * \(Q\) 'if \(S \mathbb{D E}=R\) 'and TRANS \(=C^{\prime}\),

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE CUNM L2 (S\mathbb{DE,TRANS,M ,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{T},\textrm{T},\textrm{T}
\mathbb{NFO )}

```
CHARACTER * 1 SIDE,TRANS
COM PLEX A (LDA, *),TAU (*), C (LDC, \(\left.{ }^{\star}\right), \mathrm{W} O R K(\star)\)
\(\mathbb{N} T E G E R M, N, K, L D A, L D C, \mathbb{N F O}\)
SU BROUTINE CUNML2_64 (SDE,TRANS,M,N,K,A,LDA,TAU, C,LDC,W ORK,
    \(\mathbb{N} F O\) )

CHARACTER * 1 SDE E,TRANS
COM PLEX A (LDA, *),TAU (*), C (LDC, \(\left.{ }^{\star}\right), \mathrm{W} O R K(\star)\)
\(\mathbb{N}\) TEGER*8M,N,K,LDA,LDC, \(\mathbb{N}\) FO

\section*{F95 INTERFACE}

SU BROUTINE UNM L2 (SDE,TRANS, M ], \(\mathbb{N}],[K], A,[L D A], T A U, C,[L D C]\), [W ORK], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::SIDE,TRANS
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:): ::A, C
\(\mathbb{N}\) TEGER :: M , N, K, LDA,LDC, \(\mathbb{N} F O\)
SU BROUT \(\mathbb{N} E\) UNM L2_64 (S \(\mathbb{D E}, T R A N S, \mathbb{M}], \mathbb{N}], \mathbb{K}], A,[L D A], T A U, C\),
[LDC], [WORK], [ \(\mathbb{N F O}])\)

CHARACTER ( \(\amalg E N=1\) ) : : SDE,TRANS
COM PLEX ,D \(\mathbb{I M} E N S I O N\) (:) ::TAU,W ORK
COM PLEX , D \(\mathbb{M}\) ENSION (: : : : : : A, C
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LD} \mathrm{A}, \mathrm{LDC}, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void cunm 12 (char side, chartrans, intm, intn, intk, com plex *a, int lda, com plex *tau, com plex *c, int ldc, int*info);
void cunm 12_64 (charside, chartrans, long m, long n, long k, com plex *a, long lda, com plex *tau, com plex *c, long ldc, long *info);

\section*{PURPOSE}
cunm 12 overw rites the general com plex \(m\) boy-n matrix \(C\) w ith
\(w\) here \(Q\) is a com plex untary \(m\) atrix defined as the product ofk elem entary reflectors
\[
\mathrm{Q}=\mathrm{H}(\mathrm{k})^{\prime} \ldots \mathrm{H}(2)^{\prime} \mathrm{H}(1)^{\prime}
\]
as retumed by CGELQF.Q is of orderm if \(S \mathbb{D} E=\mathbb{L}\) 'and of ordern if \(S \mathbb{D} E=R\).

\section*{ARGUMENTS}

SIDE (input)
= L': apply Q orQ 'from the Left
\(=R\) ': apply Q orQ 'from the R ight

TRANS (input)
\(=N\) ': apply Q (N o transpose)
= C ': apply Q ' (C onjugate transpose)

M (input) The num ber of row s of the \(m\) atrix \(C . M>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\).

K (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q\). IfSID \(E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0\);
ifS \(\mathbb{D} E=R{ }^{\prime}, N>=K>=0\).

A (input) (LDA,M) if \(S \mathbb{D} E=L \prime\), (LDA,N) if \(S D E=R^{\prime}\) The i-th row \(m\) ust contain the vectorw hich defines the elem entary reflector \(H\) ( \(i\) ), for \(i=1,2, \ldots, k\), as retumed by CGELQF in the firstk row sof its array argum entA. A ism odified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A. LD A >= max \((1, K)\).

TAU (input)
TAU (i) must contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CG ELQ F.

C (input/output)
On entry, them -by \(n \mathrm{~m}\) atrix C . On exit, C is overw rilten by Q * C or \(\mathrm{Q}{ }^{*} \mathrm{C}\) orC \({ }^{*} \mathrm{Q}\) 'orC \({ }^{*} \mathrm{Q}\).

LD C (input)
The leading dim ension of the aray C. LD C >= \(m a x(1, M)\).

W ORK (w orkspace)
\((\mathbb{N})\) if \(\left.S \mathbb{D} E=L^{\prime}, M\right)\) if \(S \mathbb{D} E=R^{\prime}\)
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
< 0 : if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cunm lq -overw rite the general com plex M -by-N m atrix C w ith \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R{ }^{\prime}\) TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CUNM LQ (SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
COM PLEX A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
\mathbb{NTEGER M,N,K,LDA,LDC,LW ORK,NNFO}
SUBROUT\mathbb{NE CUNMLQ_64 (S\mathbb{DE,TRANS,M ,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{T},\textrm{L},\textrm{L}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
COM PLEX A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
\mathbb{NTEGER*8M,N,K,LDA,LDC,LW ORK,INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNM LQ (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C,[L D C]\), [W ORK], [LW ORK ], [ \(\mathbb{N} F \mathrm{O}\) ])

CHARACTER (LEN=1) ::SDE,TRANS
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:): ::A, C
\(\mathbb{N}\) TEGER ::M ,N,K,LDA,LDC,LW ORK, \(\mathbb{N} F O\)
SU BROUTINE UNM LQ_64 (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}], \mathbb{K}], A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SDE,TRANS

COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::A,C
\(\mathbb{N}\) TEGER (8) :: M , N , K, LDA, LD C, LW ORK, \(\mathbb{N}\) FO

\section*{C INTERFACE}
\#include <sunperfh>
void cunm lq (char side, chartrans, intm, intn, int k, com plex *a, int lda, com plex *tau, com plex *c, int ldc, int*info);
void cunm lq_64 (char side, chartrans, long m, long n, long k, com plex *a, long lda, com plex *tau, com plex *c, long ldc, long *info);

\section*{PURPOSE}
cunm lq overw rites the general com plex M -by -N m atrix C w ith TRANS = C : \(\quad Q * * H * C \quad C * Q * *\)
where \(Q\) is a com plex unitary \(m\) atrix defined as the product ofk elem entary reflectors
\[
Q=H(k)^{\prime} \ldots H(2)^{\prime} H(1) '
\]
as retumed by CGELQF.Q is oforderM ifSIDE \(=\mathbb{L}\) 'and of orderN if \(S \mathbb{D} E=R\).

\section*{ARGUMENTS}

STDE (input)
\(=\mathrm{L}\) ': apply Q orQ \({ }^{* *} \mathrm{H}\) from the Left;
\(=R\) ': apply Q orQ ** \(_{\mathrm{H}}\) from the Right .

TRANS (input)
= N ': N o transpose, apply Q ;
= C ': C onjugate transpose, apply Q **H .
TRANS is defaulted to \(N\) 'forF \(95 \mathbb{N}\) TERFACE .

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{C} . \mathrm{M}>=0\).
\(N\) (input) The num ber of colum ns of the m atrix C. \(\mathrm{N}>=0\).
\(K\) (input) The num ber of elem entary reflectors \(w\) hose product defines the \(m\) atrix \(Q\). If \(S \mathbb{D} E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0\); if \(S \mathbb{D} E=R \prime N>=K>=0\).
 \(i\)-th row m ustcontain the vectorw hich defines the elem entary reflector \(H\) ( \(i\) ), for \(i=1,2, \ldots, k\), as retumed by CGELQF in the firstk row sof its array argum entA. A ism odified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, K)\).

TAU (input)
TAU (i) m ustcontain the scalar factor of the ele\(m\) entary reflectorH (i), as retumed by CG ELQ F.

C (input/output)
On entry, the \(M-b y-N m\) atrix \(C\). On exit, \(C\) is overw ritten by Q * C or \(\mathrm{Q} * * \mathrm{H} * \mathrm{C}\) or C * \(\mathrm{Q} * \mathrm{H}\) or C * Q .

LD C (input)
The leading dim ension of the array \(C . \operatorname{LDC}>=\) \(\max (1, M)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the amay W ORK. IfSIDE = L', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LWORK >= \(\max (1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D E} L^{\prime}\) ', and LW ORK \(>=M * N B\) if \(S \mathbb{D} E=R\) ',
w here N B is the optim alblocksize.

If LW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cunm ql-overw rite the general com plex M -by -N m atrix C w ith \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime}\) TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CUNMQL (SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
COM PLEX A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
\mathbb{NTEGER M,N,K,LDA,LDC,LW ORK,NNFO}
SUBROUT\mathbb{NE CUNMQL_64 (S\mathbb{DE,TRANS,M ,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{T},\textrm{L},\textrm{L}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
COM PLEX A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
\mathbb{NTEGER*8M,N,K,LDA,LDC,LW ORK,INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNM QL (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C,[L D C]\), [W ORK], [LW ORK], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::SDE,TRANS
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:): ::A, C
\(\mathbb{N}\) TEGER :: M , N, K,LDA,LDC,LW ORK, \(\mathbb{N} F O\)
SU BROUTINE UNM QL_64 (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}], \mathbb{K}], A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SIDE,TRANS

COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::A,C
\(\mathbb{N}\) TEGER (8) :: M , N, K,LDA,LD C,LW ORK, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void cunm ql(char side, chartrans, intm, intn, intk, com plex *a, int lda, com plex *tau, com plex *c, int ldc, int*info);
void cunm ql_64 (char side, chartrans, long m, long n, long k, com plex *a, long lda, com plex *tau, com plex *c, long ldc, long *info);

\section*{PURPOSE}
cunm qloverw rites the general com plex M -by -N m atrix C w ith TRANS = C : \(\quad Q * * H * C \quad C * Q * *\)
where \(Q\) is a com plex unitary \(m\) atrix defined as the product ofk elem entary reflectors
\[
Q=H(k) \ldots H(2) H(1)
\]
as retumed by CGEQLF. Q is oforderM ifSIDE= L'and of orderN if \(S \mathbb{D} E=R\) '.

\section*{ARGUMENTS}

STDE (input)
= L ': apply Q orQ **H from the Left;
\(=R\) ': apply Q or \(\mathrm{Q}{ }^{* * H}\) from the R ight.

TRANS (input)
= N ': N o transpose, apply Q ;
= C ': Transpose, apply Q **H .
TRANS is defaulted to \(N\) 'forF \(95 \mathbb{N}\) TERFACE .

M (input) The num ber of row sof the \(m\) atrix \(\mathrm{C} . \mathrm{M}>=0\).

N (input) The num ber of collm ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\).
\(K\) (input) The num ber of elem entary reflectors \(w\) hose product defines the \(m\) atrix \(Q\). If \(S \mathbb{D} E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0\); if \(S \mathbb{D} E=R \prime, N>=K>=0\).

A (input) The i-th colum \(n\) must contain the vector which defines the elem entary reflector \(H\) (i), for \(i=\) \(1,2, \ldots, k\), as retumed by CGEQ LF in the last \(k\) colum ns of its array argum entA. A ism odified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A . If \(\mathrm{S} \mathbb{D} \mathrm{E}=\) L', LDA \(>=\max (1, M)\); if \(S D E=R '\) LDA \(>=\) \(\max (1, \mathbb{N})\).

TAU (input)
TAU (i) \(m\) ustcontain the scalar factorof the ele\(m\) entary reflectort (i), as retumed by CG EQ LF .

C (input/output)
On entry, the \(M\) boy \(-N m\) atrix C. On exit, \(C\) is overw ritten by \(\mathrm{Q} * \mathrm{C}\) or \(\mathrm{Q} * \mathrm{H}^{*} \mathrm{C}\) or \(\mathrm{C} \mathrm{Q}_{\mathrm{Q}}{ }^{* *} \mathrm{H}\) or C Q .

LD C (input)
The leading dim ension of the array C.LD C >= \(\max (1, \mathrm{M})\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK ( 1 ) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the aray \(W\) ORK. IfSDE \(=\mathrm{L}\) ', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LW ORK \(>=\) \(\max (1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=L\) ', and LW ORK \(>=M * N B\) if \(S \mathbb{D} E=R\) ', where NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cunm qr-overw rite the general com plex M -by -N m atrix C w th
\(S \mathbb{D E}=L^{\prime} S \mathbb{D} E=R^{\prime}\) TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUTINE CUNMQR(SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
COM PLEX A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
\mathbb{NTEGERM,N,K,LDA,LDC,LW ORK,NNFO}
SU BROUT\mathbb{NE CUNMQR_64 (SDDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
COM PLEX A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
\mathbb{NTEGER*8M,N,K,LDA,LDC,LW ORK, INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNM QR (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C,[L D C]\), \(\mathbb{W}\) ORK ], [LW ORK ], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SDE,TRANS
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER :: M , N, K,LDA,LDC,LW ORK, \(\mathbb{N} F O\)
SUBROUTINE UNM QR_64 (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SIDE,TRANS

COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,C
\(\mathbb{N}\) TEGER (8) :: M , N, K,LDA,LD C,LW ORK, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void cunm qr(char side, chartrans, intm, intn, intk, com plex *a, int lda, com plex *tau, com plex *c, int ldc, int*info);
void cunm qr_64 (char side, char trans, long m, long n, long k, com plex *a, long lda, com plex *tau, com plex *c, long ldc, long *info);

\section*{PURPOSE}
cunm qroverw rites the general com plex M -by-N m atrix C w ith TRANS = C : \(\quad \mathrm{Q} * * \mathrm{H} * \mathrm{C} \quad \mathrm{C} * \mathrm{Q} * * \mathrm{H}\)
where \(Q\) is a com plex unitary \(m\) atrix defined as the product ofk elem entary reflectors
\[
Q=H(1) H(2) \ldots H(k)
\]
as retumed by CGEQRF.Q is oforderM ifSIDE = L'and of order \(N\) if \(S \mathbb{D} E=R\).

\section*{ARGUMENTS}

STDE (input)
\(=\mathrm{L}\) ': apply Q orQ \(* * \mathrm{H}\) from the Left;
\(=R\) ': apply Q or \(\mathrm{Q}{ }^{* * H}\) from the Right .

TRANS (input)
= N ': N o transpose, apply Q ;
= C ': C onjugate transpose, apply \(Q\) **H .
TRANS is defaulted to \(N\) 'forF \(95 \mathbb{N}\) TERFACE .

M (input) The num ber of row sof the \(m\) atrix \(\mathrm{C} . \mathrm{M}>=0\).

N (input) The num ber of Colm ns of the \(m\) atrix \(\mathrm{C} . \mathrm{N}>=0\).
\(K\) (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q\). If \(S \mathbb{D} E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0\); if \(S \mathbb{D} E=R \prime, N>=K>=0\).

A (input) The i-th colum \(n\) must contain the vector which defines the elem entary reflector \(H\) (i), for \(i=\) \(1,2, \ldots, k\), as retumed by CGEQRF in the first \(k\) colum ns of its array argum entA. A ism odified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A . If \(\mathrm{S} \mathbb{D} \mathrm{E}=\) L', LDA >= \(\max (1, M)\); if \(S \mathbb{D E}=R^{\prime}, L D A>=\) \(\max (1, N)\).

TAU (input)
TAU (i) \(m\) ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CGEQRF.

C (input/output)
On entry, the \(M\) by -N matrix C. On exit, C is overw rilten by \(\mathrm{Q} * \mathrm{C}\) or \(\mathrm{Q} * * \mathrm{H} * \mathrm{C}\) or \(\mathrm{C} \mathrm{Q}_{\mathrm{Q}}{ }^{* *} \mathrm{H}\) orC \({ }^{\mathrm{Q}} \mathrm{Q}\).

LD C (input)
The leading dim ension of the aray C. LD C >= max (1,M).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. IfSDE \(=\mathrm{L}\) ', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LW ORK >= \(\max (1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=L '\), and \(L W\) ORK \(>=M * N B\) if \(S \mathbb{D} E=R\) ', where NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim al size of
the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LW ORK is issued by XERBLA.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cunm r2 -overw rite the general com plex \(m\)-by-n matrix \(C\) w ith
\(Q * C\) if \(S \mathbb{D} E=L\) 'and TRANS \(=N\) ', or \(Q * C\) if \(S \mathbb{D} E=\) L'and TRANS = C', or \(C * Q\) if \(S \mathbb{D} E=R\) 'and TRANS = \(N\) ', or \(C\) * \(Q\) 'if \(S \mathbb{D E}=R\) 'and TRAN \(S=C^{\prime}\),

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE CUNMR2(S\mathbb{DE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{M},\textrm{L},\textrm{L}
\mathbb{NFO )}
CHARACTER * 1SIDE,TRANS
COM PLEX A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
INTEGERM,N,K,LDA,LDC,INFO
SUBROUTINE CUNMR2_64(SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,
\mathbb{NFO )}

```
CHARACTER * 1 SIDE,TRANS

\(\mathbb{N}\) TEGER*8 M , N , K , LDA , LD C , \(\mathbb{N}\) FO

\section*{F95 INTERFACE}

SU BROUTINE UNMR2 (SDE,TRANS, M ], \(\mathbb{N}],[K], A,[L D A], T A U, C,[L D C]\), [W ORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::SIDE,TRANS
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : : A, C
\(\mathbb{N}\) TEGER :: M , N, K, LDA, LD C, \(\mathbb{N} F O\)
SUBROUT \(\mathbb{N} E\) UNM R2_64 (S \(\mathbb{D E} E, T R A N S, \mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C\),
[LDC], [WORK],[ \(\mathbb{N} F O]\) )

CHARACTER (〔EN=1) ::SDE,TRANS
COM PLEX ,D \(\mathbb{I M} E N S I O N\) (:) ::TAU,W ORK
COM PLEX , D \(\mathbb{M}\) ENSION (: : : : : : A, C
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LD} \mathrm{A}, \mathrm{LDC}, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void cunm 22 (charside, chartrans, intm, intn, intk, com plex *a, int lda, com plex *tau, com plex *c, int ldc, int*info);
void cunm 22 _64 (char side, chartrans, long m, long n, long k , com plex *a, long lda, com plex *tau, com plex * \({ }^{\prime}\), long ldc, long *info);

\section*{PURPOSE}
cunm \(n 2\) overw rites the general com plex \(m\)-by-n m atrix \(C\) w ith
\(w\) here \(Q\) is a com plex untary \(m\) atrix defined as the product ofk elem entary reflectors
\[
\mathrm{Q}=\mathrm{H}(1)^{\prime} \mathrm{H}(2)^{\prime} \ldots \mathrm{H}(\mathrm{k})^{\prime}
\]
as retumed by CGERQF.Q is of orderm ifSIDE \(=\mathrm{L}\) 'and of ordern if \(S \mathbb{D} E=R\).

\section*{ARGUMENTS}

SIDE (input)
= L': apply Q orQ 'from the Left
\(=R\) ': apply Q orQ 'from the R ight

TRANS (input)
\(=\mathrm{N}^{\prime}:\) apply \(\mathrm{Q} \quad(\mathbb{N}\) o transpose)
= C ': apply Q ' (C onjugate transpose)

M (input) The num ber of row s of the \(m\) atrix \(C . M>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\).

K (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q\). IfSID \(E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0\);
ifS \(\mathbb{D} E=R{ }^{\prime}, N>=K>=0\).

A (input) ( \(L D A, M\) ) if \(S D E=L \prime\) ( \(L D A, N\) ) if \(S D E=R^{\prime} T\) he i-th row \(m\) ustcontain the vectorw hich defines the elem entary reflector H ( i ), for \(i=1,2, \ldots, \mathrm{k}\), as retumed by CGERQF in the lastk row s of its anay argum entA. A is modified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A. LD A >= max \((1, K)\).

TAU (input)
TAU (i) must contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CG ERQF.

C (input/output)
On entry, them -by \(n \mathrm{~m}\) atrix C . On exit, C is overw rilten by Q * C or \(\mathrm{Q}{ }^{*} \mathrm{C}\) orC \({ }^{*} \mathrm{Q}\) 'orC \({ }^{*} \mathrm{Q}\).

LD C (input)
The leading dim ension of the array C. LD C >= max (1, M).

W ORK (w orkspace)
\((\mathbb{N})\) if \(\left.S \mathbb{D} E=L^{\prime}, M\right)\) if \(S \mathbb{D} E=R^{\prime}\)
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cunm rq-overw rite the general com plex M -by -N m atrix C w th \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime}\) TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUTINE CUNM RQ (SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
COM PLEX A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
\mathbb{NTEGERM,N,K,LDA,LDC,LW ORK,NNFO}
SU BROUT\mathbb{NE CUNM RQ_64 (SDDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
COM PLEX A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
\mathbb{NTEGER*8M,N,K,LDA,LDC,LW ORK, INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNMRQ (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C,[L D C]\), \(\mathbb{W}\) ORK ], [LW ORK ], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SDE,TRANS
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER :: M , N, K,LDA,LDC,LW ORK, \(\mathbb{N} F O\)
SUBROUTINE UNMRQ_64 (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SIDE,TRANS

COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A,C
\(\mathbb{N}\) TEGER (8) :: M , N, K,LDA,LD C,LW ORK, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void cunm rq (char side, chartrans, intm, intn, intk, com plex *a, int lda, com plex *tau, com plex *c, int ldc, int*info);
void cunm rq_64 (char side, chartrans, long m, long n, long k, com plex *a, long lda, com plex *tau, com plex *c, long ldc, long *info);

\section*{PURPOSE}
cunm rq overw rites the general com plex M -by -N m atrix C w th
TRANS = C : \(\quad \mathrm{Q} * * \mathrm{H} * \mathrm{C} \quad \mathrm{C} * \mathrm{Q} * * \mathrm{H}\)
where \(Q\) is a com plex unitary \(m\) atrix defined as the product ofk elem entary reflectors
\[
\mathrm{Q}=\mathrm{H}(1) \text { 'H(2)'...H(k)' }
\]
as retumed by CGERQF.Q is oforderM ifSIDE = L'and of orderN if \(S \mathbb{D} E=R\).

\section*{ARGUMENTS}

STDE (input)
\(=\mathrm{L}\) ': apply Q orQ \({ }^{* *} \mathrm{H}\) from the Left;
\(=R\) ': apply Q orQ ** \(_{\mathrm{H}}\) from the Right .

TRANS (input)
= N ': N o transpose, apply Q ;
= C ': Transpose, apply Q **H .
TRANS is defaulted to \(N\) 'forF \(95 \mathbb{N}\) TERFACE .

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{C} . \mathrm{M}>=0\).
\(N\) (input) The num ber of colum ns of the m atrix C. \(\mathrm{N}>=0\).
\(K\) (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q\). If \(S \mathbb{D} E=L ', M \quad=K>=0\); if \(S \mathbb{D} E=R \prime, N>=K>=0\).

A (input) ( \(L D A, M\) ) if \(S \mathbb{D} E=\mathbb{L}\) ', (LDA, \(N\) ) if \(S \mathbb{D} E=R^{\prime}\) The \(i\)-th row m ustcontain the vectorw hich defines the elem entary reflector \(H\) ( \(i\) ), for \(i=1,2, \ldots, k\), as retumed by C GERQF in the lastk row s of its array argum entA. A is modified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, K)\).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectorH (i), as retumed by CGERQF.

C (input/output)
On entry, the \(M-b y-N m\) atrix \(C\). On exit, \(C\) is overw rilten by Q * C or \(\mathrm{Q} * \mathrm{H} * \mathrm{C}\) or C * \(\mathrm{Q} * \mathrm{H}\) or C * Q .

LD C (input)
The leading dim ension of the array \(C . \operatorname{LDC}>=\) \(\max (1, M)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the amay W ORK. IfSIDE = L', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LWORK >= \(\max (1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=L\) ', and LW ORK \(>=M * N B\) if \(S \mathbb{D} E=R\) ', w here N B is the optim alblocksize.

If LW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK anay, retums this value as the first
entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}

Cunm \(r z\)-overw rite the general com plex \(M\)-by \(-\mathrm{N} m\) atrix \(C\) w ith \(S \mathbb{D} E=\mathbb{L} ' S \mathbb{D} E=R^{\prime}\) TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CUNMRZ (S\mathbb{DE,TRANS,M ,N,K,L,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{M,},\textrm{L},\textrm{L}
LW ORK,\mathbb{NFO)}

```
CHARACTER * 1 SDE E,TRANS
COM PLEX A (LDA, \()\),TAU (*), C (LDC , \(\left.{ }^{*}\right), \mathrm{W} O R K(\star)\)
\(\mathbb{N}\) TEGERM,N,K,L,LDA,LDC,LW ORK, \(\mathbb{N} F O\)
SU BROUTINECUNMRZ_64 (SIDE,TRANS,M,N,K,L,A,LDA,TAU,C,LDC,
        WORK,LW ORK, \(\mathbb{N} F O\) )

CHARACTER * 1 SIDE ,TRANS

\(\mathbb{N}\) TEGER*8M,N,K,L,LDA,LDC,LWORK, \(\mathbb{N} F O\)

\section*{F95 INTERFACE}

SU BROUTINE CUNMRZ (SDE,TRANS,M,N,K,L,A,LDA,TAU,C,LDC, W ORK,LW ORK, \(\mathbb{N} F\) ) )

CHARACTER (LEN=1) ::SIDE,TRANS
COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER ::M,N,K,L,LDA,LDC,LWORK, \(\mathbb{N}\) FO
SU BROUTINE CUNMRZ_64 (SDE, TRANS,M,N,K,L,A,LDA,TAU,C,LDC, W ORK,LW ORK, \(\mathbb{N} F O\) )

CHARACTER ( \(几 E N=1\) ) : : SDE E,TRANS
COM PLEX , D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX, D \(\mathbb{I M}\) ENSION (: : : : : A , C
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{L}, \mathrm{LDA}, \mathrm{LD} \mathrm{C}, \mathrm{LW} O R K, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void cunm rz (char side, chartrans, intm, intn, intk, int 1, com plex *a, int lda, com plex *tau, com plex \({ }^{*}{ }_{\mathrm{C}}\), int \(1 d \mathrm{dc}\), int *info);
void cunm rz_64 (charside, chartrans, long m, long n, long k, long l, complex *a, long lda, com plex *tau, com plex * c, long ldc, long *info);

\section*{PURPOSE}
cunm rz overw rites the general com plex \(M\) boy \(\mathrm{N} m\) atrix C w ith TRANS = C : \(\quad Q^{* *} H * C \quad C * Q^{* *} H\)
\(w\) here \(Q\) is a com plex untary \(m\) atrix defined as the product ofk elem entary reflectors
\[
Q=H(1) H(2) \ldots H(k)
\]
as retumed by CTZRZF.Q is of orderM ifSDE \(=\mathbb{L}\) 'and of order \(N\) if \(S \mathbb{D} E=R \prime\).

\section*{ARGUMENTS}

SIDE (input)
\(=\mathbb{L}\) ': apply Q or \(\mathrm{Q} * * \mathrm{H}\) from the Left;
\(=R\) ': apply \(Q\) orQ **H from the \(R\) ight.

TRANS (input)
\(=\mathrm{N}\) ': N o transpose, apply \(Q\);
= C ': C onjugate transpose, apply \(Q^{* *} \mathrm{H}\).

M (input) The num ber of row s of the \(m\) atrix C. \(\mathrm{M}>=0\).

N (input) The num ber of colum ns of the \(m\) atrix \(\mathrm{C} . \mathrm{N}>=0\).

K (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q\). IfSID \(E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0\);
ifS \(\mathbb{D} E=R{ }^{\prime}, N>=K>=0\).

L (input) The num ber of colum \(n s\) of the \(m\) atrix \(A\) containing
the \(m\) eaningfulpart of the \(H\) ouseholder reflectors.
If \(S \mathbb{D} E=L \prime, M>=L>=0\), if \(S \mathbb{D} E=R \prime N>=L\)
\(>=0\) 。

A (input) (LDA, M) if \(S \mathbb{D} E=L \prime\) ( \(L D A, N\) ) if \(S \mathbb{D} E=R^{\prime}\) The \(i\)-th row \(m\) ustcontain the vectorw hich defines the
elem entary reflector \(H\) (i), for \(i=1,2, \ldots, k\), as
retumed by CTZRZF in the lastk row s of its array argum entA. A is modified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array \(\mathrm{A} . \mathrm{LDA}>=\) \(\max (1, K)\).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectorH (i), as retumed by CTZRZF.

C (input/output)
On entry, the \(M-b y-N m\) atrix \(C\). On exit, \(C\) is overw ritten by Q * C or \(\mathrm{Q} * * \mathrm{H} * \mathrm{C}\) or \(\mathrm{C} \mathrm{Q}^{*}{ }^{*} \mathrm{H}\) or C * Q .

LD C (input)
The leading dim ension of the aray C.LDC >= \(\max (1, M)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. If \(S \mathbb{D} E=L\) ',
LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R \prime\) LW ORK >= \(\max (1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=L '\), and LW ORK \(>=M * N B\) if \(S D E=R \prime\), where NB is the optim alblock.size.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

B ased on contributions by
A. Petitet, C om puter Science D ept., U niv . of Tenn., K noxville, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cunm tr-overw rite the general com plex M -by -N m atrix C w ith \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime} T R A N S=N^{\prime}\)

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE CUNM TR (SDE,UPLO,TRANS,M,N,A,LDA,TAU,C,LDC,W ORK,}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,UPLO,TRANS
COM PLEX A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
INTEGERM,N,LDA,LDC,LW ORK,INFO
SU BROUT\mathbb{NE CUNM TR_64 (S\mathbb{DE,UPLO,TRANS,M ,N,A,LDA,TAU,C,LDC,}}\mathbf{T},\textrm{T},\textrm{T}
W ORK,LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,UPLO,TRANS
COM PLEX A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
INTEGER*8M,N,LDA,LDC,LW ORK,INFO

```

\section*{F95 INTERFACE}

SU BROUTINE UNM TR (SDE, UPLO, [TRANS], \(\mathbb{M}], \mathbb{N}], A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1)::SDE,UPLO,TRANS
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N} T E G E R:: M, N, L D A, L D C, L W O R K, \mathbb{N F O}\)
SU BROUTINE UNM TR_64 (SDE, UPLO, [TRANS], \(\mathbb{M}], \mathbb{N}], A,[L D A], T A U, C\), [LD C ], [W ORK], [LW ORK ], [NFO ])

CHARACTER (LEN=1)::SIDE,UPLO,TRANS

COM PLEX,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX,D \(\mathbb{I M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER (8) ::M , N,LDA,LDC,LW ORK, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void cunm tr (char side, charuplo, chartrans, intm, int n, com plex *a, int lda, com plex *tau, com plex \({ }^{*}\) c, int ldc, int*info);
void cunm tr_64 (char side, charuplo, char trans, long m, long n, com plex *a, long lda, com plex *tau, com plex *c, long ldc, long *info);

\section*{PURPOSE}
cunm troverw rites the general com plex M -by -N m atrix C w ith TRANS = C : \(\quad Q * * H * C \quad C * Q * *\)
where \(Q\) is a com plex unitary \(m\) atrix of ordernq, \(w\) th \(n q=m\) if \(S \mathbb{D} E=\Sigma\) 'and \(n q=n\) if \(S \mathbb{D} E=R '^{\prime} . Q\) is defined as the product of nq-1 elem entary reflectors, as retumed by CHETRD :
if \(\mathrm{UPLO}=\mathrm{U}\) ', \(\mathrm{Q}=\mathrm{H}(\mathrm{nq}-1) \ldots\). \(\mathrm{H}(2) \mathrm{H}(1)\);
if UPLO = L', Q = H (1) H (2) ...H (nq-1).

\section*{ARGUMENTS}

SIDE (input)
\(=\mathrm{L}\) ': apply Q orQ \({ }^{* *} \mathrm{H}\) from the Left;
= R ': apply Q orQ **H from the Right.
UPLO (input)
= U :: U ppertriangle of A contains elem entary
reflectors from CHETRD; = L':Low er triangle of A
contains elem entary reflectors from CHETRD.

TRANS (input)
\(=\mathrm{N}\) ': N o transpose, apply Q;
= C ': C onjugate transpose, apply Q **H .
TRANS is defaulted to \(N\) 'forF \(95 \mathbb{N}\) TERFACE .

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{C} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\).
A (input) ( \(L D A, M\) ) if \(S \mathbb{D} E=L^{\prime}(L D A, N)\) if \(S \mathbb{D} E=R^{\prime}\) The vectors w hich define the elem entary reflectors, as retumed by CHETRD.

LD A (input)
The leading dim ension of the anay A. LDA >= \(\max (1, M)\) ifS \(\mathbb{D} E=L ; L D A>=m a x(1, N)\) if \(S \mathbb{D} E=\) R.

TAU (input)
\((M-1)\) ifSTDE \(=\mathbb{L}(N-1)\) if \(S \mathbb{D E}=R^{\prime} T A U\) (i) \(m\) ust contain the scalar factor of the elem entary reflectorH (i), as retumed by CHETRD .

C (input/output)
On entry, the \(M\) boy -N m atrix C . On exit, C is overw ritten by \(\mathrm{Q} * \mathrm{C}\) or \(\mathrm{Q} * * \mathrm{H} * \mathrm{C}\) or \(\mathrm{C} * \mathrm{Q} *{ }^{*} \mathrm{H}\) or \(\mathrm{C} * \mathrm{Q}\).

LD C (input)
The leading dim ension of the array C. LD C >= \(m\) ax (1, M).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the anay \(W\) ORK. If \(S \mathbb{D} E=L\) ', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LW ORK >= \(m a x(1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=L^{\prime}\), and LW ORK \(>=M\) *NB ifSDE \(=R\) ', where NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cupgtr-generate a complex unitary matrix Q which is defined as the product ofn-1 elem entary reflectors \(H\) (i) of ordern, as retumed by C H PTRD using packed storage

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE CUPGTR (UPLO,N,AP,TAU,Q,LDQ,W ORK,INFO)}
CHARACTER * 1 UPLO
COM PLEX AP (*),TAU (*),Q (LDQ,*),W ORK (*)
INTEGERN,LDQ,INFO
SUBROUT\mathbb{NE CUPGTR_64(UPLO,N,AP,TAU,Q,LDQ,W ORK,INFO)}
CHARACTER * 1 UPLO
COM PLEX AP (*),TAU (*),Q (LDQ,*),W ORK (*)
INTEGER*8N,LDQ,INFO

```

\section*{F95 INTERFACE}

SU BROUTINE UPGTR (UPLO, \(\mathbb{N}], A P, T A U, Q,[L D Q],[W O R K],[\mathbb{N} F O])\)
CHARACTER (LEN=1) ::UPLO
COM PLEX,D IM ENSION (:) ::AP,TAU,W ORK
COM PLEX,D IM ENSION (:,:) :: Q
\(\mathbb{N} T E G E R:: N, L D Q, \mathbb{N} F O\)

SU BROUTINE UPG TR_64 (UPLO, \(\mathbb{N}], A P, T A U, Q,[L D Q],[\mathbb{N} O R K],[\mathbb{N} F O])\)
CHARACTER (LEN=1)::UPLO
COM PLEX,D \(\mathbb{I M} E N S I O N(:):: A P, T A U, W\) ORK
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : : Q
\(\mathbb{N}\) TEGER ( 8 ) :: \(\mathrm{N}, \mathrm{LD} Q, \mathbb{N}\) FO

\section*{C INTERFACE}
\#include <sunperfh>
void cupgtr(charuplo, intn, com plex *ap, complex *tau, com plex *q, int ldq, int *info);
void cupgtr_64 (charuplo, long n, com plex *ap, com plex *tau, com plex *q, long ldq, long *info);

\section*{PURPOSE}
cupgtrgenerates a com plex unitary \(m\) atrix \(Q\) which is defined as the productofn-1 elem entary reflectors \(H\) (i) ofordern, as retumed by CH PTRD using packed storage: if \(U P L O=U ', Q=H(n-1) \ldots H(2) H(1)\), if \(U P L O=L^{\prime}, Q=H(1) H(2) \ldots H(n-1)\).

\section*{ARGUMENTS}

UPLO (input)
= U ':U ppertriangular packed storage used in previous call to CHPTRD; = L': Low er triangular packed storage used in previous call to CH PTRD .

N (input) The order of the matrix \(\mathrm{Q} . \mathrm{N}>=0\).
AP (input)
The vectors w hich define the elem entary reflectors, as retumed by CH PTRD.

TAU (input)
TAU (i) \(m\) ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CH PTRD .

Q (output)
The N -by -N unitary m atrix Q .
LD Q (input)
The leading dim ension of the array \(\mathrm{Q} . \mathrm{LD} \mathrm{Q}>=\) \(\max (1, N)\).

W ORK (w orkspace)
dim ension \((\mathbb{N}-1)\)
\(\mathbb{N}\) FO (output)
= 0: successfulexit
< 0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cupm tr-overw rite the general com plex M -by -N m atrix C w ith \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime}\) TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SU BROUTINE CUPM TR (S\mathbb{DE,UPLO,TRANS,M,N,AP,TAU,C,LDC,W ORK,}
\mathbb{NFO)}
CHARACTER * 1SIDE,UPLO,TRANS
COM PLEX AP (*),TAU (*),C (LDC ,*),W ORK (*)
\mathbb{NTEGERM,N,LDC,INFO}
SU BROUTINE CUPM TR_64(SDD E,UPLO,TRANS,M,N,AP,TAU,C,LDC,W ORK,
\mathbb{NFO)}
CHARACTER * 1SIDE,UPLO,TRANS
COM PLEX AP (*),TAU (*),C (LDC ,*),W ORK (*)
\mathbb{NTEGER*8M,N,LDC,INFO}

```

\section*{F95 INTERFACE}
```

SU BROUTINE UPM TR (SDE, UPLO, [TRANS], $\mathbb{M}], \mathbb{N}], A P, T A U, C,[L D C]$, [ W ORK], [ $\mathbb{N} F \mathrm{O}$ ])
CHARACTER (LEN=1)::SDE,UPLO,TRANS
COMPLEX,DIM ENSION (:) ::AP,TAU,W ORK
COM PLEX,D $\mathbb{M}$ ENSION (:,:) :: C
$\mathbb{N}$ TEGER ::M,N,LDC, $\mathbb{N} F O$
SU BROUTINE UPM TR_64 (SDE, UPLO, [TRANS], $\mathbb{M}], \mathbb{N}], A P, T A U, C,[L D C]$, [ $\mathrm{W} O \mathrm{RK}$ ], [ $\mathbb{N} F \mathrm{FO}$ )

```

CHARACTER (LEN=1) ::SDE,UPLO,TRANS

COMPLEX,D \(\mathbb{M}\) ENSION (:) ::AP,TAU,W ORK
COM PLEX,D \(\mathbb{M}\) ENSION (:,:) ::C
\(\mathbb{N}\) TEGER (8) ::M,N,LDC, \(\mathbb{N}\) FO

\section*{C INTERFACE}
\#include <sunperfh>
void cupm tr(char side, charuplo, chartrans, intm , int n, complex *ap, complex *tau, com plex *c, int ldc, int*info);
void cupm tr_64 (char side, charuplo, char trans, long m, long n, com plex *ap, com plex *tau, com plex *c, long ldc, long *info);

\section*{PURPOSE}
cupm troverw rites the general com plex M -by -N m atrix C w ith TRANS = C : \(\quad Q * * H * C \quad C * Q * *\)
where \(Q\) is a com plex unitary \(m\) atrix of ordernq, \(w\) th \(n q=m\) if \(S \mathbb{D} E=L\) 'and \(n q=n\) if \(S \mathbb{D} E=R\) '. \(Q\) is defined as the product of nq-1 elem entary reflectors, as retumed by CH PTRD using packed storage:
if \(\mathrm{UPLO}=\mathrm{U}\) ', \(\mathrm{Q}=\mathrm{H}(\mathrm{nq}-1) \ldots\). \(\mathrm{H}(2) \mathrm{H}(1)\);
if UPLO = L', Q = H (1) H (2) ...H (nq-1).

\section*{ARGUMENTS}

STDE (input)
\(=\mathrm{L}\) ': apply Q orQ \({ }^{* *} \mathrm{H}_{\mathrm{H}}\) from the Left;
= R ': apply Q orQ **H from the Right.
UPLO (input)
= U ':U ppertriangular packed storage used in previous call to CHPTRD ; = L ':Low er triangular packed storage used in previous call to CH PTRD .

TRANS (input)
= N ': N ○ transpose, apply Q ;
= C ': C onjugate transpose, apply Q **H .

TRANS is defaulted to \(N\) 'forF \(95 \mathbb{N}\) TERFACE .

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{C} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\).
AP (input)
\((M *(M+1) / 2)\) if \(S D E=L(\mathbb{N} *(N+1) / 2)\) if \(S D E=\)
\(R\) ' The vectors which define the elem entary
reflectors, as retumed by CHPTRD. AP ism odified
by the routine but restored on exit.
TAU (input)
or \((\mathbb{N}-1)\) if \(S \mathbb{D} E=R^{\prime} T A U(i)\) must contain the scalar factor of the elem entary reflectorH (i), as retumed by CHPTRD.
C (input/output)
On entry, the M by -N matrix C. On exit, C is overw rilten by Q * C or \(\mathrm{Q}{ }^{* *} \mathrm{H}\) * C or C \(\mathrm{Q} * * \mathrm{H}\) orC \({ }^{*} \mathrm{Q}\).

LD C (input)
The leading dim ension of the array C. LD C >= \(\max (1, \mathrm{M})\).

W ORK (w orkspace)
\(\mathbb{N})\) if \(S \mathbb{D} E=L^{\prime}(M)\) if \(S \mathbb{D} E=R^{\prime}\)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an ille-
galvalue

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
cvbrm m -variable block sparse row form atm atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NECVBRMM(TRANSA,MB,N,KB,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,BINDX,RPNTR,CPNTR,BPNTRB,BPNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,MB,N,KB,DESCRA (5),LDB,LDC,LW ORK}
INTEGER INDX(*),B\mathbb{NDX(*),RPNTR M B+1),CPNTR(KB+1),}
* BPNTRB M B),BPNTRE M B)
COM PLEX ALPHA,BETA
COMPLEX VAL (*),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NECVBRMM_64(TRANSA,M B,N,KB,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,B}\mathbb{N}DX,RPNTR,CPNTR,BPNTRB,BPNTRE,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,MB,N,KB,DESCRA (5),LDB,LDC,LW ORK}

```

```

* BPNTRB MB),BPNTREMB)
COM PLEX ALPHA,BETA
COMPLEX VAL (*),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}

SUBROUTINEVBRMM (TRANSA, MB, \(\mathbb{N}], K B, A L P H A, D E S C R A\), * VAL, \(\mathbb{N} D \mathrm{X}, \mathrm{B} \mathbb{N} D \mathrm{X}, \mathrm{RPNTR}, \mathrm{CPN} T \mathrm{R}, \mathrm{BPNTRB}, \mathrm{BPNTRE}\), * B, [LDB],BETA, C,[LDC], [W ORK],[LW ORK])
\(\mathbb{N} T E G E R\) TRANSA,MB,KB
\(\mathbb{N}\) TEGER,D \(\mathbb{I M}\) ENSION (:) :: DESCRA, \(\mathbb{N} D \mathrm{X}, \mathrm{B} \mathbb{N} D \mathrm{X}\)
\(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION (:) :: RPNTR,CPNTR,BPNTRB,BPNTRE
COMPLEX ALPHA,BETA
COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::VAL

SUBROUTINE VBRM M _64 (TRANSA, M B, \(\mathbb{N}], K B, A L P H A, D E S C R A\), * \(\quad V A L, \mathbb{N} D X, B \mathbb{N D X}, R P N T R, C P N T R, B P N T R B, B P N T R E\), * \(B,[\) LD B \(], B E T A, C,[L D C],[W O R K],[L W O R K])\)

INTEGER*8 TRANSA, MB,KB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: D E S C R A, \mathbb{N} D X, B \mathbb{N} D X\)
\(\mathbb{I N T E G E R * 8 , D \mathbb { M } E N S I O N ( : ) : : R P N T R , C P N T R , B P N T R B , B P N T R E ~}\)
COMPLEX ALPHA,BETA
COM PLEX, D \(\mathbb{M}\) ENSION (:) ::VAL
COM PLEX , D \(\mathbb{M}\) ENSION (: :) :: B , C

\section*{DESCRIPTION}
C <-alpha op (A ) B + beta C
where A LPHA and BETA are scalar, \(C\) and \(B\) are \(m\) atriges, \(A\) is a \(m\) atrix represented in variable block sparse row form at and op (A ) is one of
\(o p(A)=A\) or \(o p(A)=A^{\prime}\) or op \((A)=\operatorname{conjg}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalent to 1 if the \(m\) atrix is real.

M B \(\quad\) Num ber ofblock row s in m atrix A
\(N \quad N\) um ber of colum ns in \(m\) atrix \(C\)

K B \(\quad\) Num ber ofblock colum ns in m atrix A

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger aray
D ESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm itian ( \(A=\) CON JG (A ) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(A=-C O N J(A)\) )

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 :non-unit
1 : unit
DESCRA (4) A ray base \(N\) OT \(\mathbb{M}\) PLEM ENTED)
0 :C C++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL ( scalar array of length NN Z consisting of the block entries of A w here each block entry is a dense rectangularm atrix stored colum \(n\) by colum \(n\).
NN Z is the total num berof pointentries in all nonzero block entries of am atrix A.
\(\mathbb{N} D \mathrm{X}\) () integer anray of length BNN Z +1 where BNNZ is the num berof block entries of a m atrix A such that the I-th elem entof \(\mathbb{N}\) D X [] points to the location in VAL of the \((1,1)\) elem ent of the I-th block entry.

B IND X () integer array of length BNN Z consisting of the block colum \(n\) indiges of the block entries of \(A\) where BNNZ is the num berblock entries of a m atrix A.

RPN TR 0) integer amay of length M B+1 such thatRPN TR (I) RPN TR (1) +1 is the row index of the firstpoint row in the \(I\)-th block
row .
RPN TR \(M B+1\) ) is set to \(M+\operatorname{RPN} \operatorname{TR}(1)\) where \(M\) is the num ber of row \(s\) in \(m\) atrix \(A\).
Thus, the num berof point row sin the I-th block row is RPNTR (I+1)RPNTR (I).

CPN TR 0 integer array of length \(K B+1\) such that CPN TR (J)-CPN TR (1)+1 is the colum \(n\) index of the firstpoint colum \(n\) in the \(J\) th block colum n. CPN TR ( \(K B+1\) ) is set to \(K+C P N T R(1)\) where \(K\) is the num ber of \(c o l u m n s\) in \(m\) atrix \(A\). Thus, the num ber of point \(\infty 0\) lum ns in the \(J\) th block colum n is CPNTR ( \(\mathrm{J}+1\) )-CPNTR ( \(J\) ).

BPNTRB () integer array of length \(M B\) such thatBPNTRB (I) BPNTRB (1) +1 points to location in B IND X of the first.block entry of the I-th block row of A.

BPNTRE 0 integer anay of length \(M B\) such that BPN TRE (I) BPNTRB (1) points to location in B \(\mathbb{N}\) D X of the lastblock entry of the I-th block row of A.

B 0 rectangular array w ith firstdin ension LD B .

LD B leading dim ension ofB

BETA Scalarparam eter
C 0 rectangular anray w ith firstdim ension LD C .

LD C leading dim ension ofC
W ORK () scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is not referenced in the cumentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at: http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1.For a generalm atrix (DESCRA (1)=0), array CPN TR can be different from RPNTR. Forallotherm atrix types, RPNTR \(m\) ustequalC CN TR and a single array can be passed forboth argum ents.
2. It is know \(n\) that there exists another representation of the variable block sparse row form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of six anay instead of the seven used in the current im plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each block row in the array \(B \mathbb{N} D X\) is used instead of two arrays BPNTRB and BPNTRE.To use the routine w th this kind of variable block sparse row form at the follow ing calling sequence should be used

SUBROUTINE SVBRMM (TRANSA,MB,N,KB,ALPHA,DESCRA,
* \(\quad V A L, \mathbb{N} D X, B \mathbb{N} D X, R P N T R, C P N T R, \mathbb{A}, \mathbb{A}(2)\),
* B,LDB,BETA, C,LDC,W ORK,LW ORK )

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
cvbrsm - variable block sparse row form at triangular solve

\section*{SYNOPSIS}
```

SUBROUTINE CVBRSM(TRANSA,M B,N,UNITD,DV,ALPHA,DESCRA,

* VAL,\mathbb{NDX,B\mathbb{NDX,RPNTR,CPNTR,BPNTRB,BPNTRE,}}\mathbf{}\mathrm{ , }
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,MB,N,UNITD,DESCRA (5),LDB,LDC,LW ORK
INTEGER \mathbb{NDX(*),BINDX (*),RPNTR M B+1),CPNTR M B+1),}
* BPNTRB M B),BPNTREMB)
COMPLEX ALPHA,BETA
COM PLEX DV (*),VAL (*),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINE CVBRSM_64(TRANSA,M B ,N,UNITD,DV,ALPHA,DESCRA,
* VAL,INDX,BINDX,RPNTR,CPNTR,BPNTRB,BPNTRE,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,M B,N,UNITD,DESCRA (5),LDB,LDC,LW ORK
\mathbb{NTEGER*8 \mathbb{NDX (*),B}\mathbb{NDX (*),RPNTR M B+1),CPNTR M B+1),}}\mathbf{~}\mathrm{ ,}
* BPNTRBMB),BPNTREMB)
COM PLEX ALPHA,BETA
COMPLEX DV (*),VAL (*),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NEVBRSM (TRANSA,M B, N ],UNITD,DV,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,B}\mathbb{N}DX,RPNTR,CPNTR,BPNTRB,BPNTRE,
* B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])
INTEGER TRANSA,MB,UNITD

```

```

\mathbb{NTEGER,D IM ENSION (:) :: RPNTR,CPNTR,BPNTRB,BPNTRE}
COMPLEX ALPHA,BETA
COM PLEX,D IM ENSION (:) ::VAL,DV
COM PLEX,D IM ENSION (:,:):: B,C

```

SUBROUTINEVBRSM_64 (TRANSA, MB, \(\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A\),
* \(V A L, \mathbb{N} D X, B \mathbb{N} D X, R P N T R, C P N T R, B P N T R B, B P N T R E\),
* \(B,[\) [DB], BETA \(, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N T E G E R *} 8\) TRANSA, MB,UNITD
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: \operatorname{DESCRA}, \mathbb{N} D X, B \mathbb{N} D X\)
\(\mathbb{N} T E G E R * 8, D \mathbb{I M} E N S \mathbb{O}(:):: R P N T R, C P N T R, B P N T R B, B P N T R E\)
COMPLEX ALPHA,BETA
COMPLEX,D \(\mathbb{M} E N S I O N(:):: V A L, D V\)
COM PLEX , D \(\mathbb{M}\) ENSION (: :) :: B , C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA C C <-ALPHA D op (A ) B + BETA C } \\
& C<-A L P H A \text { op (A) D B + BETA C } \\
& \text { where A LPH A and BETA are scalar, C and B arem by n dense m atrices, } \\
& \text { D is a block diagonalm atrix, A is a unit, ornon-unit, upperor } \\
& \text { low ertriangularm atrix represented in variable block sparse row } \\
& \text { form atand op (A ) is one of }
\end{aligned}
\]
op (A) \()=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}\left(\infty n \dot{g}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(\sin m\) atrix \(A\)

N \(\quad\) Num berof colum ns in matrix C

UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 : Scale on left (row block scaling)
3 : Scale on right (colum \(n\) block scaling)

DV () A rray containing the block entries of the block diagonalm atrix D. The size of the Jth block is RPN TR ( \(\mathrm{J}+1\) )-RPN TR (J) and each block containsm atrix entries stored colum n-m ajor. The total length of aray DV is given by the form ula:
sum over J from 1 to M B:

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem ent integer array
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) )
3 :Triangular
4 : Skew (Anti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CON}\) J ( A ) )
N ote: For the routine, DESCRA (1)=3 is only supported.
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) main diagonal type
0 : non-identity blocks on the \(m\) ain diagonal
1 : identily diagonalblock
2 : diagonalblocks are densem atrices
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED )
0 :C C++ com patible
1 :Fortran com patible
D ESCRA (5) repeated indices? NOT IM PLEM EN TED )
0 : unknown
1 :no repeated indices
VAL 0 scalar array of length NN Z consisting of the block entries ofA where each block entry is a dense rectangularm atrix stored colum \(n\) by colum \(n\).
NN Z is the total num berof pointentries in allnonzero block entries of a m atrix A.
\(\mathbb{I N}\) D X 0 integer array of length BNN Z +1 where BNN \(Z\) is the num ber block entries of a \(m\) atrix A such that the I-th elem ent of \(\mathbb{N} D \mathrm{X}[]\) points to the location in VAL of the \((1,1)\) elem ent of the I-th block entry.
\(B \mathbb{N} D\) X () integer array of length BNNZ consisting of the block colum \(n\) indioes of the block entries of A where BNN Z is the num berblock entries of a m atrix A. B lock colum n indices M U ST be sorted in increasing order foreach block row .

RPN TR 0) integer aray of length M B +1 such thatRPN TR (I) RPN TR (1) +1 is the row index of the firstpoint row in the I-th block
row.
RPN TR \(M B+1\) ) is set to \(M+R P N T R(1)\) where \(M\) is the num ber of row \(s\) in square triangularm atrix \(A\).

Thus, the num berof point row s in the I-th block row is RPNTR (I+1)RPNTR (I).

NOTE: For the cumentversion CPN TR m ustequalRPN TR and a single array can be passed forboth argum ents

CPNTR 0 integeramay of length \(M B+1\) such thatCPN TR (J)-CPN TR (1) +1 is the colum \(n\) index of the firstpoint colum \(n\) in the \(J\) th block colum n. CPN TR M B+1) is set to M +CPN TR (1). Thus, the num ber of pointcolum ns in the J-th block colum n is CPNTR ( \(\mathrm{J}+1\) )-CPNTR (J).

NO TE: For the current version CPN TR m ustequal RPN TR and a single aray can be passed forboth argum ents
BPN TRB 0 integer aray of length \(M B\) such thatBPN TRB (I)-BPNTRB (1)+1 points to location in B IND X of the firstblock entry of the I-th block row of A.

BPN TRE () integer array of length \(M B\) such thatBPN TRE (I) BPN TRB (1) points to location in B \(\mathbb{N} D \mathrm{X}\) of the last.block entry of the I-th block row of A.

B 0 rectangular aray w th first dim ension LD B.

LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC
W ORK 0 scratch array of length LW ORK. On exit, ifLW ORK = -1,W ORK (1) retums the optim um size of LW ORK.

LW ORK length of ORK array. LW ORK should be at least \(\mathrm{M}=\mathrm{RPNTR} \mathrm{M} \mathrm{B}+1)\) RPNTR (1).

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=\mathrm{M} * \mathrm{~N}\) _CPUS where N _CPUS is the \(m\) axim um num berof processors available to the program .

IfLW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK amray, and no enrorm essage related to LW ORK is issued

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/n csd/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such testsm ust.be perform ed before calling this routine.
2. If \(D E S C R A\) (3)=0, the low erorupper triangular part of each diagonalblock is used by the routine depending on DESCRA (2).
3. If \(D E S C R A(3)=1\), the unit diagonalblocksm ightorm ight not.be referenced in the VBR representation of a sparse \(m\) atrix. They are notused anyw ay .
4. If \(D E S C R A\) (3)=2, diagonalblocks are considered as dense \(m\) atrices and the LU factorization \(w\) ith partialpivoting is used by the routine. WORK (1)=0 on retum if the factorization for alldiagonalblocks has been com pleted successfully, otherw ise WORK (1) = -iw here is the block num ber forw hich the LU factorization could notbe com puted.
5.The routine can be applied for solving triangular system \(s\) w hen the upper or low er triangle of the general sparse m atrix \(A\) is used. H ow erverDESCRA (1) m ustbe equalto 3 in this case. DESCRA (2) indicates w hich triangle w illbe used.
6. It is know \(n\) that there exists another representation of the variable block sparse row form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of six array instead of the seven used in the current im plem entation. Them ain difference is that only one array, IA , containing the pointers to the beginning ofeach block row in the array \(B \mathbb{N} D X\) is used instead of two arrays BPN TRB and BPN TRE.To use the routine w th this kind of variable block sparse row

SUBROUTINECVBRSM (TRANSA,MB,N,UNITD,DV,ALPHA,DESCRA,
* \(\quad V A L, \mathbb{N} D X, B \mathbb{N} D X, R P N T R, C P N T R, \mathbb{A}, \mathbb{A}(2)\),
* B,LDB,BETA, C,LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
cvm ul-com pute the scaled product of com plex vectors

\section*{SYNOPSIS}

```

    COM PLEX ALPHA,BETA
    COM PLEX X (*),Y (*),Z (*)
    \mathbb{NTEGERN,INCX,}\mathbb{N}CY,\mathbb{NCZ}
    ```

```

    COM PLEX ALPHA,BETA
    COM PLEX X (*),Y (*),Z (*)
    \mathbb{NTEGER*8N,}\mathbb{NCX,INCY,INCZ}
    F95 INTERFACE

```

```

    COMPLEX ::ALPHA,BETA
    COM PLEX,D IM ENSION (:) ::X,Y,Z
    ```


```

    COMPLEX ::ALPHA,BETA
    COM PLEX,D IM ENSION (:) ::X,Y,Z
    \mathbb{NTEGER (8)::N,INCX,\mathbb{NCY, INCZ}}\mathbf{N}={
    C INTERFACE
\#include <sunperfh>

```
void crm ul(intn, com plex *alpha, com plex *x, intincx, com plex *y, int incy, com plex *beta, com plex *z, int incz);
void cvm ul 64 (long n, com plex *alpha, com plex *x, long incx, com plex *y, long incy, com plex *beta, com plex *z, long incz);

\section*{PURPOSE}
cvm ulcom putes the scaled product of com plex vectors: \(z(i)=A L P H A * x(i) * y(i)+B E T A * z(i)\) for \(1<=\mathrm{i}<=\mathrm{N}\).

\section*{ARGUMENTS}

N (input)
Length of the vectors. \(\mathrm{N}>=0\). Retums im mediately if \(\mathrm{N}=0\).

ALPHA (input)
Scale factor on the m ultiplicand vectors.

X (input) dim ension (*)
M ultiolicand vector.
\(\mathbb{N C X}\) (input)
Stride betw een elem ents of the m ultiplicand vector X. \(\mathbb{N} C X>0\).

Y (input) dim ension (*)
M ultiplicand vector.
\(\mathbb{N C C Y}\) (input)
Stride betw een elem ents of the m ultiplicand vector
Y. \(\mathbb{N} C Y>0\).

BETA (input)
Scale factor on the product vector.

Z (input/output)
dim ension (*)
Productvector. On exit, \(z(i)=A L P H A * x(i) *\)
\(y\) (i) + BETA * \(z\) (i).
\(\mathbb{N C Z}\) (input)
Stride betw een elem ents of Z. \(\mathbb{N N C Z}>0\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dasum -Retum the sum of the absolute values of a vectorx.

\section*{SYNOPSIS}
\[
\text { DOUBLE PRECISION FUNCTION DASUM } \mathbb{N}, \mathrm{X}, \mathbb{N} C X)
\]
\(\mathbb{N}\) TEGER \(N, \mathbb{N C X}\)
DOUBLE PRECISION X (*)
DOUBLE PRECISION FUNCTION DASUM _ \(64(\mathbb{N}, \mathrm{X}, \mathbb{N} C X)\)
\(\mathbb{N} T E G E R * 8 N, \mathbb{N C X}\)
DOUBLE PRECISION X (*)

\section*{F95 INTERFACE}

REAL (8) FUNCTION ASUM ( \(\mathbb{N}], \mathrm{X},[\mathbb{N} C X])\)
\(\mathbb{N} T E G E R:: N, \mathbb{N} C X\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::X

REAL (8) FUNCTION ASUM _64 ( \(\mathbb{N}\) ], X, [ \(\mathbb{N} C X]\) )
\(\mathbb{N} T E G E R(8):: N, \mathbb{N} C X\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::X

\section*{C INTERFACE}
\#include <sunperfh>
double dasum (intn, double *x, intincx);
double dasum _64 (long n, double *x, long incx);

\section*{PURPOSE}
dasum Retum the sum of the absolute values of \(x\) where \(x\) is an n -vector.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.
X (input)
\((1+(n-1) * a b s(\mathbb{N} C X))\). On entry, the increm ented array \(X\) must contain the vectorx. U nchanged on exit.
\(\mathbb{N} C X\) (input)
O n entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{I N C X}\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

daxpy - com pute y := alpha * x + y

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDAXPY N,ALPHA,X,\mathbb{NCX,Y, NNCY)}}\mathbf{N},\textrm{N}
INTEGERN,\mathbb{NCX,INCY}
DOUBLE PRECISION ALPHA
DOUBLE PRECISION X (*),Y(*)
SU BROUT\mathbb{NE DAXPY_64 N,ALPHA,X, INCX,Y, INCY)}
INTEGER*8N,\mathbb{NCX,INCY}
DOUBLE PRECISION ALPHA
DOUBLE PRECISION X (*),Y (*)
F95 INTERFACE

```

```

\mathbb{NTEGER ::N,\mathbb{NCX,}\mathbb{NCY}}\mathbf{N}=\mathbb{N}
REAL (8) ::A LPHA
REAL (8),D IM ENSION (:) ::X,Y

```

```

\mathbb{NTEGER (8)::N,\mathbb{NCX,INCY}}\mathbf{N}={
REAL (8) ::A LPHA
REAL (8),D IM ENSION (:) ::X,Y

```

\section*{C INTERFACE}
\#include <sunperfh>
void daxpy (intn, double alpha, double *x, int incx, double *y, int incy);
void daxpy_64 (long n, double alpha, double *x, long incx, double *y, long incy);

\section*{PURPOSE}
daxpy com pute \(\mathrm{y}:=\) alpha * \(\mathrm{x}+\mathrm{y}\) w here alpha is a scalar and x and y are n -vectors.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.

\section*{A LPHA (input)}

On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
\((1+(n-1) * a b s(\mathbb{N} C X))\). Before entry, the increm ented array \(X\) must contain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X\) m ustnotbe zero. U nchanged on exit.

Y (input/output)
( \(1+(\mathrm{n}-1) * \mathrm{abs}(\mathbb{N} C Y)\) ). On entry, the increm ented aray \(Y\) m ustcontain the vectory. On exit, \(Y\) is overw rilten by the updated vectory.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y\). \(\mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

daxpyi-C om pute y = alpha * x + y

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DAXPYINZ,A,X,\mathbb{NDX,Y)}}\mathbf{N}=(
DOUBLE PRECISION A
DOUBLE PRECISION X (*),Y (*)
INTEGERNZ
INTEGER INDX(*)
SUBROUTINE DAXPY I_64 N Z,A,X,NNDX,Y)
DOUBLE PRECISION A
DOUBLE PRECISION X (*),Y (*)
INTEGER*8NZ
INTEGER*8 \mathbb{NDX (*)}
F95 IN TERFACE
SUBROUT\mathbb{NE AXPYI(NZ],[A],X,NNDX,Y)}
REAL (8) ::A
REAL (8),D IM ENSION (:) ::X,Y
\mathbb{NTEGER ::NZ}
\mathbb{NTEGER,D IM ENSION (:) :: \mathbb{NDX}}\mathbf{N}=\mp@code{N}
SUBROUTINE AXPYI_64(NZ ],[A],X,\mathbb{NDX,Y)}
REAL (8) ::A
REAL (8),D $\mathbb{I}$ ENSION (:) :: X,Y
$\mathbb{N}$ TEGER (8) ::N Z
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{I} \operatorname{ENSION(:)::\mathbb {IN}\mathrm {X}}$

```

\section*{PURPOSE}

D A X PY IC om pute \(y:=\) alpha * \(x+y\) where alpha is a scalar, \(x\) is a sparse vector, and \(y\) is a vector in fullstorage form
```

do i=1,n
y (indx (i)) = alpha * x (i) + y (indx (i))
enddo

```

\section*{ARGUMENTS}

NZ (input) - \(\mathbb{N}\) TEGER
N um ber of elem ents in the com pressed form.
U nchanged on exit.

A (input)
On entry, A (LPH A) specifies the scaling value.
U nchanged on exit. A is defaulted to 1.0D 0 forF95
\(\mathbb{I N}\) TERFACE.
X (input)
V ector containing the values of the com pressed form.
U nchanged on exit.
\(\mathbb{N} D \mathrm{X}\) (input) \(-\mathbb{N}\) TEGER
V ector containing the indioes of the com pressed
form. It is assum ed that the elem ents in \(\mathbb{N}\) D X are distinctand greater than zero. U nchanged on exit.

Y (output)
V ector on inputw hich contains the vector \(Y\) in full storage form. O n exit, only the elem ents
comesponding to the indices in \(\mathbb{N}\) D X have been
m odified.

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dbcom m -block coordinate \(m\) atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDBCOMM(TRANSA,M B,N,KB,ALPHA,DESCRA,}

* VAL,BINDX,BJNDX,BNNZ,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER TRANSA,MB,N,KB,DESCRA (5),BNNZ,LB,
* LDB,LDC,LWORK
\mathbb{NTEGER B}\mathbb{NDX (BNNZ),BJNDX (BNNZ)}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL (LB *LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINEDBCOMM_64(TRANSA,M B,N,KB,ALPHA,DESCRA,
* VAL,BINDX,BJNDX,BNNZ,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER*8 TRANSA,MB,N,KB,DESCRA (5),BNNZ,LB,
* LDB,LDC,LW ORK
INTEGER*8 B\mathbb{NDX (BNNZ),BJNDX (BNNZ)}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL (LB *LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}

SUBROUTINE BCOMM (TRANSA, MB,N,KB,ALPHA,DESCRA,VAL,BINDX,BUNX, * BNNZ,LB,B,[LDB],BETA,C,[LDC],[WORK],[LW ORK])
\(\mathbb{N} T E G E R\) TRANSA, MB,N,KB,BNNZ,LB
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: D E S C R A, B \mathbb{N D} X, B J N D\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D \(\mathbb{M}\) ENSION (:) ::VAL DOUBLE PRECISION ,D \(\mathbb{M}\) ENSION (:, :) :: B,C
* BNNZ,LB,B,[LDB],BETA,C,[LDC],[WORK],[LWORK])
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,N,KB,BNNZ,LB
\(\mathbb{N} T E G E R * 8, D \mathbb{I}\) ENSION (:) :: DESCRA,B \(\mathbb{N D} \mathrm{X}, \mathrm{B}\) JNDX
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D IM ENSION (:) ::VAL
DOUBLE PRECISION ,D IM ENSION (:, :) :: B,C

\section*{DESCRIPTION}
\[
C \text { <-alpha op (A) B + beta C }
\]
where A LPH A and BETA are scalar, C and B are dense \(m\) atrices, \(A\) is a m atrix represented in block coordinate form at and op(A) is one of \(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).
('indicates m atrix transpose)

\section*{ARGUMENTS}
\begin{tabular}{|c|c|}
\hline \multirow[t]{4}{*}{TRANSA} & A Indicates how to operate w ith the sparse \(m\) atrix \\
\hline & 0 : operate w th m atrix \\
\hline & 1 : operate w ith transpose m atrix \\
\hline & 2 : operate \(w\) th the conjugate transpose ofm atrix. 2 is equivalent to 1 if the \(m\) atrix is real. \\
\hline M B & N um berofblock row s in m atrix A \\
\hline N & \(N\) um berof colum ns in m atrix C \\
\hline K B & \(N\) um ber ofblock colum ns in m atrix A \\
\hline A LPH A & Scalarparam eter \\
\hline \multirow[t]{13}{*}{DESCRA} & 0 D escriptor argum ent. Five elem ent integer array \\
\hline & DESCRA (1) m atrix structure \\
\hline & 0 : general \\
\hline & 1 : symmetric ( \(A=A\) ) \\
\hline & \(2: \mathrm{Herm}\) Itian ( \(\mathrm{A}=\mathrm{CONJ}\) ( A ) ) \\
\hline & 3 :Triangular \\
\hline & 4 : Skew (A nti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) ) \\
\hline & 5 :D iagonal \\
\hline & 6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJG}(\mathrm{A})\) ) \\
\hline & D ESCRA (2) upper/low er triangular indicator \\
\hline & 1 : low er \\
\hline & 2 :upper \\
\hline & DESCRA (3) m ain diagonal type \\
\hline
\end{tabular}

0 :non-unit
1 : unit
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED )
0 :C C++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL ( scalar array of length LB *LB *BNN Z consisting of the non-zero block entries of \(A\), in any order. Each block is stored in standard colum n-m ajor form .
\(B \mathbb{N} D X(\) integer array of length \(B N N Z\) consisting of the block row indiaes of the block entries of .

B JND X 0 integer anray of length BNNZ consisting of the block colum \(n\) indiges of the block entries of \(A\).

BNNZ num berofblock entries

LB dim ension of dense blocks com posing A.
B 0 rectangular array with first dim ension LD B .
LD B leading dim ension ofB
BETA Scalarparam eter

C 0 rectangular array w ith firstdim ension LD C .

LD C leading dim ension ofC
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the currentversion.

LW ORK length ofW ORK array.LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov \(\mathrm{m}_{\mathrm{c}}\) csd/Staffk Rem ington/tspoblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dbdim m -block diagonal form atm atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUTINEDBD $\mathbb{M} M$ (TRANSA, MB,N,KB,ALPHA,DESCRA,

* VAL,BLDA, $\mathbb{B D} \mathbb{A} G, N B D \mathbb{I} G, L B$,
* $\quad \mathrm{B}, \mathrm{LD} B, B E T A, C, L D C, W$ ORK,LW ORK)
$\mathbb{N} T E G E R$ TRANSA,MB,N,KB,DESCRA (5),BLDA,NBD $\mathbb{A} G, L B$,
* LDB,LDC,LW ORK
$\mathbb{N} T E G E R \quad \mathbb{B D} \mathbb{I} G \mathbb{N} B D \mathbb{I} G)$
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL (LB*LB*BLDA*NBDIAG), B (LDB,*), C (LDC,*), WORK (LWORK)

```
SUBROUTINEDBD \(\mathbb{I} M\) M_64 (TRANSA, M B, N, KB, ALPHA, DESCRA,
* VAL,BLDA, \(\mathbb{B D} \mathbb{I} G, N B D \mathbb{I A}, L B\),
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,N,KB,DESCRA (5), BLDA,NBD \(\mathbb{I} G, L B\),
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R * 8 \mathbb{B D} \mathbb{A} G \mathbb{N} B \mathbb{A} G)\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL (LB*LB*BLDA*NBDIAG), B (LDB,*), C (LDC,*), WORK (LWORK)

\section*{F95 INTERFACE}

SUBROUTINEBD \(\mathbb{I} M\) (TRANSA, MB, \(\mathbb{N}], K B, A L P H A, D E S C R A, V A L, B L D A\), * \(\mathbb{B D} \mathbb{I} G, N B D \mathbb{I} G, L B, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N} T E G E R\) TRANSA, MB,KB,BLDA,NBD \(\mathbb{I A} G, L B\)
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \quad\) DESCRA, \(\mathbb{B D} \mathbb{I A}\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D IM ENSION (:) ::VAL
DOUBLE PRECISION ,D \(\mathbb{M}\) ENSION (:, :) :: B,C
SUBROUTINEBD \(\mathbb{M} M \_64\) (TRANSA, MB, \(\left.\mathbb{N}\right], K B, A L P H A, D E S C R A, V A L, B L D A\),
* \(\mathbb{B D} \mathbb{I} G, N B D \mathbb{I} G, L B, B,[L D B], B E T A, C,[L D C],[W\) ORK], [LW ORK])
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,KB,BLDA,NBDIAG,LB
\(\mathbb{N} T E G E R * 8, D \mathbb{I M} E N S \mathbb{O N}(:):: \quad\) ESCRA, \(\mathbb{B D} \mathbb{I A} G\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D \(\mathbb{M}\) ENSION (:) ::VAL
DOUBLE PRECISION ,D IM ENSION (: : : :: B,C

\section*{DESCRIPTION}
\[
C \text { <-alpha op (A) B + beta C }
\]
where A LPHA and BETA are scalar, C and B are dense \(m\) atrices, \(A\) is a m atrix represented in block diagonal form at and op(A) is one of \(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).
('indicates m atrix transpose)

\section*{ARGUMENTS}
\begin{tabular}{|c|c|}
\hline \multirow[t]{4}{*}{TRANSA} & A Indicates how to operate w th the sparse \(m\) atrix \\
\hline & 0 : operate w th m atrix \\
\hline & 1 : operate w ith transpose m atrix \\
\hline & 2 : operate \(w\) th the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real. \\
\hline M B & N um berofblock row s in m atrix A \\
\hline N & \(N\) um berof colum \(n s\) in \(m\) atrix \(C\) \\
\hline K B & \(N\) um ber ofblock colum ns in m atrix A \\
\hline A LPH A & Scalar param eter \\
\hline \multirow[t]{13}{*}{DESCRA} & () D escriptor argum ent. Five elem ent integer amay \\
\hline & DESCRA (1) m atrix structure \\
\hline & 0 : general \\
\hline & 1 : symmetric ( \(\mathrm{A}=\mathrm{A}\) ) \\
\hline & 2: Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) ) \\
\hline & 3 :Triangular \\
\hline & 4 : Skew (A nti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) ) \\
\hline & 5 :D iagonal \\
\hline & 6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJG}(\mathrm{A})\) ) \\
\hline & D ESCRA (2) upper/low er triangular indicator \\
\hline & 1 : low er \\
\hline & 2 : upper \\
\hline & D ESCRA (3) m ain diagonaltype \\
\hline
\end{tabular}

0 : non-unit
1 : unit
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL \(0 \quad\) tw o-dim ensionalLB *LB *BLD A -by-N BD IA G scalar anay consisting of the NBD IA G nonzero block diagonal in any order. Each dense block is stored in standard colum n.m ajor form .

BLD A leading block dim ension ofV A L ( ).
IBD IA G 0 integer amay of length N BD IA G consisting of the corresponding diagonaloffsets of the non-zero block diagonals ofA in VA L. Low ertriangular block diagonals have negative offsets, the \(m\) ain block diagonal has offset 0, and uppertriangular block diagonals have positive offset.

NBD IA G the num berofnon-zero block diagonals in A.
LB dim ension of dense blocks com posing A.
B 0 rectangular array w ith first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w th first dim ension LD C.

LD C leading dim ension ofC
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib org/utk/papers/sparse .ps

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dbdism - block diagonal form at triangular solve

\section*{SYNOPSIS}
```

SUBROUTINEDBD ISM (TRANSA, M B,N,UNITD,DV,ALPHA,DESCRA,

* VAL,BLDA, $\mathbb{B D} \mathbb{A} G, N B D \mathbb{I A}, \mathrm{LB}$,
* B,LDB,BETA, C,LDC,WORK,LWORK)
$\mathbb{N} T E G E R$ TRANSA, MB,N,UNITD,DESCRA (5), BLDA,NBD $\mathbb{I} G, L B$,
* LDB,LDC,LW ORK
$\mathbb{N} T E G E R \quad \mathbb{B D} \mathbb{I} G \mathbb{N} B D \mathbb{I} G)$
DOUBLE PRECISION ALPHA,BETA
D OUBLE PRECISION DV M B *LB*LB),VAL (LB*LB*BLDA,NBD IAG), B (LDB,*), C (LDC,*),
* WORK (LW ORK)
SUBROUTINEDBD ISM_64(TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,
* VAL,BLDA, $\mathbb{B D} \mathbb{I} G, N B D \mathbb{A} G, L B$,
* B,LDB,BETA,C,LDC,WORK,LW ORK)
$\mathbb{N} T E G E R * 8$ TRANSA, M B,N,UN ITD,DESCRA (5), BLDA,NBD IAG,LB,
* LDB,LDC,LWORK
$\mathbb{N} T E G E R * 8 \mathbb{B D} \mathbb{I} G \mathbb{N} B \mathbb{I} G)$
DOUBLE PRECISION ALPHA,BETA
D OUBLE PRECISION DV M B *LB*LB),VAL (LB*LB*BLDA,NBD IAG), B (LDB,*), C (LDC,*),
* $\quad$ WORK (LW ORK)

```

\section*{F95 INTERFACE}
```

SUBROUTINE BD ISM (TRANSA,M B,N,UNITD,DV,ALPHA,DESCRA,VAL,BLDA,

* \mathbb{BDIAG,NBD IAG,LB,B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])}
INTEGER TRANSA,MB,N,UNITD,BLDA,NBDIAG,LB

```

```

DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D IM ENSION (:) ::VAL,DV
DOUBLE PRECISION,D IM ENSION(:, :) :: B,C

```

SUBROUTINE BD ISM _64 (TRANSA , M B , N , UNITD , DV, ALPHA, DESCRA, VAL, BLDA,
* \(\mathbb{B D} \operatorname{IAG}, N B D I A G, L B, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N} T E G E R * 8\) TRANSA, M B , N , UN ITD , BLDA, NBD IA G , LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O}(:):: \quad \mathrm{DESCRA}, \mathbb{B D} \mathbb{I} G\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D \(\mathbb{M}\) ENSION (:) ::VAL, DV
D OUBLE PRECISION ,D \(\mathbb{M}\) ENSION (: : : : : B , C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { op (A) B + BETA } C \quad C<-A L P H A D \text { op (A) B + BETA C } \\
& C<-A L P H A \text { Op (A)D B + BETA } C
\end{aligned}
\]
where ALPHA and BETA are scalar, \(C\) and \(B\) are m by \(n\) dense \(m\) atrices, \(D\) is a block diagonalm atrix, \(A\) is a unit, ornon-unit, upperor low ertriangularm atrix represented in block diagonal form at and op (A) is one of op (A) \()=\operatorname{inv}(A)\) or op (A \()=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}\left(\operatorname{cong}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix 0 : operate \(w\) ith \(m\) atrix 1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(s\) in matrix A
\(N \quad N\) um berof colum \(n s\) in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity matrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)

DV () A rray of length M B *LB *LB containing the elem ents of the diagonalblocks of them atrix \(D\). The size of each square block is LB -by-LB and each block is stored in standard colum n-m ajor form .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay DESCRA (1) m atrix structure
            0 : general
            1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
            2 : Herm itian ( \(A=\operatorname{CONJG}(A))\)
            3 :Triangular
            4 : Skew (A nti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) )
            5 :D iagonal
            6 : Skew Herm itian ( \(A=-C O N J(A)\) )
                            N ote: For the routine, D ESCRA (1)=3 is only supported.
                            D ESCRA (2) upper/low er triangular indicator
            1 : low er
            2 :upper
DESCRA (3) m ain diagonaltype
            0 : non-identity blocks on the \(m\) ain diagonal
            1 : identity diagonalblocks
            2 : diagonalblocks are dense \(m\) atrices
            DESCRA (4) A may base \(\mathbb{N O T} \mathbb{I}\) PLEM ENTED)
            0 :C C ++ com patible
            1 :Fortran com patible
                    DESCRA (5) repeated indices? \(\mathbb{N O T} \mathbb{I}\) PLEM ENTED)
                    0 : unknown
                            1 : no repeated indices
VAL () Two-dim ensionalLB *LB *B LD A -by-N BD IA G scalaranay
consisting of the N BD IA G non-zero block diagonal.
Each dense block is stored in standard colum n-m ajor form .
B LD A Leading block dim ension ofV A L (). Should be greater
    than orequal to M B .
IBD IA G 0 integer amay of length NBD IA G consisting of the corresponding diagonal offsets of the non-zero block diagonals ofA in VA L. Low ertriangularblock diagonals have negative offsets, them ain block diagonalhas offset 0 , and upper triangularblock diagonals have positive offset. Elem ents of IBD IA G M UST be sorted in increasing order.
NBD IA G The num berofnon-zero block diagonals in A.
LB D im ension of dense blocks com posing A.
B 0 Rectangular aray with firstdim ension LD B .
LD B Leading dim ension of B .
BETA Scalarparam eter.
C 0 Rectangular array w ith first dim ension LD C .
LD C Leading dim ension of C .

W ORK 0 scratch array of length LW ORK. On exit, ifLW ORK=-1,W ORK (1) retums the optim um size ofLW ORK.

LW ORK length ofW ORK array. LW ORK should be at least M B *LB.

Forgood penform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=M B * L B * N \_C P U S\) where \(N\) _CPU \(S\) is the \(m\) axim um num berof processors available to the program .

IfLW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no enrorm essage related to LW ORK is issued by X ERBLA.

\section*{SEE ALSO}

N IST FORTRA N Sparse B las U sers G uide available at: http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such tests \(m\) ustbe perform ed before calling this routine.
2. If \(D E S C R A\) (3)=0, the low erorupper triangularpart of each diagonalblock is used by the routine depending on DESCRA (2).
3. If \(D E S C R A\) (3)=1, the unit diagonalblocksm ightorm ight notbe referenced in the BD I representation of a sparse \(m\) atrix. They are notused anyw ay .
4. If \(D E S C R A(3)=2\), diagonalblocks are considered as dense \(m\) atrices and the LU factorization \(w\) ith partialpivoting is used by the routine. WORK (1)=0 on retum if the factorization for alldiagonalblocks has been com pleted successfully, otherw ise WORK (1) = -iw here is the block
num ber forw hich the LU factorization could notbe com puted.
5. The routine can be applied for solving triangular system \(s\) w hen the upper or low er triangle of the general.sparse m atrix \(A\) is used. H ow erver \(D E S C R A\) (1) m ustbe equal to 3 in this case.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dbdsdc - com pute the singularvalue decom position (SV D ) of a
realN -by -N (upper or low er) bidiagonalm atrix B

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE DBD SDC (UPLO,COM PQ,N,D,E,U,LDU,VT,LDVT,Q,IQ,}
W ORK,IN ORK,\mathbb{NFO)}

```
CHARACTER * 1 UPLO, COMPQ
\(\mathbb{N} T E G E R N, L D U, L D V T, \mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{I}\left(^{*}\right)\), \(\mathbb{I N} O R K(*)\)
DOUBLE PRECISION D (*), E (*), U (LDU , *), VT (LDVT, \(\left.{ }^{\star}\right), \mathrm{Q}\) ( \(\left.^{*}\right)\),
WORK (*)
SU BROUTINE DBD SDC_64 (UPLO, COM PQ,N,D,E,U,LDU,VT,LDVT,Q, IQ,
    W ORK, \(\mathbb{I N} O R K, \mathbb{N} F O)\)
CHARACTER * 1 UPLO, COMPQ
\(\mathbb{N}\) TEGER* \(8 \mathrm{~N}, \mathrm{LD} \mathrm{U}, \mathrm{LDVT}, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{I}(*), \mathbb{I V} O R K(*)\)
D OUBLE PRECISION D (*), E (*), U (LDU ,*), VT( LDVT,*), Q (*),
W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE BDSDC (UPLO, COMPQ, \(\mathbb{N}], D, E, U,[L D U], V T,[L D V T], Q, I Q\), [W ORK ], [IN ORK ], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::UPLO, COM PQ
\(\mathbb{N} T E G E R:: N, L D U, L D V T, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: IQ, \(\mathbb{I N}\) ORK
REAL (8),D \(\mathbb{M}\) ENSION (:) ::D , E, Q,W ORK

REAL (8), D IM ENSION (:,:) ::U ,VT
SU BROUTINE BD SDC_64 (UPLO ,COM PQ, \(\mathbb{N}], D, E, U,[L D U], V T,[L D V T], Q\), \(\mathbb{I}, \mathbb{W}\) ORK \(],[\mathbb{N}\) ORK \(],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO, COM PQ
\(\mathbb{N}\) TEGER (8) :: N, LDU,LDVT, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:): \(: \mathbb{I}, \mathbb{I V}\) ORK
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D , E, Q, W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::U,VT

\section*{C INTERFACE}
\#include <sunperfh>
void dbdsdc (charuple, char com pq, intn, double *d, double
*e, double *u, int ldu, double *vt, int ldvt, double *q, int *iq, int *info);
void dbdsdc_64 (charuplo, charcom pq, long n, double *d, double *e, double *u, long ldu, double *vt, long ldvt, double *q, long *iq, long *info);

\section*{PURPOSE}
dbdsdc com putes the singular value decom position (SVD ) of a realN -by -N (upper or low er) bidiagonalm atrix \(\mathrm{B}: \mathrm{B}=\mathrm{U} * \mathrm{~S}\) * VT, using a divide and conquerm ethod, where \(S\) is a diagonalm atrix w ith non-negative diagonalelem ents the singular values ofB ), and U and V T are orthogonalm atrices of left and right singular vectors, respectively. SBD SD C can be used to com pute allsingular values, and optionally, singular vectors or singular vectors in com pact form .

This code \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. It w ill w ork on m achines w ith a guard digitin add/subtract, oron those binary machines w ithout guard digits which subtract like the \(C\) ray \(X-M P, C\) ray \(Y ~ M P, C\) ray \(\mathrm{C}-90\), or C ray-2. Itcould conœívably fail on hexadecim al or decin al machines w thout guard digits, butw e know of none. See SLA SD 3 fordetails.

The code cumently callSLA SD Q if singularvalues only are desired. H ow ever, it can be slightly \(m\) odified to com pute singular values using the divide and conquerm ethod.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': B is upperbidiagonal.
= L ': B is low erbidiagonal.

COM PQ (input)
Specifies w hether singularvectors are to be com puted as follow s:
\(=\mathrm{N}:\) : C om pute singular values only;
\(=\mathrm{P}\) ': C om pute singularvahues and com pute singular vectors in compact form ; = I': C om pute singularvalues and singularvectors.

N (input) The orderof the matrix \(\mathrm{B} . \mathrm{N}>=0\).

D (input/output)
O n entry, the n diagonalelem ents of the bidiagonal matrix B. On exit, if \(\mathbb{N} F O=0\), the singular values ofB.

E (input/output)
O n entry, the elem ents of \(E\) contain the offdiagonalelem ents of the bidiagonalm atrix whose SVD is desired. On exit, E has been destroyed.

U (output)
If \(\mathrm{COMPQ}=\) ' \('\) ', then: On exit, if \(\mathbb{N F O}=0\), U contains the left singular vectors of the bidiagonalm atrix. Forothervahues of \(C O M P Q, U\) is not referenced.

LD U (input)
The leading dim ension of the array \(U\). LD U >= 1 . If singular vectors are desired, then LD \(U>=\max\) ( \(1, \mathrm{~N})\).

VT (output)
If \(C O M P Q=I '\), then: \(O n\) exit, if \(\mathbb{N F F O}=0, V T\) '
contains the right singularvectors of the bidiagonalm atrix. Forothervalues of COM PQ,VT is not referenced.

LDVT (input)
The leading dim ension of the array V T. LD V T >= 1 . If singular vectors are desired, then LD V T \(>=\max\) ( 1,N ).
\(Q\) (input) If \(C O M P Q=P\) ', then: \(O n\) exit, if \(\mathbb{N} F O=0, Q\) and \(\mathbb{I Q}\) contain the left and rightsingularvectors in a com pact form, requiring \(O(N \log N\) ) space instead of \(2 * \mathrm{~N} * * 2\). In particular, \(Q\) contains all
the REAL data in LDQ \(>=\mathrm{N}^{*}(11+2 * S M L S I+\) \(8 * \mathbb{N}\) T (LOG_2 \(\mathbb{N} /(\) SM LSIZ + 1) ) )) w ords ofm em ory, where SM LSIZ is retumed by ILAENV and is equal to the m axim um size of the subproblem s at the bottom of the com putation tree (usually about 25). For othervalues of COM PQ,Q is notreferenced.

IQ (output)
If \(C O M P Q=P\) ', then: On exit, if \(\mathbb{N} F O=0, Q\) and \(I Q\) contain the leftand rightsingularvectors in a com pactform, requiring \(O(\mathbb{N} \log N\) ) space instead of \(2 * N * * 2\). In particular, IQ contains all \(\mathbb{N T E G E R ~ d a t a ~ i n ~ L D I Q ~}>=\mathrm{N} *(3 \quad+\) \(3 * \mathbb{N}\) T (LOG_2 \(\mathbb{N} /(\) SM LSSZ + 1) ) )) w ords ofm em ory, where SM LSIZ is retumed by \(\amalg \mathrm{A} A \mathrm{ENV}\) and is equal to the \(m\) axim um size of the subproblem s at the bottom of the com putation tree (usually about 25). For other values of \(C O M P Q, \mathbb{Q}\) is notreferenced.

W ORK (w orkspace)
IfCOMPQ \(=N\) 'then LW ORK \(>=(2 * N)\). IfCOMPQ \(=\)
\(P^{\prime}\) then LW ORK \(>=(6 * N)\). IfCOMPQ = \(I^{\prime}\) then
LWORK \(>=(3 * N * * 2+4 * N)\).

IW ORK (w orkspace)
dim ension ( \(8 * N\) )
\(\mathbb{N}\) FO (output)
= 0: successfulexit.
\(<0:\) if \(\mathbb{N F O}=-i\), the i-th argum enthad an ille-
galvalue.
\(>0\) : The algorithm failed to com pute an singular
value. The update process of divide and conquer
failed.

\section*{FURTHER DETAILS}

B ased on contributions by
M ing Gu and H uan Ren, C om puterScience D ivision, U niversity of

C alifomia at B erkeley, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dbdsqr-com pute the singularvahue decom position (SV D ) of a realN -by -N (upper or low er) bidiagonalm atrix B .

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DBD SQR (UPLO,N,NCVT,NRU,NCC,D,E,VT,LDVT,U,LDU,C,}
LDC,W ORK,\mathbb{NFO)}
CHARACTER * 1 UPLO
INTEGERN,NCVT,NRU,NCC,LDVT,LDU,LDC, INFO
DOUBLE PRECISION D (*),E (*),VT (LDVT,*),U (LDU,*),C (LDC,*),
W ORK (*)
SU BROUTINE D BD SQR_64 (UPLO,N,NCVT,NRU,NCC,D,E,VT,LDVT,U,LDU,
C,LDC,WORK,\mathbb{NFO)}

```

CHARACTER * 1 UPLO
\(\mathbb{N} T E G E R * 8 N, N C V T, N R U, N C C, L D V T, L D U, L D C, \mathbb{N} F O\)
DOUBLE PRECISIOND (*), E ( \(\left.{ }^{*}\right), \mathrm{VT}\left(\mathrm{LDVT},^{\star}\right)\), U (LDU, \(\left.{ }^{\star}\right), \mathrm{C}(\mathrm{LDC}, \star)\),
W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE BDSQR (UPLO, \(\mathbb{N}], \mathbb{N} C V T], \mathbb{N} R U], \mathbb{N C C}], D, E, V T,[L D V T]\), U, [LD U ], C, [LD C ], [W ORK], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::UPLO
\(\mathbb{I N} T E G E R:: N, N C V T, N R U, N C C, L D V T, L D U, L D C, \mathbb{N F O}\)
REAL (8),D \(\mathbb{M}\) ENSION (:) :: D ,E,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::VT,U,C
SUBROUTINE BDSQR_64 (UPLO, \(\mathbb{N}], \mathbb{N} C V T], \mathbb{N} R U], \mathbb{N C C}], D, E, V T,[L D V T]\), \(\mathrm{U},[\mathrm{LD} \mathrm{U}], \mathrm{C},[\mathrm{LDC}],[\mathbb{W}\) ORK],[ \(\mathbb{N F O}])\)

CHARACTER (LEN=1) :: UPLO
\(\mathbb{N}\) TEGER (8) :: N , NCVT,NRU,NCC,LDVT,LDU, LDC, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::VT,U , C

\section*{C INTERFACE}
\#include <sunperfh>
void dbdsqr(charuple, intn, intncvt, int nnu, int ncc, double *d, double *e, double *vt, int ldvt, double *u, int ldu, double * C , int ldc, int *info);
void dbdsqr_64 (char uplo, long n, long ncvt, long nnu, long ncc, double *d, double *e, double *vt, long ldvt, double *u, long ldu, double *c, long ldc, long *info);

\section*{PURPOSE}
dbdsqr com putes the singularvahe decom position (SVD) of a realN -by-N (upper orlow er) bidiagonalm atrix \(B: B=Q * S\) * \(P^{\prime}\left(P^{\prime}\right.\) denotes the transpose of \(\left.P\right)\), w here \(S\) is a diagonal \(m\) atrix \(w\) ith non-negative diagonal elem ents the singular values of \(B\) ), and \(Q\) and \(P\) are orthogonalm atrices.

The routine com putes \(S\), and optionally com putes \(U * Q, P^{\prime} \star\) V , or \(\mathrm{Q}^{\prime} * \mathrm{C}\), for given realinputm atrioes \(\mathrm{U}, \mathrm{V} \mathrm{T}\), and C .

See "C om puting Sm allSingularV alues ofB idiagonalM atrioes W th G uaranteed H igh Relative A ccuracy," by J. D em m eland W . K ahan, LAPACK W orking N ote \#3 (orSIAM J. Sci. Statist. C om put.vol.11, no.5,pp. 873-912, Sept1990) and "A ccurate singular values and differential qd algorithm \(s\), " by B. Parlett and V.Femando, TechnicalReportCPAM -554, \(M\) athem atics D epartm ent, U niversity of \(C\) alifomia at Berkeley, July 1992 for a detailed description of the algorithm .

\section*{ARGUMENTS}

UPLO (input)
\(=U\) ': B is upperbidiagonal;
\(=1 \mathrm{~L}\) ': B is low erbidiagonal.

N (input) The order of the m atrix \(\mathrm{B} . \mathrm{N}>=0\).

NCVT (input)

The num berof colum ns of them atrix V T.NCV T \(>=0\).

NRU (input)
The num ber of row sof the \(m\) atrix \(U . N R U>=0\).

NCC (input)
The num berof colum ns of the m atrix C. \(\mathrm{NCC}>=0\).

D (input/output)
O n entry, the n diagonalelem ents of the bidiagonal \(m\) atrix \(B\). On exit, if \(\mathbb{N} F O=0\), the singular values ofB in decreasing order.
E (input/output)
O n entry, the elem ents ofE contain the offdiagonalelem ents of the bidiagonalm atrix whose SVD is desired. On norm alexit ( \(\mathbb{N} F O=0\) ), E is destroyed. If the algorithm does not converge ( \(\mathbb{N}\) FO > 0), D and E w ill contain the diagonal and superdiagonal elem ents of a bidiagonalm atrix orthogonally equivalent to the one given as input. E (N) is used forw orkspace.

VT (input/output)
On entry, an N Hy \(-\mathrm{NCVT} m\) atrix VT. On exit, VT is overw rilten by \(\mathrm{P}^{\prime *} \mathrm{VT}\). VT is notreferenced if \(\mathrm{NCVT}=0\).

LDVT (input)
The leading dim ension of the array VT. LDV T >= \(\max (1, N)\) ifncV \(>0 ; L D V T>=1\) ifNCVT \(=0\).

U (input/output)
On entry, an NRU -by-N m atrix U. On exit, \(U\) is overw ritten by \(U * Q\). \(U\) is not referenced if \(N R U\) \(=0\).

LD U (input)
The leading dim ension of the array \(U\). LD U >= max ( \(1, N R U\) ).

C (input/output)
On entry, an N-by -N CC matrix C. On exit, C is overw ritten by \(Q\) '* C . C is not referenced if \(\mathrm{N} C \mathrm{C}\) \(=0\).

LD C (input)
The leading dim ension of the anay C. LD C \(>=\) \(\max (1, N)\) if \(N C C>0 ; L D C>=1\) if \(N C C=0\).

W ORK (w orkspace)
dim ension ( \(4 * N\) )
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0:\) If \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue
> 0: the algorithm did not converge; D and E contain the elem ents of a bidiagonalm atrix which is orthogonally sim ilarto the inputm atrix B; if \(\mathbb{N F O}=\) i, ielem ents ofE have notconverged to zero.

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dbelm m-block Ellipack form atm atrix \(m\) atrix \(m\) ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DBELMM (TRANSA,M B,N,KB,ALPHA,DESCRA,}

* VAL,B\mathbb{NDX,BLDA,MAXBNZ,LB,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER TRANSA,MB,N,KB,DESCRA (5),BLDA,MAXBNZ,LB,
* LDB,LDC,LWORK
INTEGER B\mathbb{NDX (BLDA,MAXBNZ)}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL (LB*LB*BLDA*M AXBNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINEDBELMM_64(TRANSA,M B,N,KB,ALPHA,DESCRA,
* VAL,BINDX,BLDA,MAXBNZ,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER*8 TRANSA,MB,N,KB,DESCRA (5),BLDA,M AXBNZ,LB,
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 B}\mathbb{N}DX(BLDA,MAXBNZ)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL (LB *LB *BLDA *M AXBNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE BELMM (TRANSA,MB, N ],KB,ALPHA,DESCRA,VAL,B\mathbb{NDX,}}\mathbf{N}=,

* BLDA,MAXBNZ,LB,B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])
\mathbb{NTEGER TRANSA,MB,KB,BLDA,MAXBNZ,LB}
INTEGER,D IM ENSION (:) :: DESCRA,B\mathbb{NDX}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D IM ENSION(:) ::VAL
DOUBLE PRECISION ,D IM ENSION (:,:) :: B,C

```
SUBROUTINEBELMM_64(TRANSA, MB, \(\mathbb{N}], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\),

BLDA,MAXBNZ,LB,B,[LDB],BETA,C,[LDC], \(\mathbb{W}\) ORK], [LW ORK])
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,KB,BLDA,MAXBNZ,LB
\(\mathbb{N} T E G E R * 8, D \mathbb{I M} E N S \mathbb{O} N(:):: \quad D E S C R A, B \mathbb{N} D X\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D \(\mathbb{M}\) ENSION (:) ::VAL
DOUBLE PRECISION ,D \(\mathbb{I M}\) ENSION (:, :) :: B,C

\section*{DESCRIPTION}
\[
C \text { <-alpha op (A) B + beta C }
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrioes, \(A\) is a m atrix represented in block Ellpack form at and op(A) is one of
```

op(A )=A or op(A )= A' or op(A )= conjg(A').

```
( 'indicates m atrix transpose)

\section*{ARGUMENTS}
\begin{tabular}{|c|c|}
\hline \multirow[t]{4}{*}{TRANSA} & A Indicates how to operate w ith the sparse m atrix \\
\hline & 0 : operate w ith m atrix \\
\hline & 1 : operate w ith transpose \(m\) atrix \\
\hline & 2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real. \\
\hline M B & N um berofblock row s in m atrix A \\
\hline N & \(N\) um berof colum ns in m atrix C \\
\hline KB & \(N\) um ber ofblock 00 lum ns in m atrix A \\
\hline A LPH A & Scalar param eter \\
\hline \multirow[t]{13}{*}{DESCRA} & () D escriptor argum ent. Five elem ent integer amay \\
\hline & DESCRA (1) m atrix structure \\
\hline & 0 : general \\
\hline & 1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) ) \\
\hline & \(2:\) Herm itian ( \(\mathrm{A}=\mathrm{CONJ}\) ( A ) ) \\
\hline & 3 :Triangular \\
\hline & 4 : Skew (Anti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) ) \\
\hline & 5 :D iagonal \\
\hline & 6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJ}\) ( A ) ) \\
\hline & D ESCRA (2) upper/low er triangular indicator \\
\hline & 1 : low er \\
\hline & 2 :upper \\
\hline & D ESCRA (3) m ain diagonal type \\
\hline
\end{tabular}

0 : non-unit
1 : unit
DESCRA (4) A ray base NOT IM PLEM ENTED )
0 :C C++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL 0 scalar array of length LB *LB *BLDA *M AXBN Z containing \(m\) atrix entries, stored 00 lum \(n-m\) ajorw thin each dense block.
\(B \mathbb{N} D X_{0} \quad\) tw o-dim ensional integerBLD A -by \(-M A X B N Z\) aray such B IND X (i,:) consists of the block colum \(n\) indices of the nonzero blocks in block row i, padded by the integer value i if the num ber of nonzero blocks is less than MAXBNZ.

BLDA leading dim ension of \(\operatorname{INDX(:,:).}\)

M A X BN Z max num berof nonzerosblocks per row .
LB row and colum \(n\) dim ension of the dense blocks com posing VAL.

B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension ofB
BETA Scalarparam eter

C 0 rectangular array w th first dim ension LD C .
LD C leading dim ension of \(C\)
W ORK () scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK aray. LW ORK is not referenced in the cumentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U sers G uide available at:
http://m ath nist.gov/n csd/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)

Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dbelsm -block E llpack form at triangular solve

\section*{SYNOPSIS}
```

SUBROUTINEDBELSM (TRANSA,MB,N,UNITD,DV,ALPHA,DESCRA,

* VAL,BINDX,BLDA,MAXBNZ,LB,
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
$\mathbb{I N} T E G E R$ TRANSA,MB,N,UNITD,DESCRA (5),BLDA,MAXBNZ,LB,
* LDB,LDC,LW ORK
$\mathbb{N} T E G E R \quad B \mathbb{N} D X(B L D A, M A X B N Z)$
DOUBLE PRECISION ALPHA,BETA
D OUBLE PREC ISION DV MB*LB*LB),VAL (LB*LB*BLDA*MAXBNZ), B (LDB,*), C (LDC,*),
* WORK (LW ORK)
SUBROUTINEDBELSM_64(TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,
* VAL,BINDX,BLDA,MAXBNZ,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
$\mathbb{N}$ TEGER*8 TRANSA, M B,N,UN ITD,DESCRA (5), BLDA, MAXBNZ,LB,
* LDB,LDC,LWORK
$\mathbb{N} T E G E R * 8$ B $\mathbb{N} D X(B L D A, M A X B N Z)$
DOUBLE PRECISION ALPHA,BETA
D OUBLE PREC ISION DV M B *LB*LB),VAL (LB*LB*BLDA*MAXBNZ), B (LDB,*), C (LDC,*),
* $\quad$ WORK (LW ORK)

```

\section*{F95 INTERFACE}

SUBROUTINE BELSM (TRANSA, MB, \(\mathbb{N}], \operatorname{UN} I T D, D V, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\),
* BLDA,MAXBNZ,LB,B,[LDB],BETA,C,[LDC], [W ORK], [LW ORK])
\(\mathbb{N} T E G E R\) TRANSA, MB,UNITD, BLDA,MAXBNZ,LB
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \quad D E S C R A, B \mathbb{N} D X\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION ,D \(\mathbb{I M}\) ENSION (:) ::VAL,DV
DOUBLE PRECISION,D \(\mathbb{M}\) ENSION (:, :) :: B,C

SUBROUT \(\mathbb{N} E \operatorname{BELSM}\) _64 (TRANSA, M B, \(\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\),
* \(B L D A, M A X B N Z, L B, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,UNITD, BLDA, MAXBNZ,LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: D E S C R A, B \mathbb{N} D X\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION ,D \(\mathbb{M}\) ENSION (:) ::VAL, DV
DOUBLE PRECISION ,D \(\mathbb{M}\) ENSION (: :) :: B , C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { op (A) B + BETA } C \quad C<-A L P H A D \text { op (A) B + BETA C } \\
& C<-A L P H A \text { Op (A)D B + BETA } C
\end{aligned}
\]
where ALPHA and BETA are scalar, \(C\) and \(B\) are m by \(n\) dense \(m\) atrices, \(D\) is a block diagonalm atrix, \(A\) is a unit, ornon-unit, upperor low ertriangularm atrix represented in block Elhoack form at and op (A ) is one of op (A) \()=\operatorname{inv}(A)\) or op (A \()=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}\left(\infty n \dot{g}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix 0 : operate \(w\) ith \(m\) atrix 1 : operate \(w\) th transpose \(m\) atrix 2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(s\) in \(m\) atrix \(A\)

N \(\quad\) Num berof colum ns in matrix C

UN ITD Type of scaling:
1 : Identity matrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)

DV () A may of the length M B *LB *LB consisting of the block entries ofblock diagonalm atrix \(D\) w here each block is stored in standard colum n-m ajor form .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general

1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2: Herm itian ( \(A=\operatorname{CONJ}(A)\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) )
5 :D iagonal
6 : Skew Herm titian ( \(A=-\operatorname{CON}\) J ( \(A\) ) )
N ote: For the routine, D ESCRA (1)=3 is only supported.
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 : upper
DESCRA (3) m ain diagonal type
0 : non-identity blocks on the \(m\) ain diagonal
1 : identity diagonalblocks
2 : diagonalblocks are dense \(m\) atrices
DESCRA (4) A ray base \(\mathbb{N O T} \mathbb{M}\) PLEM ENTED )
0 : C C C+ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? \(\mathbb{N} O T \mathbb{M}\) PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () scalar array of length LB *LB *BLD A *M A X BN Z containing \(m\) atrix entries, stored colum \(n-m\) ajorw thin each dense block.

B \(\mathbb{N}\) D X () tw o-dim ensionalintegerB LD A boy-M A X BN Z array such B IND X ( \(i\), : ) consists of the block colum \(n\) indices of the nonzero blocks in block row i, padded by the integer value iif the num ber ofnonzero blocks is less than M A X BN Z. The block colum \(n\) indioesM U ST be sorted in increasing order foreach block row.

BLDA leading dim ension ofB INDX (:,:).

M AXBNZ max num berofnonzerosblocks per row .
LB row and colum \(n\) dim ension of the dense blocks com posing A.

B 0 rectangular array w ith first dim ension LD B .

LD B leading dim ension of \(B\)

BETA Scalarparam eter

C 0 rectangular aray w ith first dim ension LD C .

LD C leading dim ension of C

W ORK () scratch array of length LW ORK.
On exit, ifLW ORK \(=-1, W\) ORK (1) retums the minim um
size ofLW ORK.

LW ORK length ofW ORK anay.LW ORK should be at least M B *LB.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=\mathrm{M} \mathrm{B} * \mathrm{LB} * \mathrm{~N}\) _CPU \(S\) where \(\mathrm{N} \_\)CPUS is the \(m\) axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W O RK array, and no errorm essage related to LW ORK is issued by XERBLA .

\section*{SEE ALSO}

\section*{N IST FO RTRA N Sparse B las U ser's G uide available at:} http:/m ath nist.gov/m cso/Staff/K Rem ington/Espblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib org/utk/papers/sparse _ps

\section*{NOTES /BUGS}
1.N o test for singularity ornear-singularity is included in this routine. Such tests m ust.be perform ed before calling this routine.
2. If \(D E S C R A(3)=0\), the low er or upper triangularpart of each diagonalblock is used by the routine depending on DESCRA (2) .
3. If \(D E S C R A(3)=1\), the unitdiagonalblocksm ightorm ight notbe referenced in the B EL representation of a sparse \(m\) atrix. They are notused anyw ay .
4. If \(D E S C R A(3)=2\), diagonalblocks are considered as dense m atrices and the LU factorization w ith partialpivoting is used by the routine. WORK (1)=0 on retum if the factorization for alldiagonalblocks has been com pleted successfully, otherw ise WORK (1) = -iw here i is the block num ber forw hich the LU factorization could notbe com puted.
5.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse
m atrix \(A\) is used. H ow erver DESCRA (1) m ust.be equalto 3 in this case.

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dbsam m -block sparse colum n m atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUTINEDBSCMM(TRANSA,M B,N,KB,A LPHA,DESCRA,

* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,MB,N,KB,DESCRA (5),LB,}
* LDB,LDC,LW ORK
\mathbb{NTEGER BINDX (BNNZ),BPNTRB (KB),BPNTRE (KB)}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL (LB *LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINEDBSCMM_64(TRANSA,M B,N,KB,ALPHA,DESCRA,
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER*8 TRANSA,MB,N,KB,DESCRA (5),LB,
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 B\mathbb{NDX (BNNZ),BPNTRB (KB),BPNTRE (KB)}}\mathbf{(K)}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL (LB *LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
where:BNNZ = BPNTRE (KB)-BPNTRB (1)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE BSCMM (TRANSA,M B, N ],KB,ALPHA,DESCRA,VAL,B INDX,}

* BPNTRB,BPNTRE,LB,B,[LD B ],BETA,C,[LDC], [W ORK ], [LW ORK ])
INTEGER TRANSA,MB,KB,LB
\mathbb{NTEGER,D IM ENSION (:) :: DESCRA,B NNDX,BPNTRB,BPNTRE}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,DIM ENSION(:) ::VAL
DOUBLE PRECISION,D IM ENSION (:,:) :: B,C

```

SUBROUT \(\mathbb{N} E \operatorname{BSCM} M \_64(T R A N S A, M B, \mathbb{N}], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\),
* BPNTRB,BPNTRE,LB,B, [LDB],BETA,C, [LDC], [WORK], [LWORK])
\(\mathbb{N}\) TEGER*8 TRANSA, MB, KB,LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: \quad \mathrm{DESCRA}, \mathrm{B} \mathbb{N} D \mathrm{X}, \mathrm{BPNTRB}, \mathrm{BPN} T R E\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,DIMENSION (:) ::VAL
D OUBLE PRECISION ,D \(\mathbb{M}\) ENSION (: : : : : B , C

\section*{DESCRIPTION}
C <-alpha op (A ) B + beta C
where A LPHA andBETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, \(A\) is a m atrix represented in block sparse colum n form at and op (A ) is one of \(o p(A)=A \quad\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix.
2 is equivalentto 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(s\) in matrix A

N \(\quad \mathrm{N}\) um berof colum ns in \(m\) atrix \(C\)

K B \(\quad\) Num ber ofblock colum ns in m atrix A

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(A=A\) )
2: Herm Itian ( \(A=C O N J(A)\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(A=-C O N J\) ( \(A\) ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper

DESCRA (3) m ain diagonaltype
0 : non-unit
1 :unit
DESCRA (4) A ray base NOT \(\mathbb{M}\) PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 :unknown
1 : no repeated indices
VAL () scalar array of length \(\mathrm{LB} * \mathrm{LB} * \mathrm{BNN} Z\) consisting of the block entries stored collm n-m ajorw thin each dense block .
\(B \operatorname{IND}\) X (integer array of length BNNZ consisting of the block row indioes of the block entries ofA .

BPN TRB 0 integer aray of length \(K B\) such that BPN TRB (J) BPN TRB (1)+1 points to location in B IND X of the firstblock entry of the J-th block colum n of A.
BPNTRE ( integeramay of length \(K B\) such that BPN TRE (J) BPN TRB (1) points to location in B IN D X of the last.block entry of the J-th block colum n of A.

LB dim ension of dense blocks com posing A.
B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension of \(B\)
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension ofC
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the current version.

LW ORK length ofW ORK array. LW ORK is notreferenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U sers G uide available at:
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse .ps

\section*{NOTES/BUGS}

It is know \(n\) that there exists another representation of the block sparse colum \(n\) form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each block colum \(n\) in the arrays VAL and B INDX is used instead of tw o arrays BPN TRB and BPN TRE. To use the routine \(w\) ith this kind ofblock sparse colum \(n\) form at the follow ing calling sequence should be used

CALL SBSCMM (TRANSA, MB,N,KB,ALPHA,DESCRA, * \(\quad V A L, B \mathbb{N D}, \mathbb{A}, \mathbb{A}(2), L B\), * B,LDB,BETA, C,LDC,WORK,LWORK )

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dbscsm -block sparse colum n form at triangular solve

\section*{SYNOPSIS}

SUBROUTINEDBSCSM (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
\(\mathbb{N} T E G E R\) TRANSA,MB,N,UNITD,DESCRA (5), LB,
* LDB,LDC,LWORK
\(\mathbb{N} T E G E R \quad B \mathbb{N} D X(B N N Z)\), BPNTRB \(M B)\), BPNTRE \(M\) B)
DOUBLE PRECISION ALPHA, BETA
DOUBLE PRECISION DV MB*LB*LB),VAL (LB*LB*BNNZ),B(LDB,*),C(LDC,*),WORK
(LW ORK)
SUBROUTINEDBSCSM_64(TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA, * VAL,BINDX,BPNTRB,BPNTRE,LB, * B,LDB,BETA, C,LDC,WORK,LWORK) \(\mathbb{N} T E G E R * 8\) TRANSA, MB,N,UNITD,DESCRA (5), LB, * LDB,LDC,LW ORK \(\mathbb{N} T E G E R * 8 \operatorname{B} \mathbb{N} D X(B N N Z), B P N T R B(M B)\), BPNTRE \(M B\) ) DOUBLE PRECISION ALPHA,BETA DOUBLE PRECISION DV MB*LB*LB),VAL (LB*LB*BNNZ),B(LDB,*),C(LDC,*),WORK (LW ORK)
where: \(\mathrm{BNNZ}=\mathrm{BPNTRE}\) M B)-BPNTRB (1)

\section*{F95 INTERFACE}

SUBROUTINEBSCSM (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,VAL,B \(\mathbb{N} D X\), * BPNTRB,BPNTRE,LB,B,[LDB],BETA,C,[LDC],[WORK],[LWORK])
\(\mathbb{N} T E G E R\) TRANSA, MB,N,UNTID,LB
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: D E S C R A, B \mathbb{N} D X, B P N T R B, B P N T R E\)
D OUBLE PRECISION ALPHA, BETA

DOUBLEPRECISION,D \(\mathbb{M} E N S I O N(:):: V A L, D V\)
DOUBLE PRECISION ,D IM ENSION (:, :) :: B , C

SU BROUTINE BSCSM_64 (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,VAL,BINDX,
* BPNTRB,BPNTRE,LB,B, [LDB],BETA,C, [LDC], [W ORK], [LWORK])
\(\mathbb{N}\) TEGER*8 TRANSA, M B , N , UNITD , LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{I} N(:):: D E S C R A, B \mathbb{N} D X, B P N T R B, B P N T R E\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D \(\mathbb{M} E N S I O N(:):: V A L, D V\)
DOUBLE PRECISION ,D IM ENSION (:, :) :: B , C

\section*{DESCRIPTION}
```

C<-ALPHA OP(A)B + BETA C C <-ALPHA D Op (A)B + BETA C
C<-ALPHA Op(A)D B + BETA C

```
where ALPHA and BETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense \(m\) atrices, \(D\) is a block diagonalm atrix, \(A\) is a unit, ornon-unit, upperor low er triangularm atrix represented in block sparse colum n form at and op (A ) is one of op (A) \()=\operatorname{inv}(A)\) or op (A \()=\operatorname{inv}(A)\) or op (A) \()=\operatorname{inv}\left(\operatorname{cong}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRA NSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate w ith m atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(s\) in \(m\) atrix \(A\)
\(N \quad N\) um berof colum ns in matrix C

UN ITD Type of scaling:
1 : Identity \(m\) atrix (argum entD V [] is ignored)
2 : Scale on left (row block scaling)
3 : Scale on right (colum n block scaling)

DV () A nray of the length M B *LB *LB consisting of the block entries ofblock diagonalm atrix \(D\) where each block is stored in standard colum \(n-m\) ajor form .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Fíve elem ent integer anay

DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
\(2:\) Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) )
3 :Triangular
4 : Skew (Anti)-Symm etric ( \(A=-A\) )
5 : D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\operatorname{CONJ}\) (A))
N ote:For the routine, DESCRA \((1)=3\) is only supported.

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 : non-identity blocks on the \(m\) ain diagonal
1 : identity diagonalblocks
2 : diagonalblocks are densem atrices
DESCRA (4) A ray base (NOT IM PLEM ENTED)
0 : C C + + com patible
1 :Fortran com patible
D ESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL 0 scalar array of length LB *LB *BNN Z consisting of the block entries stored colum \(n-m\) ajorw thin each dense block.
\(B \mathbb{N} D X 0 \quad\) integer amray of length BNNZ consisting of the block row indices of the block entries ofA.
The block row indicesM U ST be sorted
in increasing order foreach block colum \(n\).
BPNTRB () integer array of length \(M B\) such that BPN TRB (J) BPN TRB (1)+1 points to location in B IND X of the first.block entry of the J-th block colum n of A.

BPNTRE ( integer anay of length M B such that
BPN TRE (J)-BPN TRB (1) points to location in B IND X of the last.block entry of the Jth block colum n of A .

LB dim ension of dense blocks com posing A.
B 0 rectangular array w ith firstdin ension LD B.

LD B leading din ension ofB
BETA Scalarparam eter
C 0 rectangular array with first dim ension LD C .

LD C leading dim ension of \(C\)

W ORK () scratch array of length LW ORK.
On exit, if LW ORK \(=-1\), W ORK (1) retums the optim um size ofLW ORK.

LW ORK length ofW ORK array.LW ORK should be at least M B*LB.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK >=M B *LB*N_CPU \(S\) where N_CPU \(S\) is the \(m\) axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/n cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.
2. If \(D E S C R A\) (3)=0, the low erorupper triangularpart of each diagonalblock is used by the routine depending on DESCRA (2).
3. If \(D E S C R A\) (3)=1, the unitdiagonalblocksm ightorm ight not.be referenced in the BSC representation of a sparse \(m\) atrix. They are notused anyw ay.
4. If \(D E S C R A\) (3)=2, diagonalblocks are considered as dense \(m\) atrices and the LU factorization \(w\) ith partialpivoting is
used by the routine. WORK (1)=0 on retum if the factorization foralldiagonalblocks has been com pleted successfully, otherw ise WORK (1) = -iw here i is the block num ber forw hich the LU factorization could notbe com puted.
5.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse \(m\) atrix A is used. H ow erverDESCRA (1) m ustbe equalto 3 in this case.

6 . It is know \(n\) that there exists another representation of the block sparse colum n form at (see for exam ple Y Saad, "Tterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each block colum \(n\) in the arrays VAL and B \(\mathbb{N D} D\) is used instead of tw o arrays BPN TRB and BPN TRE. To use the routine \(w\) ith this kind ofblock sparse colum \(n\) form at the follow ing calling sequence should be used

CALL SBSCSM (TRANSA,MB,N,UNITD,DV,ALPHA,DESCRA, * \(\quad V A L, B \mathbb{N} D X, \mathbb{A}, \mathbb{A}(2), L B\),
* B,LDB,BETA, C,LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dbsmm -block sparse row form atm atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUTINEDBSRMM(TRANSA,M B,N,KB,A LPHA,DESCRA,

* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER TRANSA,MB,N,KB,DESCRA (5),LB,
* LDB,LDC,LW ORK
\mathbb{NTEGER BINDX (BNNZ),BPNTRB MB),BPNTRE MB)}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL (LB *LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
SUBROUTINEDBSRMM_64(TRANSA, MB,N,KB,ALPHA,DESCRA,
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,WORK,LWORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,N,KB,DESCRA (5),LB,
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R * 8 \operatorname{B} \mathbb{N} D X(B N N Z), B P N T R B(M B)\), BPNTRE \(M B)\)
DOUBLE PRECISION ALPHA, BETA
D OUBLE PRECISION VAL (LB *LB*BNNZ), B (LDB,*), C (LDC, \(\left.{ }^{\star}\right)\), W ORK (LW ORK)
where: BNNZ = BPNTRE M B)-BPNTRB (1)

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE BSRMM (TRANSA,M B, N ],KB,ALPHA,DESCRA,VAL,B INDX,}

* BPNTRB,BPNTRE,LB,B,[LDB],BETA,C,[LDC], [W ORK], [LW ORK ])
INTEGER TRANSA,MB,KB,LB
\mathbb{NTEGER,D IM ENSION (:) :: DESCRA,B NNDX,BPNTRB,BPNTRE}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D IM ENSION(:) ::VAL
DOUBLE PRECISION,D IM ENSION (:,:) :: B,C

```

SUBROUT \(\mathbb{N} E \operatorname{BSRM} M \_64(T R A N S A, M B, \mathbb{N}], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\),
* BPNTRB,BPNTRE,LB,B, [LDB],BETA,C, [LDC], [W ORK], [LWORK])
\(\mathbb{N}\) TEGER*8 TRANSA, MB, KB,LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: \quad \mathrm{DESCRA}, \mathrm{B} \mathbb{N} D \mathrm{X}, \mathrm{BPNTRB}, \mathrm{BPN} T R E\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,DIMENSION (:) ::VAL
D OUBLE PRECISION ,D \(\mathbb{M}\) ENSION (: : : : : B , C

\section*{DESCRIPTION}
C <-alpha op (A ) B + beta C
where A LPH A and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, \(A\) is a m atrix represented in block sparse row form at and op (A ) is one of \(o p(A)=A \quad\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix \(A\) is real.

M B \(\quad\) Num ber ofblock row \(s\) in matrix A

N \(\quad\) Num berof colum ns in matrix C

K B \(\quad\) Number ofblock colum ns in m atrix A

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(A=A\) )
2: Herm Itian ( \(A=C O N J(A)\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(A=-C O N J\) ( \(A\) ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper

DESCRA (3) main diagonal type
0 : non-unit
1 :unit
DESCRA (4) A ray base NOT \(\mathbb{M}\) PLEM ENTED)
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () scalar array of length LB *LB*BNNZ consisting of the block entries stored colum n-m ajorw thin each dense block .
\(B \operatorname{IND}\) X (integer array of length BNNZ consisting of the block colum n indices of the block entries of A.

BPN TRB () integeramay of length \(M B\) such that BPN TRB (J) BPN TRB (1)+1 points to location in B IND X of the first.block entry of the \(J\)-th block row of A.
BPN TRE () integer array of length \(M B\) such that BPN TRE (J)-BPN TRB (1) points to location in B IND X of the lastblock entry of the \(J\) th block row ofA.

LB dim ension of dense blocks com posing A.
B 0 rectangular array w th first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w th first dim ension LD C.
LD C leading dim ension of \(C\)
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the currentversion.

LW ORK length ofW ORK array. LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B lasU ser's G uide available at:
htep://m ath nist.gov/m csd/Staff/k Rem ington/Aspblas/
"D ocum ent forthe B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse дps

\section*{NOTES /BUGS}

It is know \(n\) that there exists another representation of the block sparse row form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem s",W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. The \(m\) ain difference is that only one array, \(\mathbb{A}\), containing the pointers to the beginning of each block row in the amays \(V A L\) and \(B \mathbb{N D X}\) is used instead of tw o arrays BPN TRB and BPN TRE. To use the routine w ith this kind ofblock sparse row form at the follow ing calling sequence should be used

CALL SBSRMM (TRANSA, MB,N,KB,ALPHA,DESCRA, * \(V A L, B \mathbb{N} D X, \mathbb{I}, \mathbb{I A}(2), L B\),
* \(\quad \mathrm{B}, \mathrm{LD} B, B E T A, C, L D C, W\) ORK,LWORK )

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dbsrsm -block sparse row form at triangular solve

\section*{SYNOPSIS}

SUBROUTINEDBSRSM (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* \(\quad \mathrm{B}, \mathrm{LD} B, B E T A, C, L D C, W\) ORK,LWORK)
\(\mathbb{N} T E G E R\) TRANSA,MB,N,UNITD,DESCRA (5), LB,
* LDB,LDC,LWORK
\(\mathbb{N} T E G E R \quad B \mathbb{N} D X(B N N Z), B P N T R B(M B)\),BPNTRE \(M\) B)
DOUBLE PRECISION ALPHA, BETA
DOUBLE PRECISION DV MB*LB*LB),VAL (LB*LB*BNNZ),B(LDB,*),C(LDC,*),WORK
(LW ORK)

SUBROUTINEDBSRSM_64(TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA, * VAL,BINDX,BPNTRB,BPNTRE,LB, * B,LDB,BETA, C,LDC,WORK,LW ORK) \(\mathbb{N} T E G E R * 8\) TRANSA, MB,N,UNITD,DESCRA (5), LB, * LDB,LDC,LWORK \(\mathbb{N} T E G E R * 8 \operatorname{B} \mathbb{N} D X(B N N Z), B P N T R B(M B), B P N T R E M B)\) DOUBLE PRECISION ALPHA, BETA DOUBLE PRECISION DV M B *LB*LB) ,VAL (LB*LB*BNNZ), B (LDB,*), C (LDC,*), WORK (LW ORK)
where: \(\operatorname{BNN} Z=B P N T R E M B)-B P N T R B(1)\)

\section*{F95 INTERFACE}

SUBROUTINEBSRSM (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,VAL,B \(\mathbb{N} D X\), * BPNTRB,BPNTRE,LB,B,[LDB],BETA,C,[LDC],[WORK],[LWORK])
\(\mathbb{N} T E G E R\) TRANSA,MB,N,UNITD,LB
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: D E S C R A, B \mathbb{N} D X, B P N T R B, B P N T R E\)
DOUBLE PRECISION ALPHA,BETA

DOUBLEPRECISION,D \(\mathbb{M} E N S I O N(:):: V A L, D V\)
DOUBLE PRECISION ,D IM ENSION (:, :) :: B , C

SU BROUTINE BSRSM _64 (TRANSA, MB,N, UN ITD, DV, ALPHA,DESCRA, VAL, B \(\mathbb{N} D X\),
* BPNTRB,BPNTRE,LB,B, [LDB],BETA,C, [LDC], [W ORK], [LWORK])
\(\mathbb{N}\) TEGER*8 TRANSA, M B , N , UNITD , LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{I} N(:):: D E S C R A, B \mathbb{N} D X, B P N T R B, B P N T R E\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D \(\mathbb{M} E N S I O N(:):: V A L, D V\)
DOUBLE PRECISION ,D IM ENSION (:, :) :: B , C

\section*{DESCRIPTION}
```

C<-ALPHA OP(A)B + BETA C C <-ALPHA D Op (A)B + BETA C
C<-ALPHA Op(A)D B + BETA C

```
where A LPHA andBETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense \(m\) atrices, \(D\) is a block diagonalm atrix, \(A\) is a unit, ornon-unit, upperor low er triangularm atrix represented in block sparse row form at form atand op (A ) is one of
op \((A)=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}\left(\operatorname{An} \bar{g}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix
0 : operate w ith m atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(s\) in \(m\) atrix \(A\)
\(N \quad N\) um berof colum ns in matrix C

UN ITD Type of scaling:
1 : Identity \(m\) atrix (argum entD V [] is ignored)
2 : Scale on left (row block scaling)
3 : Scale on right (colum n block scaling)

DV () A mray of the length M B *LB *LB consisting of the block entries ofblock diagonalm atrix \(D\) where each block is stored in standard colum \(n-m\) ajor form .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Fíve elem ent integer anay

DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm tian ( \(\mathrm{A}=\mathrm{CONJG}\) (A))
3 :Triangular
4 : Skew (Anti)-Symm etric ( \(A=-A\) )
5 : D iagonal
6 : Skew Herm Hian ( \(\mathrm{A}=-\mathrm{CONJG}(\mathrm{A})\) )
N ote:For the routine, DESCRA \((1)=3\) is only supported.

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 : non-identity blocks on the \(m\) ain diagonal
1 : identity diagonalblocks
2 : diagonalblocks are densem atrices
DESCRA (4) A ray base (NOT IM PLEM ENTED)
0 : C C + + com patible
1 :Fortran com patible
D ESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL 0 scalar array of length LB *LB *BNN Z consisting of the block entries stored colum \(n-m\) ajorw thin each dense block.
\(B \mathbb{N} D X 0 \quad\) integer amray of length BNNZ consisting of the block collum \(n\) indiges of the block entries of A.
The block colum \(n\) indices M U ST be sorted in increasing order foreach block row .

BPNTRB () integeramay of length \(M B\) such that BPN TRB (J) BPN TRB (1)+1 points to location in B IND X of the firstblock entry of the J-th block row of A.

BPN TRE () integer aray of length \(M B\) such that BPN TRE (J)-BPN TRB (1) points to location in B IND X of the lastblock entry of the \(J\) th block row of A.

LB dim ension of dense blocks com posing A.
B 0 rectangular anray with firstdim ension LD B .

LD B leading din ension ofB
BETA Scalarparam eter
C 0 rectangular array with first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch array of length LW ORK.
On exit, if LW ORK \(=-1\), W ORK (1) retums the optim um size ofLW ORK.

LW ORK length ofW ORK array.LW ORK should be at least M B*LB.

Forgood perform ance, LW O RK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK >=M B *LB*N_CPU \(S\) where N_CPU \(S\) is the \(m\) axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/n cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.
2. If \(D E S C R A\) (3)=0, the low erorupper triangularpart of each diagonalblock is used by the routine depending on DESCRA (2).
3. If \(D E S C R A\) (3)=1, the unitdiagonalblocksm ightorm ight not.be referenced in the \(B S R\) representation of a sparse \(m\) atrix. They are notused anyw ay.
4. If \(D E S C R A\) (3)=2, diagonalblocks are considered as dense \(m\) atrices and the LU factorization \(w\) ith partialpivoting is
used by the routine. WORK (1)=0 on retum if the factorization foralldiagonalblocks has been com pleted successfully, otherw ise WORK (1) = -iw here i is the block num ber forw hich the LU factorization could notbe com puted.
5.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse \(m\) atrix A is used. H ow erverDESCRA (1) m ustbe equalto 3 in this case.

6 . It is know \(n\) that there exists another representation of the block sparse row form at (see forexam ple Y Saad, "Tterative M ethods forSparse LinearSystem s",W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each block row in the arrays VAL and B \(\mathbb{N} D X\) is used instead of tw o arrays BPN TRB and BPN TRE. To use the routine \(w\) ith this kind ofblock sparse row form at the follow ing calling sequence should be used

CALL DBSRSM (TRANSA,MB,N,UNITD,DV,ALPHA,DESCRA, * \(V A L, B \mathbb{N} D X, \mathbb{A}, \mathbb{A}(2), L B\),
* B,LDB,BETA, C,LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- NOTES

\section*{NAME}
dcnvcor-com pute the convolution or correlation of real
vectors

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DCNVCOR (CNVCOR,FOUR,NX,X,\mathbb{FX,}\mathbb{NCX,NY,NPRE,M,Y,}}\mathbf{N},\textrm{N},\textrm{N}
\mathbb{F}Y,\mathbb{NC}C1Y,\mathbb{N}C2Y,NZ,K,Z,\mathbb{FZ},\mathbb{N}C1Z,\mathbb{NC}2Z,WORK,LW ORK)
CHARACTER * 1 CNVCOR,FOUR

```

```

K,\mathbb{FZ,}\mathbb{N}C1Z,\mathbb{NC2Z,LW ORK}
DOUBLE PRECISION X (*),Y (*),Z (*),W ORK (*)
SUBROUT\mathbb{NE DCNVCOR_64 (CNVCOR,FOUR,NX,X,\mathbb{FX,}\mathbb{NCX,NY,NPRE,M,Y,}}\mathbf{N},

```

```

CHARACTER * 1 CNVCOR,FOUR

```

```

K,\mathbb{FZ,INC1Z,INC2Z,LW ORK}
DOUBLE PRECISION X (*),Y (*),Z (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE CNVCOR (CNVCOR,FOUR, \(\mathbb{N} X], X, \mathbb{F X},[\mathbb{N C X}], N Y, N P R E, M, Y\), \(\mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N} C 2 Y, N Z, K, Z, \mathbb{F} Z, \mathbb{N C} 1 Z, \mathbb{N C} 2 Z, W\) ORK, (LW ORK ])

CHARACTER (LEN=1) ::CNVCOR,FOUR
\(\mathbb{N} T E G E R:: N X, \mathbb{F} X, \mathbb{N} C X, N Y, N P R E, M, \mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N} C 2 Y\),
\(\mathrm{NZ}, \mathrm{K}, \mathbb{F} Z, \mathbb{N} \mathrm{C} 1 \mathrm{Z}, \mathbb{I N C} 2 \mathrm{Z}, \mathrm{LW}\) ORK
REAL (8),D \(\mathbb{M}\) ENSION (:) :: X,Y,Z,W ORK

SU BROUTINE CNVCOR_64 (CNVCOR,FOUR, \(\mathbb{N} X], X, \mathbb{F X},[\mathbb{N} C X], N Y, N P R E, M\),

CHARACTER (LEN=1) ::CNVCOR,FOUR
\(\mathbb{N} \operatorname{TEGER}(8):: N X, \mathbb{F X}, \mathbb{N} C X, N Y, N P R E, M, \mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N} C 2 Y\), \(\mathrm{N} Z, \mathrm{~K}, \mathbb{F} Z, \mathbb{N} \mathrm{C} 1 \mathrm{Z}, \mathbb{N} \mathrm{C} 2 \mathrm{Z}, \mathrm{LW}\) ORK
REAL (8),D \(\mathbb{M}\) ENSION (:) ::X,Y,Z,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void denvcor(char cnvcor, char four, intnx, double *x, int ifx, int incx, intny, int npre, intm, double * \(y\), intify, int incly, int inc2y, intnz, intk, double \({ }^{z}\), int ify, int inclz, intinc \(2 z\), double \(*_{\text {w ork, }}\) int lw ork);
void denvcor_64 (char cnvcor, char four, long nx, double *x, long ifx, long incx, long ny, long npre, long m, double *y, long ify, long inc1y, long inc \(2 y\), long \(n z\), long \(k\), double *z, long ifz, long inc1z, long inc2z, double *w ork, long lw ork);

\section*{PURPOSE}
dcnvcorcom putes the convolution or comelation of realvectors.

\section*{ARGUMENTS}

CNVCOR (input)
V 'or \(\mathrm{V}^{\prime}\) if convolution is desired, R ' or 'r' if comelation is desired.

FOUR (input)
T 'or t'if the Fourier transform \(m\) ethod is to be used, D 'or d'if the com putation should be done directly from the definition. The Fourier transform m ethod is generally faster, but itm ay introduce noticeable errors into certain results, notably w hen both the filter and data vectors consistentirely of integers or vectors where ele\(m\) ents ofeither the filtervector or a given data vectordiffer significantly in \(m\) agnitude from the 1 -norm of the vector.

NX (input)
Length of the filtervector. \(\mathrm{NX}>=0\). DCNVCOR
w ill retum im m ediately if \(\mathrm{NX}=0\).

X (input)
Filtervector.

IFX (input)
Index of the firstelem entofX. \(\mathrm{NX}>=\mathbb{F} \mathrm{X}>=1\).
\(\mathbb{N} C X\) (input)
Stride betw een elem ents of the filtervector in X . \(\mathbb{N} C X>0\).

NY (input)
Length of the inputvectors. NY \(>=0\). DCNVCOR w ill retum im m ediately if \(\mathrm{N} Y=0\).
NPRE (input)
The num ber of im plicit zeros prepended to the \(Y\) vectors. NPRE \(>=0\).

M (input)
Num ber of inputvectors. \(\mathrm{M}>=0\). DCNVCOR will retum imm ediately if \(M=0\).

Y (input)
Inputvectors.

IFY (input)
Index of the firstelem entof \(. \mathrm{NY}>=\mathbb{F} Y>=1\).
\(\mathbb{N} C 1 Y\) (input)
Stride betw een elem ents of the inputvectors in Y.
\(\mathbb{I N C} 1 \mathrm{Y}>0\).
\(\mathbb{I N C} 2 Y\) (input)
Stride betw een the inputvectors in \(Y . \mathbb{N} C 2 Y>0\).

N Z (input)
Length of the outputvectors. NZ \(>=0\). DCNVCOR w ill retum im m ediately if \(\mathrm{NZ}=0\). See the N otes section below for inform ation abouthow this argu\(m\) ent interacts \(w\) ith \(N X\) and \(N Y\) to control circular versus end-off shifting.

K (input)
N um berof \(Z\) vectors. \(\mathrm{K}>=0\). If \(\mathrm{K}=0\) then DCNVCOR will retum immediately. If \(K<M\) then only the firstK inputvectors \(w\) ill.be processed. If \(K>M\) then \(M\) inputvectors \(w\) ill.be processed.

Z (output)
Resultvectors.

FZ (input)
Index of the firstelem entofZ. NZ >= \(\mathbb{F Z}\) >=1.
\(\mathbb{N} C 1 Z\) (input)
Stride betw een elem ents of the output vectors in Z. \(\mathbb{N} C 1 Z>0\).
\(\mathbb{N} C 2 Z\) (input)
Stride betw een the outputvectors in Z. \(\mathbb{N N}\) C 2 Z > 0 .

W ORK (input/output)
Scratch space. Before the first call to DCNVCOR w th particular values of the integer argum ents the firstelem entofW ORK mustbe set to zero. If W ORK is written betw een calls to DCNVCOR or if DCNVCOR is called w ith different values of the integer argum ents then the firstelem entofW ORK \(m\) ustagain be setto zero before each call. If W ORK has notbeen w rilten and the sam e values of the integer argum ents are used then the firstelem entofW ORK to zero. This can avoid certain initializations that store their results into \(W\) ORK, and avoiding the initialization can \(m\) akeD CNVCOR nun faster.

LW ORK (input)
Length ofW ORK. LW ORK >= 4*M AX NX NY,NZ)+15.

\section*{NOTES}

If any vector overlaps a w ritable vector, eitherbecause of argum ent aliasing or ill-chosen values of the various \(\mathbb{I N} C\) argum ents, the results are undefined and \(m\) ay vary from one nun to the next.

Them ost com \(m\) on form of the com putation, and the case that executes fastest, is applying a filtervectorX to a series of vectors stored in the colum ns of \(Y\) w ith the resultplaced into the colum nsof \(Z\). In that case, \(\mathbb{N} C X=1, \mathbb{N} C 1 Y=1\), \(\mathbb{N} C 2 Y>=N Y, \mathbb{N} C 1 Z=1, \mathbb{N} C 2 Z>=N Z\). A nothercomm on form is applying a filtervectorX to a series of vectors stored in the row sof \(Y\) and store the result in the row of \(Z\), in which case \(\mathbb{N} C X=1, \mathbb{N} C 1 Y>=N Y, \mathbb{N} C 2 Y=1, \mathbb{N} C 1 Z>=N Z\), and \(\mathbb{N} C 2 Z=1\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dcnvcor2 - com pute the convolution or comelation of real
m atrices

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DCNVCOR2 (CNVCOR,M ETHOD,TRANSX,SCRATCHX,TRANSY,}
SCRATCHY,M X,NX,X,LDX,MY,NY,M PRE,NPRE,Y,LDY,M Z,NZ,Z,
LD Z,W ORK NN,LW ORK)
CHARACTER * 1 CNVCOR,M ETHOD,TRANSX, SCRATCHX, TRANSY,
SCRATCHY
DOUBLE COM PLEX W ORK\mathbb{N (*)}
\mathbb{N TEGER M X,NX,LDX,M Y,NY,M PRE,NPRE,LDY,M Z,NZ, LD Z,}
LW ORK
DOUBLE PRECISION X (LDX,*),Y (LDY,*),Z (LD Z,*)
SU BROUTINE DCNVCOR2_64 CNVCOR,M ETHOD,TRANSX,SCRATCHX,TRANSY,
SCRATCHY,M X,NX,X,LDX,MY,NY,M PRE,NPRE,Y,LDY,M Z,NZ,Z,
LD Z,W ORK\mathbb{N,LW ORK)}
CHARACTER * 1 CNVCOR,M ETHOD,TRANSX, SCRATCHX, TRANSY,
SCRATCHY
DOUBLE COM PLEX W ORK \mathbb{N (*)}
\mathbb{NTEGER*8M X,NX,LDX,M Y,NY,M PRE,NPRE,LDY,M Z,NZ,LD Z,}
LW ORK
DOUBLE PRECISION X (LDX,\star),Y (LDY,\star),Z (LD Z,*)

```

\section*{F95 INTERFACE}

SU BROUTINE CNVCOR2 (CNVCOR,METHOD,TRANSX, SCRATCHX,TRANSY, SCRATCHY, MX], NX],X,[LDX], MY], \(\mathbb{N} Y\) ],MPRE,NPRE, Y, [LDY], \(\mathbb{M} Z], \mathbb{N} Z], Z,[L D Z], W\) ORK \(\mathbb{N},[L W O R K])\)

CHARACTER (LEN=1) :: CNVCOR, METHOD, TRANSX, SCRATCHX,
TRANSY,SCRATCHY
COMPLEX (8),D \(\mathbb{I M} E N S I O N(:):: W\) ORK \(\mathbb{N}\)
\(\mathbb{N} T E G E R:: M X, N X, L D X, M Y, N Y, M P R E, N P R E, L D Y, M Z, N Z\),
LD Z,LW ORK
REAL (8),D IM ENSIO N (:,:) ::X,Y,Z

SU BROUTINE CNVCOR2_64 CNVCOR,METHOD,TRANSX,SCRATCHX,TRANSY, SCRATCHY, MX], NX],X, [LDX], M Y ], NY],MPRE,NPRE,Y, [LDY], \(\mathbb{M} Z], \mathbb{N} Z], Z,[L D Z], W\) ORK \(\mathbb{N},[L W O R K])\)

CHARACTER (LEN=1) :: CNVCOR, METHOD, TRANSX, SCRATCHX,
TRANSY,SCRATCHY
COMPLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK \(\mathbb{N}\)
\(\mathbb{N}\) TEGER (8) ::M X,NX,LDX,MY,NY,MPRE,NPRE,LDY,M Z,NZ,
LD Z, LW ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) :: X, Y,Z

\section*{C INTERFACE}
\#include <sunperfh>
void denvcor2 (charcnvcor, charm ethod, char transx, char scratchx, chartransy, char scratchy, intm \(x\), int \(n x\), double * \(x\), int ldx, intm \(y\), intny, int m pre, intnpre, double \(* y\), int ldy, intm \(z\), intnz, double *z, int ldz, doublecom plex *w orkin, int lw ork);
void denvcor2_64 (charcnvcor, charm ethod, char transx, char scratchx, char transy, char scratchy, long mx, long \(n x\), double *x, long ldx, long my, long ny, long \(m\) pre, long npre, double *y, long ldy, long m z, long nz, double *z, long ldz, doublecom plex *w orkin, long lw ork);

\section*{PURPOSE}
dcnvcor2 com putes the convolution or comelation of real m atrices.

\section*{ARGUMENTS}
```

CNVCOR (input)
V 'or も'to com pute convolution, R 'or 'r' to
com pute comelation.

```

METHOD (input)
T'or t'if the Fourier transform \(m\) ethod is to
be used, D 'or d'to com pute directly from the definition.

TRANSX (input)
\(N\) 'or \(h\) 'if \(X\) is the filterm atrix, \(T\) ' or \(t^{\prime}\) if transpose \((X)\) is the filterm atrix.

SCRATCHX (input)
N 'or h'ifX m ustbe preserved, S'or S'ifX can be used as scratch space. The contents of \(X\) are undefined after retuming from a call in which X is allow ed to be used for scratch.

TRANSY (input)
N 'or h'if \(Y\) is the inputm atrix, \(T\) 'or \(\mathrm{t}^{\prime}\) if transpose \((Y)\) is the inputm atrix.
SCRATCHY (input)
N 'or h'ifY m ustbe preserved, S 'or s 'if Y
can be used as scratch space. The contents of \(Y\) are undefined after retuming from a call in which \(Y\) is allow ed to be used for scratch.

M X (input)
Num ber of row s in the filterm atrix. \(\mathrm{M} \mathrm{X}>=0\).

NX (input)
N um ber of colum ns in the filterm atrix. \(\mathrm{NX}>=0\).

X (input) dim ension (LD X ,NX)
On entry, the filterm atrix. U nchanged on exit if SCRATCHX is N' or h', undefined on exitif SCRATCHX is \(S^{\prime}\) 'or \(\mathrm{S}^{\prime}\).

LD \(X\) (input)
Leading dim ension of the array that contains the filterm atrix.

M Y (input)
N um ber of row s in the inputm atrix. \(\mathrm{M} Y>=0\).
NY (input)
\(N\) um ber of colum ns in the inputm atrix. \(N Y>=0\).
M PRE (input)
N um ber of im plicit zeros to prepend to each row of the inputm atrix. M PRE \(>=0\).

NPRE (input)
\(N\) um berof im plicit zeros to prepend to each colum n of the inputm atrix. NPRE \(>=0\).

Y (input) dim ension (LD Y ,*)
Inputm atrix. U nchanged on exit if SCRATCHY is \(N^{\prime}\) or \(h^{\prime}\), undefined on exitifSCRATCHY is \(S^{\prime}\) or \(S^{\prime}\).

LD Y (input)
Leading dim ension of the array that contains the inputm atrix.

M Z (input)
N um ber of row s in the output m atrix. \(\mathrm{M} \mathrm{Z} \mathrm{>=0}\). D CNV COR 2 w ill retum im m ediately ifM \(\mathrm{Z}=0\).

N Z (input)
N um ber of colum ns in the outputm atrix. \(\mathrm{NZ}>=0\). D CNV COR2 w ill retum im m ediately if \(\mathrm{NZ}=0\).

Z (output)
dim ension (LD Z, *)
Resultm atrix.

LD Z (input)
Leading dim ension of the array that contains the resultm atrix. LD Z >= M AX (1, M Z).

W ORK \(\mathbb{N}\) (input/output)
(input/scratch) dim ension (LW ORK)
On entry for the first call to D CNVCOR2, W ORK \(\mathbb{N}\) (1)
m ust contain 0.0. A fter the first call, W ORK \(\mathbb{N}\) (1)
m ustbe set to 0.0 iff \(W\) ORK IN has been altered since the lastcall to this subroutine or if the sizes of the arrays have changed.

LW ORK (input)
Length of the w ork vector. If the FFT is to be used then forbestperform ance LW ORK should be at least 30 w ords longerthan the am ount of \(m\) em ory needed to hold the trig tables. If the FFT is not used, the value ofLW ORK is unim portant.

\section*{Contents}
- NAME
- SYNOPSIS

\title{
- F95 INTERFACE
}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dcoom m - coordinate m atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDCOOMM(TRANSA,M,N,K,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,JNDX,NNZ,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER TRANSA,M,N,K,DESCRA (5),NNZ
* LDB,LDC,LWORK
\mathbb{NTEGER INDX NNZ),UNDX NNZ)}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
SUBROUTINEDCOOMM_64(TRANSA, M,N,K,ALPHA,DESCRA,
* VAL, \(\mathbb{N} D X, J N D X, N N Z\),
* B,LDB,BETA,C,LDC,WORK,LWORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, \(\mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{DESCRA}(5), \mathrm{NNZ}\)
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R * 8 \mathbb{N} D X(\mathbb{N} Z), \mathcal{N} D X(\mathbb{N} Z)\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL (NNZ), B (LDB,*), C (LDC, \(\left.{ }^{\star}\right), \mathrm{W}\) ORK (LW ORK)

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NECOOMM(TRANSA,M, N ],K,ALPHA,DESCRA,}

* VAL,\mathbb{NDX, NNDX,NNZ,B,[LDB],BETA,C,[LDC],}
* [W ORK], [LW ORK])
INTEGER TRANSA,M,K,NNZ

```

```

DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D IM ENSION (:) ::VAL
DOUBLE PRECISION,D IM ENSION (:,:) :: B,C

```

SUBROUTINECOOMM_64(TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A\),
* VAL, \(\mathbb{N} D X, \operatorname{JNDX}, N N Z, B,[L D B], B E T A, C,[L D C]\),
* [W ORK], [LW ORK])
\(\mathbb{I N T E G E R *}\) TRANSA, M, K, NNZ
\(\mathbb{N} T E G E R * 8, D \mathbb{I M}\) ENSION(:):: DESCRA, \(\mathbb{N} D X, J N D X\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,DIM ENSION (:) ::VAL
DOUBLE PRECISION,D \(\mathbb{M}\) ENSION (: : : : \(:\), C

\section*{DESCRIPTION}
C <-alpha op (A) B + beta C
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrioes, \(A\) is a \(m\) atrix represented in coordinate form at and op (A) is one of \(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{con} \dot{g}\left(A^{\prime}\right)\).
( 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRAN SA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate w ith m atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M \(\quad\) um berof row \(s\) in \(m\) atrix A
\(\mathrm{N} \quad \mathrm{N}\) um berof colum ns in \(m\) atrix C

K \(\quad \mathrm{Num}\) berof colum ns in \(m\) atrix \(A\)

A LPH A Scalarparam eter
DESCRA () D escriptor argum ent. Five elem ent integer array
DESCRA (1) m atrix structure
0 :general
1 : symmetric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm ( \(\mathrm{A}=\mathrm{CONJG}\) (A))
3 :Triangular
4 : Skew (Anti)-Symm etric ( \(A=-A\) )
5 : D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJ}\) ( A ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper

DESCRA (3) main diagonal type
0 : non-unit
1 : unit
DESCRA (4) A ray base NOT IM PLEM ENTED)
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 :no repeated indices
VAL () scalar array of length NNZ consisting of the non-zero entries of \(A\), in any order.
\(\mathbb{I N D X}\) () integer array of length NNZ consisting of the comesponding row indices of the entries of A.

JND X () integer amray of length NNZ consisting of the corresponding colum \(n\) indioes of the entries of A.

NN Z number of non-zero elem ents in A.
B 0 rectangular array w th first dim ension LD B.
LD B leading din ension ofB

BETA Scalarparam eter
C 0 rectangular anray with firstdim ension LD C.

LD C leading dim ension of \(C\)
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is notreferenced in the current version.

\section*{SEE ALSO}

N IST FORTRA N Sparse B las U sers G uide available at:
http://m ath nistgov/m csd/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib.org/utk/papers/sparse.ps

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dcopy -C opy x to y

```

\section*{SYNOPSIS}

```

\mathbb{NTEGERN,}\mathbb{NCX,\mathbb{NCY}}\mathbf{T}\mathrm{ \}
DOUBLE PRECISION X (*),Y (*)
SUBROUT\mathbb{NEDCOPY_64 N,X,\mathbb{NCX,Y,INCY)}}\mathbf{N}=\mathbf{N}
INTEGER*8N,\mathbb{NCX,INCY}
DOUBLE PRECISION X (*),Y (*)

```
F95 INTERFACE
    SU BROUTINE COPY ( \(\mathbb{N}\) ],X, \([\mathbb{N C X}], Y,[\mathbb{N} C Y])\)
    \(\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y\)
    REAL (8), D IM ENSION (:) :: X,Y
    SU BROUTINE COPY_64 (N ],X, [ \(\mathbb{N} C X], Y,[\mathbb{N} C Y])\)
    \(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{N} C X, \mathbb{N} C Y\)
    REAL (8), D \(\mathbb{I M}\) ENSION (:) :: X,Y
C INTERFACE
    \#include <sunperfh>
    void doopy (intn, double *x, int incx, double *y, int incy);
    void doopy_64 (long n, double *x, long incx, double *y, long
        incy);

\section*{PURPOSE}
dcopy C opy \(x\) to \(y\) where \(x\) and \(y\) are \(n\)-vectors.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N mustbe at least one for the subroutine to have any visible effect. U nchanged on exit.
X (input)
\((1+(n-1) \star \operatorname{abs}(\mathbb{N} C X))\). Before entry, the increm ented array X m ust contain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{I N C X}\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{I N C X}\) m ustnotbe zero. U nchanged on exit.

Y (output)
( \(1+(m-1) * \operatorname{abs}(\mathbb{N} C Y)\) ). On entry, the increm ented array \(Y\) m ustcontain the vectory. On exit, \(Y\) is overw rilten by the vectorx.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y . \mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
dcosqb - synthesize a Fourier sequence from its representation in term s of a cosine series \(w\) th odd \(w\) ave num bers. The CO SQ operations are unnorm alized inverses of them selves, so a call to COSQF follow ed by a call to CO SQ B w illm ultiply the inputsequence by 4 * \(N\).

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DCOSQB N,X,W SAVE)}

```
\(\mathbb{N}\) TEGER N
DOUBLE PRECISION X (*),W SAVE (*)
SU BROUTINEDCOSQB_64 \(\mathbb{N}, \mathrm{X}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER*8 N
DOUBLE PRECISION X (*), W SAVE (*)

F95 INTERFACE
SUBROUTINE COSQB ( \(\mathbb{N}\) ],X,W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::X,W SAVE

SU BROUTINE COSQB_64 (N ],X,W SAVE)
\(\mathbb{N} T E G E R(8):: N\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::X,W SAVE

\section*{C INTERFACE}
\#include <sunperfh>
void dcosqb (intn, double *x, double *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. These subroutines are \(m\) ostefficientw hen \(N\) is a product of sm allprim es. \(\mathrm{N}>=0\).

X (input/output)
On entry, an aray of length \(N\) containing the sequence to be transform ed. On exit, the quarterw ave cosine synthesis of the input.
W SAVE (input)
O n entry, an array with dim ension of at least (3 *
\(\mathrm{N}+15\) ) that has been in itialized by D C O SQ I.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
dcosqf-com pute the Fourier coefficients in a cosine series representation w th only odd w ave num bers. The COSQ operations are unnorm alized inverses of them selves, so a call to COSQF followed by a call to COSQB w illm ultiply the input sequence by 4 * N .

\section*{SYNOPSIS}
\[
\text { SUBROUTINEDCOSQF } \mathbb{N}, \mathrm{X}, \mathrm{~W} \text { SAVE) }
\]
\(\mathbb{N}\) TEGER N
DOUBLE PRECISIONX (*) , W SAVE (*)
SU BROUTINEDCOSQF_64 \(\mathbb{N}, \mathrm{X}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER*8N
DOUBLE PRECISION X (*), W SAVE (*)
F95 INTERFACE
SU BROUTINE COSQF ( \(\mathbb{N}\) ],X,W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::X,W SAVE

SU BROUTINE COSQF_64 ( \(\mathbb{N}\) ],X,W SAVE)
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}\)
REAL (8), D IM ENSION (:) ::X,W SAVE

\section*{C INTERFACE}
\#include <sunperfh>
void doosqf(intn, double *x, double *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. These subroutines are \(m\) ostefficientw hen \(N\) is a product of sm allprim es. \(\mathrm{N}>=0\).

X (input/output)
On entry, an aray of length \(N\) containing the sequence to be transform ed. On exit, the quarter-w ave cosine transform of the input.
W SAVE (input)
O n entry, an array with dim ension of at least (3
* \(\mathrm{N}+15\) ) that has been initialized by D C O SQ I.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
dcosqi-initialize the array W SA VE, which is used in both COSQF and COSQB.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DCOSQIN,W SAVE)}

```
\(\mathbb{N}\) TEGER N
DOUBLE PRECISION W SAVE (*)
SUBROUTINEDCOSQI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)

INTEGER*8N
DOUBLE PRECISION W SAVE (*)

\section*{F95 INTERFACE}

SUBROUTINECOSQIN,W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W SAVE

SUBROUTINE COSQ I_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
\(\mathbb{N} T E G E R(8):: N\)
REAL (8), D IM ENSION (:) ::W SAVE

\section*{C INTERFACE}
\#include < sunperfh>
void doosqi(intn, double *w save);
void doosqi_ 64 (long n, double *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. The m ethod is m ost efficientw hen N is a product of sm allprim es.

W SAVE (input)
On entry, an array ofdim ension ( 3 * \(\mathrm{N}+15\) ) or greater. D C O SQ I needs to be called only once to intialize W SAVE before calling DCOSQF and/or DCOSQB if N and W SAVE rem ain unchanged betw een these calls. Thus, subsequent transform \(s\) or inverse transforms of sam e size can be obtained faster than the first since they do not require initialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
dcost-com pute the discrete Fourier cosine transform of an even sequence. The COST transform s are unnorm alized inverses of them selves, so a call of COST follow ed by another call of C O ST w illm ultiply the input sequence by 2 * (N-1).

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDCOST N,X,W SAVE)}

```
\(\mathbb{N}\) TEGER N
DOUBLE PRECISION X (*),W SAVE (*)
SUBROUTINEDCOST_64 \(\mathbb{N}, \mathrm{X}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER*8N
DOUBLE PRECISION X (*), W SAVE (*)

F95 INTERFACE
SU BROUTINE COST ( \(\mathbb{N}\) ], \(X, W\) SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::X,W SAVE

SU BROUTINE COST_64 (N ],X,W SAVE)
\(\mathbb{N}\) TEGER (8) :: N
REAL (8),D \(\mathbb{M}\) ENSION (:) ::X,W SAVE

\section*{C INTERFACE}
\#include <sunperfh>
void doost(intn, double *x, double *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. These subroutines are most efficient when \(\mathrm{N}-1\) is a productofsm allprin es. \(\mathrm{N}>=2\).

X (input/output)
On entry, an aray of length \(N\) containing the sequence to be transform ed. On exit, the cosine transform of the input.
W SAVE (input)
On entry, an array with dim ension of at least (3
* \(\mathrm{N}+15\) ), initialized by D COSTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
dcosti-initialize the array W SAVE, which is used in CO ST .

\section*{SYNOPSIS}
SU BROUTINE DCOSTIN,W SAVE)
\(\mathbb{N}\) TEGER N
DOUBLE PRECISION W SAVE (*)
SUBROUTINEDCOSTI_64 N,W SAVE)
\(\mathbb{N}\) TEGER*8N
DOUBLE PRECISION W SAVE (*)
F95 INTERFACE
SU BROUTINE COSTIN,W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL (8),D IM ENSION (:) ::W SAVE
SUBROUTINECOSTI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER (8) ::N
REAL (8), D IM ENSION (:) ::W SAVE

\section*{C INTERFACE}
\#include < sunperfh>
void doosti(intn, double *w save);
void doosti_ 64 (long n, double *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. The \(m\) ethod is \(m\) ostefficientw hen \(N-1\) is a product of sm allprim es. \(\mathrm{N}>=2\).

W SAVE (input)
On entry, an array ofdim ension ( 3 * \(\mathrm{N}+15\) ) or greater. DCOSTI is called once to initialize W SAVE before calling DCOST and need notbe called again betw een calls to DCOST ifN andW SAVE rem ain unchanged. Thus, subsequent transform s of sam e size can be obtained faster than the first since they do not require initialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dcscm \(m\)-com pressed sparse colum \(n\) form atm atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDCSCMM(TRANSA,M,N,K,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER TRANSA,M,N,K,DESCRA (5),
* LDB,LDC,LWORK
INTEGER INDX NNZ),PNTRB(K),PNTRE (K)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NEDCSCMM _64(TRANSA,M ,N,K,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER*8 TRANSA,M,N,K,DESCRA (5),
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 }\mathbb{N}DX(NNZ),PNTRB(K),PNTRE (K)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
where NN Z = PN TRE (K)-PN TRB (1)

\section*{F95 INTERFACE}

SUBROUTINECSCMM (TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N} D X\), * PNTRB,PNTRE, B, [LDB],BETA,C, [LDC], [WORK], [WWORK])
\(\mathbb{N}\) TEGER TRANSA, M, K
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \quad\) ESCRA, \(\mathbb{N} D X, P N T R B, P N T R E\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D \(\mathbb{M}\) ENSION(:) ::VAL

DOUBLE PRECISION ,D IM ENSION (:, :) :: B , C

SUBROUTINECSCMM_64 (TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N D X}\),
* PNTRB, PNTRE,B, [LDB],BETA,C, [LDC], [WORK], [LWORK])
\(\mathbb{N}\) TEGER*8 TRANSA, M, K
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{N} N(:):: \operatorname{DESCRA}, \mathbb{N} D X\), PNTRB, PNTRE
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION ,D \(\mathbb{M}\) ENSION (:) ::VAL
D OUBLE PRECISION ,D IM ENSION (:, :) :: B , C

\section*{DESCRIPTION}
\[
C<- \text { alpha op (A) B + beta C }
\]
where A LPH A and BETA are scalar, C and B are dense m atriges, \(A\) is a m atrix represented in com pressed sparse colum \(n\) form at and \(o p(A)\) is one of \(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRAN SA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate w th m atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in matrix A

N \(\quad\) um berof colum ns in matrix C

K \(\quad \mathrm{N}\) um berof colum ns in \(m\) atrix \(A\)

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem ent integer anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(A=-C O N J\) ( \(A\) ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er

2 :upper
DESCRA (3) main diagonal type
0 :non-unit
1 : unit
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED)
0 :C C ++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM EN TED )
0 : unknown
1 : no repeated indices

VAL () scalar array of length NN Z consisting of nonzero entries ofA.

IND X \(0 \quad\) integer array of length NN Z consisting of the row indices of nonzero entries ofA .

PN TRB 0 integer amray of length \(K\) such thatPN TRB (J)-PN TRB (1)+1 points to location in VA L of the firstnonzero elem ent in colum n J .
PN TRE 0 integer array of length \(K\) such thatPN TRE (J)-PN TRB (1) points to location in V A L of the lastnonzero elem ent in colum n J .

B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension ofB

BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C.
LD C leading dim ension of \(C\)

W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the currentversion.

LW ORK length ofW ORK array.LW ORK is notreferenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov fin csd/Staffk Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee,
http://w w w netlib .org/utk/papers/sparse .ps

\section*{NOTES/BUGS}

It is know \(n\) that there exists another representation of the com pressed sparse colum n form at (see forexam ple Y Saad, "IterativeM ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each colum \(n\) in the arrays VA L and \(\mathbb{N} D \mathrm{X}\) is used instead oftw o arraysPN TRB and PN TRE.To use the routine \(w\) th this kind of sparse colum \(n\) form at the follow ing calling sequence should be used

SUBROUTINE SCSCMM (TRANSA, M,N,K,ALPHA,DESCRA,
* \(V A L, \mathbb{N} D X, \mathbb{A}, \mathbb{I}(2), B, L D B, B E T A\),
* C,LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dcscsm - com pressed sparse colum \(n\) form at triangular solve

\section*{SYNOPSIS}
```

SUBROUTINEDCSCSM(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,

* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER TRANSA,M,N,UNITD,DESCRA (5),
* LDB,LDC,LW ORK
\mathbb{NTEGER INDX NNZ),PNTRB (M),PNTREM)}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION DV M),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
SUBROUTINEDCSCSM_64(TRANSA, M,N,UNITD,DV,ALPHA,DESCRA,
* VAL, \(\mathbb{N} D \mathrm{X}, \mathrm{PNTRB}, \mathrm{PNTRE}\),
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, M,N,UNITD,DESCRA (5),
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R * 8 \mathbb{N} D X(N N Z), \operatorname{PNTRB}(M)\), PNTRE \(M\) )
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION DV M),VAL \(\mathbb{N N Z}\) ), B (LDB,*), C (LDC, \(\left.{ }^{\star}\right), W\) ORK (LW ORK)
where \(N N Z=P N T R E M) P N T R B(1)\)

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE CSCSM (TRANSA,M, N ],UNITD,DV,ALPHA,DESCRA,VAL, INDX,}

```
* PNTRB,PNTRE, B, [LDB],BETA,C, [LDC], [W ORK], [LW ORK])
\(\mathbb{N}\) TEGER TRANSA, M, UNITD
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: D E S C R A, \mathbb{N} D \mathrm{X}, \mathrm{PN} T R B, \operatorname{PNTRE}\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D \(\mathbb{M}\) ENSION (:) ::VAL,DV
DOUBLE PRECISION ,D IM ENSION (: :) :: B, C

SUBROUT \(\mathbb{N} E \operatorname{CSC} M\) _ 64 (TRANSA \(, ~ M, ~ \mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X\),
* PNTRB,PNTRE, B, [LDB],BETA, C , [LDC], [W ORK], [LW ORK])
\(\mathbb{N}\) TEGER*8TRANSA, M , UN ITD
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: \operatorname{DESCRA}, \mathbb{N} D X, \operatorname{PNTRB}, \operatorname{PNTRE}\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D \(\mathbb{I M}\) ENSION (:) ::VAL, DV
DOUBLE PRECISION ,D \(\mathbb{M}\) ENSION (:, :) :: B , C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { op }(A) B+B E T A C \\
& C<-A L P H A \text { OP }(A) D B+B E T A C
\end{aligned}
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense matrices, \(D\) is a diagonalscaling \(m\) atrix, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix represented in com pressed sparse colum n form atand op (A ) is one of op (A) \()=\operatorname{inv}(A)\) or op (A \()=\operatorname{inv}(A)\) or op (A) \()=\operatorname{inv}\left(\operatorname{con} \dot{g}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix 0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in \(m\) atrix \(A\)

N \(\quad\) umberof colum \(n s\) in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum \(n\) scaling)
4 : A utom atic colum n scaling (see section NOTES for furtherdetails)

DV () A rray of length M containing the diagonalentries of the scaling diagonalm atrix D.

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem ent integer anay DESCRA (1) m atrix structure
\[
\begin{aligned}
& 0 \text { : general } \\
& 1 \text { : sym m etric ( } A=A \text { ) } \\
& 2: \text { H erm itian ( } A=\operatorname{CON} \text { JG (A }) \text { ) } \\
& 3 \text { : Triangular } \\
& 4 \text { : Skew (A nti)-Sym m etric ( } A=-A \text { ) } \\
& 5 \text { : D iagonal } \\
& 6 \text { : Skew Herm itian ( } A=-\operatorname{CON} \text { JG (A ) ) }
\end{aligned}
\]

N ote: For the routine, D ESCRA (1)=3 is only supported.

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonaltype
0 : non-unit
1 :unit
DESCRA (4) A may base \(\mathbb{N O T} \mathbb{M}\) PLEM ENTED)
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? \(N\) OT \(\mathbb{M}\) PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () scalar array of length N N Z consisting of nonzero entries ofA.
\(\mathbb{N} D \mathrm{X}\) () integer array of length N N Z consisting of the row indices of nonzero entries of . (R ow indigesM UST be sorted in increasing order for each colum n).

PNTRB () integer amay of length \(M\) such thatPN TRB (J) PN TRB (1)+1 points to location in VA L of the firstnonzero elem ent in colum \(n \mathrm{~J}\).

PN TRE () integer array of length \(M\) such thatPN TRE (J)-PN TRB (1) points to location in VA L of the lastnonzero elem ent in colum \(n \mathrm{~J}\).

B 0 rectangular array w ith first dim ension LD B .

LD B leading dim ension ofB

BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch array of length LW ORK.

On exit, if LW ORK \(=-1, W\) ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK array. LW ORK should be at leastM .
Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=M\) *N_CPUS where N_CPUS is the m axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
htep://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such tests \(m\) ust.be perform ed before calling this routine.
2. If UN ITD \(=4\), the routine scales the colum ns of A such that their 2 -norm s are one. The scaling \(m\) ay im prove the accuracy of the com puted solution. C orresponding entries of VA L are changed only in the particular case. On retum D V \(m\) atrix stored as a vector contains the diagonalm atrix by which the colum ns have been scaled. UN ITD = 3 should be used for the next calls to the routine \(w\) ith overw ritten VA L and DV.

WORK (1)=0 on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here i is the colum n num berw hich 2 -norm is exactly zero.
3. If \(D E S C R A(3)=1\) and \(U N\) ITD < 4, the unitdiagonalelem ents m ightorm ightnotbe referenced in the C SC representation
of a sparse \(m\) atrix. They are notused anyw ay in these cases. ButifU N ITD = 4, the unitdiagonalelem ents M U ST be referenced in the CSC representation.
4.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general.sparse \(m\) atrix A is used. H ow ever DESCRA (1) m ustbe equal to 3 in this case.
5. It is know \(n\) that there exists another representation of the com pressed sparse colum n form at (see for exam ple Y Saad, "Iterative M ethods forSparse LinearSystem \(s\) ", W PS, 1996). Its data structure consists of three anray instead of the fourused in the cumentim plem entation. Them ain difference is thatonly one amray, IA , containing the pointers to the beginning ofeach colum \(n\) in the arrays VA L and \(\mathbb{I N D X}\) is used instead of tw o amaysPNTRB and PN TRE.To use the routine \(w\) th this kind of sparse colum \(n\) form at the follow ing calling sequence should be used

SUBROUTINESCSCSM (TRANSA, M,N,UNITD,DV,ALPHA,DESCRA,
* \(V A L, \mathbb{N} D X, \mathbb{A}, \mathbb{A}(2), B, L D B, B E T A\),
* \(\quad\), LDC,\(W\) ORK,LW ORK )

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dcssmm -com pressed sparse row form atm atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DCSRMM(TRANSA,M,N,K,ALPHA,DESCRA,}

* VAL, INDX,PNTRB,PNTRE,
* B,LDB,BETA,C,LDC,W ORK,LWORK)
INTEGER TRANSA,M,N,K,DESCRA (5),
* LDB,LDC,LWORK
INTEGER INDX (NNZ),PNTRB(M),PNTRE(M)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL NNZ),B (LDB,*),C (LDC,*),WORK (LWORK)
SUBROUT\mathbb{NE D CSRMM_64(TRANSA,M,N,K,ALPHA,DESCRA,}
* VAL,INDX,PNTRB,PNTRE,
* B,LDB,BETA,C,LDC,WORK,LW ORK)
INTEGER*8 TRANSA,M,N,K,DESCRA (5),
* LDB,LDC,LWORK
\mathbb{NTEGER*8 INDX (NNZ),PNTRB(M),PNTRE(M)}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL NNZ),B (LDB,*),C (LDC,*),WORK (LW ORK)

```
where \(N N Z=P N T R E(M)-P N T R B(1)\)

\section*{F95 INTERFACE}
```

SUBROUTINE CSRMM (TRANSA,M, N ],K,ALPHA,DESCRA,VAL, INDX,

* PNTRB,PNTRE,B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])
INTEGER TRANSA,M,K
INTEGER,D IM ENSION (:) :: DESCRA, INDX,PNTRB,PNTRE
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D IM ENSION(:) ::VAL
DOUBLE PRECISION,D IM ENSION (:,:) :: B,C

```

SUBROUTINE CSRMM_64 (TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N D X}\),
* PNTRB, PNTRE, B, [LDB],BETA, C , [LDC], [WORK], [LW ORK])

IN TEGER*8TRANSA, M, K
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: \operatorname{DESCRA}, \mathbb{N} D X, \operatorname{PNTRB}, \operatorname{PNTRE}\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION ,D \(\mathbb{M}\) ENSION (:) ::VAL
DOUBLE PRECISION ,D \(\mathbb{M}\) ENSION (:, :) :: B, C

\section*{DESCRIPTION}
\[
C<- \text { alpha op (A) B + beta C }
\]
where A LPH A andBETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, A is a m atrix represented in com pressed sparse row form at and \(o p(A)\) is one of \(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRA N SA Indicates how to operate \(w\) th the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate w th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalentto 1 ifm atrix is real.

M \(\quad N\) um ber of row \(s\) in \(m\) atrix A

N \(\quad\) Num berof colum ns in \(m\) atrix \(C\)

K \(N\) umberof colum \(n s\) in \(m\) atrix \(A\)

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(A=A\) )
2 : Herm itian ( \(A=C O N J(A)\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(A=-C O N J\) ( \(A\) ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper

DESCRA (3) main diagonal type
0 : non-unit
1 : unit
DESCRA (4) A ray base NOT \(\mathbb{M}\) PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 :no repeated indices
VAL ( scalar array of length NN Z consisting of nonzero entries ofA.
\(\mathbb{I N D X} 0 \quad\) integer array of length NN Z consisting of the colum \(n\) indioes of nonzero entries of \(A\).

PN TRB () integer array of length \(M\) such thatPN TRB (J) PN TRB (1)+1
points to location in VA L of the firstnonzero elem ent in row J .
PNTRE ( integerarray of length \(M\) such thatPNTRE (J)-PNTRB (1) points to location in V A L of the lastnonzero elem ent in row J .

B 0 rectangular array w th first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C \(0 \quad\) rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC
W ORK () scratch array of length LW ORK.W ORK is not referenced in the curment version.

LW ORK length ofW ORK aray. LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/n cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:

\section*{NOTES/BUGS}

It is know \(n\) that there exists another representation of the com pressed sparse row form at (see forexam ple Y Saad, "Herative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three anray instead of the fourused in the currentim plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each row in the arrays VA L and \(\mathbb{N D} \mathrm{X}\) is used instead of tw o arrays PN TRB and PN TRE. To use the routine w th this kind of com pressed sparse row form at the follow ing calling sequence should be used

SUBROUTINESCSRMM (TRANSA, M, N, K,ALPHA,DESCRA,
* \(\quad V A L, \mathbb{N} D X, \mathbb{A}, \mathbb{I A}(2), B, L D B, B E T A\),
* C,LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dcsrsm - com pressed sparse row form at triangular solve

\section*{SYNOPSIS}
```

SUBROUTINEDCSRSM(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,

* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER TRANSA,M,N,UNITD,DESCRA (5),
* LDB,LDC,LW ORK
\mathbb{NTEGER INDX (NNZ),PNTRB (M),PNTREM)}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION DV M),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
SUBROUTINEDCSRSM_64(TRANSA, M,N,UNITD,DV,ALPHA,DESCRA,
* VAL, \(\mathbb{N} D \mathrm{X}, \mathrm{PNTRB}, \mathrm{PNTRE}\),
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, \(M, N, U N I T D, D E S C R A(5)\),
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R * 8 \mathbb{N} D X(N N Z), \operatorname{PNTRB}(M)\), PNTRE \(M\) )
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION DV M),VAL \(\mathbb{N N Z}\) ), B (LDB,*), C (LDC, \(\left.{ }^{\star}\right), W\) ORK (LW ORK)
where \(N N Z=P N T R E M) P N T R B(1)\)

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE CSRSM (TRANSA,M, N ],UNITD,DV,ALPHA,DESCRA,VAL, INDX,}

```
* PNTRB,PNTRE, B, [LDB],BETA,C, [LDC], [W ORK], [LW ORK])
\(\mathbb{I N}\) TEGER TRANSA, M, UN ITD
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: D E S C R A, \mathbb{N} D \mathrm{X}, \mathrm{PN} T R B, \operatorname{PNTRE}\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D \(\mathbb{M}\) ENSION (:) ::VAL,DV
DOUBLE PRECISION ,D IM ENSION (: :) :: B, C

SUBROUTINECSRSM_64 (TRANSA, M, \(\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X\),
* PNTRB,PNTRE, B, [LDB],BETA, C , [LDC], [W ORK], [LW ORK])
\(\mathbb{N}\) TEGER*8TRANSA, M , UN ITD
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: D E S C R A, \mathbb{N D} X, \operatorname{PNTRB}, \operatorname{PNTRE}\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D \(\mathbb{I M}\) ENSION (:) ::VAL, DV
DOUBLE PRECISION ,D IM ENSION (:, :) :: B, C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { op }(A) B+B E T A C \\
& C<-A L P H A \text { OP }(A) D B+B E T A C
\end{aligned}
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense matrices, \(D\) is a diagonalscaling \(m\) atrix, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix represented in com pressed sparse row form atand op (A ) is one of op (A) \(=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}\left(\operatorname{conjg}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicates \(m\) atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix 0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in \(m\) atrix \(A\)
\(N \quad N\) um berof colum \(n s\) in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum \(n\) scaling)
4 : A utom atic row scaling (see section N O TES for furtherdetails)

DV () A ray of length M containing the diagonalentries of the scaling diagonalm atrix D.

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem ent integer anay DESCRA (1) m atrix structure
\[
\begin{aligned}
& 0 \text { : general } \\
& 1 \text { : sym m etric ( } A=A \text { ) } \\
& 2: \text { H erm itian ( } A=\operatorname{CON} \text { JG (A }) \text { ) } \\
& 3 \text { : Triangular } \\
& 4 \text { : Skew (A nti)-Sym m etric ( } A=-A \text { ) } \\
& 5 \text { : D iagonal } \\
& 6 \text { : Skew Herm itian ( } A=-\operatorname{CON} \text { JG (A ) ) }
\end{aligned}
\]

N ote: For the routine, only DESCRA (1)=3 is supported.

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 : non-unit
1 :unit
DESCRA (4) A ray base \(\mathbb{N O T} \mathbb{M}\) PLEM ENTED )
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? \(\mathbb{N O T} \mathbb{I}\) PLEM ENTED)
0 : unknown
1 : no repeated indices
VAL () scalar array of length N N Z consisting of nonzero entries ofA.
\(\mathbb{I N D X ( ) \quad i n t e g e r a m a y ~ o f ~ l e n g t h ~ N ~ N ~ Z ~ c o n s i s t i n g ~ o f ~ t h e ~ c o l u m ~ n ~}\) indices of nonzero entries ofA (colum n indices M U ST be sorted in increasing order for each row )

PNTRB () integer amay of length \(M\) such thatPN TRB (J) PN TRB (1)+1 points to location in VA L of the firstnonzero elem ent in row J.

PN TRE () integer amay of length M such thatPNTRE (J)-PN TRB (1) points to location in VA L of the lastnonzero elem ent in row J.

B 0 rectangular array w ith first dim ension LD B .

LD B leading dim ension ofB

BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch array of length LW ORK.

On exit, ifLW ORK \(=-1, W\) ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK array. LW ORK should be at leastM .
Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=M\) *N_CPUS where N_CPUS is the m axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
htep://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such tests \(m\) ustbe perform ed before calling this routine.
2. IfUN ITD \(=4\), the routine scales the row s of \(A\) such that their 2 -nom s are one. The scaling \(m\) ay im prove the accuracy of the com puted solution. C orresponding entries ofV A L are changed only in the particular case. On retum D V m atrix stored as a vector contains the diagonalm atrix by which the row s have been scaled. U N ITD = 2 should be used for the next calls to the routine w ith overw ritten VAL and DV.

WORK (1)=0 on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here i is the row num berw hich 2 -norm is exactly zero.
3. If \(\operatorname{DESCRA}(3)=1\) and UN ITD < 4, the unitdiagonalelem ents \(m\) ightorm ightnotbe referenced in the CSR representation of a sparse m atrix. They are notused anyw ay in these cases.

ButifUN ITD \(=4\), the unit diagonalelem ents M U ST be referenced in the CSR representation.
4.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse m atrix \(A\) is used. H ow ever \(\operatorname{DESCRA}\) (1) m ustbe equal to 3 in this case.
5. It is know \(n\) that there exists another representation of the com pressed sparse row form at (see for exam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, IA , containing the pointers to the beginning ofeach row in the amaysVA L and \(\mathbb{N} D \mathrm{X}\) is used instead of tw o arrays PN TRB and PN TRE. To use the routine \(w\) ith this kind of com pressed sparse row form at the follow ing calling sequence should be used

SUBROUTINESCSRSM (TRANSA,M,N,UNTID,DV,ALPHA,DESCRA,
* \(\quad V A L, \mathbb{N} D X, \mathbb{A}, \mathbb{I A}(2), B, L D B, B E T A, C\),
* LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

ddiam m -diagonal form atm atrix-m atrix m ultiply

```

\section*{SYNOPSIS}
```

SUBROUTINEDDIAMM(TRANSA,M,N,K,ALPHA,DESCRA,

* VAL,LDA,D\mathbb{AG,NDIAG,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,M,N,K,DESCRA (5),LDA,NDIAG,}
* LDB,LDC,LWORK
\mathbb{NTEGER IDIAG NDIAG)}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL (LDA,ND IAG),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
SUBROUTINEDDIAMM_64(TRANSA, M,N,K,ALPHA,DESCRA,
* VAL,LDA, \(\mathbb{D} \mathbb{I} G, N D \mathbb{A} G\),
* B,LDB,BETA, C,LDC,WORK,LWORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, \(M, N, K, D E S C R A(5), L D A, N D I A G\),
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R * 8 \mathbb{D} \mathbb{A} G(\mathbb{N} \mathbb{I} G)\)
DOUBLE PRECISION ALPHA, BETA
DOUBLE PRECISION VAL (LDA,NDIAG), B (LDB,*), C (LDC, \(\left.{ }^{\star}\right), W\) ORK (LW ORK)

\section*{F95 INTERFACE}

SUBROUTINED \(\operatorname{IAM}\) M (TRANSA, \(M, \mathbb{N}], K, A L P H A, D E S C R A, V A L,[L D A]\), * \(\mathbb{D} \mathbb{I} G, N D \mathbb{A}, \mathrm{~B},[\mathrm{LDB}], \mathrm{BETA}, \mathrm{C},[\mathrm{LDC}],[\mathbb{W} O R K],[\mathrm{LW} O R K])\)
\(\mathbb{N} T E G E R\) TRANSA, \(\mathrm{M}, \mathrm{K}, \mathrm{ND} \mathbb{I} G\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \quad \mathrm{DESCRA}, \mathbb{D} \mathbb{A} G\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D IM ENSION (:, :) :: VAL, B, C
SU BROU TINEDIAMM_64 (TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L,[L D A]\), * \(\mathbb{D} \mathbb{I} G, N D \mathbb{A} G, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)

\section*{DESCRIPTION}

C <-alpha op (A) B + beta C
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrioes, \(A\) is a \(m\) atrix represented in diagonal form at and op (A ) is one of
\[
o p(A)=A \quad \text { or } \operatorname{op}(A)=A^{\prime} \text { or op }(A)=\operatorname{con} \dot{g}\left(A^{\prime}\right) .
\]
( 'indicatesm atrix transpose)
TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate w ith m atrix
1 : operate \(w i\) ith transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in \(m\) atrix A
N \(\quad N\) um berof colum ns in \(m\) atrix \(C\)
K \(\quad \mathrm{Num}\) berof colum ns in \(m\) atrix \(A\)

A LPH A Scalarparam eter
DESCRA () D escriptor argum ent. Five elem ent integer array
0 :general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm itian ( \(\mathrm{A}=\mathrm{CONJG}\) (A))
3 :Triangular
4 : Skew (Anti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJG}\) ( A ))
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) main diagonal type
0 :non-unit
1 :unit
DESCRA (4) A ray base \(\mathbb{N O T} \mathbb{I}\) PLEM ENTED )
0 :C C ++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 :unknown
1 : no repeated indices

VAL ( ) tw o-dim ensionalLD A boy ND IA G array such thatV A L (:I) consists of non-zero elem ents on diagonal ID IA G (I) of A. D iagonals in the low er triangularpart of A are padded from the top, and those in the upper triangularpart are padded from the bottom.

LDA leading dim ension ofVAL,m ustbe GE.M \(\mathbb{N} \mathbb{M}, K\) )
ID IA G () integer array of length ND IA G consisting of the comesponding diagonal offsets of the non-zero diagonals ofA in VA L. Low ertriangular diagonals have negative offsets, them ain diagonal has offset 0 , and upper triangular diagonals have positive offset.

ND IA G num berof non-zero diagonals in A.
B 0 rectangular array with first dim ension LD B .
LD B leading dim ension ofB

BETA Scalarparam eter
C 0 rectangular array w th first dim ension LD C .

LD C leading dim ension ofC
W ORK () scratch array of length LW ORK.W ORK is not referenced in the current version.

LW ORK length ofW ORK array. LW ORK is not referenced in the current version.

\section*{SEE ALSO}

N IST FO RTRAN Sparse B las U ser's G uide available at:
http://m ath nistgov/m csd/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib.org/utk/papers/sparse.ps

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

ddiasm -diagonal form at triangular solve

```

\section*{SYNOPSIS}
```

SUBROUTINEDDIASM(TRANSA,M ,N,UN ITD,DV,ALPHA,DESCRA,

* VAL,LDA,\mathbb{D IAG,NDIAG,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,M,N,UNITD,DESCRA (5),LDA,ND IAG,}
* LDB,LDC,LWORK
\mathbb{NTEGER IDIAG NDIAG)}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION DV M ),VAL (LDA,NDIAG),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINEDDIA SM_64(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,
* VAL,LDA,\mathbb{D IAG,NDIAG,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER*8 TRANSA,M,N,UNITD,DESCRA (5),LDA,NDIAG,}
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 \mathbb{D IAG NDIAG)}}\mathbf{N}\mathrm{ )}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION DV M ),VAL (LDA,NDIAG),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}
```

SUBROUTINEDIASM (TRANSA,M, N ],UNITD,DV,ALPHA,DESCRA,VAL,

* [LDA],\mathbb{D IAG,NDIAG,B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])}
INTEGER TRANSA,M,NDIAG
INTEGER,D IM ENSION (:) :: DESCRA,\mathbb{D IAG}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D IM ENSION (:) :: DV
D OUBLE PRECISION,D IM ENSION (:,:) :: VAL,B,C

```
SUBROUTINEDIASM_64(TRANSA,M, N ],UNITD,DV,ALPHA,DESCRA,VAL,
* [LDA], \(\mathbb{D} \mathbb{I A G}, N D \mathbb{I A G}, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N}\) TEGER*8 TRANSA, M,ND IAG
\(\mathbb{N} T E G E R * 8, D \mathbb{M}\) ENSION (:) :: DESCRA, \(\mathbb{D} \mathbb{I A}\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D IM ENSION (:) :: DV
D OUBLE PRECISION,D IM ENSION (:, :) :: VAL,B,C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { op(A)B + BETA } C \quad C<-A L P H A D \text { op(A)B+BETA C } \\
& C<-A L P H A \text { op(A)D B + BETA } C
\end{aligned}
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense \(m\) atrices, \(D\) is a diagonalscaling \(m\) atrix, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix represented in diagonal form at and op (A ) is one of
\(\mathrm{op}(\mathrm{A})=\operatorname{inv}(\mathrm{A})\) or op (A) \(=\operatorname{inv}(A)\) or op (A) \()=\operatorname{inv}\left(\right.\) (oonjg ( \(\left.\left.A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate w th transpose m atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in \(m\) atrix A

N \(\quad N\) um berof colum ns in matrix C
UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 :Scale on left (row scaling)
3 : Scale on right (colum \(n\) scaling)
4 :A utom atic row scaling (see section NOTES for furtherdetails)

DV 0 A ray of length M containing the diagonalentries of the scaling diagonalm atrix D .

A LPH A Scalarparam eter
DESCRA 0 D escriptor argum ent. Five elem ent integer aray
DESCRA (1) m atrix structure
0 : general

1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm itian ( \(A=C O N J(A)\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \((A=-A\) )
5 :D iagonal
6 : Skew Herm titian ( \(A=-\operatorname{CON}\) J ( \(A\) ) )
N ote: For the routine, only D ESCRA (1)=3 is supported.

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 :non-unit
1 :unit
DESCRA (4) A ray base \(\mathbb{N O T} \mathbb{M}\) PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? (NOT \(\mathbb{M}\) PLEM ENTED)
0 : unknown
1 : no repeated indices
VAL () tw o-dim ensionalLD A boy-ND IA G aray such thatVAL(:,I) consists of non-zero elem ents on diagonal ID IA G (I) of A. D iagonals in the low er triangularpart of A are padded from the top, and those in the upper triangularpart are padded from the bottom.

LDA leading dim ension ofVAL, m ustbe GE.M \(\mathbb{N}(M, K)\)

ID IA G () integer anay of length ND IA G consisting of the corresponding diagonaloffsets of the non-zero diagonals ofA in VAL. Low ertriangular diagonals have negative offsets, them ain diagonalhas offset 0 , and uppertriangular diagonals have positive offset. Elem ents of \(\mathbb{D}\) IA G ofM UST be sorted in increasing order.

ND IA G num berofnon-zero diagonals in A.

B 0 rectangular array w ith firstdim ension LD B .

LD B leading dim ension of \(B\)

BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch amay of length LW ORK. On exit, if LW ORK = -1,W ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK aray. LW ORK should be at leastM.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=M\) *N_CPU \(S\) where \(N\) _CPU \(S\) is the \(m\) axim um num berof processors available to the program .

IfLW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK anray, retums this value as the firstentry of theW ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov fn csd/Staff/k Rem ington/tspoblas/
"D ocum ent for the Basic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib.org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. No test for singularity ornear-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.
2. If U N ITD \(=4\), the routine scales the row sofA such that their 2 -norm sare one. The scaling \(m\) ay im prove the accuracy of the com puted solution. C orresponding entries ofV A L are changed only in the particular case. On retum DV m atrix stored as a vector contains the diagonalm atrix by w hich the row s have been scaled. U N ITD \(=2\) should be used for the next calls to the routine w ith overw ritten VA L and DV .

WORK ( 1 )=0 on retum if the scaling has been com pleted successfully, otherw ise \(W O R K(1)=-i w\) here \(i\) is the row num berw hich 2 -norm is exactly zero.
3. If \(D E S C R A(3)=1\) and \(U N\) ITD \(<4\), the unitdiagonalelem ents m ightorm ightnotbe referenced in the D IA representation of a sparse m atrix. They are notused anyw ay in these cases. ButifUN ITD = 4, the unit diagonalelem ents M U ST be referenced in the \(D \mathbb{I A}\) representation.
4.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse
m atrix \(A\) is used. H ow ever DESCRA (1) m ustbe equal to 3 in this case.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ddisna - com pute the reciprocalcondition num bers for the eigenvectors of a real sym \(m\) etric or com plex \(H\) em itian \(m\) atrix or forthe leftor right singular vectors of a general mby \(-\mathrm{n} m\) atrix

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DD ISNA (O B,M N,D,SEP, INFO)}

```
CHARACTER * 1 Job
\(\mathbb{N}\) TEGER \(\mathrm{M}, \mathrm{N}, \mathbb{N} F O\)
D OUBLE PRECISION D (*), SEP (*)
SU BROUTINE DD ISNA_64 (JOB,M,N,D,SEP, IN FO )
CHARACTER * 1 JOB
\(\mathbb{N}\) TEGER*8M,N, \(\mathbb{N} F O\)
D OUBLE PRECISION D (*), SEP (*)

\section*{F95 INTERFACE}

SU BROUTINED ISNA (OOB,M,N,D,SEP, [NFO])
CHARACTER (LEN=1) :: JOB
\(\mathbb{N} T E G E R:: M, N, \mathbb{N} F O\)
REAL (8), D IM ENSION (:) ::D ,SEP

SU BROUTINE D ISNA_64 (JO B, M , N, D , SEP, [iN FO ])
CHARACTER (LEN=1) :: JOB
\(\mathbb{N}\) TEGER (8) ::M , N , \(\mathbb{N} F O\)
REAL (8),D IM ENSION (:) ::D ,SEP

\section*{C INTERFACE}
\#include <sunperfh>
void ddisna (char job, intm , intn, double *d, double *sep, int*info);
void ddisna_64 (char jंjb, long m, long n, double *d, double *sep, long *info);

\section*{PURPOSE}
ddisna com putes the reciprocal condition num bers for the eigenvectors of a realsym m etric or com plex H erm itian m atrix orforthe left or rightsingular vectors of a general m-by-n matrix. The reciprocalcondition num ber is the gap' betw een the corresponding eigenvalue or singular value and the nearest other one.

The bound on the error, \(m\) easured by angle in radians, in the I-th com puted vector is given by

SLAMCH (E')* (ANORM /SEP (I))
where \(A N O R M=2-\) norm \((A)=\max (\operatorname{abs}(D(\mathcal{D}))\). SEP (I) is not allow ed to be sm allerthan SLAM CH (E')*ANORM in orderto lim it the size of the errorbound.

SD ISN A m ay also be used to com pute enrorbounds for eigenvectors of the generalized sym \(m\) etric definite eigenproblem .

\section*{ARGUMENTS}

JOB (input)
Specifies forw hich problem the reciprocal condition num bers should be com puted:
\(=E\) ': the eigenvectors of a sym m etric/H erm itian m atrix;
= IL ': the leftsingular vectors of a general
\(m\) atrix;
\(=\mathrm{R}\) ': the rightsingularvectors of a general \(m\) atrix.
\(M\) (input) The num ber of row \(s\) of the \(m\) atrix. \(M>=0\).

N (input) If \(\mathrm{OB}=\mathrm{L}\) 'or R ', the num ber of colum ns of the \(m\) atrix, in which case \(N>=0\). Ignored if \(\mathrm{JO} \mathrm{B}=\) E'

D (input) dim ension ( \(m\) in \(M, N\) )) if \(30 B=L\) ' or \(R^{\prime}\) The eigenvalues (if \(\mathrm{JOB}=\mathrm{E}\) ) or singular values (if \(J O B=L\) ' or \(R\) ) of the matrix, in either increasing or decreasing order. If singular values, they \(m\) ustbe non-negative.

SEP (output)
dimension ( \(m\) in \((M, N)\) ) if \(J O B=L^{\prime}\) or \(R^{\prime}\) The reciprocal condition num bers of the vectors.
\(\mathbb{I N} F O\) (output)
= 0: successfulexit.
<0: if \(\mathbb{N F O}=-\) i, the \(i\)-th argum ent had an illegal value.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

ddot-com pute the dotproductoftw o vectors }\textrm{x}\mathrm{ and }\textrm{y}

```

\section*{SYNOPSIS}
\[
\text { DOUBLE PRECISION FUNCTION DDOT } \mathbb{N}, \mathrm{X}, \mathbb{N} C X, Y, \mathbb{N} C Y)
\]
\(\mathbb{N}\) TEGERN, \(\mathbb{N C X}, \mathbb{N} C Y\)
DOUBLE PRECISION X (*), Y (*)
DOUBLE PRECISION FUNCTION DDOT_64 \(\mathbb{N}, \mathrm{X}, \mathbb{N} C X, Y, \mathbb{N} C Y)\)
\(\mathbb{N} T E G E R * 8 N, \mathbb{N} C X, \mathbb{N C Y}\)
DOUBLE PRECISION X (*), Y (*)
F95 INTERFACE
REAL (8) FUNCTION DOT ( \(\mathbb{N}], \mathrm{X},[\mathbb{N C X}], \mathrm{Y},[\mathbb{N C Y}]\) )
\(\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y\)
REAL (8),D IM ENSIO N (:) :: X,Y
REAL (8) FUNCTION DOT_64 (N ],X, [ \(\mathbb{N} C X], Y,[\mathbb{N} C Y])\)
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{I N C X}, \mathbb{N} C Y\)
REAL (8),D \(\mathbb{I M}\) ENSIO N (:) :: X,Y
C INTERFACE
\#include <sunperfh>
double ddot(intn, double *x, int incx, double *y, intincy);
double ddot_64 (long n, double *x, long incx, double *y, long

\section*{PURPOSE}
ddot com pute the dot productof \(x\) and \(y\) where \(x\) and \(y\) are n-vectors.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. IfN is not positive then the function retums the value 0.0. U nchanged on exit.
\(X\) (input)
( \(1+(\mathrm{n}-1) * \mathrm{abs}(\mathbb{N} C X)\) ). On entry, the increm ented array \(X\) must contain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of X. \(\mathbb{N} C X\) m ustnotbe zero. U nchanged on exit.
\(Y\) (input)
( \(1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C Y)\) ). On entry, the increm ented array \(Y\) m ust contain the vectory. U nchanged on exit.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of Y. \(\mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS

\section*{NAME}
ddoti-C om pute the indexed dotproduct.

\section*{SYNOPSIS}

DOUBLE PRECISION FUNCTION DDOTINZ, X, \(\mathbb{N D D} \mathrm{X}, \mathrm{Y})\)
DOUBLE PRECISION X (*) , Y (*)
\(\mathbb{N}\) TEGER NZ
\(\mathbb{I N} T E G E R \mathbb{N} D X(*)\)

DOUBLE PRECISION FUNCTION DDOTI_64 \(\mathbb{N} Z, X, \mathbb{N} D X, Y)\)
DOUBLE PRECISION X (*), Y ( \({ }^{*}\) )
\(\mathbb{N} T E G E R * 8 N Z\)
\(\mathbb{I N}\) TEGER*8 \(\mathbb{I N D X ( * )}\)
F95 \(\mathbb{I N}\) TERFACE
DOUBLE PRECISION FUNCTION DOTI(NZ],X, \(\mathbb{N} D X, Y)\)
REAL (8),D \(\mathbb{M}\) ENSION (:) :: X,Y
\(\mathbb{N}\) TEGER ::NZ

DOUBLE PRECISION FUNCTION DOTI_64 ( \(\mathbb{N} Z], X, \mathbb{N} D X, Y\) )
REAL (8),D \(\mathbb{M}\) ENSION (:) :: X,Y
\(\mathbb{N}\) TEGER (8) ::NZ
\(\mathbb{N}\) TEGER (8),D \(\mathbb{I}\) ENSION (:) :: \(\mathbb{N} D \mathrm{X}\)

\section*{PURPOSE}
fullstorage form.
```

dot $=0$
do $i=1, n$
$\operatorname{dot}=\operatorname{dot}+x(i)$ * $y($ ind $x(i))$
enddo

```

\section*{ARGUMENTS}

N Z (input)
N um ber of elem ents in the com pressed form .
U nchanged on exit.
\(X\) (input)
V ector in com pressed form . U nchanged on exit.
\(\mathbb{N} D X\) (input)
\(V\) ector containing the indiaes of the com pressed
form. It is assum ed that the elem ents in \(\mathbb{N} D X\) are distinctand greaterthan zero. U nchanged on exit.

Y (input)
V ector in fullstorage form. O nly the elem ents corresponding to the indices in \(\mathbb{N}\) D X w illbe accessed.

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

delmm -E llpack form atm atrix-m atrix m ultiply

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DELLMM(TRANSA,M,N,K,A LPHA,DESCRA,}

* VAL,\mathbb{NDX,LDA,MAXNZ,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER TRANSA,M,N,K,DESCRA (5),LDA,MAXNZ,
* LDB,LDC,LWORK
INTEGER INDX(LDA,MAXNZ)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL (LDA,MAXNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NEDELLMM_64(TRANSA,M,N,K,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,LDA,MAXNZ,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER*8 TRANSA,M,N,K,DESCRA (5),LDA,MAXNZ,}
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 }\mathbb{N}DX(LDA,MAXNZ)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL (LDA,MAXNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE ELLMM (TRANSA,M, N ],K,ALPHA,DESCRA,VAL, INDX,}

* [LDA],MAXNZ,B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])
\mathbb{NTEGER TRANSA,M,K,MAXNZ}
INTEGER,D\mathbb{M ENSION (:) :: DESCRA}
INTEGER,D\mathbb{M ENSION (:,:):: \mathbb{NDX}}\mathbf{N}=\mp@code{N}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D IM ENSION (:,:) :: VAL,B,C

```
SUBROUTINE ELLMM _64(TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N} D X\),

\section*{DESCRIPTION}
where A LPH A and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrioes, A is a m atrix represented in Ellpack form at form at and op (A) is one of
\(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conj}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate w ith m atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate w ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M \(\quad\) um berof row \(s\) in \(m\) atrix A

N \(\quad\) Num berof colum ns in \(m\) atrix \(C\)

K \(\quad N\) um berof \(c o l u m n s\) in \(m\) atrix A

A LPH A Scalarparam eter
DESCRA () D escriptor argum ent. Five elem ent integer aray
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(A=A\) )
2 : H erm itian ( \(\mathrm{A}=\mathrm{CONJG}\) (A ))
3 :Triangular
4 : Skew (Anti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJ}\) ( A ))
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonaltype
0 : non-unit
1 :unit

DESCRA (4) A ray base NOT IM PLEM ENTED)
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? (NOT IM PLEM ENTED)
0 : unknown
1 : no repeated indices
VAL ( 0 tw o-dim ensionallD A -by M A XNZ array such thatVA L ( \(\mathrm{I}, \mathrm{:}\) ) consists of non-zero elem ents in row IofA, padded by zero values if the row contains less than M AXN Z .
\(\mathbb{I N D X} 0 \quad\) tw o-dim ensional integer LD A by -M A XN Z aray such \(\mathbb{N} D \mathrm{X}\) ( \(I\), :) consists of the colum n indices of the nonzero elem ents in row \(I\), padded by the integer value I if the num berof nonzeros is less than M AXNZ.

LD A leading dim ension ofVAL and \(\mathbb{N D} X\).

MAXNZ max num berofnonzeros elem ents per row .
B 0 rectangular array w th first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w th first dim ension LD C .

LD C leading dim ension of \(C\)
W ORK ( scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK anay. LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dellsm -E llpack form at triangular solve

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDELLSM(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,}

* VAL,INDX,LDA,MAXNZ,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,M,N,UNITD,DESCRA (5),LDA,MAXNZ,}
* LDB,LDC,LWORK
INTEGER INDX (LDA,MAXNZ)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION DV M),VAL (LDA,MAXNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NE DELLSM _64(TRANSA,M ,N,UNTID,DV,A LPHA,DESCRA,}
* VAL, \mathbb{NDX,LDA,MAXNZ,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER*8 TRANSA,M,N,UNITD,DESCRA (5),LDA,MAXNZ,
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 INDX (LDA,MAXNZ)}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION DV M ),VAL (LDA,MAXNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}

SUBROUTINE ELLSM (TRANSA, M, \(\mathbb{N}], U N \mathbb{I T}, D V, A L P H A, D E S C R A, V A L\), * \(\mathbb{N} D X,[L D A], M A X N Z, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{I N T E G E R}\) TRANSA, M, MAXNZ
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: DESCRA
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:, :) :: \(\mathbb{N D X}\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D \(\mathbb{M}\) ENSION (:) :: DV
DOUBLE PRECISION,D \(\mathbb{I M} E N S I O N(:,:):\) VAL,B,C

SUBROUTINE ELLSM _64(TRANSA, M, \(\mathbb{N}], U N \mathbb{T} D, D V, A L P H A, D E S C R A, V A L\),
* \(\mathbb{N} D X,[L D A], M A X N Z, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N} T E G E R * 8\) TRANSA, M, MAXNZ
\(\mathbb{N} T E G E R * 8, D \mathbb{M}\) ENSION (:) :: DESCRA
\(\mathbb{N} T E G E R * 8, D \mathbb{M}\) ENSION (: : : : : \(\mathbb{N} D X\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D IM ENSION (:) :: DV
D OUBLE PREC ISION,D IM ENSION (: : : :: VAL, B, C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { op (A)B + BETA C } \quad C<-A L P H A D \text { op (A) B + BETA C } \\
& C<-A L P H A \text { op(A)D B + BETA C }
\end{aligned}
\]
where A LPHA and BETA are scalar, C and B are m by n dense \(m\) atrices, \(D\) is a diagonalscaling \(m\) atrix, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix represented in Ellpack form at and op (A) is one of \(\mathrm{op}(\mathrm{A})=\operatorname{inv}(\mathrm{A})\) or \(\mathrm{op}(\mathrm{A})=\operatorname{inv}\left(\mathrm{A}^{\prime}\right)\) or \(\mathrm{op}(\mathrm{A})=\operatorname{inv}\left(\mathrm{conjg}\left(\mathrm{A}^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix 0 : operate with m atrix 1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in \(m\) atrix A
N \(\quad\) Uum berof colum ns in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity \(m\) atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)
4 :A utom atic row scaling (see section N O TES for furtherdetails)

DV 0 A ray of length M containing the diagonalentries of the scaling diagonalm atrix D .

A LPH A Scalarparam eter
DESCRA 0 D escriptor argum ent. Five elem ent integer aray DESCRA (1) m atrix structure
```

    0 :general
    1 : symm etric ( }A=A\mathrm{ ) )
    2:H erm Itian (A = CON JG (A ))
    3:Triangular
    4 :Skew (A nti)-Symm etric (A=-A )
    5 :D iagonal
    6:Skew Herm titian (A= CON JG (A ) )
    N ote:For the routine, only D ESCRA (1)=3 is supported.
    D ESCRA (2) upper/low er triangular indicator
        1 : low er
        2 :upper
    DESCRA (3) m ain diagonaltype
        0:non-unit
        1 :unit
    DESCRA (4) A ray base NOT IM PLEM ENTED )
        0 : C C ++ com patible
        1 :Fortran com patible
    DESCRA (5) repeated indices? (NOT IM PLEM ENTED)
        0 :unknow n
        1: no repeated indices
    VAL () tw o-dim ensionalLD A foy M A X N Z array such thatV A L (I,:)
consists of non-zero elem ents in row IofA, padded by
zero values if the row contains less than M AXN Z .
INDX () tw o-dim ensionalintegerLD A boy-M A XN Z array such
\mathbb{N D X (I,:) consists of the colum n indiges of the}
nonzero elem ents in row I, padded by the integer
value I if the num berofnonzeros is less than M A XN Z .
The colum n indices M U ST be sorted in increasing order
foreach row .
LDA leading dim ension ofV A L and \mathbb{ND X .}
M A X N Z m ax num ber ofnonzeros elem ents per row .
B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension ofC

```
    W ORK () scratch amay of length LW ORK.
        On exit, ifLW ORK \(=-1, W\) ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK array. LW ORK should be at leastM .
Forgood penform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=\mathrm{M}\) *N_CPUS where N_CPUS is the m axim um num berof processors available to the program .

IfLW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of theW ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
htep://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such tests \(m\) ustbe perform ed before calling this routine.
2. IfUN ITD \(=4\), the routine scales the row s of \(A\) such that their 2 -nom s are one. The scaling \(m\) ay im prove the accuracy of the com puted solution. C orresponding entries ofV A L are changed only in the particular case. On retum DV m atrix stored as a vector contains the diagonalm atrix by w hich the row s have been scaled. U N ITD = 2 should be used for the next calls to the routine w ith overw ritten VAL and DV.

WORK ( 1 )=0 on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here i is the row num berw hich 2 -norm is exactly zero.
3. If \(\operatorname{DESCRA}(3)=1\) and U N ITD < 4, the unitdiagonalelem ents \(m\) ightorm ightnotbe referenced in the ELL representation of a sparse m atrix. They are notused anyw ay in these cases.

ButifUN ITD \(=4\), the unit diagonalelem ents M U ST be referenced in the ELL representation.
4.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse m atrix A is used. H ow ever DESCRA (1) m ustbe equal to 3 in this case.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
dezftb - com putes a periodic sequence from its Fourier coefficients. D EZFTB is a sim plified butslow erversion of DFFTB .

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DEZFTB N,R,A ZERO,A,B,W SAVE)}

```

\section*{\(\mathbb{N}\) TEGER N}

DOUBLE PRECISION AZERO
DOUBLE PRECISION R (*), A (*), B (*), W SAVE ( \({ }^{*}\) )
SU BROUTINEDEZFTB_64 \(\mathbb{N}, R, A Z E R O, A, B, W\) SAVE)
\(\mathbb{N}\) TEGER*8N
DOUBLE PRECISION AZERO
DOUBLE PRECISION R (*),A (*), B (*), W SAVE ( \({ }^{*}\) )

\section*{F95 INTERFACE}

SU BROUTINE DEZFTB \(\mathbb{N}, R, A Z E R O, A, B, W\) SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL (8) ::A ZERO
REAL (8),D IM ENSION (:) ::R,A,B,W SAVE
SU BROUTINEDEZFTB_64 \(\mathbb{N}, R, A Z E R O, A, B, W\) SAVE)
\(\mathbb{N} T E G E R(8):: N\)
REAL (8) ::AZERO
REAL (8),D IM ENSION (:) ::R,A,B,W SAVE

\section*{C INTERFACE}
\#include < sunperfh>
void dezftb (intn, double *r, double azero, double *a, double *b, double *w save);
void dezftb_64 (long n, double *r, double azero, double *a, double *b, double *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be synthesized. The m ethod ism ostefficientw hen N is the product of smallprim es. \(N>=0\).

R (output)
On exit, the Fourier synthesis of the inputs.
AZERO (input)
On entry, the constant Fourier coefficient A 0 . U nchanged on exit.

A (input/output)
On entry, array that contains the rem aining Fourier coefficients. On exit, these arrays are unchanged.

B (input/output)
On entry, amay that contains the rem aining Fourier coefficients. On exit, these arrays are unchanged.

W SAVE (input)
On entry, an array with dim ension of at least (3 * \(\mathrm{N}+15\) ), in inialized by D EZFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
dezfff-com putes the Fourier coefficients of a periodic sequence. DEZFTF is a sim plified butslow erversion of DFFTF.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DEZFTF N,R,AZERO,A,B,W SAVE)}

```

\section*{\(\mathbb{N}\) TEGER N}

DOUBLE PRECISION AZERO
DOUBLE PRECISION R (*) , A (*) , B (*) , W SAVE (*)
SUBROUTINEDEZFTF_64 \(\mathbb{N}, R, A Z E R O, A, B, W\) SAVE)
\(\mathbb{N}\) TEGER*8N
DOUBLE PRECISION AZERO
DOUBLE PRECISION R (*),A (*), B (*), W SAVE ( \({ }^{*}\) )

\section*{F95 INTERFACE}

SU BROUTINE DEZFTF \(\mathbb{N}, R, A Z E R O, A, B, W\) SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL (8) ::A ZERO
REAL (8),D IM ENSION (:) ::R,A,B,W SAVE
SUBROUTINEDEZFTF_64 \(\mathbb{N}, R, A Z E R O, A, B, W\) SAVE)
\(\mathbb{N} T E G E R(8):: N\)
REAL (8) ::AZERO
REAL (8),D \(\mathbb{I}\) ENSION (:) ::R,A,B,W SAVE

\section*{C INTERFACE}
\#include < sunperfh>
void dezfff(intn, double *r, double azero, double *a, double *b, double *w save);
void dezftf_64 (long \(n\), double *r, double azero, double *a, double *b, double *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. The m ethod is m ostefficientw hen N is the product of sm allprim es. \(N>=0\).
\(R\) (input/output)
O \(n\) entry, a real array of length \(N\) containing the sequence to be transform ed. On exit, \(R\) is unchanged.

AZERO (output)
On exit, the sum from \(i=1\) to \(i=n\) ofr(i) \(/ n\).

A (input/output)
On entry, array that contains the rem aining Fourier coefficients. On exit, these amays are unchanged.

B (input/output)
On entry, array that contains the rem aining Fourier coefficients. On exit, these arrays are unchanged.

W SAVE (input)
O n entry, an array with dim ension of at least (3 * \(\mathrm{N}+15\) ), initialized by D EZFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
dezfti-initializes the array W SAVE, which is used in both DEZFTF and DEZFTB.

\section*{SYNOPSIS}

SUBROUTINEDEZFTIN,W SAVE)
\(\mathbb{N}\) TEGER N
DOUBLE PRECISION W SAVE (*)
SUBROUTINEDEZFTI_64 N, W SAVE)
\(\mathbb{N}\) TEGER*8 N
DOUBLE PRECISION W SAVE (*)
F95 INTERFACE
SU BROUTINE DEZFTIN,W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL (8),D IM ENSION (:) ::W SAVE
SUBROUTINEDEZFTI_64 N,W SAVE)
\(\mathbb{N}\) TEGER (8) ::N
REAL (8), D IM ENSION (:) ::W SAVE

\section*{C INTERFACE}
\#include <sunperfh>
void dezfti(intn, double *w save);
void dezfti_ 64 (long n, double *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. \(\mathrm{N}>=0\).

W SAVE (input)
O n entry, an array w ith a dim ension of at least (3
* \(\mathrm{N}+15\) ). The sam ew ork array can be used for both DEZFTF and DEZFTB as long as N remains unchanged. D ifferent W SAVE arrays are required fordifferentvalues of \(N\). This initialization does not have to be repeated betw een calls to DEZFTF orDEZFTB as long as N and W SAVE remain unchanged, thus subsequent transform \(s\) can be obtained faster than the first.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
dfft2b -com pute a periodic sequence from its Fourier coefficients. The D FFT operations are unnom alized, so a call ofD FFT2F follow ed by a callof D FFT2B w ill multiply the input sequence by \(\mathrm{M} * \mathrm{~N}\).

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDFFT2B(PLACE,M,N,A,LDA,B,LDB,W ORK,LW ORK)}
CHARACTER * 1 PLACE
\mathbb{NTEGERM,N,LDA,LDB,LW ORK}
DOUBLE PRECISION A (LDA,*),B (LDB,*),W ORK (*)
SUBROUT\mathbb{NE DFFT2B_64(PLACE,M ,N,A,LDA,B,LDB,W ORK,LW ORK)}
CHARACTER * 1 PLACE
INTEGER*8 M ,N,LDA,LDB,LW ORK
DOUBLE PRECISION A (LDA,*),B (LDB,*),W ORK (*)
F95 INTERFACE
SUBROUT\mathbb{NE FFT2B (PLACE, M ], N ],A, [LDA ],B,[LDB],W ORK,LW ORK)}
CHARACTER (LEN=1)::PLACE
\mathbb{NTEGER ::M ,N,LDA,LDB,LW ORK}
REAL (8),D IM ENSION (:) ::W ORK
REAL (8),D IM ENSION (:,:) ::A,B
SUBROUT\mathbb{NE FFT2B_64(PLACE, M ], N ],A, [LDA ],B,[LDB],W ORK,LW ORK)}
CHARACTER (LEN=1)::PLACE
\mathbb{NTEGER (8)::M ,N,LDA,LDB,LW ORK}
REAL (8),D IM ENSION (:) ::W ORK
REAL (8),D IM ENSION (:,:) ::A ,B

```

\section*{C INTERFACE}
\#include <sunperfh>
void dfft2b (charplace, intm , intn, double *a, int lda, double *b, int ldlb, double *w ork, int lw ork);
void dfffib_64 (charplace, long m, long n, double *a, long lda, double *b, long ldlo, double *w ork, long Iw ork);

\section*{ARGUMENTS}

\section*{PLACE (input)}

C haracter. IfPLA CE = 'I'or 'i' (for in-place), the input and outputdata are stored in anray A. IfPLACE = \(0^{\prime}\) or \(b^{\prime}\) (for out-of-place), the input data is stored in array \(B\) while the output is stored in A.

M (input) Integer specifying the num ber of row \(s\) to be transform ed. It is m ost efficientw hen M is a productofsm allprim es. \(M>=0\); when \(M=0\), the subroutine retums im mediately w ithoutchanging any data.

N (input) Integerspecifying the num ber of colum ns to be transform ed. It is \(m\) ostm ostefficientw hen \(N\) is a productofsm allprim es. \(\mathrm{N}>=0\); when \(\mathrm{N}=0\), the subroutine retums im \(m\) ediately \(w\) thout changing any data.

A (input/output)
Realarray ofdim ension ( \(L D A, N\) ). On entry, the tw o-dim ensional array \(A(L D A, N)\) contains the input data to be transform ed if an in-place transform is requested. O therw ise, it is not referenced. U pon exit, results are stored in \(A(1: M, 1: N)\).

LD A (input)
Integer specifying the leading dim ension ofA. If an out-of-place transform is desired LDA \(>=\mathrm{M}\). Else if an in-place transform is desired LDA >= 2* \(\mathrm{M} / 2+1\) )

B (input/output)
Realarray of dim ension ( \(2 *\) LD \(B, N\) ). On entry, if an out-of-place transform is requested \(B\) contains the inputdata. O therw ise, \(B\) is not referenced. \(B\) is unchanged upon exit.

LD B (input)
Integer. If an out-of-place transform is desired, \(2 *\) LD B is the leading dim ension of the amay B which contains the data to be transform ed and \(2 * \operatorname{LDB}>=2 *(2+1)\). O therw ise it is notreferenced.

W ORK (input/output)
O ne-dim ensional real array of length at least LW ORK. On input, W ORK m usthave been initialized by D FFT2I.

LW ORK (input)
Integer. LW ORK >= \(M+2 \star \mathrm{~N}+\mathrm{MAX} \mathrm{M}, 2 \star \mathrm{~N})+30\) )

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
dfft2f-com pute the Fourier coefficients of a periodic sequence. The DFFT operations are unnorm alized, so a call ofD FFT2F follow ed by a callof D FFT2B w ill multiply the input sequence by \(\mathrm{M} * \mathrm{~N}\).

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE DFFT2F(PLACE,FULL,M,N,A,LDA,B,LDB,W ORK,LW ORK)}
CHARACTER * 1 PLACE,FULL
\mathbb{NTEGERM,N,LDA,LDB,LW ORK}
DOUBLE PRECISION A (LDA,*),B (LDB,*),W ORK (*)
SUBROUT\mathbb{NE DFFT2F_64(PLACE,FULL,M ,N,A,LDA,B,LDB,W ORK,LW ORK)}
CHARACTER * 1 PLACE,FULL
INTEGER*8 M ,N ,LDA,LDB,LW ORK
DOUBLE PRECISION A (LDA,*),B (LDB,*),W ORK (*)

```
F95 INTERFACE
SU BROUTINE FFT2F (PLACE,FULL, \(\mathbb{M}], \mathbb{N}], A,[L D A], B,[L D B], W\) ORK,
    LW ORK)
CHARACTER (LEN=1) ::PLACE,FULL
INTEGER ::M,N,LDA,LDB,LW ORK
REAL (8),D IM ENSION (:) ::W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A , B
SU BROUTINE FFT2F_64 (PLACE,FULL, M ], \(\mathbb{N}], A,[L D A], B,[L D B], W O R K\),
    LW ORK)
CHARACTER (LEN=1) :: PLACE,FULL
\(\mathbb{N}\) TEGER (8) ::M ,N,LDA,LDB,LW ORK
REAL (8),D IM ENSION (:) ::W ORK

REAL (8), D \(\mathbb{M} \operatorname{ENSION}(:,:\) ) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void dffi2f(charplace, char full, intm, intn, double *a, int lda, double *b, intldb, double *w ork, int lw ork);
void dffi2f 64 (charplace, char fill, long m, long n, double *a, long lda, double *b, long ldb, double *w ork, long lw ork);

\section*{ARGUMENTS}

PLACE (input)
Character. IfPLA CE = I'or 1' (for in-place), the input and outputdata are stored in array A. IfPLACE = \(\mathrm{O}^{\prime}\) or \(\mathrm{b}^{\prime}\) (for out-of-place), the input data is stored in array B while the output is stored in A.

FULL (input)
Indicatesw hether or not to generate the full result \(m\) atrix. \(F^{\prime}\) or ' \(\mathrm{I}^{\prime} \mathrm{w}\) ill cause D FFT 2F to generate the full resultm atrix. O therw ise only a partial \(m\) atrix that takes advantage of sym \(m\) etry w illbe generated.
\(M\) (input) Integer specifying the num ber of rows to be transform ed. It is m ost efficientw hen M is a productofsm allprim es. \(M>=0\); when \(M=0\), the subroutine retums im mediately w ithoutchanging any data.

N (input) Integerspecifying the num ber of colum ns to be transform ed. It is m ostm ostefficientw hen N is a productofsm allprim es. \(\mathrm{N}>=0\); when \(\mathrm{N}=0\), the subroutine retums im \(m\) ediately \(w\) thout changing any data.

A (input/output)
O n entry, a tw o-dim ensional aray A (LDA,N) that contains the data to be transform ed. U pon exit, A is unchanged if an out-of-place transform is done. If an in-place transform w ith partial result is requested, \(A(1:(M / 2+1) \star 2,1 \mathbb{N})\) w ill contain the transform ed results. If an in-place transform w ith full result is requested, \(A(1: 2 * M, 1: N) w i l l\)
contain com plete transform ed results.
LD A (input)
Leading dim ension of the array containing the data to be transform ed. LDA \(m\) ust be even if the transform ed sequences are to be stored in A.

IfPLACE \(=\left(\mathrm{O}^{\prime}\right.\) 'or \(\left.\mathrm{b}^{\prime}\right)\) LDA \(>=\mathrm{M}\)
IfPLACE \(=\) ( \(I\) 'or \({ }^{\prime}\) ') LD A m ustbe even. If

FULL is not ( \(\mathrm{F}^{\prime}\) or f ), LDA \(>=(\mathrm{M} / 2+1\) ) 2

B (input/output)
U pon exi, a tw o-dim ensionalarray B ( \(2 \star\) LD B , N ) that contains the transform ed results if an out-ofplace transform is done. O therw ise, B is not used.

If an out-of-place transform is done and FU LL is not \(\mathrm{F}^{\prime}\) or \(\mathrm{'}^{\prime}\) ' \(\mathrm{B}(1:(\mathrm{M} / 2+1) * 2,1 \mathbb{N})\) w ill contain the partial transform ed results. IfFU \(L L=F\) 'or ' f ', \(\mathrm{B}(1: 2 * \mathrm{M}, 1 \mathbb{N})\) w ill contain the com plete transform ed results.

LD B (input)
\(2 *\) LD B is the leading dim ension of the array \(B\). If an in-place transform is desired LD B is ignored.

IfPLACE is ( 0 'or \(b^{\prime}\) ) and
FULL is ( F 'or \({ }^{\prime}\) ' ), LD B \(>=\mathrm{M}\)

FU LL is not ( F 'or \(\mathrm{If}^{\prime}\) ), LD B \(>=\mathrm{M} / 2+1\)

N ote thateven though LD B is used in the argum ent list, 2*LD B is the actual leading dim ension ofB.

W ORK (input/output)
O ne-dim ensional real array of length at least
LW ORK. On input, W ORK m usthave been initialized by D FFT2I.

LW ORK (input)
Integer. LW ORK >= \(M+2 \star N+M A X(M, 2 \star N)+30)\)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
dfft2i-initialize the array W SA VE, which is used in both the forw ard and backw ard transform s.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DFFT2IM ,N,W ORK)}

```
\(\mathbb{I N}\) TEGERM,N
DOUBLE PRECISION W ORK (*)
SU BROUTINE DFFT2I_64 (,\(N\), W ORK)
\(\mathbb{N}\) TEGER*8 M , N
DOUBLE PRECISION W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE FFT2IM ,N,W ORK)
\(\mathbb{N} T E G E R:: M, N\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ORK

SU BROUTINE FFT2I_64 M ,N,W ORK)
\(\mathbb{N} T E G E R(8):: M, N\)
REAL (8),D IM ENSION (:) ::W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void dffl2i(intm , intn, double *w ork);
void dfft2i_64 (long m, long n, double *w ork);

\section*{ARGUMENTS}

M (input) N um ber of row s to be transform ed. \(\mathrm{M}>=0\).

N (input) N um ber of colum ns to be transform ed. \(\mathrm{N}>=0\).

W ORK (input/output)
On entry, an aray ofdim ension \(M+2 * N+M A X M\), \(2 * N\) ) +30 ) orgreater. D FFT2I needs to be called only once to initialize aray W O RK before calling DFFT2F and/or D FFT2B if M, N andW ORK rem ain unchanged betw een these calls. Thus, subsequent transform s or inverse transform s of sam e size can be obtained faster than the first since they do not require initialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
dfft3b -com pute a periodic sequence from its Fourier coefficients. The D FFT operations are unnorm alized, so a call of DFFT 3F follow ed by a callofD FFT 3B w illm ultiply the input sequence by \(M * N * K\).

\section*{SYNOPSIS}
```

SUBROUTINEDFFT3B(PLACE,M,N,K,A,LDA,B,LDB,W ORK,LW ORK)
CHARACTER * 1 PLACE
INTEGERM,N,K,LDA,LDB,LW ORK
DOUBLE PRECISION A (LDA,N,*),B (LDB ,N,*),W ORK (*)
SU BROUT\mathbb{NE DFFT3B_64(PLACE,M ,N,K,A,LDA,B,LD B,W ORK,LW ORK)}
CHARACTER * 1 PLACE
INTEGER*8 M ,N,K,LDA,LDB,LW ORK
DOUBLE PRECISION A (LDA,N,\star),B(LDB,N,*),W ORK (*)

```
F95 INTERFACE
SU BROUTINE FFT3B (PLACE, \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], B,[L D B], W\) ORK,
    LW ORK)
CHARACTER (LEN=1) ::PLACE
\(\mathbb{N} T E G E R:: M, N, K, L D A, L D B, L W\) ORK
REAL (8),D IM ENSION (:) ::W ORK
REAL (8), D IM ENSIO N (:,:,:) ::A , B
SU BROUTINE FFT3B_64 (PLACE, \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], B,[L D B], W\) ORK,
    LW ORK)
CHARACTER (LEN=1) ::PLACE
\(\mathbb{N}\) TEGER (8) ::M , N, K, LDA, LD B, LW ORK
REAL (8),D IM ENSION (:) ::W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void dffll3b (charplace, intm, intn, intk, double *a, int lda, double *b, int ldb, double *W ork, int lw ork);
void dffi3b_64 (charplace, long m, long n, long k, double
*a, long lda, double *b, long ldb, double *w ork, long lw ork);

\section*{ARGUMENTS}

PLACE (input)
Selectan in-place ('I'or i') or out-of-place ( \(\mathrm{D}^{\prime}\) 'or \(\mathrm{b}^{\prime}\) ) transform .

M (input) Integer specifying the num ber of row s to be transform ed. It is m ost efficientw hen M is a productofsm allprim es. \(M>=0\); when \(M=0\), the subroutine retums im mediately w ithoutchanging any data.

N (input) Integerspecifying the num ber of colum ns to be transform ed. It is m ost efficientw hen N is a productof.sm allprim es. \(\mathrm{N}>=0\); w hen \(\mathrm{N}=0\), the subroutine retums im mediately w ithoutchanging any data.

K (input) Integer specifying the num ber of planes to be transform ed. It is m ost efficientw hen K is a product ofsm allprim es. \(\mathrm{K}>=0\); when \(K=0\), the subroutine retums im mediately w ithoutchanging any data.

A (input/output)
O n entry, the three-dim ensional aray A (LD A, \(N, K\) ) contains the data to be transform ed if an in-place transform is requested. O therw ise, it is not referenced. Upon exit, results are stored in A \((\mathbb{M}, 1 \mathbb{N}, 1: K)\).

LD A (input)
Integer specifying the leading dim ension of A. If an out-of-place transform is desired LDA \(>=M\). Else if an in-place transform is desired LDA \(>=\) 2* ( \(/ 2+1\) )

B (input/output)
Realaray ofdim ension \(B(2 * L D B, N, K)\). On entry, if an out-ofplace transform is requested
\(\left.B\left(1: 2^{\star} M / 2+1\right), 1 \mathbb{N}, 1: K\right)\) contains the input data.
O therw ise, \(B\) is not referenced. \(B\) is unchanged upon exit.

LD B (input)
If an out-of-place transform is desired, \(2 * \mathrm{LDB}\) is
the leading dim ension of the array \(B\) which contains the data to be transform ed and \(2 *\) LD B \(>=\) \(2 * M / 2+1)\). O therw ise it is not referenced.

W ORK (input/output)
O ne-dim ensional real anray of length at least LW ORK. On input, W ORK m usthave been initialized by DFFT3I.

LW ORK (input)
Integer. LW ORK >= \(M+2 *(\mathbb{N}+K)+4 * K+45)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}

\section*{dffl3f-com pute the Fourier coefficients of a real periodic} sequence. The D FFT operations are unnorm alized, so a call of DFFT 3F follow ed by a callofD FFT 3B w illm ultiply the input sequence by \(M * N * K\).

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDFFT3F(PLACE,FULL,M,N,K,A,LDA,B,LDB,W ORK,LW ORK)}
CHARACTER * 1 PLACE,FULL
INTEGERM,N,K,LDA,LDB,LW ORK
DOUBLE PRECISION A (LDA,N,*),B (LDB ,N,*),W ORK (*)
SU BROUT\mathbb{NE DFFT3F_64 (PLACE,FULL,M ,N ,K,A LD A ,B,LDB,W ORK,}
LW ORK)
CHARACTER * 1 PLACE,FULL
INTEGER*8M,N,K,LDA,LDB,LW ORK
DOUBLE PRECISION A (LDA,N,\star),B (LDB,N,*),W ORK (*)

```

\section*{F95 INTERFACE}

SUBROUTINE FFT3F (PLACE,FULL, \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], B\), [LDB], W ORK,LW ORK)

CHARACTER (LEN=1) ::PLACE,FULL
\(\mathbb{N}\) TEGER ::M,N,K,LDA,LDB,LW ORK
REAL (8),D IM ENSION (:) ::W ORK
REAL (8), D IM ENSION (:,:,:) ::A,B
SU BROUTINE FFT3F_64 (PLACE,FULL, \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], B,[L D B]\), W ORK,LW ORK)

CHARACTER (LEN=1) ::PLACE,FULL
\(\mathbb{N}\) TEGER (8) ::M ,N,K,LDA,LDB,LW ORK

REAL (8), D IM ENSION (:) ::W ORK
REAL (8), D IM ENSION (:,:,:) ::A , B

\section*{C INTERFACE}
\#include <sunperfh>
void dffllef(charplace, char full, intm, intn, int , double *a, int lda, double *b, int ldb, double *w ork, intlw ork);
void dffl3f 64 (charplace, char full, long \(m\), long \(n\), long k, double *a, long lda, double *b, long ldb, double *w ork, long lw ork);

\section*{ARGUMENTS}

PLACE (input)
Selectan in-place (I'or li) or out-of-place ( D 'or b ) transform .

FU LL (input)
Selecta full (F' or 'I') or partial (' )
representation of the results. If the caller
selects fiull representation then an \(\mathrm{M} \times N \times K\) real array w ill transform to produce an M xN xK com plex array. If the caller does not select full representation then an \(\mathrm{M} \times N \times K\) real array \(w i l l\) transform to a \(M / 2+1) x N\) xK complex array that takes advantage of the sym \(m\) etry properties of a transform ed realsequence.
\(M\) (input) Integer specifying the num ber of row s to be transform ed. It is most efficientw hen \(M\) is a productofsm allprim es. \(\mathrm{M}>=0\); when \(\mathrm{M}=0\), the subroutine retums im mediately withoutchanging any data.

N (input) Integer specifying the num ber of colum ns to be transform ed. It is m ost efficientw hen N is a product ofsm allprim es. \(\mathrm{N}>=0\); when \(\mathrm{N}=0\), the subroutine retums im mediately without changing any data.
\(K\) (input) Integer specifying the num ber of planes to be transform ed. It is m ost efficientw hen K is a productofsm allprim es. \(K>=0\); when \(K=0\), the subroutine retums im \(m\) ediately without changing any data.

A (input/output)
O n entry, a three-dim ensional array A (LD A \(N, K\) )
that contains input data to be transform ed. On
exit, if an in-place transform is done and FU LL is
not \(\mathrm{F}^{\prime}\) or \(\left.\mathrm{I}^{\prime}, \mathrm{A}\left(1: 2^{*} \mathrm{M} / 2+1\right), 1 \mathbb{N}, 1: K\right)\) w illcon-
tain the partialtransform ed results. If FU LL \(=\) F'or ' \(\mathrm{I}^{\prime}, \mathrm{A}(1: 2 \star \mathrm{M}, 1 \mathbb{N}, 1: \mathrm{K})\) w illcontain the com plete transform ed results.

LD A (input)
Leading dim ension of the array containing the data to be transform ed. LD A must be even if the transform ed sequences are to be stored in A.

IfPLACE \(=\left(\right.\) ( 'or \(\left.^{\prime}\right)\) LD A \(>=\mathrm{M}\)
IfPLACE \(=\left(\right.\) I'or \({ }^{\prime}\) ) LD A m ustbe even. If
\(\left.F U L L=\left(F^{\prime} \text { or }{ }^{\prime}\right)^{\prime}\right), L D A>=2 * M\)

FULL is not ( \(\mathrm{F}^{\prime}\) or \(\mathrm{I}^{\prime}\) ), LDA \(>=2 \star \mathrm{M} / 2+1\) )
B (input/output)
U pon exit, a three-dim ensional array \(B(2 * L D B, N, K)\) that contains the transform ed results if an out-of-place transform is done. O therw ise, \(B\) is not used.

If an out-of-place transform is done and FULL is not \(\mathrm{F}^{\prime}\) or \(\left.\mathrm{I}^{\prime}, \mathrm{B}\left(1 \cdot 2^{\star} \mathrm{M} / 2+1\right), 1 \mathbb{N}, 1 \mathrm{~K}\right) \mathrm{w}\) illcontain the partialtransform ed results. If FU LL \(=\) F 'or 'f', B ( \(1: 2 * \mathrm{M}, 1\) N, \(1 \mathrm{~K})\) w ill contain the com plete transform ed results.

LD B (input)
\(2 * L D B\) is the leading dim ension of the array \(B\). If an in-place transform is desired LD B is ignored.

IfPLACE is ( 0 'or b') and

FULL is ( F 'or ' F ), then LD B >=M
FULL is not ( F ' or ' F ), then \(\mathrm{LD} \mathrm{B}>=\mathrm{M} / 2+1\)
\(N\) ote thateven though LD B is used in the argum ent list, \(2 *\) LD \(B\) is the actual leading dim ension ofB.

W ORK (input/output)
O ne-dim ensional real array of length at least
LW ORK. W ORK m usthave been initialized by DFFT 3I.

LW ORK (input)
Integer. LW ORK >= \(M+2 \star(\mathbb{N}+K)+4 * K+45)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
dffl3i-initialize the array W SAVE, which is used in both DFFT3F and DFFT3B.

\section*{SYNOPSIS}

SUBROUTINEDFFT3IM,N,K,WORK)
\(\mathbb{N}\) TEGERM, N , K
DOUBLE PRECISION W ORK (*)
SU BROUTINE DFFT3I_64 \(M, N, K, W\) ORK)
\(\mathbb{N}\) TEGER*8 M , N , K
DOUBLE PRECISION W ORK ( \({ }^{( }\))
F95 INTERFACE
SU BROUTINE FFT3IM , N, K, W ORK)
\(\mathbb{N} T E G E R:: M, N, K\)
REAL (8), D IM ENSION (:) ::W ORK

SU BROUTINE FFT3I_64 M , N, K , W ORK)
\(\mathbb{N} T E G E R(8):: M, N, K\)
REAL (8),D IM ENSION (:) ::W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void dffl3i(intm , intn, intk, double *w ork);
void dfftii 64 (long m, long n, long k, double *w ork);

\section*{ARGUMENTS}
\(M\) (input) N um ber of row s to be transform ed. \(\mathrm{M}>=0\).

N (input) N um ber of colum ns to be transform ed. \(\mathrm{N}>=0\).

K (input) N um ber of planes to be transform ed. \(\mathrm{K}>=0\).

W ORK (input/output)
O n entry, an array ofdim ension \(M+2 *(\mathbb{N}+K)+\) 30) or greater. D FFT 3I needs to be called only once to initialize array \(W\) ORK before calling DFFT3F and/or DFFT3B ifM , N, K and W ORK rem ain unchanged betw een these calls. Thus, subsequent transform s or inverse transform s of sam e size can be obtained fasterthan the first since they do not require initialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
dfftb -com pute a periodic sequence from its Fourier coefficients. The D FFT operations are unnorm alized, so a call of D FFTF follow ed by a callofD FFTB w ill multiply the input sequence by N .

\section*{SYNOPSIS}
\[
\text { SU BROUTINE DFFTB } \mathbb{N}, \mathrm{X}, \mathrm{~W} \text { SAVE) }
\]
\(\mathbb{N}\) TEGER N
DOUBLE PRECISIONX (*), W SAVE (*)
SU BROUTINE DFFTB_64 \(\mathbb{N}, \mathrm{X}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER*8N
DOUBLE PRECISION X (*) , W SAVE (*)

\section*{F95 INTERFACE}

SU BROUTINE FFTB ( \(\mathbb{N}\) ],X,W SAVE)
\(\mathbb{N}\) TEGER :: N
REAL (8),D \(\mathbb{M}\) ENSION (:) ::X,W SAVE
SUBROUTINE FFTB_64 ( \(\mathbb{N}\) ],X,W SAVE)
\(\mathbb{N}\) TEGER (8) :: N
REAL (8), D IM ENSION (:) ::X,W SAVE

\section*{C INTERFACE}
\#include <sunperfh>
void dfftb (intn, double *x, double *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product ofsm allprim es. \(\mathrm{N}>=0\).

X (input) On entry, an array of length N containing the sequence to be transform ed.

W SAVE (input) O n entry, W SAVE m ustbe an array ofdim ension (2 * \(\mathrm{N}+15\) ) orgreater and m usthave been initialized by D FFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
dfflf-com pute the Fourier coefficients of a periodic sequence. The FFT operations are unnorm alized, so a callof D FFTF follow ed by a callofD FFTB w ill multiply the input sequence by N .

\section*{SYNOPSIS}
```

SUBROUTINEDFFTF N,X,W SAVE)

```
\(\mathbb{N}\) TEGER N
DOUBLE PRECISIONX (*), W SAVE (*)
SU BROUTINE DFFTF_64 \(\mathbb{N}, \mathrm{X}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER*8N
DOUBLE PRECISION X (*) , W SAVE (*)

\section*{F95 INTERFACE}

SUBROUTINE FFTF ( \(\mathbb{N}\) ],X,W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::X,W SAVE
SU BROUTINE FFTF_64 (N ], X,W SAVE)
\(\mathbb{N}\) TEGER (8) :: N
REAL (8),D \(\mathbb{M}\) ENSION (:) ::X,W SAVE

\section*{C INTERFACE}
\#include <sunperfh>
void dfff(intn, double *x, double *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product ofsm allprim es. \(\mathrm{N}>=0\).
\(X\) (input) \(O n\) entry, an array of length \(N\) containing the sequence to be transform ed.

W SAVE (input) O n entry, W SAVE m ustbe an array ofdim ension (2 * \(\mathrm{N}+15\) ) orgreater and m usthave been initialized by D FFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
dffti-initialize the array W SAVE, which is used in both DFFTF and DFFTB.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DFFTIN,W SAVE)}

```
\(\mathbb{N}\) TEGER N
DOUBLE PRECISION W SAVE (*)
SU BROUTINEDFFTI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER*8 N
DOUBLE PRECISION W SAVE (*)

\section*{F95 INTERFACE}

SU BROUTINE FFTIN, W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W SAVE

SU BROUTINE FFTI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
\(\mathbb{N} T E G E R(8):: N\)
REAL (8),D IM ENSION (:) ::W SAVE

\section*{C INTERFACE}
\#include < sunperfh>
void dfflidintn, double *w save);
void dffti_ 64 (long n, double *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. \(\mathrm{N}>=0\).

W SAVE (input)
O n entry, an array ofdim ension ( \(2 * N+15\) ) or greater. D FFT I needs to be called only once to initialize array W ORK before calling D FFTF and/or D FFTB if N and W SAVE rem ain unchanged betw een these calls. Thus, subsequent transform \(s\) or inverse transform s of sam e size can be obtained faster than the first since they do not require indialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE

\section*{NAME} dfftopt - com pute the length of the closest fast FFT

\section*{SYNOPSIS}
\(\mathbb{N}\) TEGER FUNCTION DFFTOPT (LEN)
\(\mathbb{N}\) TEGER LEN
\(\mathbb{N}\) TEGER*8FUNCTION DFFTOPT_64 (LEN)
\(\mathbb{N}\) TEGER*8 LEN

F95 INTERFACE
\(\mathbb{N}\) TEGER FUNCTION DFFTOPT (LEN)
\(\mathbb{N}\) TEGER ::LEN
\(\mathbb{N}\) TEGER (8) FUNCTION DFFTO PT_64 (LEN)
\(\mathbb{N} T E G E R(8):: L E N\)
C INTERFACE
\#include <sunperfh>
intdfftopt(int len);
long dfffopt 64 (long len);

\section*{PURPOSE}

Fourier transform algorithm s , including those used in Perform ance L ibrary, w ork bestw ith vector lengths that are products of sm all prim es. Forexam ple, an FFT of length \(32=2 * * 5 \mathrm{w}\) ill nun fasterthan an FFT of prime length 31 because 32 is a productofsm allprim es and 31 is not. If your application is such that you can taper or zero pad your vector to a larger length then this function \(m\) ay help you select a better length and run your FFT faster.

DFFTOPT will retum an integerno sm aller than the input argum entN that is the closestnum ber that is the product of sm allprim es. D FFTO PT w ill retum 16 foran input of \(\mathrm{N}=16\) and retum 18=2*3*3 foran inputof \(\mathrm{N}=17\).

N ote that the length com puted here is not guaranteed to be optim al, only to be a product of sm allprim es. A lso, the value retumed \(m\) ay change as the underlying
FFT sbecom e capable of handling larger prim es. For exam ple, passing in \(N=51\) to day \(w\) ill retum \(52=2 \star 2 \star 13\) rather than \(51=3 * 17\) because the FFT s in Perform ance Library do not have fast radix 17 code. In the future, radix 17 code m ay be added
and then \(\mathrm{N}=51 \mathrm{w}\) ill retum 51 .

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dffiz - initialize the trigonom etric w eight and factor tables or com pute the forw ard FastFourier T ransform of a double precision sequence.

\section*{SYNOPSIS}

\(\mathbb{N}\) TEGER \(\mathbb{I O P T}, N, \mathbb{F} A C(*), L W O R K, \mathbb{E R R}\)
DOUBLE COM PLEX Y (*)
DOUBLE PRECISION X (*), SCALE, TRIGS (*), WORK (*)
SU BROUTINEDFFTZ_64 (TOPT,N,SCALE, X,Y,TRIGS, FAC,W ORK,LW ORK, ERR)
\(\mathbb{N} T E G E R * 8 \mathbb{I} P T, N, \mathbb{F A C}(*), L W O R K, \mathbb{E R R}\)
DOUBLE COM PLEX Y (*)
D OUBLE PRECISION X (*), SCALE, TRIGS (*),WORK (*)

\section*{F95 INTERFACE}

SU BROUTINE FFT (IOPT,N,SCALE,X,Y,TRIGS, IFAC,W ORK, [LW ORK ], ERR)
\(\mathbb{N}\) TEGER, \(\mathbb{N} T E N T(\mathbb{N})::\) IOPT
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N})\),OPTIONAL ::N,LW ORK
REAL (8), \(\mathbb{N}\) TENT \((\mathbb{N})\),OPTIONAL ::SCALE
REAL (8), \(\mathbb{N} T E N T(\mathbb{N})\),D \(\mathbb{I M}\) ENSION (:) ::X
COM PLEX (8), \(\mathbb{I N T E N T}\) (OUT),D \(\mathbb{M}\) ENSION (:) ::Y
REAL (8), \(\mathbb{N} T E N T(\mathbb{N} O U T)\), D \(\mathbb{M}\) ENSION (:) ::TRIGS
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{I M}\) ENSION (:) :: \(\mathbb{F A C}\)
REAL (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:) ::W ORK
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(O U T):: \mathbb{E R R}\)

SU BROUTINE FFT_64 (IOPT, \(\mathbb{N}],[S C A L E], X, Y, T R I G S, \mathbb{F A C}, W\) ORK, [LW ORK ], \(\mathbb{E R R}\) )
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N}):: \operatorname{IOPT}\)
\(\mathbb{N} T E G E R(8), \mathbb{N} T E N T(\mathbb{N})\), OPTIONAL :: N , LW ORK
REAL (8), \(\mathbb{N} T E N T(\mathbb{N})\), OPTIONAL :: SCALE
REAL (8), \(\mathbb{N} \operatorname{TENT}(\mathbb{N}), D \mathbb{M} E N S I O N(:):: X\)
COM PLEX (8), \(\mathbb{N}\) TENT (OUT), D \(\mathbb{M}\) ENSION (:) ::Y
REAL (8), \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} E N S I O N(:):: T R I G S\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N} O U T), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{F A C}\)
REAL (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:) ::W ORK
\(\mathbb{N}\) TEGER (8), \(\mathbb{N}\) TENT (OUT) :: \(\mathbb{E R R}\)

\section*{C INTERFACE}
\#include <sunperfh>
void dfflz_ (int*iopt, int*n, double *scale, double *x, doublecom plex *y, double *trigs, int *ifac, double *W ork, int *lw ork, int *ienc);
void dfftz_64_ (long *iopt, long *n, double *scale, double \({ }^{*} \mathrm{x}\), doublecom plex *y, double *trigs, long *ifac, double *w ork, long *lw ork, long *ienc);

\section*{PURPOSE}
dfftz initializes the trigonom etric w eight and factor tables or com putes the forw ard FastFourier T ransform of a double precision sequence as follow s:
\[
\mathrm{N}-1
\]

Y (k) = scale * SUM W *X (i)
\(=0\)
where
k ranges from 0 to \(\mathrm{N}-1\)
\(i=\operatorname{sqnt}(-1)\)
isign \(=-1\) for forw ard transform
\(W=\exp \left(i s i g n \star i^{\star} j^{\star} k \star 2 \star p i N N\right)\)
In real-to-com plex transform of length \(N\), the \((\mathbb{N} / 2+1)\) com plex output data points stored are the positive-frequency half of the spectrum of the D iscrete Fourier T ransform. The other half can be obtained through com plex conjugation and therefore is notstored.

\section*{ARGUMENTS}

IO PT (input)
Integer specifying the operation to be perform ed:

IO PT \(=0\) com putes the trigonom etric weight table and factor table
IO PT = -1 com putes forw ard FFT

N (input)
Integer specifying length of the input sequence \(X\). N is \(m\) ostefficientw hen it is a product of sm all prim es. \(\mathrm{N}>=0\). U nchanged on exit.

SCALE (input)
D ouble precision scalarby w hich transform results are scaled. Unchanged on exit. SCALE is defaulted to 1.0 D 0 forF95 \(\mathbb{I N}\) TERFA CE .

X (input) On entry, X is a realanay whose first N elem ents contain the sequence to be transform ed.

Y (output)
D ouble com plex array w hose first \(\mathbb{N} / 2+1\) ) elem ents contain the transform results. \(X\) and \(Y ~ m\) ay be the sam e array starting at the sam e \(m\) em ory location, in which case the dim ension of \(\mathrm{X} m\) ustbe at least \(2 *(N / 2+1)\). O therw ise, it is assum ed that there is no overlap betw een \(X\) and \(Y\) in \(m\) em ory.

TR IG S (input/output)
D ouble precision array of length \(2 * N\) that contains the trigonom etric w eights. The w eights are com puted w hen the routine is called w ith \(\mathbb{I O}\) PT \(=0\) and they are used in subsequent calls w hen 10 PT \(=-1\). U nchanged on exit.

FAC (input/output)
Integer array of dim ension at least 128 that contains the factors of N . The factors are com puted when the routine is called w ith IO PT = 0 and they are used in subsequent calls where \(10 P T=-1\). U nchanged on exit.

W ORK (w orkspace)
D ouble precision array ofdim ension at least N . The user can also choose to have the routine allocate its ow n w orkspace (see LW ORK).

LW ORK (input)
Integer specifying workspace size. IfLW ORK = 0, the routine w illallocate its ow n w orkspace.

ERR (output)
On exit, integer \(\mathbb{E R R}\) has one of the follow ing
values:
0 = norm alretum
\(-1=10 P T\) is not 0 or -1
\(-2=\mathrm{N}<0\)
\(-3=(L W O R K\) is not 0) and (LW ORK is less than N)
\(-4=m\) em ory allocation forw orkspace failed

\section*{SEE ALSO}
ffl

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO
- CAUTIONS

\section*{NAME}
dfftz2 -initialize the trigonom etric weight and factor
tables or com pute the tw o-dim ensional forw ard FastFourier
\(T\) ransform of a tw o-dim ensional double precision array.

\section*{SYNOPSIS}

SU BROUTINE DFFTZ2 (IOPT,N1,N2,SCALE, X,LDX,Y,LDY,TRIGS, FAC,WORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER \(\mathbb{I O P T}, N 1, N 2, L D X, L D Y, \mathbb{F A C}(*), L W O R K, \mathbb{E R R}\)
DOUBLE COM PLEX Y (LDY,*)
D OUBLE PRECISION X (LDX,*), SCALE, TRIGS (*), WORK (*)
SU BROUTINE DFFTZ2_64 (IOPT,N1,N2,SCALE, X,LDX,Y,LDY,TRIGS, FAC,W ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N} T E G E R * 8 \mathbb{I} P T, N 1, N 2, L D X, L D Y, \mathbb{F} A C(*), L W O R K, \mathbb{E R R}\)
DOUBLE COM PLEX Y (LDY,*)
D OUBLE PRECISIONX (LDX,*),SCALE,TRIGS (*),WORK (*)

\section*{F95 INTERFACE}

SU BROUTINE FFT2 (LOPT, \(\mathbb{N} 1], \mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S\), FAC,W ORK, [LW ORK], ERR)
\(\mathbb{N}\) TEGER, \(\mathbb{N} T E N T(\mathbb{N}):: \mathbb{I O P T}\)
\(\mathbb{N}\) TEGER, \(\mathbb{N} T E N T(\mathbb{N})\),OPTIONAL ::N1,N2,LDX,LDY,LW ORK
REAL (8), \(\mathbb{N}\) TENT \((\mathbb{N})\),OPTIONAL :: SCALE
\(\operatorname{REAL}(8), \mathbb{N} T E N T(\mathbb{N}), D \mathbb{M}\) ENSION \((:,:):: X\)
COM PLEX (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:,:) :: Y
REAL (8), \(\mathbb{N}\) TENT ( \(\mathbb{N} O U T\) ), D \(\mathbb{I M}\) ENSION (:) ::TRIGS
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{I M}\) ENSION (:) :: \(\mathbb{F A C}\)

REAL (8), \(\mathbb{N} T E N T\) (OUT),D \(\mathbb{M}\) ENSION (:) ::W ORK
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(O U T):: \mathbb{E R R}\)
SU BROUTINE FFT2_64 (DPPT, \(\mathbb{N} 1], \mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S\), تAC,W ORK, [LW ORK], ERR)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N}):: \mathbb{I O P T}\)
\(\mathbb{N}\) TEGER (8), \(\mathbb{N}\) TENT \((\mathbb{N})\), OPT IONAL ::N 1,N2,LDX,LDY,LW ORK
REAL (8), \(\mathbb{N}\) TENT \((\mathbb{N})\), OPTIONAL :: SCALE
REAL (8), \(\mathbb{N} T E N T(\mathbb{N}), D \mathbb{I}\) ENSION (:,:) ::X
COM PLEX (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:,:) ::Y
REAL (8), \(\mathbb{N}\) TENT ( \(\mathbb{N} O U T\) ), D \(\mathbb{M}\) ENSION (:) ::TRIGS
\(\mathbb{N} T E G E R(8), \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M}\) ENSION (:) :: \(\mathbb{F}\) AC
REAL (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{I M}\) ENSION (:) ::W ORK
\(\mathbb{N} T E G E R(8), \mathbb{N} T E N T(O U T):: \mathbb{E R R}\)

\section*{C INTERFACE}
\#include < sunperfh>
void dfftz2_ (int*iopt, int*n1, int *n2, double *scale, double *x, int*ldx, doublecom plex *y, int*ldy, double *trigs, int *ifac, double *work, int *lw ork, int *ien);
void dfftz2_64_ (long *iopt, long *n1, long *n2, double *scale, double *x, long *ldx, doublecom plex *y, long *ldy, double *trigs, long *ifac, double *w ork, long *lw ork, long *ien);

\section*{PURPOSE}
dfftz2 initializes the trigonom etric weight and factor tables orcom putes the tw o-dim ensional forw ard FastFourier \(T\) ransform of a tw o-dim ensional double precision array. In com puting the tw o-dim ensionalFFT, one-dim ensionalFFT s are computed along the columns of the input array. O ne-dim ensionalFFT s are then com puted along the row s of the interm ediate results.

> N 2-1 N 1-1
\(Y(k 1, k 2)=\) scale * SUM SUM W 2*W \(1 * X(\mathcal{1},-2)\)
j2=0 \(\quad \mathfrak{l}=0\)
where
k 1 ranges from 0 to N 1-1 and \(k 2\) ranges from 0 to \(\mathrm{N} 2-1\)
\(i=\operatorname{sqrt}(-1)\)
isign \(=-1\) for forw ard transform
W \(1=\exp \left(\right.\) isign*i* \(\left.{ }^{1} * k 1 * 2 * p i / N 1\right)\)
W \(2=\exp \left(i s i g n \star i^{\star}{ }^{2}{ }^{2} k 2 * 2 * p i N 2\right)\)
In real-to-com plex transform of length \(N 1\), the \((N 1 / 2+1\) ) com -
plex output data points stored are the positive-frequency half of the spectrum of the D iscrete Fourier T ransform. The other half can be obtained through com plex conjugation and therefore is not stored.

\section*{ARGUMENTS}

IO PT (input)
Integer specifying the operation to be perform ed:
IO PT \(=0\) com putes the trigonom etric \(w\) eight table and factor table
IO PT \(=-1\) com putes forw ard FFT

N 1 (input)
Integer specifying length of the transform in the first dim ension. N 1 is m ostefficientw hen it is a productofsm allprim es. N \(1>=0\). U nchanged on exit.

N 2 (input)
Integer specifying length of the transform in the second dim ension. N 2 is m ostefficientw hen it is a productofsm allprim es \(\mathrm{N} 2>=0\). U nchanged on exit.

SCALE (input)
D ouble precision scalarby which transform results are scaled. Unchanged on exit. SCALE is defaulted to 1.0 D 0 forF95 \(\mathbb{N}\) TERFACE.

X (input) X is a double com plex aray of din ensions (LD X , N 2 ) that contains input data to be transform ed. X and \(Y\) can be the sam e array.

LD X (input)
Leading dim ension of X . LD X >=N1 if X is not the sam e aray as Y.Else, LD X = 2*LD Y. U nchanged on exit.

Y (output)
\(Y\) is a double com plex array of dim ensions (LD Y, \(N 2\) ) that contains the transform results. \(X\) and \(Y\) can be the sam e array starting at the sam e \(m\) em ory location, in which case the input data are overw ritten by their transform results. O therw ise, it is assum ed that there is no overlap betw een \(X\) and \(Y\) in \(m\) em ory.

LD Y (input)
Leading dim ension of Y . LD Y \(>=\mathrm{N} 1 / 2+1\) U nchanged on exit.

TR IG S (input/output)
D ouble precision array of length 2* (N1+N2) that contains the trigonom etric w eights. The w eights are com puted when the routine is called w ith IO PT
= 0 and they are used in subsequent calls w hen
IO PT \(=-1\). U nchanged on exit.

IFAC (input/output)
Integer array ofdim ension at least \(2 * 128\) that
contains the factors ofN 1 and N2. The factors are com puted when the routine is called w ith IO PT
= 0 and they are used in subsequent calls w hen IO PT \(=-1\). U nchanged on exit.

W ORK (w orkspace)
D ouble precision array of dimension at least
\(\operatorname{MAX}(\mathbb{N} 1,2 * N 2)\) where \(N C P U S\) is the num berof threads used to execute the routine. The user can also choose to have the routine allocate its own w orkspace (see LW ORK).
LW ORK (input)
Integer specifying workspace size. If LW ORK = 0, the routine w illallocate its ow n w orkspace.
\(\mathbb{E R R}\) (output)
On exit, integer \(\mathbb{E R R}\) has one of the follow ing
values:
0 = norm alretum
\(-1=\mathbb{I O P T}\) is not0 or -1
\(-2=\mathrm{N} 1<0\)
\(-3=\mathrm{N} 2<0\)
\(-4=(\operatorname{LDX}<\mathrm{N} 1)\) or (LD X notequal2*LD Y when X and
\(Y\) are sam e array)
\(-5=(\mathbb{L D Y}<\mathrm{N} 1 / 2+1)\)
\(-6=(L W O R K\) not equal 0) and (LWORK <
\(\operatorname{MAX}(\mathbb{N} 1,2 * N 2)\) )
\(-7=m\) em ory allocation failed

\section*{SEE ALSO}
fft

\section*{CAUTIONS}

On exit, outputarray \(Y(1: L D Y, 1 \mathbb{N} 2)\) is overw rilten.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO
- CAUTIONS

\section*{NAME}
dfftz3-initialize the trigonom etric weight and factor
tables or com pute the three-dim ensional forw ard FastFourier
Transform of a three-dim ensional double com plex array.

\section*{SYNOPSIS}

SU BROUT \(\mathbb{N} E\) DFFTZ 3 (IOPT,N 1,N 2,N 3, SCA LE, X ,LD X 1, LD X 2, Y ,LD Y 1, LD Y 2, TRIGS, \(\mathbb{F} A C, W\) ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER IOPT,N1,N2,N3,LDX1,LDX2,LDY1, LDY2, \(\mathbb{F A C}\) (*) \(^{*}\),
LW ORK, \(\mathbb{E R R}\)
DOUBLE COM PLEX Y (LDY1,LDY \(2, *\) )
DOUBLE PRECISION X (LDX1, LDX2, *), SCALE, TRIGS ( \(\left.{ }^{( }\right), \mathrm{W} O R K(\star)\)
SU BROUTINE DFFTZ3_64 (TOPT,N1,N2,N 3, SCALE,X,LDX 1,LD X 2, Y, LD Y 1, LD Y 2, TRIGS, FAC,WORK,LWORK, ERR)
\(\mathbb{N}\) TEGER*8 \(\mathbb{I O}\) PT,N1,N2,N3,LD X 1, LD X 2, LD Y 1, LD Y 2, \(\mathbb{F} A C(\star)\), LW ORK, \(\mathbb{E R R}\)
D OUBLE COM PLEX Y (LD Y 1,LDY 2, *)
DOUBLE PRECISION X (LDX1,LDX2,*),SCALE,TRIGS (*),W ORK (*)

\section*{F95 INTERFACE}

SUBROUTINE FFT3 (TOPT, \(\mathbb{N} 1], \mathbb{N} 2], \mathbb{N} 3],[S C A L E], X,[L D X 1], L D X 2, Y,[L D Y 1]\), LDY 2, TR IG S, \(\mathbb{F A C}, \mathrm{W} O R K\), [LW ORK], \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER, \(\mathbb{N} T E N T(\mathbb{N})::\) IOPT,LDX 2 ,LD Y 2
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N})\),OPTIONAL ::N1, N 2, N 3, LDX1, LD Y 1,
LW ORK
REAL (8), \(\mathbb{N}\) TENT \((\mathbb{N})\), OPTIONAL ::SCALE
REAL (8), \(\mathbb{N}\) TENT ( \(\mathbb{N}\) ), D \(\mathbb{I M}\) ENSION (:,:) ::X

COM PLEX（8）， \(\mathbb{I N T E N T}(O U T), D \mathbb{M} E N S I O N(:,:):: Y\)
REAL（8）， \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} E N S I O N(:):: T R I G S\)
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} E N S I O N(:):: \mathbb{F A C}\) REAL（8）， \(\mathbb{N}\) TENT（OUT），D \(\mathbb{M} \operatorname{ENSION}(:):\) W ORK \(\mathbb{N}\) TEGER， \(\mathbb{N} T E N T(O U T):: \mathbb{E R R}\)

SU BROU T \(\mathbb{N} E\) FFT3＿64（ \(\mathbb{O}\) PT， \(\mathbb{N} 1], \mathbb{N} 2], \mathbb{N} 3],[S C A L E], X,[L D X 1], L D X 2, Y\) ， ［LD Y 1］，LD Y 2，TR IG S， تAC，WORK，［LWORK］，正RR）
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N}):: \operatorname{IOPT}, L D X 2, L D Y 2\)
\(\mathbb{N}\) TEGER（8）， \(\mathbb{N}\) TENT \((\mathbb{N})\) ，OPT IONAL ：：N 1，N \(2, N 3, L D X 1, L D Y 1\) ，
LW ORK
REAL（8）， \(\mathbb{N} T E N T(\mathbb{N})\) ，OPTIONAL ：：SCALE
REAL（8）， \(\mathbb{N} \operatorname{TENT}(\mathbb{N})\) ，D \(\mathbb{M}\) ENSION（：，：）：：X
COM PLEX（8）， \(\mathbb{N}\) TENT（OUT），D \(\mathbb{M} E N S I O N(:,:):: Y\)
REAL（8）， \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} E N S I O N(:):: T R I G S\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N} O U T), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{F A C}\)
REAL（8）， \(\mathbb{N}\) TENT（OUT），D \(\mathbb{M} E N S I O N(:):\) W ORK
\(\mathbb{N}\) TEGER（8）， \(\mathbb{N}\) TENT（OUT）：： \(\mathbb{E R R}\)

\section*{C INTERFACE}
\＃include＜sunperfh＞
void dfflz3＿（int＊iopt，int＊n1，int＊n2，int＊n3，double ＊scale，double＊x，int＊ldx1，int＊ldx2，doub－ lecom plex＊y，int＊ldy1，int＊ldy2，double＊trigs， int＊ifac，double＊W ork，int＊lw ork，int＊ierr）；
void dffiz3＿64＿（long＊iopt，long＊n1，long＊n2，long＊n3， double＊scale，double＊x，long＊ldx1，long＊ldx2， doublecom plex＊y，long＊ldy1，long＊ldy2，double ＊trigs，long＊ifac，double＊w ork，long＊lw ork， long＊ienc）；

\section*{PURPOSE}
dfflz3 initializes the trigonom etric weight and factor tables or computes the three－dim ensional forw ard Fast Fourier T ransform of a three－dim ensional double com plex aray．

N 3－1 N 2－1 N 1－1
Y \((k 1, k 2, k 3)=\) scale＊SUM SUM SUM W 3＊W 2＊W 1＊X（式，飞，そ）
\(\mathfrak{j}=0 \quad \mathfrak{2}=0 \quad \mathfrak{j}=0\)
where
k 1 ranges from 0 to \(\mathrm{N} 1-1 ; \mathrm{k} 2\) ranges from 0 to \(\mathrm{N} 2-1\) and \(k 3\)
ranges from 0 to \(\mathrm{N} 3-1\)
\(i=\operatorname{sqnt}(-1)\)

\section*{ARGUMENTS}

IO PT (input)
Integer specifying the operation to be perform ed:
IO PT \(=0\) com putes the trigonom etric \(w\) eight table and factortable
IO P T \(=-1\) com putes forw ard FFT

N 1 (input)
Integer specifying length of the transform in the first dim ension. N 1 ism ostefficientw hen it is a product of.sm allprim es. N1>=0. U nchanged on exit.

N 2 (input)
Integerspecifying length of the transform in the second dim ension. N 2 ism ostefficientw hen it is a productofsm allprim es. N \(2>=0\). U nchanged on exit.

N 3 (input)
Integerspecifying length of the transform in the third dim ension. N 3 ism ostefficientw hen itis a productofsm allprim es. N \(3>=0\). U nchanged on exit.

SCALE (input)
D ouble precision scalarby which transform results are scaled. Unchanged on exit. SCALE is defaulted to 1 .OD 0 forF \(95 \mathbb{N}\) TERFA CE .

X (input) X is a double precision aray of dim ensions (LD X 1, LDX2, N3) that contains input data to be transform ed. \(X\) can be sam e array as \(Y\).

LD X 1 (input)
firstdim ension of \(X\). IfX is notsame array as Y, LDX1 >= N1 Else, LDX1 = 2*LDY1 Unchanged on exit.

LD X 2 (input)
second dim ension ofX. LDX \(2>=\) N 2 Unchanged on exit.

Y (output)
Y is a double com plex aray of dim ensions (LD Y 1,
LD Y 2, N 3) that contains the transform results. X and \(Y\) can be the sam \(e\) array starting at the sam \(e\) m em ory location, in which case the input data are overw rilten by their transform results. O therw ise, it is assum ed that there is no overlap betw een \(X\) and \(Y\) in \(m\) em ory.

\section*{LD Y 1 (input)}
firstdim ension ofY. LD Y \(1>=\) N \(1 / 2+1\) U nchanged on exit.

LD Y 2 (input)
second dim ension of \(Y\). If \(X\) and \(Y\) are the same aray, LD Y \(2=\) LD X 2 Else LD Y \(2>=\mathrm{N} 2\) U nchanged on exit.

\section*{TR IG S (input/output)}

D ouble precision array of length 2 * (N \(1+\mathrm{N} 2+\mathrm{N} 3)\) that contains the trigonom etric w eights. The w eights are com puted when the routine is called w ith IO PT = 0 and they are used in subsequent calls w hen IO PT \(=-1\). U nchanged on exit.

FAC (input/output)
Integeramay ofdim ension at least \(3 * 128\) that
contains the factors of \(1, \mathrm{~N} 2\) and N 3 . The factors are com puted \(w\) hen the routine is called \(w\) ith IO PT \(=0\) and they are used in subsequent calls when IO PT \(=-1\). U nchanged on exit.

W ORK (w orkspace)
D ouble precision array of dim ension at least MAX \(\mathbb{N}, 2 * N 2,2 * N 3)+16 * N 3) * N C P U S\) where NCPUS is the num ber of threads used to execute the routine. The user can also choose to have the routine allocate its ow n w orkspace (see LW ORK).

LW ORK (input)
Integer specifying workspace size. If LW ORK \(=0\), the routine w illallocate its ow n w orkspace.
\(\mathbb{E R R}\) (output)
On exit, integer \(\mathbb{E R R}\) has one of the follow ing
values:
0 = norm alretum
\(-1=10\) PT is not 0 or -1
\(-2=\mathrm{N} 1<0\)
\(-3=\mathrm{N} 2<0\)
\(-4=\) N \(3<0\)
\(-5=(\mathbb{L D} X 1<\mathrm{N} 1)\) or ( \(\mathbb{L D} \mathrm{X}\) notequal \(2 *\) LD \(Y\) when X
and \(Y\) are sam e array)
\(-6=(\mathbb{L D} 2<\mathrm{N} 2)\)
\(-7=(\mathbb{L} \mathrm{Y} 1<\mathrm{N} 1 / 2+1)\)
-8 = (LD Y \(2<N 2\) ) or (LD Y 2 notequal LD X 2 when \(X\) and \(Y\) are sam e array)
\(-9=(\mathbb{L W} O R K\) not equal 0) and (LW ORK < (MAX (N,2*N 2,2*N 3) + 16*N 3))
\(-10=m\) em ory allocation failed

\section*{SEE ALSO}
ff

\section*{CAUTIONS}

On exit, outputsubanay \(Y(1: L D Y 1,1 \mathbb{N} 2,1 \mathbb{N} 3)\) is overw ritten.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dfftzm -initialize the trigonom etric weight and factor tables or com pute the one-dim ensional forw ard FastF ourier \(T\) ransform of a set of double precision data sequences stored in a tw o-dim ensional array.

\section*{SYNOPSIS}
 LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER \(\mathbb{I O P T}, N 1, N 2, L D X, L D Y, \mathbb{F A C}(*), L W\) ORK, \(\mathbb{E R R}\)
D OUBLE PRECISION X (LDX,*), SCALE, TRIGS (*), W ORK (*)
DOUBLE COM PLEX Y (LDY,*)
SU BROUTINE DFFTZM_64 (TOPT,N1,N2, SCALE, X,LDX,Y,LDY,TRIGS, FFAC,W ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N} \operatorname{TEGER} * 8 \mathbb{I} P \mathrm{PT}, \mathrm{N} 1, N 2, L D \mathrm{X}, \mathrm{LD} \mathrm{Y}, \mathbb{F} A C(*), L W\) ORK, \(\mathbb{E R R}\)
DOUBLE PRECISION X (LDX,*), SCALE, TRIGS (*), W ORK (*)
DOUBLE COM PLEX Y (LDY,*)

\section*{F95 INTERFACE}

SU BROUTINE FFTM (IOPT, \(\mathbb{N} 1], \mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S\), ㅍAC,W ORK, [LW ORK], \(\mathbb{E R R}\) )
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N})::\) IOPT
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N})\),OPTIONAL ::N1,N2,LDX,LDY,LW ORK
REAL (8), \(\mathbb{N}\) TENT \((\mathbb{N})\),OPTIONAL :: SCALE
REAL (8), \(\mathbb{N}\) TENT ( \(\mathbb{N}\) ), D \(\mathbb{I M}\) ENSION (:,:) ::X
COM PLEX (8), \(\mathbb{I N T E N T}(\mathrm{OUT}), \mathrm{D} \mathbb{I}\) ENSION (: : : : : : Y
REAL (8), \(\mathbb{N}\) TENT ( \(\mathbb{N} O U T\) ), D \(\mathbb{I M}\) ENSION (:) ::TRIGS
\(\mathbb{N}\) TEGER, \(\mathbb{N}\) TENT ( \(\mathbb{N} O U T\) ), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{F A C}\)

SUBROUTINE FFTM _64 (IOPT, \(\mathbb{N} 1], \mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S\), تAC, W ORK, [LW ORK], ERR)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N}):: \mathbb{I O P T}\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} T E N T(\mathbb{N}), O P T \mathbb{I} N A L:: N 1, N 2, L D X, L D Y, L W\) ORK
REAL (8), \(\mathbb{N} T E N T(\mathbb{N})\),OPTIONAL :: SCALE
\(\operatorname{REAL}(8), \mathbb{N} T E N T(\mathbb{N}), D \mathbb{M}\) ENSION \((:,:):: X\)
COM PLEX (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:,:) :: Y
REAL (8), \(\mathbb{N}\) TENT ( \(\mathbb{N} O U T\) ), D \(\mathbb{M}\) ENSION (:) ::TRIGS
\(\mathbb{N} T E G E R(8), \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{I M}\) ENSION (:) :: \(\mathbb{F A C}\) REAL (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:) ::W ORK \(\mathbb{N} T E G E R(8), \mathbb{N} T E N T(O U T):: \mathbb{E R R}\)

\section*{C INTERFACE}
\#include < sunperfh>
void dfflzm _ (int *iopt, int *m, int *n, double *scale, dou-
ble *x, int *ldx, doublecom plex *y, int *ldy, dou-
ble *trigs, int *ifac, double *w ork, int *lw ork, int*enc);
void dfflem _64_ (long *iopt, long *m, long *n, double *scale, double *x, long *ldx, doublecom plex *y, long *ldy, double *trigs, long *ifac, double
*w ork, long *lw ork, long *ien);

\section*{PURPOSE}
dfflzm initializes the trigonom etric weight and factor tables or com putes the one-dim ensional forw ard FastFourier Transform of a setofdouble precision data sequences stored in a tw o-dim ensional array:

> N 1-1

Y \((k, l)=\) scale * SUM W *X (jl)
\[
\dot{j} 0
\]
where
\(k\) ranges from 0 to N 1-1 and lranges from 0 to N 2-1
\(i=\operatorname{sqrt}(-1)\)
isign \(=-1\) for forw ard transform
\(W=\exp \left(i s i g n * i^{\star} j^{*} k * 2 \star\right.\) piN 1)
In real-to-com plex transform of length \(N 1\), the \((N 1 / 2+1)\) com -
plex output data points stored are the positive-frequency half of the spectrum of the discrete Fourier transform. The other half can be obtained through com plex conjugation and therefore is not stored.

\section*{ARGUMENTS}

IO PT (input)
Integer specifying the operation to be perform ed:
IO PT \(=0\) com putes the trigonom etric weight table and factor table
IO \(P T=-1\) com putes forw ard FFT

N 1 (input)
Integer specifying length of the input sequences. N 1 is m ostefficientw hen it is a productofsm all prim es. N \(1>=0\). U nchanged on exit.

N 2 (input)
Integerspecifying num ber of input sequences. N 2 \(>=0\). U nchanged on exit.

SCALE (input)
D ouble precision scalarby which transform results are scaled. Unchanged on exit. SCALE is defaulted to 1.0 D 0 forF \(95 \mathbb{I N}\) TERFACE .

X (input) X is a double precision array of dim ensions (LD X , \(N\) 2) that contains the sequences to be transform ed stored in its colum ns.

\section*{LD X (input)}

Leading dim ension of \(X\). If \(X\) and \(Y\) are the same aray, LDX \(=2 *\) LD \(Y\) Else LD \(X>=N 1\) Unchanged on exit.

Y (output)
\(Y\) is a double com plex array of dim ensions (LD Y, N 2 ) that contains the transform results of the inputsequences. \(X\) and \(Y\) can be the sam \(e\) array starting at the sam e \(m\) em ory location, in which case the input sequences are overw rilten by their transform results. O therw ise, it is assum ed that there is no overlap betw een \(X\) and \(Y\) in \(m\) em ory.

LD Y (input)
Leading dim ension of Y . LD Y \(>=\mathrm{N} 1 / 2+1\) U nchanged on exit.

TR IG S (input/output)
D ouble precision array of length \(2 * \mathrm{~N} 1\) that con-
tains the trigonom etric w eights. The w eights are
com puted when the routine is called with \(\mathrm{IO} \mathrm{PT}=0\) and they are used in subsequent calls when IO PT = -1. U nchanged on exit.

IFAC (input/output)
Integer array of dim ension at least 128 that contains the factors of N 1 . The factors are com puted when the routine is called w ith \(\mathbb{I O}\) PT \(=0\) and they are used in subsequent calls when IOPT \(=-1\). U nchanged on exit.

W ORK (w orkspace)
D ouble precision array ofdin ension at least N 1. The user can also choose to have the routine allocate its ow n w orkspace (see LW ORK).

LW ORK (input)
Integerspecifying workspace size. IfLW ORK \(=0\),
the routine w illallocate its ow n w orkspace.
\(\mathbb{E R R}\) (output)
On exit, integer \(\mathbb{E R R}\) has one of the follow ing
values:
0 = norm alretum
\(-1=\mathbb{O P T}\) is not 0 or -1
\(-2=\mathrm{N} 1<0\)
\(-3=N 2<0\)
\(-4=(\mathbb{L D X}<\mathrm{N} 1)\) or (LD X notequal2*LD Y when X and
Y are sam e array)
\(-4=(\operatorname{LDY}<\mathrm{N} 1 / 2+1)\)
\(-6=(\mathbb{L W} O R K\) notequal0) and (LWORK <N1)
\(-7=m\) em ory allocation failed

\section*{SEE ALSO}
fft

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgbbrd -reduce a realgeneralm -by-n band \(m\) atrix \(A\) to upper
bidiagonal form B by an orthogonal transform ation

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DGBBRD NECT,M,N,NCC,KL,KU,AB,LDAB,D,E,Q,LDQ,}
PT,LDPT,C,LDC,W ORK,INFO)
CHARACTER * 1 VECT
\mathbb{N TEGERM,N,NCC,KL,KU,LDAB,LDQ,LDPT,LDC,INFO}
DOUBLE PRECISION AB (LDAB,*), D (*), E (*), Q (LDQ ,*),
PT (LDPT,*),C (LDC,*),W ORK (*)
SU BROUTINE DGBBRD_64 NECT,M ,N ,NCC,KL,KU,AB,LDAB,D,E,Q,LDQ,
PT,LDPT,C,LDC,W ORK,INFO)
CHARACTER * 1 VECT
\mathbb{NTEGER*8M,N,NCC,KL,KU,LDAB,LDQ,LDPT,LD C, INFO}
DOUBLE PRECISION AB (LDAB,*), D (*), E (*), Q (LDQ \&),
PT (LDPT,*),C (LDC,*),W ORK (*)

```
F95 INTERFACE
    SU BROUTINE GBBRD \(N E C T, M, \mathbb{N}], \mathbb{N C C}], K L, K U, A B,[L D A B], D, E, Q\),
        [LD Q ], PT, [LDPT ], C , [LD C ], [W ORK ], [ \(\mathbb{N} F \mathrm{~F}]\) )
    CHARACTER (LEN=1) ::VECT
    \(\mathbb{N} T E G E R:: M, N, N C C, K L, K U, L D A B, L D Q, L D P T, L D C, \mathbb{N F O}\)
    REAL (8),D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK
    REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::AB, \(\mathrm{Q}, \mathrm{PT}, \mathrm{C}\)
    SU BROUTINE GBBRD_64 \(N E C T, M, \mathbb{N}], \mathbb{N C C}], K L, K U, A B,[\operatorname{LDAB}], D, E\),
        \(Q,[L D Q], P T,[L D P T], C,[L D C],[W O R K],[\mathbb{N F O}])\)

CHARACTER ( \(L E N=1\) ) : : VECT
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{NCC}, \mathrm{KL}, \mathrm{KU}, \mathrm{LD} A B, L D Q, L D P T, L D C, \mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::AB, Q,PT, C

\section*{C INTERFACE}
\#include <sunperfh>
void dgbbord (charvect, intm, intn, intncc, int kl, int
ku, double *ab, intldab, double *d, double *e, double *q, int ldq, double *pt, int ldpt, double \({ }^{*} \mathrm{C}\), int \(1 d \mathrm{c}\), int *info);
void dgbbrd_64 (charvect, long m, long n, long ncc, long kl, long ku, double *ab, long ldab, double *d, double
*e, double *q, long ldq, double *pt, long ldpt, double * c , long ldc, long *info);

\section*{PURPOSE}
dgbbid reduces a realgeneralm -by-n band m atrix A to upper bidiagonal form \(B\) by an orthogonaltransform ation: \(Q^{\prime *} A\) * \(P=B\).

The routine com putes B, and optionally form s Q or P', or com putes \(Q\) *C for given \(m\) atrix \(C\).

\section*{ARGUMENTS}

\section*{VECT (input)}

Specifies w hether ornot the \(m\) atrices \(Q\) and \(P\) 'are
to be form ed. \(=\mathrm{N}\) ': do not form Q orP';
\(=Q\) ': form \(Q\) only;
\(=P^{\prime}:\) form \(P^{\prime}\) 'only;
\(=B^{\prime}:\) form both.

M (input) The num ber of row s of the matrix \(A . M>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NCC (input)
The num berof colum ns of the m atrix \(C . N C C>=0\).

KL (input)
The num ber of subdiagonals of the m atrix A.KL >= 0 .

KU (input)
The num ber of superdiagonals of the m atrix A. K U \(>=0\).

AB (input/output)
DOUBLE PRECISION array, dimension (LDAB,N) On entry, the \(m\)-by-n band \(m\) atrix \(A\), stored in row \(s 1\) to \(K L+K U+1\). The \(j\) th colum n of \(A\) is stored in the \(j\) th column of the array AB as follows: \(A B(k u+1+i-j)=A(i, j)\) for \(\max (1, j\) \(\mathrm{ku})<=i<=m\) in \((m, j+k l)\). On exit, \(A\) is overw rilten by values generated during the reduction.

LDAB (input)
The leading dim ension of the aray A. LD AB >= \(K L+K U+1\).

D (output)
D OUBLE PRECISION aray, dim ension (m in \(M, N)\) ) The diagonalelem ents of the bidiagonalm atrix \(B\).

E (output)
D O U BLE PREC ISIO N aray, dim ension ( \(m\) in \((M, N)-1\) ) The superdiagonalelem ents of the bidiagonalm atrix \(B\).

Q (output)
DOUBLE PRECISION aray, dim ension (LDQ, M) IfVECT = \(Q\) ' or \(B\) ', them -by \(m\) orthogonalm atrix \(Q\). If \(\mathrm{VECT}=\mathrm{N}\) 'or P ', the array Q is not referenced.

LDQ (input)
The leading dim ension of the anay \(Q\). LDQ >= \(m\) ax \((1, M)\) if \(V E C T=Q\) 'or \(B\) '; LD \(Q>=1\) otherw ise.

PT (output)
D OUBLE PRECISION array, dim ension (LDPT,N) If VECT
\(=\mathrm{P}\) 'or B ', the \(\mathrm{n}-\mathrm{by}-\mathrm{n}\) orthogonalm atrix P '. If
VECT = N 'or Q ', the amay PT is not referenced.
LDPT (input)
The leading dim ension of the array PT. LD PT >= \(\mathrm{max}(1, \mathrm{~N})\) if VECT \(=\mathrm{P}\) 'or B ; LDPT >= 1 otherw ise.

C (input/output)
DOUBLE PRECISION aray, dimension (LDCNCC) On entry, an \(m\)-by-ncc matrix \(C\). On exit, \(C\) is overw rilten by Q *C. C is notreferenced if \(\mathrm{NCC}=\) 0.

LD C (input)
The leading dim ension of the array \(C\). LD \(C>=\) \(\max (1, M)\) if \(N C C>0 ; L D C>=1\) if \(N C C=0\). W ORK (w orkspace)

D OUBLE PRECISIO N aray, dim ension (2*MAX \(M, N)\) )
\(\mathbb{N} F O\) (output)
= 0: successfulexit.
<0: if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgbcon -estim ate the reciprocal of the condition num ber of a real general band \(m\) atrix \(A\), in either the 1-norm or the infinity-norm,

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGBCON NORM,N,NSUB,NSUPER,A,LDA, PPIVOT,ANORM,}
RCOND,W ORK,W ORK2,INFO)
CHARACTER * 1 NORM
\mathbb{NTEGER N,NSUB,NSUPER,LDA, INFO}
\mathbb{NTEGER \mathbb{PIVOT (*),W ORK2(*)}}\mathbf{(*)}
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION A (LDA,*),W ORK (*)
SUBROUT\mathbb{NEDGBCON_64 NORM,N,NSUB,NSUPER,A,LDA, \mathbb{PIVOT,ANORM,}}\mathbf{N},\mp@code{N},
RCOND,W ORK,WORK2, INFO)
CHARACTER * 1NORM
\mathbb{NTEGER*8 N,NSUB,NSUPER,LDA, NNFO}
INTEGER*8 \mathbb{PIVOT (*),W ORK2 (*)}
DOUBLE PRECISION ANORM,RCOND
DOU BLE PRECISION A (LDA,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GBCON \(\mathbb{N} O R M, \mathbb{N}], N S U B, N S U P E R, A,[L D A], \mathbb{P} \mathbb{I} O T, A N O R M\), RCOND, [W ORK], [W ORK2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::NORM
\(\mathbb{N} T E G E R:: N, N S U B, N S U P E R, L D A, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}, \mathrm{W}\) ORK2
REAL (8) ::ANORM,RCOND

SU BROUTINE GBCON_64 \(\mathbb{N} O R M, \mathbb{N}], N S U B, N S U P E R, A,[L D A], \mathbb{P} I V O T, A N O R M\), RCOND, [WORK], [WORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::NORM
\(\mathbb{N}\) TEGER (8) :: N , N SUB, N SUPER, LDA, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}, \mathrm{W} O R K 2\)
REAL (8) ::ANORM,RCOND
REAL (8), D \(\mathbb{I M}\) ENSION (:) ::W ORK
REAL (8), D \(\mathbb{I M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dgbcon (charnorm, intn, intnsub, int nsuper, double
*a, int lda, int *ipivot, double anorm , double
*rcond, int*info);
void dgbcon_64 (charnorm, long n, long nsub, long nsuper, double *a, long lda, long *ịívot, double anorm , double *rcond, long *info);

\section*{PURPOSE}
dgbcon estim ates the reciprocal of the condition num berofa real general band matrix A, in either the 1 -nom orthe infinity-norm, using the LU factorization computed by SGBTRF.

A \(n\) estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as
\(\operatorname{RCOND}=1 /(\operatorname{norm}(A) * \operatorname{norm}(\operatorname{inv}(A)))\).

\section*{ARGUMENTS}

NORM (input)
Specifies w hether the 1-norm condition num ber or the infinity-norm condition num ber is required:
= ' 'or \(\mathrm{O}^{\prime}\) : 1-norm;
= I': Infinity-norm .

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

N SU B (input)
The num ber of subdiagonals \(w\) thin the band of \(A\).
N SUB \(>=0\) 。

\section*{N SU PER (input)}

The num ber of superdiagonals w ithin the band of A. N SU PER \(>=0\).

A (input) D etails of the LU factorization of the band \(m\) atrix A, as com puted by SGBTRF. U is stored as an upper triangularband \(m\) atrix \(w\) ith N SU B \(+N\) SU PER superdiagonals in row s 1 to NSUB+NSUPER+1, and them ultipliers used during the factorization are stored in row sN SU B +N SU PER + 2 to \(2 *\) N SU B + N SU PER +1 .

LD A (input)
The leading dim ension of the anay A. LDA >= \(2 *\) N SU B + N SU PER +1 .
\(\mathbb{P I V O T}\) (input)
The pivot indices; for \(1<=i<=N\), row \(i\) of the \(m\) atrix \(w\) as interchanged w ith row \(\mathbb{P}\) IV OT (i).

ANORM (input)
IfNORM = ' 1 'or 0 ', the 1 -norm of the original \(m\) atrix \(A\). IfNORM = \(I\) ', the infinity-norm of the originalm atrix A.

RCOND (output)
The reciprocal of the condition number of the \(m\) atrix \(A\), com puted as RCOND \(=1 /(\) norm (A) * nom (inv (A))).

W ORK (w orkspace)
dim ension ( \(3 * \mathrm{~N}\) )

W ORK 2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgbequ - com pute row and colum \(n\) scalings intended to equilibrate an M by -N band m atrix A and reduce its condition num ber

\section*{SYNOPSIS}
```

SUBROUTINEDGBEQU M,N,KL,KU,A,LDA,R,C,ROW CN,
COLCN,AMAX,INFO)

```
\(\mathbb{N}\) TEGER M, N, KL, KU, LDA, \(\mathbb{N}\) FO
DOUBLE PRECISION ROW CN, COLCN,AMAX
DOUBLE PRECISION A (LDA , \(\left.{ }^{\star}\right), R(\star), C(\star)\)
SU BROUTINEDGBEQU_64M,N,KL,KU,A,LDA,R,C,ROW CN,
    COLCN,AMAX, \(\mathbb{N} F O\) )
\(\mathbb{N}\) TEGER*8M,N,KL,KU,LDA, \(\mathbb{N} F O\)
DOUBLE PRECISION ROW CN, COLCN,AMAX
DOUBLE PRECISION A (LDA ,*), R (*), C (*)

\section*{F95 INTERFACE}

SU BROUTINE GBEQU ( \(\mathbb{M}], \mathbb{N}], K L, K U, A,[L D A], R, C\),
ROW CN,COLCN,AMAX,[ \(\mathbb{N} F O]\) )
\(\mathbb{N}\) TEGER :: \(\mathrm{M}, \mathrm{N}, \mathrm{KL}, \mathrm{KU}, \mathrm{LD} A, \mathbb{N} F \mathrm{O}\)
REAL (8) ::ROW CN,COLCN,AMAX
REAL (8),D \(\mathbb{I}\) ENSION (:) ::R,C
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A

SU BROUTINE GBEQU_64 (M) ROW CN,COLCN,AMAX, \(\mathbb{N} F O]\) )
\(\mathbb{N} T E G E R(8):: M, N, K L, K U, L D A, \mathbb{N} F O\)
REAL (8) :: ROW CN, COLCN,AMAX
REAL (8), D \(\mathbb{M}\) ENSION (:) :: R , C
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dgbequ (intm, intn, intkl, int ku, double *a, int
lda, double *r, double * c , double *row cn, double
*colcn, double *am ax, int *info);
void dgbequ_64 (long m, long n, long kl, long ku, double *a, long lda, double *r, double *c, double *row cn, double *colcn, double *am ax, long *info);

\section*{PURPOSE}
dgbequ com putes row and colum n scalings intended to equilibrate an M boy -N band m atrix A and reduce its condition num ber. \(R\) retums the row scale factors and \(C\) the colum \(n\) scale factors, chosen to try to \(m\) ake the largestelem ent in each row and colum \(n\) of the \(m\) atrix \(B \quad w\) ith elem ents \(B(i, j)=R(i) \star A(i, j) * C(i)\) have absolute value 1.

R (i) and C (i) are restricted to be betw een SM LN UM = sm allest safe num ber and B IG N UM = largestsafe num ber. U se of these scaling factors is notguaranteed to reduce the condition num berofA butw orks w ellin practice.

\section*{ARGUMENTS}

M (input) The num ber of row s of the matrix \(A . M>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

KL (input)
The num ber of subdiagonals \(w\) thin the band of \(A\). \(\mathrm{KL}>=0\) 。

KU (input)
The num ber of superdiagonals \(w\) thin the band of \(A\). \(K U>=0\) 。

A (input) The band \(m\) atrix \(A\), stored in row 1 to \(K L+K U+1\). The jth colum n of \(A\) is stored in the \(j\) th column of the array \(A\) as follow s: A \((k u+1+i-j, j)=A(i, j)\)
form ax \((1, j \mathrm{j} k)<=i<=m\) in \((m, j+k l)\).

LD A (input)
The leading dim ension of the aray A. LDA >= \(K L+K U+1\).

R (output)
If \(\mathbb{N} F O=0\), or \(\mathbb{N} F O>M, R\) contains the row scale
factors forA.
C (output)
If \(\mathbb{N} F O=0, C\) contains the colum \(n\) scale factors forA.
ROW CN (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O>M, R O W C N\) contains the ratio
of the sm allest \(R\) (i) to the largest \(R\) (i). If
ROW CN >= 0.1 and AM AX is neither too large nor too
sm all, it is notw orth scaling by \(R\).
COLCN (output)
If \(\mathbb{N} F O=0, C O L C N\) contains the ratio of the sm allest C (i) to the largestC (i). IfC OLCN >=0.1, it is notw orth scaling by \(C\).

\section*{AM AX (output)}

A bsolute value of largestm atrix elem ent. If A M A X is very close to overflow orvery close to underflow , the m atrix should be scaled.
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
<0: if \(\mathbb{I N}\) FO = -i, the i-th argum ent had an illegalvalue
>0: if \(\mathbb{N} F O=i\), and \(i\) is
\(<=\mathrm{M}\) : the i-th row ofA is exactly zero
> M : the ( \(j-\mathrm{M}\) ) -th collum n ofA is exactly zero

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dgbm v -perform one of them atrix-vectoroperations y :=
alpha*A *x + beta*y ory = alpha*A *x + beta* y

```

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE DGBMV (TRANSA,M,N,NSUB,NSUPER,A LPHA,A,LDA,X, NNCX,}
BETA,Y,\mathbb{NCY)}
CHARACTER * 1 TRANSA
\mathbb{NTEGERM,N,NSUB,NSUPER,LDA,}\mathbb{N}CX,\mathbb{NCY}
DOUBLE PRECISION ALPHA,BETA
D OUBLE PRECISION A (LDA,*),X (*),Y (*)
SU BROUT\mathbb{NE DGBM V_64 (TRANSA ,M ,N ,N SUB,N SUPER,A LPHA ,A,LDA,X,}
INCX,BETA,Y,\mathbb{NCY)}
CHARACTER * 1 TRANSA
\mathbb{NTEGER*8M,N,NSUB,NSUPER,LDA, INCX,INCY}
DOUBLE PRECISION ALPHA,BETA
D OU BLE PRECISION A (LDA,*),X (*),Y (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GBMV ([TRANSA], \(\mathbb{M}], \mathbb{N}], N S U B, N S U P E R, A L P H A, A,[L D A], X\), \([\mathbb{N} C X], B E T A, Y,[\mathbb{N C Y}])\)

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N} T E G E R:: M, N, N S U B, N S U P E R, L D A, \mathbb{N C X}, \mathbb{N} C Y\)
REAL (8) ::ALPHA,BETA
REAL (8), D IM ENSION (:) :: X,Y
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A
SU BROUTINE GBMV_64 ([TRANSA], \(\mathbb{M}], \mathbb{N}], N \operatorname{SUB}, N \operatorname{SUPER}, A L P H A, A,[L D A]\),
\(\mathrm{X},[\mathbb{N} C X], B E T A, Y,[\mathbb{N} C Y])\)

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N} T E G E R(8):: M, N, N S U B, N S U P E R, L D A, \mathbb{N} C X, \mathbb{N} C Y\)
REAL (8) ::ALPHA,BETA
REAL (8), D \(\mathbb{M}\) ENSION (:) :: X , Y
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dgbm v (Char transa, intm, intn, intnsub, int nsuper, double alpha, double *a, int lda, double *x, int incx, double beta, double *y, int incy);
void dgbm v_64 (char transa, long m, long n, long nsub, long nsuper, double alpha, double *a, long lda, double *x, long incx, double beta, double *y, long incy);

\section*{PURPOSE}
dgbm v perform sone of the \(m\) atrix-vector operations \(y:=\) alpha*A *x + beta*y ory := alpha*A *x + beta*y, where alpha and beta are scalars, \(x\) and \(y\) are vectors and \(A\) is an \(m\) by \(n\) band \(m\) atrix, w ith nsub sub-diagonals and nsupersuperdiagonals.

\section*{ARGUMENTS}

TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=\mathrm{N}\) 'or h ' \(\mathrm{y}:=\) alpha*A \(* \mathrm{x}+\) beta* y .
TRANSA = T'ort' \(\mathrm{y}:=\) alpha*A *x + beta* y .

TRANSA \(=\) C'ort' \(y:=\) alpha*A *x + beta* \(y\).
U nchanged on exit.

TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.
M (input)
O \(n\) entry, M specifies the num berof row \(s\) of the \(m\) atrix \(A . M>=0\). U nchanged on exit.

N (input)

O n entry, \(N\) specifies the num ber of colum ns of the \(m\) atrix \(A . N>=0\). U nchanged on exit.

NSUB (input)
On entry, NSUB specifies the number of subdiagonals of them atrix A.NSUB \(>=0\). U nchanged on exit.

N SU PER (input)
On entry, N SU PER specifies the num ber of superdiagonals of the m atrix A. N SU PER \(>=0\). U nchanged on exit.
ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
Before entry, the leading (nsub + nsuper+1) by \(n\) part of the array A m ust contain the matrix of coefficients, supplied colum \(n\) by colum \(n\), with the leading diagonal of the \(m\) atrix in row (nsuper+ 1 ) of the array, the first super-diagonal starting at position 2 in row nsuper, the firstsubdiagonal starting atposition 1 in row (nsuper + 2 ), and so on. Elem ents in the array A that do not comespond to elem ents in the band matrix (such as the top leftnsuperby nsupertriangle) are not referenced. The follow ing program segm ent w ill transfer a band \(m\) atrix from conventional full \(m\) atrix storage to band storage:
\[
\text { DO } 20, \mathrm{~J}=1, \mathrm{~N}
\]
\[
K=N S U P E R+1-J
\]
\[
\text { DO } 10, I=\text { M AX }(1, J-N \text { SU PER }), M \mathbb{N}(M, J+
\]
N SUB )
\[
A(K+I, J)=m \operatorname{atrix}(I, J)
\]

10 CONTINUE
20 CONTINUE
U nchanged on exit.
LDA (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A \(>=\) (
nsub + nsuper+ 1 ). U nchanged on exit.
X (input)
\((1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C X))\) when TRANSA \(=\mathrm{N}\) 'or
\(h^{\prime}\) and at least \((1+(m-1) * a b s(\mathbb{N} C X))\)
otherw ise. Before entry, the increm ented amay \(X\)
m ustcontain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X<>0\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then \(Y\) need notbe set on input. U nchanged on exit.

Y (input/output)
\((1+(m-1) \star \operatorname{abs}(\mathbb{N} C Y))\) when TRANSA \(=\mathrm{N}\) 'or \(h^{\prime}\) and at least \((1+(n-1) * a b s(\mathbb{N} C Y))\)
otherw ise. Before entry, the increm ented array \(Y\) m ust contain the vectory. On exit, Y is overw ritten by the updated vectory.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents ofY. \(\mathbb{N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}

> dgbrfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is banded, and provides errorbounds and backw ard error estim ates for the solution

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DGBRFS (TRANSA,N,KL,KU,NRHS,A,LDA,AF,LDAF,}
\mathbb{PNOT,B,LDB,X,LDX,FERR,BERR,W ORK,W ORK 2, INFO)}
CHARACTER * 1 TRANSA
INTEGERN,KL,KU,NRHS,LDA,LDAF,LDB,LDX, INFO
INTEGER \mathbb{PIVOT (*),W ORK2 (*)}
D OUBLE PRECISION A (LDA ,*),AF (LDAF,*), B (LDB,*), X (LDX,*),
FERR (*),BERR (*),W ORK (*)
SUBROUT\mathbb{NE DGBRFS_64 (TRANSA,N,KL,KU,NRHS,A,LDA,AF,LDAF,}
\mathbb{PIVOT,B,LDB,X,LDX,FERR,BERR,W ORK,W ORK2, INFO)}

```
CHARACTER * 1 TRANSA
\(\mathbb{N}\) TEGER*8N,KL,KU,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER*8 \(\mathbb{P} \mathbb{I V O T}(*), W\) ORK 2 ( \(\left.{ }^{( }\right)\)
D OUBLE PRECISION A (LDA, *), AF (LDAF,*), B (LDB ,*), X (LDX,*),
FERR (*), BERR (*), W ORK (*)

\section*{F95 INTERFACE}

SUBROUTINE GBRFS ([TRANSA], \(\mathbb{N}], K L, K U, \mathbb{N} R H S], A,[L D A], A F\), [LDAF], \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}], \mathrm{X},[\mathrm{LD} \mathrm{X}], \mathrm{FERR}, \mathrm{BERR},[\mathrm{W}\) ORK ], [W ORK 2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N}\) TEGER ::N,KL,KU,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{I} O T, W O R K 2\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A \(, A F, B, X\)

SU BROUTINE GBRFS_64 ([TRANSA], \(\mathbb{N}], K L, K U, \mathbb{N} R H S], A,[L D A]\), \(A F,[L D A F], \mathbb{P} \mathbb{I} O T, B,[L D B], X,[L D X], F E R R, B E R R,[W O R K]\), [W ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N} T E G E R(8):: N, K L, K U, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N F O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T, W\) ORK2
REAL (8), D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,AF,B,X

\section*{C INTERFACE}
\#include <sunperfh>
void dgbrfs (chartransa, intn, int kl, int ku, int nrhs, double *a, int lda, double *af, intldaf, int *ipívot, double *b, int ldb, double *x, int ldx, double * ferr, double *berr, int *info);
void dgbrfs_64 (chartransa, long n, long kl, long ku, long nrhs, double *a, long lda, double *af, long ldaf, long *ipivot, double *b, long ldb, double *x, long ldx, double *ferr, double *berr, long *info);

\section*{PURPOSE}
dgbris im proves the com puted solution to a system of linear equations when the coefficientm atrix is banded, and provides errorbounds and backw ard error estim ates for the solution.

\section*{ARGUMENTS}

TRANSA (input)
Specifies the form of the system ofequations:
\(=\mathrm{N}: \mathrm{A} * \mathrm{X}=\mathrm{B} \quad\) (N o transpose)
\(=T{ }^{\prime}: A * * T X=B \quad\) (Transpose)
\(=C^{\prime}: A * * H * X=B \quad\) (C onjugate transpose \(=T\) ran spose)

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).

K L (input)

The num berof subdiagonals w ithin the band of A. \(K L>=0\).

KU (input)
The num ber of superdiagonals within the band of A. \(K U>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the \(m\) atrioes \(B\) and \(X\). NRH \(S>=0\).

A (input) The originalband m atrix A, stored in row s 1 to \(K L+K U+1\). The \(j\) th column ofA is stored in the \(j\) th colum n of the array A as follow s: A (ku+1+i\(j, j)=A(i, j)\) form ax \((1, j \mathrm{jku})<=i<=m\) in \((n, j+k l)\).

LD A (input)
The leading dim ension of the array A. LDA >= K L+KU+1.

AF (input)
D etails of the LU factorization of the band \(m\) atrix A , as com puted by SG BTRF. U is stored as an upper triangularband \(m\) atrix \(w\) th \(K L+K U\) superdiagonals in row \(s 1\) to \(K L+K U+1\), and the \(m\) ultipliers used during the factorization are stored in row S \(K L+K U+2\) to \(2 * K L+K U+1\).

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(2 * K L * K U+1\).
\(\mathbb{P I V O T}\) (input)
The pivotindices from SGBTRF; for \(1<=i<=N\), row \(i\) of the \(m\) atrix \(w\) as interchanged w ith row \(\mathbb{P}\) IV OT (i).
\(B\) (input) The righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, N)\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by SGBTRS. On exit, the im proved solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the array X . LDX >= \(\max (1, N)\).

\section*{FERR (output)}

The estim ated forw ard errorbound for each solution vector \(X(\mathcal{)}\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{O}), \operatorname{FERR}(\underset{)}{(1)}\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in \((X(\mathcal{J})-X\) TRU \(E\) ) divided by the \(m\) agnitude of the largestelem ent in \(\mathrm{X}(\mathcal{j})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard emror of each solution vector \(X\) (i) (i.e., the sm allest relative change in any elem entofA orB thatm akes X (i) an exactsolution).

W ORK (w orkspace)
dim ension ( \(3 \star \mathrm{~N}\) )

W ORK 2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
```

dgbsv - com pute the solution to a real system of linear
equations A * X = B,where A is a bandm atrix oforderN
W th K L subdiagonals and K U superdiagonals, and }X\mathrm{ and }B\mathrm{ are
N -by-N RH S m atrices

```

\section*{SYNOPSIS}

```

\mathbb{NTEGERN,KL,KU,NRHS,LDA,LDB, INFO}
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION A (LDA,*),B (LDB,*)
SUBROUT\mathbb{NEDGBSV_64 N,KL,KU,NRHS,A,LDA,\mathbb{PIVOT,B,LDB,}}\mathbf{N},\textrm{N},\textrm{L}
\mathbb{NFO)}
NNTEGER*8 N,KL,KU,NRHS,LDA,LDB,NNFO
INTEGER*8 \mathbb{PIVOT (*)}
DOUBLE PRECISION A (LDA,*),B (LDB,*)

```

\section*{F95 INTERFACE}

SU BROUTINE GBSV ( \(\mathbb{N}], K L, K U, \mathbb{N} R S], A,[L D A], \mathbb{P} \mathbb{I V} \operatorname{T}, \mathrm{B},[\mathrm{LDB}]\), [ \(\mathbb{N}\) FO ])
\(\mathbb{N}\) TEGER ::N,KL,KU,NRHS,LDA,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V O T}\)
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B
SU BROUTINEGBSV_64 (N) \(\mathbb{N}, \mathrm{KL}, \mathrm{KU}, \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I} O T, B\),
[LDB], [ \(\mathbb{N F O}\) ])
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{KL}, \mathrm{KU}, \mathrm{NRH}, \mathrm{LDA}, \mathrm{LD} B, \mathbb{N} F \mathrm{O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P I V O T}\)
REAL (8), D IM ENSION (:,:) ::A , B

\section*{C INTERFACE}
\#include <sunperfh>
void dgbsv (intn, int kl, intku, intnrhs, double *a, int lda, int *ipívot, double *b, int ldb, int *info);
void dgbsv_64 (long n, long kl, long ku, long nrhs, double
*a, long lda, long *ipivot, double *b, long ldb, long *info);

\section*{PURPOSE}
dgbsv com putes the solution to a real system of linear equations A * \(X=B\), where A is a band \(m\) atrix of orderN w th K L subdiagonals and \(K U\) superdiagonals, and \(X\) and \(B\) are \(N\)-byNRH S m atrices.

The LU decom position w ith partialpivoting and row interchanges is used to factorA asA \(=\mathrm{L} * \mathrm{U}\), where L is a productof perm utation and unit low er triangularm atrioes with \(K L\) subdiagonals, and \(U\) is uppertriangularw ith \(K L+K U\) superdiagonals. The factored form of \(A\) is then used to solve the system of equations \(A * X=B\).

\section*{ARGUMENTS}

N (input) The num ber of linearequations, ie., the order of them atrix A. \(N>=0\).

KL (input)
The num ber of subdiagonals w ithin the band of A. \(\mathrm{KL}>=0\).

KU (input)
The num ber of superdiagonals w ithin the band of A. \(K U>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS >=0.

A (input/output)
On entry, the m atrix A in band storage, in row s
\(K L+1\) to \(2 * K L+K U+1\); row s 1 to \(K L\) of the anay need notbe set. The \(j\) th column of A is stored in the \(j\) th collumn of the array A as follows: \(A(K L+K U+1+i-j, j)=A(i, j)\) for \(\max (1, j\) \(K U)<=i<=m\) in ( \(N, j+K L\) ) On exit, details of the factorization: \(U\) is stored as an upper triangular band \(m\) atrix \(w\) ith \(K L+K U\) superdiagonals in row \(s 1\) to \(K L+K U+1\), and the \(m\) ultipliers used during the factorization are stored in rows \(K L+K U+2\) to \(2 \star K L+K U+1\). See below for further details.

LD A (input)
The leading dim ension of the array A. LDA >= \(2 * K L+K U+1\).
\(\mathbb{P I V O T}\) (output)
The pivot indices that define the perm utation \(m\) atrix \(P\); row i of the \(m\) atrix \(w\) as interchanged w ith row \(\mathbb{P I V O T}\) (i).

B (input/output)
On entry, the \(N-b y-N R H S\) righthand sidem atrix \(B\). On exit, if \(\mathbb{N} F O=0\), the \(N\) by \(-N R H S\) solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i, U(i, i)\) is exactly zero. The factorization has been com pleted, but the factor U is exactly singular, and the solution has notbeen com puted.

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, when \(M=N=6, K L=2, K U=1\) :

On entry: Onexit:
u36
* * + + + + * * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55 u66
a21 a32 a43 a54 a65 * m 21 m 32 m 43 m 54 m 65 * a31 a42 a53 a64 * * m31 m 42 m 53 m 64 * *

A rray elem entsm arked * are notused by the routine; ele\(m\) entsm arked + need notbe seton entry, but are required by the routine to store elem ents of \(U\) because of fill-in resulting from the row interchanges.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgbsvx -use the LU factorization to com pute the solution to a real system of linearequations \(A * X=B, A * * T * X=B\), orA \(*^{*}{ }_{H} * \mathrm{X}=\mathrm{B}\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGBSVX (FACT,TRANSA,N,KL,KU,NRHS,A,LDA,AF,}
LDAF,\mathbb{PIVOT,EQUED,R,C,B,LDB,X,LDX,RCOND,FERR,}
BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 FACT,TRANSA,EQUED
INTEGERN,KL,KU,NRHS,LDA,LDAF,LDB,LDX, INFO
INTEGER \mathbb{PIVOT (*),W ORK2 (*)}
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA,*),AF (LDAF,*),R (*),C (*),B (LDB,*),
X (LDX,*),FERR (*),BERR (*),W ORK (*)
SUBROUT\mathbb{NEDGBSVX_64\&ACT,TRANSA,N,KL,KU,NRHS,A,LDA,AF,}
LDAF,\mathbb{PIVOT,EQUED,R,C,B,LDB,X,LDX,RCOND,FERR,}
BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1FACT,TRANSA,EQUED
\mathbb{NTEGER*8N,KL,KU,NRHS,LDA,LDAF,LDB,LDX,INFO}
INTEGER*8 \mathbb{PIVOT (*),W ORK2 (*)}
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA,*),AF (LDAF,*),R (*),C (*),B (LD B,*),
X (LDX,*),FERR (*),BERR (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GBSVX \(\mathbb{E A C T},[T R A N S A], \mathbb{N}], K L, K U, \mathbb{N R H S}], A,[L D A]\), AF, [LDAF], \(\mathbb{P} \mathbb{I V O T}, E Q U E D, R, C, B,[L D B], X,[L D X]\), RCOND,FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::FACT,TRANSA, EQUED
\(\mathbb{N}\) TEGER :: N, KL, KU, NRHS,LDA, LDAF, LDB, LD X , \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{V} O T, W O R K 2\)
REAL (8) ::RCOND
REAL (8), D \(\mathbb{M} \operatorname{ENSION}(:):: R, C, F E R R, B E R R, W\) ORK
REAL (8), D \(\mathbb{M}\) ENSION (: : : ) : : A , AF , B, X

SU BROUTME GBSVX_64 (FACT, [TRANSA], \(\mathbb{N}], K L, K U, \mathbb{N R H S}], A\), \([L D A], A F,[L D A F], \mathbb{P} \mathbb{I V O T}, E Q U E D, R, C, B,[L D B], X,[L D X]\), RCOND, FERR, BERR, \(\mathbb{W}\) ORK], [W ORK 2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::FACT,TRANSA, EQUED
\(\mathbb{N}\) TEGER (8) :: N, KL, KU,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N T E G E R}(8), \mathrm{D} \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T, W\) ORK 2
REAL (8) :: RCOND
REAL (8), D \(\mathbb{M} \operatorname{ENSION}(:):: R, C, F E R R, B E R R, W\) ORK
REAL (8), D \(\mathbb{M}\) ENSION (: ::) ::A, AF, B, X

\section*{C INTERFACE}
\#include <sunperfh>
void dgbsvx (char fact, chartransa, intn, intkl, int ku, int nrhs, double *a, int lda, double *af, int ldaf, int*ipivot, char equed, double * \(x\), double \({ }^{*} \mathrm{c}\), double *b, int ldb, double *x, int ldx, double *rcond, double *ferr, double *berr, int *info);
void dgbsvx_64 (char fact, chartransa, long n, long kl, long ku, long nrhs, double *a, long lda, double *af, long ldaf, long *ipivot, char equed, double *r, double *c, double *b, long ldb, double *x, long \(l d x\), double *rcond, double *ferr, double *berr, long *info);

\section*{PURPOSE}
dgbsvx uses the LU factorization to com pute the solution to a real system of linear equations \(A * X=B, A * * T * X=B\), orA \(* * H * X=B\), where \(A\) is a band \(m\) atrix of order \(N\) w ith \(K L\) subdiagonals and \(K U\) superdiagonals, and \(X\) and \(B\) are \(N\) byN R H S m atrioes.

E mrorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed by this subroutine:
1. IfFACT \(=\) E', real scaling factors are computed to
equilibrate
the system :
TRANS \(=N^{\prime}: \operatorname{diag}(\mathbb{R}) * A * \operatorname{diag}(C) \quad * \operatorname{inv}(\operatorname{diag}(C)) * X=\) \(\operatorname{diag}(R) * B\)

TRANS \(=T\) ': \((\operatorname{diag}(R) * A * \operatorname{diag}(C)) * * T * \operatorname{inv}(\operatorname{diag}(R)) * X=\) diag (C) *B

TRANS \(=C^{\prime}:(\operatorname{diag}(\mathbb{R}) * A * \operatorname{diag}(C)) * * H * \operatorname{inv}(\operatorname{diag}(\mathbb{R})) * X=\) diag (C) *B
W hether or not the system w illbe equilibrated depends on the
scaling of them atrix A, but if equilibration is used, A is
overw rilten by diag \((\mathbb{R}) * A\) *diag \((C)\) and \(B\) by \(\operatorname{diag}(R) * B\) (ff TRANS = N )
ordiag (C)*B (if TRANS = T'or C) .
2. IfFACT = N 'or E', the LU decom position is used to factor the
m atrix A (afterequilibration ifFACT = E) as
\[
A=L * U,
\]
where \(L\) is a product of perm utation and unit low er triangular
\(m\) atricesw ith \(\mathrm{K} L\) subdiagonals, and \(U\) is upper triangular w ith
K L+KU superdiagonals.
3. If som eU \((i, i)=0\), so that \(U\) is exactly singular, then the routine
retums w ith \(\mathbb{N} F O=i .0\) therw ise, the factored form of \(A\) is used
to estim ate the condition num ber of the \(m\) atrix \(A\). If the reciprocal of the condition num ber is less than \(m\) achine precision,
\(\mathbb{N} F O=N+1\) is retumed as a waming, but the routine stillgoes on
to solve for X and com pute error bounds as described below .
4. The system of equations is solved for X using the factored form of A.
5. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw and error estim ates
for it.
6. Ifequilibration \(w\) as used, the \(m\) atrix \(X\) is prem ultiplied by
diag (C) (iftRANS = N) ordiag \((\mathbb{R})\) (ifTRANS = \(T^{\prime}\) or C) so
that it solves the original system before equilibration.

\section*{ARGUMENTS}

FACT (input)
Specifies whether ornot the factored form of the \(m\) atrix A is supplied on entry, and if not, w hether the m atrix A should be equilibrated before it is factored. \(=\mathrm{F}\) ': On entry, AF and \(\mathbb{P I V O T}\) contain the factored form of A. IfEQUED is not \(N\) ', the \(m\) atrix A has been equilibrated w th scaling factors given by \(R\) and \(C . A, A F\), and \(\mathbb{P} I V O T\) are not m odified. \(=\mathrm{N}\) ': The m atrix A w illbe copied to A F and factored.
= E ': The matrix A will be equilibrated if necessary, then copied to AF and factored.
TRANSA (input)
Specifies the form of the system of equations. = N : : A * \(\mathrm{X}=\mathrm{B} \quad\) N \(\circ\) transpose)
\(=T\) ': A ** \(T\) * \(\mathrm{X}=\mathrm{B}\) ( T ranspose)
\(=C: A * * H * X=B \quad\) (Transpose)

TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.

N (input) The num ber of linear equations, i.e., the order of them atrix A. \(\mathrm{N}>=0\).

KL (input)
The num ber of subdiagonals w thin the band of A.
\(\mathrm{KL}>=0\) 。

KU (input)
The num ber of superdiagonals \(w\) ithin the band of \(A\). \(K U>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber
of collm ns of the m atrices B and X. NRHS \(>=0\).

A (input/output)
O \(n\) entry, the \(m\) atrix A in band storage, in row \(s 1\)
to \(K L+K U+1\). The \(j\) th colum \(n\) ofA is stored in the \(j\) th colum \(n\) of the anray A as follow s: A ( \(K U+1+i-\) \(j, j)=A(i, 7)\) form ax \((1, j K U)<=i<=m\) in \((N, j+k l)\)

IfFACT = F'and EQUED is not \(N\) ', then \(A\) must have been equilibrated by the scaling factors in \(R\)
and/orC. A is notm odified if FACT = F'or N', or ifFACT = E'andEQUED = N 'on exit.

Onexit, ifEQUED ne. \(N\) ', A is scaled as fol-
low s: EQUED = R': A := diag \((\mathbb{R}) * A\)
EQUED = C': A : A * diag (C)
EQUED \(=B: A:=\operatorname{diag}(\mathbb{R}) * A * \operatorname{diag}(C)\).

LD A (input)
The leading dim ension of the aray A. LDA >= \(K L+K U+1\).

AF (input/output)
If \(F A C T=F\) ', then \(A F\) is an input argum ent and on entry contains details of the LU factorization of the band \(m\) atrix \(A\), as com puted by SGBTRF. U is stored as an upper triangularband \(m\) atrix \(w\) ith \(K \mathrm{~L}+\mathrm{K} \mathrm{U}\) superdiagonals in row s 1 to \(\mathrm{K} L+K \mathrm{~K}+1\), and the multi liers used during the factorization are stored in row \(s K L+K U+2\) to \(2 * K L+K U+1\). If EQUED
ne. \(N\) ', then AF is the factored form of the equilibrated \(m\) atrix \(A\).

If FACT \(=\mathrm{N}\) ', then AF is an output argum ent and on exit retums details of the LU factorization of A.

IfFACT = E', then AF is an output argum ent and on exit retums details of the LU factorization of the equilibrated \(m\) atrix A (see the description of \(A\) forthe form of the equilibrated \(m\) atrix).

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(2 * K L+K U+1\).
\(\mathbb{P I V O T}\) (input)
IfFACT = F ', then \(\mathbb{P} \mathbb{I V O T}\) is an input argum ent and on entry contains the pivot indioes from the factorization \(\mathrm{A}=\mathrm{L} * \mathrm{U}\) as com puted by SGBTRF; row i of the \(m\) atrix \(w\) as interchanged \(w\) th row \(\mathbb{P}\) IV OT (i).

If FACT \(=N^{\prime}\), then \(\mathbb{P} \mathbb{I V O T}\) is an output argum ent and on exit contains the pivot indioes from the factorization \(\mathrm{A}=\mathrm{L} * \mathrm{U}\) of the originalm atrix A .

IfFACT = \(E\) ', then \(\mathbb{P} \mathbb{I V O T}\) is an output argum ent and on exit contains the pivot indioes from the factorization \(A=L * U\) of the equilibrated \(m\) atrix
A.

EQUED (input)
Specifies the form of equilibration thatw as done.
\(=N\) : N 0 equilibration (alw ays true ifFACT \(=\) \(\mathrm{N})\).
\(=R\) ': Row equilibration, ie., A has been prem ultiplied by diag \((R)\). = C ': C olum n equilibration, ie., A has been postm ultiplied by diag (C). = B': B oth row and colum n equilibration, ie., A has.been replaced by diag \((\mathbb{R})\) * A * diag (C). EQUED is an inputargum entifFACT= F '; otherw ise, it is an outputargum ent.
R (input/output)
The row scale factors for \(A\). IfEQUED \(=R^{\prime}\) or \(B\) ', \(A\) is multiplied on the left by diag \((\mathbb{R})\); if EQUED = N 'or \(C\) ', \(R\) is notaccessed. \(R\) is an input argum ent ifFACT = \(F\) '; otherw ise, \(R\) is an outputargum ent. IfFACT = F'and EQUED = R'or \(B\) ', each elem entofR \(m\) ustbe positive.

C (input/output)
The colum \(n\) scale factors for \(A\). IfEQUED = C 'or
B', A is multiplied on the rightby diag (C ) ; if \(E Q U E D=N\) 'or \(R\) ', \(C\) is notaccessed. \(C\) is an input argum ent ifFACT = F '; otherw ise, C is an outputargum ent. IfFACT = F'and EQUED = C'or \(B\) ',each elem entofC \(m\) ustbe positive.

B (input/output)
On entry, the righthand side matrix B. On exit, ifEQUED \(=N^{\prime}\) ', \(B\) is notm odified; ifTRANSA \(=N^{\prime}\) and \(E Q U E D=R^{\prime}\) or \(B^{\prime}, B\) is overw ritten by \(\operatorname{diag}(\mathbb{R}) * B\); if TRANSA \(=T\) 'or \(C\) 'and \(E Q U E D=C^{\prime}\) or \(B^{\prime}, B\) is overw rilten by diag ( \(C\) )*B.

LD B (input)
The leading dim ension of the array \(B\). LD B >= \(\max (1, \mathbb{N})\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=N+1\), the \(N\)-by \(-\mathrm{NRH} S\) solution
\(m\) atrix \(X\) to the original system ofequations.
\(N\) ote that \(A\) and \(B\) arem odified on exit if EQUED
ne. \(\mathrm{N}^{\prime}\), and the solution to the equilibrated
system is inv (diag (C))*X ifTRANSA \(=N\) 'andEQUED
\(=C\) 'or \(B^{\prime}\), orinv (diag \(\left.(R)\right) * X\) ifTRANSA \(=T\) 'or \(C\) 'and EQUED \(=R\) 'or \(B\) '.

LD X (input)
The leading dim ension of the array X . LD X >=
\(\max (1, \mathbb{N})\).

RCOND (output)
The estim ate of the reciprocal condition num berof the \(m\) atrix A after equilibration (if done). If
RCOND is less than them achine precision (in particular, ifRCOND \(=0\) ), the \(m\) atrix is singular to w orking precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO \(>0\).

\section*{FERR (output)}

The estim ated forw ard enorbound for each solution vector \(X(\mathcal{)}\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(\mathrm{X}(\mathcal{)}, \mathrm{FERR}()\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{H})\)-XTRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{)}\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector \(X(j)\) (i.e., the sm allest relative change in any elem entofA orB thatm akesX ( \(\mathcal{(})\) an exactsolution).

W ORK (w orkspace)
dim ension ( \(3 * \mathrm{~N}\) ) On exit, W ORK (1) contains the reciprocal pivot grow th factornorm (A)/norm (U). The "m ax absolute elem ent" norm is used. If W ORK (1) ism uch less than 1 , then the stability of the LU factorization of the (equilibrated) matrix A could be poor. This also \(m\) eans that the solution X , condition estim atorRCOND, and forw ard error bound FERR could be unreliable. If factorization fails w ith \(0<\mathbb{N} F O<=N\), then \(W\) ORK (1) contains the reciprocal pivot grow th factor for the leading
\(\mathbb{N}\) FO colum ns of A.
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfinlexit
<0: if \(\mathbb{N N}\) FO \(=-\) i, the i-th argum enthad an illegalvalue
> 0 : if \(\mathbb{N F O}=\mathrm{i}\), and i is
\(<=\mathrm{N}: \mathrm{U}(i, i)\) is exactly zero. The factorization
has been completed, but the factor \(U\) is exactly
singular, so the solution and error bounds could not be com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1\) : U is nonsingular, butRCOND is less than \(m\) achine precision, \(m\) eaning that the \(m\) atrix is singular to w orking precision. N evertheless, the solution and error bounds are com puted because there are a num ber of situations w here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dgbte2 - com pute an LU factorization of a real \(m\)-by-n band \(m\) atrix A using partialpivoting \(w\) th row interchanges

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGBTF2(M,N,KL,KU,AB,LDAB,\mathbb{P}\mathbb{V},\mathbb{NFO)}}\mathbf{N}=()
INTEGERM,N,KL,KU,LDAB,INFO
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(})
DOUBLE PRECISION AB (LDAB,*)
SUBROUT\mathbb{NE DGBTF2_64 M ,N,KL,KU,AB,LDAB, IP IV,INFO)}
\mathbb{N TEGER*8M,N,KL,KU,LDAB,INFO}
\mathbb{NTEGER*8 P\mathbb{IV (*)}}\mathbf{*})
DOUBLE PRECISION AB (LDAB,*)
F95 INTERFACE

```

```

\mathbb{NTEGER ::M ,N,KL,KU,LDAB,\mathbb{NFO}}\mathbf{N}=\mp@code{L}
INTEGER,D IM ENSION (:) :: \mathbb{PIV}
REAL (8),D IM ENSION (:,:) ::AB

```

```

\mathbb{NTEGER (8)::M,N,KL,KU,LDAB,INFO}
INTEGER (8),D IM ENSION (:) :: \mathbb{PIV}
REAL (8),D IM ENSION (:,:) ::AB

```
void dgbtf2 (intm , intn, intkl, intku, double *ab, int ldab, int *ịív, int *info);
void dgbtf2_64 (long m , long n, long kl, long ku, double *ab, long ldab, long *ịív, long *info);

\section*{PURPOSE}
dgbtf2 com putes an LU factorization ofa real \(m\)-by-n band \(m\) atrix A using partialpivoting \(w\) ith row interchanges.

This is the unblocked version of the algorithm, calling Level2 BLAS.

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
KL (input)
The num ber of subdiagonals w ithin the band of A.
\(\mathrm{KL}>=0\).
KU (input)
The num ber of superdiagonals w ithin the band of A. \(K U>=0\).

AB (input/output)
O n entry, them atrix \(A\) in band storage, in row \(s\) \(K L+1\) to \(2 * K L+K U+1\); row s 1 to \(K L\) of the anay need not.be set. The \(j\) th column ofA is stored in the \(j\) th column of the array AB as follows: \(A B(k l+k u+1+i-j, j)=A(i, j)\) for \(m a x(1, j\) \(\mathrm{ku})<=i<=m\) in \((\mathrm{m}, \mathrm{j}+\mathrm{kl})\)

On exit, details of the factorization: U is stored as an upper triangular band \(m\) atrix \(w\) ith \(K L+K U\) superdiagonals in row s 1 to \(K L+K U+1\), and the \(m u l-\) tipliers used during the factorization are stored in row s \(K L+K U+2\) to \(2 * K L+K U+1\). See below for furtherdetails.

The leading dim ension of the array A B . LD AB >=
\(2 * K L+K U+1\).

IPIV (output)
The pivot indices; for \(1<=i<=m\) in \(M, N\) ), row i of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P} \mathbb{V}\) (i).
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0\) : if \(\mathbb{N N F O}=-\) i, the \(i\)-th argum ent had an illegalvalue \(>0\) : if \(\mathbb{N} F O=+i, U(i, i)\) is exactly zero. The factorization has been com pleted, but the factor \(U\) is exactly singular, and division by zero will occur if it is used to solve a system ofequations.

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, \(w\) hen \(M=N=6, K L=2, K U=1\) :

On entry: On exit:
```

    * * * + + + * * * u14 u25
    u36
* * + + + + * * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55
u66
a21 a32 a43 a54 a65 * m 21 m 32 m 43 m 54 m 65
*
a31 a42 a53 a64 * * m 31 m 42 m 53 m 64 *

```
*

A ray elem entsm arked * are notused by the routine; ele\(m\) entsm arked + need notbe seton entry, but are required by the routine to store elem ents of \(U\), because of fill-in resulting from the row interchanges.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dgbtrf-com pute an LU factorization of a real \(m-b y-n\) band \(m\) atrix A using partialpivoting \(w\) th row interchanges

\section*{SYNOPSIS}

```

INTEGERM,N,KL,KU,LDAB,INFO
INTEGER \mathbb{PIVOTM IN M N))}
DOUBLE PRECISION AB (LDAB,N)

```

```

\mathbb{N TEGER*8M,N,KL,KU,LDAB,INFO}
INTEGER*8\mathbb{PIVOT N})
DOUBLE PRECISION AB (LDABN)

```

\section*{F95 INTERFACE}

SU BROUTINE GBTRF M, \(\mathbb{N}], K L, K U, A B,[L D A B], \mathbb{P} \mathbb{I} O T,[\mathbb{N F O}])\)
\(\mathbb{N} T E G E R:: M, N, K L, K U, L D A B, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL (8),D IM ENSION (:,:) ::AB

SU BROUTINE GBTRF_64M, N ],KL,KU,AB, [LDAB], \(\mathbb{P} \mathbb{I V O T},[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER ( 8 ) :: M, N, KL, KU, LDAB, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8), D IM ENSION (:,:) ::AB
\#include <sunperfh>
void dgbtrf(intm , intn, int kl, intku, double *ab, int ldab, int *ipivot, int*info);
void dgbtrf_64 (long m , long n, long kl, long ku, double *ab, long ldab, long *ịíivot, long *info);

\section*{PURPOSE}
dgbtrf com putes an LU factorization of a real \(m-b y-n\) band \(m\) atrix A using partialpivoting \(w\) ith row interchanges.

This is the blocked version of the algorithm, calling Level 3 BLAS .

\section*{ARGUMENTS}

M (input) Integer
The num ber of row sof the m atrix A. M \(>=0\).
N (input) Integer
The num berof \(c o l u m\) ns of the \(m\) atrix \(A . N>=0\).

K L (input) Integer
The num berof subdiagonals w ithin the band of A. \(K L>=0\).

KU (input) Integer
The num ber of superdiagonals \(w\) ithin the band of A.
\(K U>=0\).

AB (input/output) D ouble precision array of dim ension (LD AB,N).
On entry, the \(m\) atrix A in band storage, in row \(s\) \(K L+1\) to \(2 * K L+K U+1\); row s 1 to \(K L\) of the aray need not.be set. The \(j\) th column ofA is stored in the \(j\) th column of the array \(A\) as follows: AB \((\mathbb{K L}+K U+1+I-J, J)=A(I, J)\) for MAX \((1, J-\) \(K U)<=\mathbb{K}=M \mathbb{N}(M, J+K L)\)

O n exit, details of the factorization: U is stored as an upper triangular band \(m\) atrix \(w\) ith \(K L+K U\) superdiagonals in row 1 to \(K L+K U+1\), and the \(m\) ultipliers used during the factorization are stored in row \(s K L+K U+2\) to \(2 * K L+K U+1\). See below for furtherdetails.

LD A B (input) Integer array ofdim ension \(M \mathbb{N} M, N\) )
The leading dim ension of the array AB. LD AB >= 2*K L+KU +1.
\(\mathbb{P} \mathbb{V} O T\) (output) Integerarray ofdim ension \(M \mathbb{N} \mathrm{M}, \mathbb{N}\) )
The pivotindioes; for \(1<=I<=M \mathbb{N} M, N\) ), row \(I\) of the \(m\) atrix \(w\) as interchanged \(w\) th row \(\mathbb{P} \mathbb{I V O T}\) (I).
\(\mathbb{I N} F O\) (output) Integer
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) I, the I-th argum enthad an illegalvalue \(>0:\) if \(\mathbb{N} F O=+\mathbb{I}, U(I, I)\) is exactly zero. The factorization has been com pleted, but the factor \(U\) is exactly singular, and division by zero will occur if it is used to solve a system of equations.

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, when \(M=N=6, K L=2, K U=1\) :

On entry: On exit:
```

    * * * + + + * * * u14 u25
    u36
* * + + + + * * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55
u66
a21 a32 a43 a54 a65 * m 21 m 32 m 43 m 54 m 65
*
a31 a42 a53 a64 * * m31 m 42 m 53 m 64 *
*

```

A rray elem entsm arked * are notused by the routine; ele\(m\) entsm arked + need notbe seton entry, but are required by the routine to store elem ents of \(U\) because of fill-in resulting from the row interchanges.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dgbtrs -solve a system of linearequations A * X = B orA '

```
* \(\mathrm{X}=\mathrm{B}\) w ith a generalband m atrix A using the LU factoriza-
tion com puted by SGBTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DGBTRS (TRANSA,N,NSUB,NSUPER,NRHS,A,LDA, IPIVOT,B,}
LDB,\mathbb{NFO )}
CHARACTER * 1 TRANSA
INTEGER N,NSUB,NSUPER,NRHS,LDA,LDB, INFO
\mathbb{NTEGER \mathbb{PIVOT(*)}}\mathbf{(})
DOUBLE PRECISION A (LDA,*),B (LDB,*)
SUBROUT\mathbb{NEDGBTRS_64(TRANSA,N,NSUB,NSUPER,NRHS,A,LDA, \mathbb{PIVOT,}}\mathbf{N},\textrm{N},\textrm{N}
B,LDB,INFO)
CHARACTER * 1 TRANSA
INTEGER*8 N,NSUB,NSUPER,NRHS,LDA,LDB,INFO
\mathbb{NTEGER*8 P\mathbb{IVOT (*)}}\mathbf{(})
DOUBLE PRECISION A (LDA,*),B (LDB,*)

```

\section*{F95 INTERFACE}

SU BROUTINE GBTRS ([TRANSA], \(\mathbb{N}], N S U B, N S U P E R, ~ N R H S], A,[L D A]\), \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}],[\mathbb{N F O}])\)

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N}\) TEGER :: N,N SUB,NSUPER,NRHS,LDA,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V O T}\)
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A , B
SU BROUTINE GBTRS_64 ([TRANSA], \(\mathbb{N}], N S U B, N S U P E R, ~ \mathbb{N} R H S], A,[L D A]\),
```

\mathbb{PNOT,B,[LDB],[\mathbb{NFO])}}\mathbf{~}/\mp@code{M}

```

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N}\) TEGER (8) :: N , N SUB , N SUPER, NRHS,LDA, LD B, \(\mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A, B

\section*{C INTERFACE}
\#include <sunperfh>
void dgbtrs (chartransa, intn, intnsub, int nsuper, int nihs, double *a, int lda, int *ịivot, double *b, intldb, int*info);
void dgbtrs_64 (chartransa, long n, long nsub, long nsuper, long nれs, double *a, long lda, long *ipívot, double *b, long ldb, long *info);

\section*{PURPOSE}
dgbtrs solves a system of linear equations
\(A * X=B\) or \(A^{\prime} * X=B\) w th a general band \(m\) atrix \(A\) using the LU factorization com puted by SGBTRF .

\section*{ARGUMENTS}

TRANSA (input)
Specifies the form of the system ofequations. =
\(\mathrm{N}^{\prime}: A * X=B \quad\) N o transpose)
\(=T\) ': A * \(\mathrm{X}=\mathrm{B} \quad\) (Transpose)
\(=C: A\) * \(X=B\) (C onjugate transpose \(=\) Transpose)

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

N SUB (input)
The num ber of subdiagonals w thin the band of A. N SUB \(>=0\) 。

\section*{N SU PER (input)}

The num ber of superdiagonals \(w\) thin the band of A. N SU PER > \(=0\).

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of the m atrix B. NRH S \(>=0\).

A (input) D etails of the LU factorization of the band \(m\) atrix
A , as com puted by SG B TRF. U is stored as an upper triangularband m atrix w ith N SU B + N SU PER superdiagonals in row s 1 to N SU B + N SUPER +1 , and them ultipliers used during the factorization are stored in row s N SU B +N SU PER +2 to \(2 \star\) N SU B + N SU PER +1 .

LD A (input)
The leading dim ension of the anay A. LDA >= \(2 * N\) SU B \(+N\) SU PER +1 .
IPIVOT (input)
The pivotindices; for \(1<=i<=N\), row i of the m atrix \(w\) as interchanged w ith row IPIVOT (i).

B (input/output)
On entry, the right hand sidem atrix B. On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

IN FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgebak - form the rightor lefteigenvectors of a real gen-
eral matrix by backw ard transform ation on the com puted
eigenvectors of the balanced \(m\) atrix outputby SG EBA L

\section*{SYNOPSIS}

```

CHARACTER * 1 JOB,SIDE
\mathbb{NTEGERN,}\mathbb{NO,}\mathbb{H}I,M,LDV,\mathbb{NFO}
DOUBLE PRECISION SCALE (*),V (LDV ,*)

```

```

CHARACTER * 1 JOB,S\mathbb{DE}
\mathbb{NTEGER*8N,}\mathbb{NO},\mathbb{H}I,M,LDV,\mathbb{NFO}
D OUBLE PRECISION SCALE (*),V (LDV ,*)

```

\section*{F95 INTERFACE}
 [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1):: JOB,SDE
\(\mathbb{N}\) TEGER :: \(N, \mathbb{L} O, \mathbb{H} \mathrm{I}, \mathrm{M}, \mathrm{LDV}, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::SCALE
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::V
SU BROUTINE GEBAK_64 (JOB,SIDE, \(\mathbb{N}], \mathbb{I} O, \mathbb{H} I, S C A L E, \mathbb{M}], V,[L D V]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1):: JOB ,SDE
\(\mathbb{N} T E G E R(8):: N, \mathbb{L O}, \mathbb{H} \mathrm{I}, \mathrm{M}, \mathrm{LDV}, \mathbb{N} F O\)

REAL (8),D IM ENSION (:) ::SCALE
REAL (8),D IM ENSION (:,:) ::V

\section*{C INTERFACE}
\#include < sunperfh>
void dgebak (char job, charside, intn, int ilo, int ini, double *scale, int \(m\), double *v, intldv, int *info);
void dgebak_64 (char jं.b, charside, long n, long ilo, long ihí, double *scale, long m, double *v, long ldv, long *info);

\section*{PURPOSE}
dgebak form sthe rightor lefteigenvectors of a real general \(m\) atrix by backw ard transform ation on the com puted eigenvectors of the balanced m atrix outputby SG EBA L .

\section*{ARGUMENTS}

JOB (input)
Specifies the type of backw ard transform ation
required: = N ', do nothing, retum im m ediately;
\(=\mathrm{P}\) ', do backw ard transform ation for perm utation only; = S', do backw ard transform ation for scaling only; = B ', do backw ard transform ations for both perm utation and scaling. JO B m ust.be the sam e as the argum ent JO B supplied to SG EBA L .

SIDE (input)
\(=\mathrm{R}^{\prime}: \mathrm{V}\) contains righteigenvectors;
\(=\mathrm{L} \cdot \mathrm{V}\) contains lefteigenvectors.
N (input) The num ber of row sof the m atrix \(\mathrm{V} . \mathrm{N}>=0\).

ШО (input)
The integers \(\mathbb{I I O}\) and \(\mathbb{H}\) I determ ined by SG EBA L. 1
\(<=\mathbb{L O}<=\mathbb{H} \mathrm{I}<=\mathrm{N}\), if \(\mathrm{N}>0 ; \mathbb{H}=1\) and \(\mathbb{H} \mathrm{I}=0\), if
\(\mathrm{N}=0\).

IH I (input)
See the description for IIO .
SCALE (input)
D etails of the perm utation and scaling factors, as
retumed by SGEBAL.

M (input) The num ber of colum \(n s\) of the \(m\) atrix \(V . M>=0\).

V (input/output)
O n entry, the \(m\) atrix of right or lefteigenvectors to be transform ed, as retumed by SHSEIN or STREVC. On exit, \(V\) is overw rilten by the transform ed eigenvectors.

LDV (input)
The leading dim ension of the anray V.LDV >= \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-i\), the i-th argum enthad an illegalvalue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dgebal-balance a general realm atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGEBAL (JOB,N,A,LDA, IOO,\mathbb{H}I,SCA LE, INFO )}
CHARACTER * 1 JOB

```

```

DOUBLE PRECISION A (LDA,*),SCALE (*)
SUBROUT\mathbb{NEDGEBAL_64(JOB,N,A,LDA,}\mathbb{NO},\mathbb{H}I,SCALE, IN FO )
CHARACTER * 1 J B
\mathbb{NTEGER*8N,LDA,}\mathbb{NO},\mathbb{H}I,\mathbb{N}FO
DOUBLE PRECISION A (LDA,*),SCALE (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE GEBAL (JOB, $\mathbb{N}], A,[L D A], \mathbb{I} O, \mathbb{H} I, S C A L E,[\mathbb{N F O}])$
CHARACTER (LEN=1):: JOB
$\mathbb{N}$ TEGER :: $\mathrm{N}, \mathrm{LD} A, \mathbb{H O}, \mathbb{H} \mathrm{I}, \mathbb{N} F O$
REAL (8),D $\mathbb{I M}$ ENSION (:) ::SCALE
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::A
SU BROUTINE GEBAL_64 (DOB, $\mathbb{N}], A,[L D A], \mathbb{L O}, \mathbb{H} I, S C A L E,[\mathbb{N} F O$ ])
CHARACTER (LEN=1) :: JOB
$\mathbb{N}$ TEGER (8) :: N, LDA, $\mathbb{L} O, \mathbb{H} \mathrm{I}, \mathbb{I N F O}$
REAL (8),D $\mathbb{M}$ ENSION (:) ::SCALE
REAL (8),D IM ENSION (:,:) ::A

```

\section*{C INTERFACE}
\#include <sunperfh>
void dgebal(char j̣ंb, intn, double *a, intlda, int *io, int *ihi, double *scale, int *info);
void dgebal 64 (char job, long n, double *a, long lda, long *ilo, long *ihi, double *scale, long *info);

\section*{PURPOSE}
dgebalbalances a general real m atrix A. This involves, first, perm uting A by a sm ilarity transform ation to isolate eigenvalues in the first 1 to IIO -1 and last IH I+1 to N ele\(m\) ents on the diagonal; and second, applying a diagonal sim ilarity transform ation to row sand colum ns \(\mathbb{H O}\) to \(\mathbb{H}\) Ito \(m\) ake the rows and columns as close in norm aspossible. Both steps are optional.

B alancing \(m\) ay reduce the 1-norm of the \(m\) atrix, and im prove the accuracy of the com puted eigenvalues and/oreigenvectors.

\section*{ARGUMENTS}
\(J O B\) (input)
Specifies the operations to be perform ed on A:
= N ': none: simply set \(\Pi \mathrm{O}=1\), \(\mathbb{H} \mathrm{I}=\mathrm{N}\), SCALE (I) \(=1.0\) fori= \(1, \ldots, N ;=P\) ': perm ute only;
= S ': scale only;
= \(\mathrm{B}:\) : both perm ute and scale.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
A (input/output)
On entry, the inputm atrix A. On exit, A is overw rilten by the balanced \(m\) atrix. If \(J O B=N\) ', \(A\) is not referenced. See FurtherD etails.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

世O (output)
IIO and \(\mathbb{H}\) I are set to integers such thaton exit
A \((i, 1)=0\) if \(i>j a n d j=1, \ldots\), ILO -1 or \(I=\)
\(\mathbb{H} \mathrm{I}+1, \ldots, \mathrm{~N}\). If \(\mathrm{OB}=\mathrm{N}\) 'or \(\mathrm{S}^{\prime}, \mathbb{I} \mathrm{HO}=1\) and \(\mathbb{H} \mathrm{I}\)
\(=\mathrm{N}\).
IH I (output)
See the description for IIO .
SCALE (output)
D etails of the perm utations and scaling factors applied to \(A\). IfP ( \(j\) ) is the index of the row and colum \(n\) interchanged \(w\) th row and colum \(n\) jand \(D(1)\) is the scaling factorapplied to row and column \(j\) then SCALE \((j)=P(j) \quad\) for \(j=1, \ldots, I L O-1=D(j)\) for \(j=\mathbb{L O}, \ldots, \mathbb{H} I=P(j) \quad\) for \(j=\mathbb{H} I+1, \ldots, N\). The order in which the interchanges are m ade is N to \(\mathbb{H} \mathrm{I}+1\), then 1 to \(\mathbb{L O} \mathrm{O}\).
\(\mathbb{I N} F O\) (output)
= 0: successfulexit.
<0: if \(\mathbb{N N}\) FO = -i, the i-th argum enthad an illegalvahue.

\section*{FURTHER DETAILS}

The perm utations consist of row and column interchanges which put the \(m\) atrix in the form
\[
\left.\begin{array}{c}
(\mathrm{T} 1 \mathrm{X} \\
\mathrm{PAP}
\end{array}\right)=\left(\begin{array}{ll}
0 & \mathrm{~B} \\
\mathrm{P}
\end{array}\right) .
\]
where T1 and T2 are uppertriangularm atrices whose eigenvalues lie along the diagonal. The colum \(n\) indices \(\Pi \mathrm{HO}\) and IH Im ark the starting and ending colum ns of the subm atrix B. Balancing consists of applying a diagonal sim ilarity transform ation inv \((D) * B * D\) to \(m\) ake the 1 -norm \(s\) of each row of \(B\) and its comesponding colum n nearly equal. The outputm atrix is
\(\left(\begin{array}{lll}(11 & X * D & Y\end{array}\right)\)
\(\left(\begin{array}{lll}0 & \operatorname{inv}(D) * B * D & \operatorname{inv}(D) * Z\end{array}\right)\).
\(\left(\begin{array}{lll}0 & 0 & T 2\end{array}\right)\)

Inform ation about the perm utations P and the diagonalm atrix \(D\) is retumed in the vectorSCA LE.

This subroutine is based on the E ISPACK routine BA LANC.
M odified by Tzu-Y iChen, C om puterScience D ívision, U niversity of
C alifomia atB erkeley, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dgebrd - reduce a general realM -by -N m atrix A to upper or low erbidiagonal form B by an orthogonal transform ation

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGEBRD M,N,A,LDA,D,E,TAUQ,TAUP,W ORK,LW ORK,\mathbb{NFO)}}\mathbf{M}\mathrm{ (N,N}
INTEGERM,N,LDA,LW ORK,INFO
DOUBLE PRECISION A (LDA,*), D (*), E (*), TAUQ (*), TAUP (*),
W ORK (*)
SU BROUT\mathbb{NE DGEBRD_64M,N,A,LDA,D ,E,TAUQ,TAUP,W ORK,LW ORK,}
\mathbb{NFO)}
INTEGER*8M,N,LDA,LW ORK,\mathbb{NFO}
DOUBLE PRECISION A (LDA,*), D (*), E (*), TAUQ (*), TAUP (*),
W ORK (*)

```

\section*{F95 INTERFACE}

SUBROUTINE GEBRD ( \(\mathbb{M}], \mathbb{N}], A,[L D A], D, E, T A U Q, T A U P,[W O R K],[L W\) ORK ], [ \(\mathbb{N}\) FO ])
\(\mathbb{N} T E G E R:: M, N, L D A, L W O R K, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::D ,E,TAUQ,TAUP,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A
SU BROUTINE GEBRD_64 (M) \(\mathbb{M}], A,[L D A], D, E, T A U Q, T A U P,[W O R K]\), [LW ORK], [ \(\mathbb{N F O}\) ])
\(\mathbb{N} T E G E R(8):: M, N, L D A, L W O R K, \mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D ,E,TAUQ,TAUP,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void dgebrd (intm , intn, double *a, int lda, double *d, double *e, double *tauq, double *taup, int *info);
void dgebrd_64 (long m , long n, double *a, long lda, double
*d, double *e, double *tauq, double *taup, long
*info);

\section*{PURPOSE}
dgebrd reduces a general realM -by N m atrix A to upper or lower bidiagonal form \(B\) by an orthogonaltransform ation: \(\mathrm{Q} * * \mathrm{~T} * \mathrm{~A} * \mathrm{P}=\mathrm{B}\).

Ifm >= \(n, B\) is upperbidiagonal; ifm < \(n, B\) is low erbidiagonal.

\section*{ARGUMENTS}

M (input) The num ber of row s in the m atrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of collm ns in them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O \(n\) entry, the \(M-b y-N\) generalm atrix to be reduced. On exit, if \(m>=n\), the diagonal and the first superdiagonal are overw rilten w ith the upperbidiagonal \(m\) atrix \(B\); the elem ents below the diagonal, \(w\) th the array \(T A U Q\), represent the orthogonal \(m\) atrix \(Q\) as a productofelem entary reflectors, and the elem ents above the first superdiagonal, w th the aray TAUP, represent the orthogonal \(m\) atrix \(P\) as a product of elem entary reflectors; if \(\mathrm{m}<\mathrm{n}\), the diagonaland the first subdiagonalare overw ritten \(w\) ith the low er bidiagonal \(m\) atrix \(B\); the elem ents below the firstsubdiagonal, w ith the array TAUQ, represent the orthogonalm atrix \(Q\) as a product ofelem entary reflectors, and the elem ents above the diagonal, w th the amay TA U P, represent the orthogonalm atrix \(P\) as a productofelem entary reflectors. See FurtherD etails.

The leading dim ension of the amay A. LDA >= \(\max (1, M)\).

D (output)
The diagonalelem ents of the bidiagonalm atrix B :
\(D(i)=A(i, i)\).

E (output)
The off-diagonalelem ents of the bidiagonalm atrix
\(B:\) ifm \(>=n, E(i)=A(i, i+1)\) for \(i=1,2, \ldots, n-\)
\(1 ;\) ifm \(<n, E(i)=A(i+1, i)\) for \(i=1,2, \ldots m-1\).

TAUQ (output)
The scalar factors of the elem entary reflectors
which represent the orthogonal matrix Q. See
FurtherD etails.

TAUP (output)
The scalar factors of the elem entary reflectors which represent the orthogonal \(m\) atrix \(P\). See FurtherD etails.

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, W\) ORK ( 1 ) retums the optim al
LW ORK.

LW ORK (input)
The length of the array \(W\) ORK. LW ORK \(>=\) \(\max (1, M, N)\). For optim um perform ance LW ORK >= \((M+N) \star N B\), where N B is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK aray, retums this value as the first entry of the W ORK array, and no emorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{I N F O}=-i\), the \(i\) th argum enthad an illegalvalue.

\section*{FURTHER DETAILS}

The \(m\) atrices \(Q\) and \(P\) are represented as products of elem entary reflectors:

Ifm \(>=n\), \(Q=H(1) H(2) \ldots H(n)\) and \(P=G(1) G(2) \ldots G(n-1)\)

Each H (i) and G (i) has the form :
\(H(i)=I-\operatorname{tanq} * v^{*} v^{\prime}\) and \(G(i)=I-\operatorname{tanp} * u * u^{\prime}\)
where tauq and taup are realscalars, and \(v\) and \(u\) are real vectors; \(v(1: i-1)=0, v(i)=1\), and \(v(i+1 m)\) is stored on exitin \(A(i+1 m, i) ; u(1: i)=0, u(i+1)=1\), and \(u(i+2 m)\) is stored on exit in A ( \(\mathbf{i}, \mathrm{i}+2 \mathrm{n}\) ); tauq is stored in TA U Q (i) and taup in TA UP (i).
```

Ifm < n,
$Q=H(1) H(2) \ldots H(m-1)$ and $P=G(1) G(2) \ldots G(m)$

```

Each H (i) and G (i) has the form :
\(H(i)=I-\operatorname{tanq} * v^{*} v^{\prime}\) and \(G(i)=I-\operatorname{taup} * u^{*} u^{\prime}\)
where tauq and taup are realscalars, and \(v\) and \(u\) are real vectors; \(v(1: i)=0, v(i+1)=1\), and \(v(i+2 \mathrm{~m})\) is stored on exitin \(A(i+2 m, i) ; u(1: i-1)=0, u(i)=1\), and \(u(i+1 m)\) is stored on exitin A (i,i+1 n); tauq is stored in TA U Q (i) and taup in TA UP (i).

The contents ofA on exitare illustrated by the follow ing exam ples:
\(m=6\) and \(n=5(m>n): \quad m=5\) and \(n=6(m<n):\)
( d e ul ul u1) ( \(d\) u1 u1 u1 u1
ul)
( v1 d e u2 u2) (e d u2 u2 u2
u2)
( v1 v2 d e u3 ) (v1 e d u3 u3
u3 )
( v1 v2 v3 d e ) (v1 v2 e d u4
u4)
( v1 v2 v3 v4 d ) (v1 v2 v3 e d u5 )
( v1 v2 v3 v4 v5 )
where \(d\) and e denote diagonal and off-diagonal elem ents of \(B\), videnotes an elem ent of the vectordefining \(H\) (i), and ui an elem entof the vectordefining G (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgecon -estim ate the reciprocal of the condition num ber of a general real matrix \(A\), in either the 1 -norm orthe infinity-norm, using the LU factorization com puted by SGETRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGECON NORM,N,A,LDA,ANORM,RCOND,W ORK,W ORK2,\mathbb{NFO)}}\mathbf{N}\mathrm{ (N,A}
CHARACTER * 1NORM
\mathbb{NTEGERN,LDA,}\mathbb{N}FO
INTEGER W ORK2 (*)
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION A (LDA,*),W ORK (*)
SUBROUT\mathbb{NEDGECON_64 NORM,N,A,LDA,ANORM,RCOND,W ORK,W ORK2,}
\mathbb{NFO)}
CHARACTER * 1 NORM
INTEGER*8N,LDA,INFO
\mathbb{NTEGER*8 W ORK2 (*)}
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION A (LDA,*),W ORK (*)

```
F95 INTERFACE
    SU BROUTINE GECON \(\mathbb{N} O R M, \mathbb{N}], A,[L D A], A N O R M, R C O N D,[W O R K],[W\) ORK2],
        [ \(\mathbb{N} F O\) ])
    CHARACTER (LEN=1)::NORM
    \(\mathbb{N} T E G E R:: N, L D A, \mathbb{N} F O\)
    \(\mathbb{I N T E G E R , D} \mathbb{I M}\) ENSION (:) ::W ORK2
    REAL (8) ::ANORM,RCOND
    REAL (8),D IM ENSION (:) ::W ORK

SU BROUTINE GECON_64 \(\mathbb{N} O R M, \mathbb{N}], A,[L D A], A N O R M, R C O N D,[W O R K],[\mathbb{W} O R K 2]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::NORM
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{LD} A, \mathbb{N}\) FO
\(\mathbb{N}\) TEGER (8), D \(\mathbb{I M} E N S I O N(:):: W\) ORK 2
REAL (8) ::ANORM, RCOND
REAL (8), D \(\mathbb{I M}\) ENSION (:) ::W ORK
REAL (8), D \(\mathbb{I M}\) ENSION (:,:) :: A

\section*{C INTERFACE}
\#include <sunperfh>
void dgecon (charnorm, intn, double *a, int lda, double anorm, double *rcond, int *info);
void dgecon_64 (charnorm , long n, double *a, long lda, double anorm , double *roond, long *info);

\section*{PURPOSE}
dgecon estim ates the reciprocal of the condition num ber of a general real matrix A, in either the 1 -norm or the infinity-norm, using the LU factorization com puted by SGETRF .

A \(n\) estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as
\(\operatorname{RCOND}=1 /(\operatorname{norm}(A) * \operatorname{norm}(\operatorname{inv}(A)))\).

\section*{ARGUMENTS}

NORM (input)
Specifies w hether the 1 -norm condition num ber or the infinity-norm condition num ber is required:
\(=\eta^{\prime}\) or \(\mathrm{D}^{\prime}\) : 1-norm;
= I ': Infinity-norm .

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input) The factors \(L\) and \(U\) from the factorization \(A=\) P*L*U as com puted by SG ETRF.

LD A (input)
The leading dim ension of the array \(A . L D A>=\) \(\max (1, N)\).

\section*{ANORM (input)}

IfNORM = 1 'or 0 ', the 1 -nom of the original
\(m\) atrix \(A\). IfNORM = \(I\) ', the infinity-norm of the originalm atrix A.

RCOND (output)
The reciprocal of the condition num ber of the
\(m\) atrix \(A\), computed as RCOND \(=1 /(\) norm (A) * norm (inv (A))).

W ORK (w orkspace)
dim ension ( \(4 * N\) )

W ORK 2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgeequ -com pute row and colum n scalings intended to equilibrate an \(\mathrm{M}-b y-\mathrm{N} m\) atrix A and reduce its condition num ber

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGEEQUM,N,A,LDA,R,C,ROW CN,COLCN,AMAX,}
\mathbb{NFO)}
\mathbb{NTEGERM,N,LDA,}\mathbb{NFO}
DOUBLE PRECISION ROW CN,COLCN,AM AX
DOUBLE PRECISION A (LDA,*),R (*),C (*)
SU BROUT\mathbb{NEDGEEQU_64M,N,A,LDA,R,C,ROW CN,COLCN,AMAX,}
\mathbb{NFO)}
\mathbb{NTEGER*8M,N,LDA,INFO}
DOUBLE PRECISION ROW CN,COLCN,AMAX
DOUBLE PRECISION A (LDA,*),R (*),C (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GEEQU (M ], \(\mathbb{N}\) ],A, [LDA],R,C,ROW CN,COLCN, AMAX, \([\mathbb{N F O}])\)
\(\mathbb{N}\) TEGER ::M,N,LDA, \(\mathbb{N} F O\)
REAL (8) ::ROW CN,COLCN,AMAX
REAL (8),D \(\mathbb{M}\) ENSION (:) ::R,C
REAL (8), D \(\mathbb{M}\) ENSIO N (:,:) ::A
SU BROUTINE GEEQU_64 (M ], \(\mathbb{N}], A,[L D A], R, C, R O W C N, C O L C N\), AMAX, [ \(\mathbb{N F O}\) ])
\(\mathbb{N} T E G E R(8):: M, N, L D A, \mathbb{N} F O\)

REAL (8) ::ROW CN, COLCN,AMAX
REAL (8),D \(\mathbb{I}\) ENSION (:) ::R,C
REAL (8),D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dgeequ (intm, intn, double *a, int lda, double *r, double *c, double *row cn, double *colen, double *am ax, int*info);
void dgeequ_64 (long \(m\), long \(n\), double *a, long lda, double *r, double *c, double *row cn, double *colen, double *am ax, long *info);

\section*{PURPOSE}
dgeequ com putes row and colum n scalings intended to equilibrate an M -by-N m atrix A and reduce its condition num ber. R retums the row scale factors and \(C\) the colum \(n\) scale factors, chosen to tey to \(m\) ake the largestelem ent in each row and column of the matrix \(B\) with elements \(B(i, \gamma)=R(i) * A(i, \lambda) * C(j)\) have absolute value 1 .
\(R\) (i) and C (i) are restricted to be betw een SM LN UM \(=\) sm allest safe num ber and B IG N UM = largest safe num ber. U se of these scaling factors is notguaranteed to reduce the condition num berofA butw orks w ellin practice.

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(A . M>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input) The \(M\)-by -N m atrix w hose equilibration factors are to be com puted.

LDA (input)
The leading dim ension of the array A. LDA >= \(m a x(1, M)\).

R (output)
If \(\mathbb{N F O}=0\) or \(\mathbb{N} F O>M, R\) contains the row scale
factors forA.

C (output)

If \(\mathbb{N} F O=0, C\) contains the colum \(n\) scale factors forA.

ROW CN (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O>M, R O W C N\) contains the ratio of the sm allest \(R\) (i) to the largest \(R\) (i). If ROW CN >= 0.1 and AM AX is nether too large nortoo sm all, it is notw orth scaling by R .

COLCN (output)
If \(\mathbb{N} F O=0, C O L C N\) contains the ratio of the sm allest C (i) to the largestC (i). IfC O LCN >=0.1, it is notw orth scaling by \(C\).
AMAX (output)
A bsolute value of largestm atrix elem ent. IfA M A X
is very close to overflow orvery close to underflow , the m atrix should be scaled.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvahue
\(>0\) : if \(\mathbb{N F O}=i\), and \(i\) is
\(<=\mathrm{M}\) : the i-th row ofA is exactly zero
> M : the ( \((-\mathrm{M})\) )-th collum n ofA is exactly zero

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dgees -com pute foran N boy-N real nonsymm etric m atrix A,
the eigenvalues, the realSchur form T, and, optionally, the
m atrix ofSchurvectors Z

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGEES (OOBZ,SORTEV,SELECT,N,A,LDA,NOUT,W R,W I,Z,}
LD Z,W ORK,LDW ORK,W ORK 3,\mathbb{NFO)}

```
CHARACTER * 1 JOBZ, SORTEV
\(\mathbb{N}\) TEGER N,LDA,NOUT,LD Z,LDW ORK, \(\mathbb{N} F\) O
LOG ICAL SELECT
LO G ICAL W ORK 3 (*)
DOUBLE PRECISION A (LDA,*),W R (*),W I(*), Z (LDZ,*),W ORK (*)
SU BROUTINE DGEES_64 (JOBZ,SORTEV,SELECT,N,A,LDA,NOUT,WR,W I, Z,
    LD Z,W ORK,LDW ORK,W ORK 3, \(\mathbb{N} F\) ) )
CHARACTER * 1 JOBZ, SORTEV
\(\mathbb{N}\) TEGER*8N,LDA,NOUT,LDZ,LDW ORK, \(\mathbb{N}\) FO
LOG ICAL*8 SELECT
LOG ICAL*8W ORK 3 (*)
DOUBLE PRECISION A (LDA,*),W R (*),W I(*), Z (LDZ,*),W ORK (*)

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N E}\) GEES (JOBZ, SORTEV ,SELECT, \(\mathbb{N}], A,[L D A], N O U T, W R, W I, Z\), [LD Z ], [W ORK ], [LDW ORK ], [W ORK 3], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1)::JOBZ,SORTEV
\(\mathbb{N}\) TEGER ::N,LDA,NOUT,LDZ,LDW ORK, \(\mathbb{N} F O\)
LOG ICAL :: SELECT
LOG ICAL,D IM ENSION (:) ::W ORK 3

REAL (8), D \(\mathbb{M} E N S I O N(:):: W R, W I, W O R K\)
REAL (8), D IM ENSION (:,:) ::A, Z

SU BROUTINE GEES_64 (OBZ,SORTEV, SELECT, \(\mathbb{N}], A,[L D A], N O U T, W R, W I\), Z, [LD Z], \([\mathbb{W} O R K],[L D W O R K],[W O R K 3],[\mathbb{N} F O])\)

CHARACTER ( \(L E N=1\) ) : : JOBZ, SORTEV
\(\mathbb{N}\) TEGER (8) :: N , LDA , NOUT,LD Z, LDW ORK, \(\mathbb{N} F O\)
LOGICAL (8) :: SELECT
LOGICAL (8), D \(\mathbb{M}\) ENSION (:) ::WORK 3
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W R,W I,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A, Z

\section*{C INTERFACE}
\#include <sunperfh>
void dgees(char jobz, char sortev, int(*select) (double,double), intn, double *a, int lda, int *nout, double *W r, double *W i, double *z, intldz, int*info);
void dgees_64 (char jobz, char sortev, long (*select) (double,double), long n, double *a, long lda, long *nout, double *w r, double *w i, double *z, long ldz, long *info);

\section*{PURPOSE}
dgees com putes for an \(N\) boy -N realnonsym \(m\) etric \(m\) atrix \(A\), the eigenvalues, the real Schur form \(T\), and, optionally, the \(m\) atrix of Schurvectors Z. This gives the Schur factorization \(A=Z * T *(Z * * T)\).

Optionally, italso orders the eigenvalues on the diagonal of the realSchur form so that selected eigenvalues are at the top left. The leading colum ns of \(Z\) then form an orthonorm albasis forthe invariantsubspace coresponding to the selected eigenvalues.

A matrix is in real Schur form if it is upper quasitriangularw ith 1 -by-1 and 2 -by -2 blocks. 2 -by -2 blocksw ill be standardized in the form
[ a b ]
\(\left[\begin{array}{ll}\mathrm{c} & \mathrm{a}\end{array}\right]\)
where \(\mathrm{b}^{*} \mathrm{c}<0\). The eigenvalues of such a block are a +squt (bc).

\section*{ARGUMENTS}

JO BZ (input)
\(=\mathrm{N}\) ': Schurvectors are notcom puted;
\(=\mathrm{V}\) : Schurvectors are com puted.

SORTEV (input)
Specifies w hether or not to order the eigenvalues
on the diagonalof the Schur form . = N ': Eigenvalues are notordered;
\(=S\) ': Eigenvalues are ordered (see SELECT).

SELECT (input)
SELECT m ustbe declared EXTERNAL in the calling subroutine. If SORTEV \(=S^{\prime}\), SELECT is used to selecteigenvalues to sort to the top leftof the Schur form. If SORTEV = \(N^{\prime}\), SELECT is not referenced. A n eigenvalue W R ( 1\()+\operatorname{sqnt}(-1) * W I(1)\) is selected ifSELECT (W R ( 7 ) , \(\mathrm{N} I(\mathcal{j})\) ) is tue; ie., if either one of a com plex conjugate pair of eigenvalues is selected, then both com plex eigenvalues are selected. N ote that a selected com plex eigenvalue m ay no longer satisfy SELECT \((\mathbb{W} R(\mathcal{J}) \mathbb{W} I(\mathcal{J})=\) .TRUE.afterordering, since ordering may change the value of com plex eigenvalues (especially if the eigenvalue is ill-conditioned); in this case \(\mathbb{N}\) FO is set to \(N+2\) (see \(\mathbb{N}\) FO below).

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).

A (input/output)
On entry, the \(N\) boy -N m atrix A. On exit, A has been overw ritten by its realSchur form \(T\).

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

NOUT (output)
If SORTEV = N',NOUT = 0. IfSORTEV = S', NOUT
= num ber of eigenvalues (aftersorting) forw hich
SELECT is true. (C om plex conjugate pairs forw hich
SELECT is true foreithereigenvalue countas 2.)

\section*{W R (output)}

W R and W I contain the real and im aginary parts, respectively, of the com puted eigenvahues in the sam e order that they appear on the diagonal of the output Schur form T. C om plex conjugate pairs of eigenvalueswill appear consecutively w ith the
eigenvalue having the positive in aginary part first.

W I (output)
See the description forW \(R\).
Z (output)
If \(\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{Z}\) contains the orthogonalm atrix Z of Schurvectors. If \(J 0 \mathrm{BZ}=\mathrm{N}^{\prime}, \mathrm{Z}\) is notreferenced.
LD Z (input)
The leading dim ension of the array Z . LD \(\mathrm{Z}>=1\); if \(\mathrm{JO} \mathrm{BZ}=\mathrm{V}\) ', LD \(\mathrm{Z}>=\mathrm{N}\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) contains the optim al LDW ORK.

LDW ORK (input)
The dimension of the amay \(W\) ORK. LDW ORK >= \(m a x(1,3 \star N)\). For good perform ance, LDW ORK must generally be larger.

If LD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 3 (w orkspace)
dim ension ( N\() \mathrm{N}\) ot referenced if \(S O\) RTEV \(=\mathrm{N}^{\prime}\).
\(\mathbb{N F O}\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue.
\(>0:\) if \(\mathbb{N F O}=i\), and \(i\) is
\(<=N\) : the Q R algorithm failed to com pute all the eigenvalues; elem ents \(1: \mathbb{H}-1\) and i+ \(1 \mathbb{N}\) ofW R and W I contain those eigenvalues which have converged; if \(J O B Z=V^{\prime}, Z\) contains the \(m\) atrix which reduces \(A\) to its partially converged Schur form . \(=\mathrm{N}+1\) : the eigenvalues could not be reordered because som e eigenvalues w ere too close to separate (the problem is very ill-conditioned); \(=\mathrm{N}+2\) : after reordering, roundoff changed values of som e com plex eigenvalues so that leading eigenvalues in the Schur form no longersatisfy SELECT=TRUE. This could also be caused by underflow due to scaling.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dgeesx - com pute foran N -by-N realnonsymm etric m atrix A,
the eigenvalues, the realSchur form T, and, optionally, the
m atrix ofSchurvectors Z

```

\section*{SYNOPSIS}
```

SUBROUTINE DGEESX (JOBZ,SORTEV,SELECT,SENSE,N,A,LDA,NOUT,W R,
W I, Z,LD Z,SRCONE,RCONV,W ORK,LDW ORK,IN ORK2,LDW RK2,BW ORK 3,
\mathbb{NFO)}
CHARACTER * 1 JOBZ,SORTEV,SEN SE
INTEGERN,LDA,NOUT,LDZ,LDW ORK,LDW RK2, INFO
\mathbb{NTEGER IN ORK2(*)}
LOG ICAL SELECT
LOG ICAL BW ORK3(*)
DOUBLE PRECISION SRCONE,RCONV
DOUBLE PRECISION A (LDA,*),W R (*),W I(*),Z (LD Z,*),W ORK (*)
SU BROUTINE DGEESX_64 (JOBZ,SORTEV,SELECT,SENSE,N,A,LDA,NOUT,
W R,W I, Z,LD Z,SRCONE,RCONV,W ORK,LDW ORK,IN ORK2,LDW RK2,
BW ORK 3, \mathbb{NFO)}

```
CHARACTER * 1 JOBZ, SORTEV, SEN SE
\(\mathbb{N}\) TEGER*8N,LDA,NOUT,LD Z,LDW ORK,LDW RK 2, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R * 8 \mathbb{I N}\) ORK 2 (*)
LOG ICAL*8 SELECT
LOG ICAL*8BW ORK 3 (*)
DOUBLE PRECISION SRCONE,RCONV
D OUBLE PRECISION A (LDA, \(\left.{ }^{\star}\right), \mathrm{W} R(*), W I(*), Z(L D Z, *), W\) ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GEESX ( JOBZ , SORTEV, SELECT, SENSE, \(\mathbb{N}\) ],A, [LDA],NOUT,

W R, W I, Z, [LDZ], SRCONE,RCONV, [W ORK], [LDW ORK], [IW ORK2], [LDW RK2], [BW ORK 3], [ \(\mathbb{N F O}\) ])

CHARACTER (โEN=1) :: JOBZ, SORTEV , SEN SE
\(\mathbb{N}\) TEGER : : N, LDA, NOUT,LDZ,LDW ORK,LDW RK2, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{I W}\) ORK2
LOGICAL :: SELECT
LOGICAL,D \(\mathbb{M}\) ENSION (:) ::BW ORK 3
REAL (8) :: SRCONE,RCONV
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W R , W I, W ORK
REAL (8), D \(\mathbb{M}\) ENSION (: : : : : : A , Z
```

SU BROUTINE GEESX_64 (DOBZ, SORTEV, SELECT, SENSE, $\mathbb{N}$ ],A, [LDA ],NOUT,
W R, W I, Z, [LDZ], SRCONE,RCONV, $\mathbb{W} O R K],[L D W O R K],[\mathbb{W} O R K 2]$,
[LDW RK 2], [BW ORK 3], [ $\mathbb{N F O}$ ])

```
CHARACTER (LEN=1) :: JOBZ, SORTEV, SEN SE
\(\mathbb{N} T E G E R(8):: N\), LDA, NOUT,LDZ,LDW ORK, LDW RK2, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{I N} O R K 2\)
LOGICAL (8) :: SELECT
LOG ICAL (8), D \(\mathbb{I M} E N S I O N(:):: B W O R K 3\)
REAL (8) :: SRCONE, RCONV
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W R , W I, W ORK
REAL (8), D \(\mathbb{M}\) ENSION (: : : : : : A , Z

\section*{C INTERFACE}
\#include <sunperfh>
void dgeesx (char jobz, char sortev, int(*select) (double,double), char sense, intn, double *a, int lda, int *nout, double *W \(r\), double \({ }^{*}\) w i, double *z, intldz, double *srcone, double *rconv, int*info);
void dgeesx_64 (char jंbz, char sortev, long (*select) (double,double), charsense, long n, double *a, long lda, long *nout, double *w r, double *w i, double * z, long ldz, double *srcone, double *roonv, long *info);

\section*{PURPOSE}
dgeesx com putes for an \(N\) boy \(-N\) real nonsym \(m\) etric \(m\) atrix \(A\), the eigenvahues, the realSchur form T, and, optionally, the \(m\) atrix of Schurvectors \(Z\). This gives the Schur factorization \(A=Z * T *(Z * * T)\).

O ptionally, italso orders the eigenvalues on the diagonal of the realSchur form so thatselected eigenvalues are at
the top left; com putes a reciprocalcondition num ber for the average of the selected eigenvalues (RCONDE); and com putes a reciprocal condition num ber for the right invariant subspace comesponding to the selected eigenvalues (RCONDV). The leading colum ns of \(Z\) form an orthonorm al basis for this invariant subspace.

For further explanation of the reciprocal condition num bers RCONDE and RCONDV, see Section 4.10 of the LAPACK U sers' G uide (w here these quantities are called s and sep respectively).

A realm atrix is in realSchur form if it is upper quasitriangularw ith 1-by-1 and 2 -by- 2 blocks. 2 -by- 2 blocksw ill be standardized in the form
[ a b ]
[ \(\mathrm{c} a \mathrm{a}\) ]
\(w\) here \(b^{*} c<0\). The eigenvalues of such a block are a +sqrt(bc).

\section*{ARGUMENTS}

JOBZ (input)
= N ':Schurvectors are not com puted;
= V ':Schurvectors are com puted.

SORTEV (input)
Specifies w hether ornot to order the eigenvalues on the diagonal of the Schur form . = N ': Eigenvalues are not ordered;
= S':E igenvahues are ordered (see SELECT) .
SELECT (input)
SELECT mustbe declaredEXTERNAL in the calling subroutine. If SORTEV = S', SELECT is used to selecteigenvalues to sort to the top left of the Schurform. If SORTEV = N', SELECT is notreferenced. An eigenvalue \(W\) R \((j)+\operatorname{sqnt}(-1) * W I(j)\) is selected if SELECT \((\mathbb{N} R(\mathcal{J}), N I(\mathcal{J})\) is true; ie., if either one of a com plex conjugate pair of eigenvalues is selected, then both are. N ote that a selected com plex eigenvalue \(m\) ay no longer satisfy \(\operatorname{SELECT} \mathbb{N} R(\mathcal{J}, \mathbb{N} I(\mathcal{j})=\) TRUE.after ordering, since ordering \(m\) ay change the value of com plex eigenvalues (especially if the eigenvalue is illconditioned); in this case \(\mathbb{I N F O} \mathrm{m}\) ay be set to \(\mathrm{N}+3\) (see \(\mathbb{I N} F O\) below ).

D eterm ines which reciprocal condition num bers are com puted. = N ': N one are com puted;
\(=\mathrm{E}\) ': C om puted for average of selected eigenvalues only;
= V ': C om puted for selected right invariant subspace only;
\(=\mathrm{B}\) ': C om puted forboth. If SEN \(S E=\mathrm{E}^{\prime}, \mathrm{V}^{\prime}\) or B',SORTEV mustequal \(S^{\prime}\).

N (input) The order of the m atrix A \(. \mathrm{N}>=0\).

A (input/output)
On entry, the \(N\) boy \(-N m\) atrix A. On exit, A is overw ritten by its realSchur form T.
LDA (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

NOUT (output)
If SORTEV \(=\mathrm{N}\) ', NOUT \(=0\). If SORTEV \(=\mathrm{S}\) ', NOUT
= num ber of eigenvalues (aftersorting) forw hich
SELECT is true. (C om plex conjugate pairs forw hich
SELECT is true fore thereigenvalue countas 2 .)
W R (output)
W R and W Icontain the real and im aginary parts, respectively, of the com puted eigenvalues, in the sam e order that they appearon the diagonal of the output Schur form T. C om plex conjugate pairs of eigenvalues appear consecutively \(w\) ith the eigenvalue having the positive im aginary part first.

W I (output)
See the description forW R.
Z (output)
If \(\mathrm{OBBZ}=\mathrm{V}\) ', Z contains the orthogonalm atrix Z of Schurvectors. If JO BZ \(=N^{\prime}\), Z is notreferenced.

LD \(Z\) (input)
The leading dim ension of the array \(Z\). LD \(Z>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD \(\mathrm{Z}>=\mathrm{N}\).

SRCONE (output)
If SENSE = E' or B', SRCONE contains the reciprocal condition num ber for the average of the selected eigenvalues. N ot referenced if SEN SE = N 'or V '.

RCONV (output)
If SEN SE = V 'or B ', RCONV contains the reciprocal condition num ber for the selected right invariant subspace. N ot referenced if \(\operatorname{SEN} S E=\mathrm{N}^{\prime}\) or E'.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the array W ORK. LDW ORK >= \(\max (1,3 \star \mathrm{~N})\). A lso, if SENSE = E'or V 'or B',
LDW ORK >=N+2*NOUT* \((\mathbb{N}-N O U T)\), where NOUT is the
num ber of selected eigenvalues com puted by this routine. N ote that \(\left.\mathrm{N}+2{ }^{*} \mathrm{NOUT} \mathrm{N}^{*} \mathrm{~N}+\mathrm{NOUT}\right)<=\mathrm{N}+\mathrm{N} * \mathrm{~N} / 2\).
For good perform ance, LDW ORK mustgenerally be larger.

IW ORK 2 (w orkspace/output)
N ot referenced if SEN \(S E=\mathrm{N}\) 'or \(\mathrm{E}^{\prime}\). . On exit, if \(\mathbb{N} F O=0, \mathbb{I}\) ORK 2 (1) retums the optim alld W RK 2.

LD W RK 2 (input)
The dim ension of the array \(\mathbb{I W}\) ORK 2. LDW RK 2 >= 1;
if SENSE \(=V\) 'or \(B^{\prime}\),LDW RK2 \(>=\) NOUT* \((N-N O U T)\).

BW ORK 3 (w orkspace)
dim ension ( \(\mathbb{N}\) ) N ot referenced if \(S O R T E V=\mathrm{N}^{\prime}\).
\(\mathbb{I N F O}\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum ent had an illegalvalue.
\(>0:\) if \(\mathbb{I N F O}=i\), and \(i\) is
\(<=N\) : the \(Q R\) algorithm failed to com pute all the eigenvalues; elem ents \(1: \mathbb{I} \mathrm{O}-1\) and i+ 1 N ofW R and W I contain those eigenvalues which have converged; if \(\mathrm{JOBZ}=\mathrm{V}\) ', Z contains the transform ation w hich reduces A to its partially converged Schur form. \(=\mathrm{N}+1\) : the eigenvalues could not be reordered because some eigenvalues were too close to separate (the problem is very ill-conditioned); = \(\mathrm{N}+2\) : after reordering, roundoff changed values of som e com plex eigenvalues so that leading eigenvalues in the Schur form no longer satisfy SELEC T= TRU E. This could also be caused by underflow due to scaling.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgeev - com pute for an N -by- N real nonsym \(m\) etric \(m\) atrix \(A\), the eigenvalues and, optionally, the left and/orright eigenvectors

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DGEEV (OOBVL,NOBVR,N,A,LDA,W R,W I,VL,LDVL,VR,LDVR,} W ORK,LDW ORK, $\mathbb{N} F O$ )

```

CHARACTER * 1 JobvL, JOBVR
\(\mathbb{N}\) TEGER \(N, L D A, L D V L, L D V R, L D W O R K, \mathbb{N} F O\)
DOUBLE PRECISION A (LDA, \()^{*}\), W R (*), W \(I\left({ }^{\star}\right), ~ V L(L D V L, \star)\), VR(LDVR,*), WORK (*)

SU BROUTINEDGEEV_64 (JOBVL, JOBVR,N,A,LDA,WR,W I, VL,LDVL,VR, LDVR, W ORK, LDW ORK, \(\mathbb{N} F O\) )

CHARACTER * 1 JOBVL, JOBVR
\(\mathbb{N} T E G E R * 8 N, L D A, L D V L, L D V R, L D W O R K, \mathbb{N} F O\)
DOUBLE PRECISION A (LDA, *), WR(*), W I(*), VL (LDVL, \(\left.{ }^{*}\right)\),
VR (LDVR,*),W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GEEV (JOBVL, JOBVR, \(\mathbb{N}\) ],A, [LDA],W R,W I, VL, [LDVL],VR, [LDVR], [W ORK], [LD W ORK], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1)::JOBVL, JOBVR
\(\mathbb{N} T E G E R:: N, L D A, L D V L, L D V R, L D W O R K, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::WR,W I,W ORK
REAL (8),D IM ENSION (:,:) ::A,VL,VR
SU BROUTINE GEEV_64 (JOBVL, JOBVR, \(\mathbb{N}], A,[L D A], W R, W I, V L,[L D V L]\),

VR, [LDVR], [W ORK], [LDW ORK], [ \(\mathbb{N F O}])\)

CHARACTER ( \(L E N=1\) ) :: JOBVL, OBVR
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{LD} A, L D V L, L D V R, L D W O R K, \mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W R , W I, W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A , VL, VR

\section*{C INTERFACE}
\#include <sunperfh>
void dgeev (char jobvl, char jobvr, int \(n\), double *a, int lda, double *w r, double *w i, double *vl, int ldvl, double *Vr, int ldvr, int*info);
void dgeev_64 (char j̀jbvl, char jobvr, long n, double *a, long lda, double *W r, double *W i, double *vl, long ldvl, double *vr, long ldvr, long *info);

\section*{PURPOSE}
dgeev com putes for an \(N\) boy -N realnonsym \(m\) etric \(m\) atrix \(A\), the eigenvalues and, optionally, the left and/or righteigenvectors.

The righteigenvectorv ( \()\) ) ofA satisfies
A * \(\mathrm{v}(\mathrm{J})=\operatorname{lam} \mathrm{bda}(\mathrm{j}) * \mathrm{v}(\boldsymbol{j})\)
w here lam bda ( 7 ) is its eigenvalue.
The lefteigenvectoru ( \()\) of ) satisfies

where \(u(j) * * H\) denotes the conjugate transpose ofu ( \(\mathcal{j}\) ).

The com puted eigenvectors are norm alized to have Euclidean norm equal to 1 and largest com ponent real.

\section*{ARGUMENTS}
\(J 0 \mathrm{BVL}\) (input)
\(=\mathrm{N}\) : lefteigenvectors of \(A\) are not com puted;
\(=\mathrm{V}\) ': lefteigenvectors of A are com puted.

JOBVR (input)
\(=\mathrm{N}\) ': righteigenvectors of \(A\) are notcom puted;
\(=\mathrm{V}\) ': righteigenvectors of A are com puted.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).

A (input/output)
On entry, the \(N\) boy -N m atrix A. On exit, A has
been overw ritten.

LDA (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

W R (output)
W R and W Icontain the real and im aginary parts, respectively, of the com puted eigenvalues. Com plex conjugate pairs of eigenvalues appear consecutively w ith the eigenvalue having the positive im aginary part first.

\section*{W I (output)}

See the description forW R.
VL (output)
If \(\mathrm{JOBVL}=\mathrm{V}\) ', the left eigenvectors \(\mathrm{u}(\mathrm{y})\) are stored one afteranother in the colum ns of V L, in the sam e order as theireigenvalues. If \(\mathrm{JOBVL}=\) N ', VL is not referenced. If the \(j\) th eigenvalue is real, then \(u(\eta)=V L(: r)\), the \(j\) th column of \(V L\). If the \(j\) th and ( \(j+1\) )-steigenvalues form a com plex conjugate pair, then \(u(\mathcal{j})=V L(:, i)+\) \(i^{\star} V L(: j+1)\) and
\(u(j+1)=V L(:, 7)-i \star V L(:, j+1)\).
LDVL (input)
The leading dim ension of the array VL. LD V L >=1; if JO BVL = V', LDVL >= N .

VR (input)
If \(\mathrm{JOBVR}=\mathrm{V}\) ', the right eigenvectors \(\mathrm{V}(\mathrm{y})\) are stored one afteranother in the colum ns of VR, in the sam e order as theireigenvalues. If \(J 0 B V R=\) \(N^{\prime}, \mathrm{VR}\) is not referenced. If the \(j\) th eigenvalue is real, then \(v(\mathcal{I})=V R(:, \gamma)\), the \(j\) th column of \(V R\). If the \(j\) th and ( \(j+1\) )-steigenvalues form a com plex conjugate pair, then \(\mathrm{V}(\mathcal{j})=\operatorname{VR}(:, 7)+\) i*VR (: \(1 \mathrm{j}+1\) ) and
\(\mathrm{V}(j+1)=\operatorname{VR}(:, 7)-\mathrm{i} \star \mathrm{V}(:, j+1)\).
LDVR (input)
The leading dim ension of the array VR. LD V \(\mathrm{R}>=1\); if \(\mathrm{JOBVR}=\mathrm{V}\) ', LDVR \(>=\mathrm{N}\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al
LDW ORK.
LDW ORK (input)

The dimension of the amray \(W\) ORK. LDW ORK >= \(\mathrm{max}(1,3 * \mathrm{~N})\), and if \(\mathrm{JOBVL}=\mathrm{V}\) 'orJobVR \(=\mathrm{V}\) ', LDW ORK >= 4*N. Forgood perform ance, LDW ORK must generally be larger.

IfLDW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{N}\) FO \(=-\)-i, the \(i\)-th argum enthad an illegalvalue.
> 0: if \(\mathbb{N} F O=i\), the \(Q R\) algorithm failed to com -
pute all the eigenvalues, and no eigenvectors have
been com puted; elem ents i+ \(1 \mathbb{N}\) of R and \(W\) I contain eigenvalues which have converged.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgeevx - com pute foran \(N\)-by -N realnonsym \(m\) etric \(m\) atrix \(A\), the eigenvalues and, optionally, the left and/or right eigenvectors

\section*{SYNOPSIS}
```

SUBROUTINE DGEEVX (BALANC,JOBVL,JOBVR,SENSE,N,A,LDA,W R,W I,VL,
LDVL,VR,LDVR,\PiO,IHI,SCALE,ABNRM,RCONE,RCONV,W ORK,
LDW ORK,INORK2,\mathbb{NFO)}
CHARACTER * 1 BALANC,JOBVL,JOBVR,SENSE

```

```

\mathbb{NTEGER IN ORK2(*)}
DOUBLE PRECISION ABNRM
DOUBLE PRECISION A (LDA,*), W R (*), W I(*), VL (LDVL,*),
VR(LDVR,*),SCALE (*),RCONE (*),RCONV (*),W ORK (*)
SU BROUTINE DGEEVX_64(BALANC,JO BVL,JOBVR,SEN SE,N,A,LDA,W R,W I,
VL,LDVL,VR,LDVR,\PiO,HI,SCALE,ABNRM,RCONE,RCONV,W ORK,
LDW ORK,INORK2,\mathbb{NFO)}

```
CHARACTER * 1 BALANC, JOBVL, JOBVR, SENSE
\(\mathbb{N}\) TEGER*8 N,LDA,LDVL,LDVR, \(\mathbb{L O}, \mathbb{H} I, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER*8 \(\mathbb{I N}\) ORK 2 (*)
DOUBLE PRECISION ABNRM
DOUBLE PRECISION A (LDA,*), WR(*), W I(*), VL (LDVL,*),
VR (LDVR,*),SCALE (*), RCONE (*),RCONV (*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GEEVX (BALANC, JOBVL, JOBVR, SENSE, \(\mathbb{N}], A,[L D A], W R, W\) I, VL, [LDVL],VR, [LDVR], \(\mathbb{L} O, \mathbb{H} I, S C A L E, A B N R M, R C O N E, R C O N V\), [W ORK ], [LDW ORK], [IW ORK 2], [ \(\mathbb{N} F O\) ])

CHARACTER ( \(\llcorner E N=1\) ) : : BA LANC, JOBVL, JO BVR, SEN SE
\(\mathbb{N}\) TEGER :: N, LDA ,LDVL,LDVR, \(\mathbb{L} O, \mathbb{H} I, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{I W} O R K 2\)
REAL (8) ::ABNRM
REAL (8), D \(\mathbb{M} E N S I O N(:):: W R, W I, S C A L E, R C O N E, R C O N V, W O R K\)
REAL (8), D \(\mathbb{M}\) ENSION (: : : : : : A, VL, VR
 W I, VL, [LDVL], VR, [LDVR], ILO, IHI, SCALE, ABNRM,RCONE,RCONV, \([\mathbb{W} O R K],[L D W O R K],[\mathbb{W}\) ORK 2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::BALANC, OBVL, JOBVR, SENSE
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{LD} A, L D V L, L D V R, \Pi O, \mathbb{H} I, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{I N} O R K 2\)
REAL (8) ::ABNRM
REAL (8), D \(\mathbb{M} E N S I O N(:):: W R, W I, S C A L E, R C O N E, R C O N V, W O R K\)
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A, VL, VR

\section*{C INTERFACE}
\#include <sunperfh>
void dgeevx (charbalanc, char jobvl, char jobvr, charsense, int \(n\), double *a, int lda, double \({ }_{\mathrm{W}} \mathrm{r}\), double *\({ }_{\mathrm{w}} \mathrm{i}_{\text {, }}\) double *Vl, int ldvl, double *Vr, int ldvr, int *ilo, int *ihi, double *scale, double *abnrm , double *roone, double *roonv, int *info);
void dgeevx_64 (charbalanc, char jobvl, char jobvr, char sense, long \(n\), double *a, long lda, double *W r, double *W i, double *vl, long ldvl, double *vr, long ldvr, long *ilo, long *ihi, double *scale, double *abnrm, double *roone, double *rconv, long *info);

\section*{PURPOSE}
dgeevx com putes for an \(N\) boy -N real nonsymm etric \(m\) atrix \(A\), the eigenvalues and, optionally, the left and/or right eigenvectors.

Optionally also, it com putes a balancing transform ation to im prove the conditioning of the eigenvalues and eigenvectors ( \(\mathbb{L} O, \mathbb{H} I\), SCALE , and A BN RM ), reciprocal condition num bers for the eigenvalues (RCONDE), and reciprocal condition num bers for the right eigenvectors (RCONDV).

The righteigenvectorv ( \()\) ) of A satisfies
\[
\text { A * } \mathrm{v}(\mathcal{\jmath})=\operatorname{lam} \operatorname{bda}(\mathcal{j}) * v(\mathcal{j})
\]
where \(\operatorname{lam} \operatorname{bda}(j)\) is its eigenvalue.
The lefteigenvectoru ( \()\) ) of A satisfies
\[
\mathrm{u}(\mathrm{j}) * * \mathrm{H} * \mathrm{~A}=\operatorname{lam} \operatorname{bda}(\mathrm{j}) * \mathrm{u}(\hat{)})^{* *} \mathrm{H}
\]
where \(u(j) * * H\) denotes the conjugate transpose of \(u(j)\).
The com puted eigenvectors are norm alized to have Euclidean norm equalto 1 and largestcom ponent real.

B alancing a \(m\) atrix \(m\) eans perm uting the row \(s\) and colum \(n s\) to m ake itm ore nearly upper triangular, and applying a diagonalsim ilarity transform ation \(D\) * \(A\) * \(D\) ** \((-1)\), where \(D\) is a diagonalm atrix, to \(m\) ake its row \(s\) and colum ns closer in norm and the condition num bers of its eigenvalues and eigenvectors sm aller. The com puted reciprocalcondition num bers comespond to the balanced \(m\) atrix. Perm uting row \(s\) and colum ns will not change the condition num bers (in exact arithm etic) but diagonal scaling will. For further explanation of balancing, see section 4.102 of the LA PA CK U sers'G uide.

\section*{ARGUMENTS}

BALANC (input)
Indicates how the inputm atrix should be diagonally scaled and/orperm uted to im prove the conditioning of its eigenvalues. \(=\mathrm{N}\) ': D o not diagonally scale orperm ute;
\(=\mathrm{P}\) ':Perform perm utations to m ake the m atrix \(m\) ore nearly upper triangular. D o notdiagonally scale; = S':D iagonally scale the m atrix, ie. replace \(A\) by \(D * A * D * *(-1)\), where \(D\) is a diagonal \(m\) atrix chosen to \(m\) ake the row \(s\) and colum ns of \(A\) m ore equal in norm .D o notperm ute; \(=\mathrm{B}\) ':Both diagonally scale and perm ute A.

C om puted reciprocalcondition num bers \(w\) illbe for the \(m\) atrix afterbalancing and/orperm uting. Per\(m\) uting does not change condition num bers (in exact arithm etic), butbalancing does.

JOBVL (input)
\(=N\) ': lefteigenvectors of A are not com puted;
\(=\mathrm{V}\) ': lefteigenvectors of A are com puted. If
SEN SE = E 'or B', JO BV L m ust= V'.
\(J 0 B V R\) (input)
= N ': righteigenvectors of \(A\) are not com puted;
\(=\mathrm{V}\) : righteigenvectors of A are com puted. If SEN SE = E'or B', JO BV R m ust= V'.

SENSE (input)
D eterm ines which reciprocalcondition num bers are com puted. = N ': N one are com puted;
\(=\mathrm{E}: \mathrm{C}\) om puted foreigenvalues only;
\(=\mathrm{V}\) ':C om puted for righteigenvectors only;
\(=B: C\) om puted foreigenvalues and right eigenvectors.

If SEN SE = E 'or B ', both leftand right eigenvectors \(m\) ust also be com puted ( \(\mathrm{JO} \mathrm{BVL}=\mathrm{V}\) 'and \(\mathrm{JO} B V R=V\) ).

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
A (input/output)
On entry, the \(N-b y-N\) m atrix A. On exit, A has
been overw ritten. If \(J 0 B V L=V\) 'orJOBVR \(=V\) ',
A contains the realSchur form of the balanced version of the inputm atrix A.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, \mathbb{N})\).

\section*{W R (output)}

W R and W I contain the real and im aginary parts, respectively, of the com puted eigenvalues. Com plex conjugate pairs of eigenvahues w ill appear consecutively w th the eigenvalue having the positive im aginary part first.

W I (output)
See the description forW R.
VL (output)
If \(\mathrm{JOBVL}=\mathrm{V}\) ', the left eigenvectors \(\mathrm{u}(\mathrm{y})\) are stored one after another in the colum ns of V L, in the sam e order as theireigenvalues. If \(\mathrm{JOBVL}=\) N ', VL is not referenced. If the \(j\) th eigenvalue is real, then \(u(j)=V L(:, ~ j)\), the \(j\) th column of \(V L\). If the \(j\) th and ( \(j+1\) )-steigenvalues form a com plex conjugate pair, then \(u(\mathcal{J})=\mathrm{VL}(:, 7)+\) i*V L (:,j+1) and \(u(j+1)=V L(:, 7)-i \star V L(:, j+1)\).

LDVL (input)
The leading dim ension of the array VL. LD V L >=1; if \(\mathrm{JOBVL}=\mathrm{V}\) ', LDVL \(>=\mathrm{N}\) 。

\section*{VR (output)}

If \(\mathrm{OB} \mathrm{BR}=\mathrm{V}\) ', the right eigenvectors \(\mathrm{V}(\mathrm{I})\) are stored one after another in the colum ns of VR, in the sam e order as theireigenvalues. If \(J 0 B V R=\) N ', VR is not referenced. If the jth eigenvalue is real, then \(v(\mathcal{F})=V R\left(:, ~ \mathcal{I}^{\prime}\right)\), the \(j\) th column of VR. If the \(j\) th and ( \(j+1\) )-steigenvalues form \(a\) com plex conjugate pair, then \(\mathrm{v}(\mathcal{1})=\mathrm{VR}(:, 1)+\) i \({ }^{*}\) VR (: \(\ddagger+1\) ) and

LDVR (input)
The leading dim ension of the amay VR.LDVR >=1, and if \(J O B V R=V^{\prime}, L D V R>=N\).

ㅍO (output)
HO and IH Iare integervahes determ ined when A w as balanced. The balanced \(A(i, j)=0\) if \(I>J\) and \(J=1, \ldots, I \mathrm{HO}-1\) or \(I=\mathbb{H} I+1, \ldots, N\).

IH I (output)
See the description of IIO .

SCALE (output)
D etails of the perm utations and scaling factors applied w hen balancing A. IfP ( \(\mathcal{D}\) ) is the index of the row and colum \(n\) interchanged \(w\) ith row and colum \(n j\) and \(D(\mathcal{)}\) ) is the scaling factor applied to row and colum \(n j\) then \(\operatorname{SCALE}(J)=P(J)\), for \(J=1, \ldots, \Pi O-1=D(J), \quad\) for \(J=\Pi O, \ldots, \mathbb{H} I=\) \(P(J) \quad\) for \(J=\mathbb{H} I+1, \ldots, N\). The order in which the interchanges are \(m\) ade is \(N\) to \(\mathbb{H} \mathrm{I}+1\), then 1 to ILO-1.

\section*{ABNRM (output)}

The one-norm of the balanced \(m\) atrix the \(m\) axim um of the sum of absolute values of elem ents of any colum n).

RCONE (output)
RCONE ( \(\mathcal{j}\) ) is the reciprocal condition num ber of the jth eigenvalue.

RCONV (output)
RCONV (ㄱ) is the reciprocalcondition num ber of the jth righteigenvector.

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dim ension of the amay \(W\) ORK. IfSENSE \(=N^{\prime}\)
or E', LDW ORK >= max ( \(1,2 * \mathrm{~N}\) ), and if \(\mathrm{JOBVL}=\mathrm{V}^{\prime}\) orJOBVR = V',LDW ORK >= 3*N. IfSENSE = V' or B', LDW ORK >= \(N * \mathbb{N}+6)\). Forgood perform ance, LD W ORK m ustgenerally be larger.

IfLD W ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim al size of
the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by X ERBLA .

IV ORK 2 (w orkspace)
dim ension ( 2 * \(\mathrm{N}-2\) ) IfSEN SE = N 'or E', notreferenced.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue.
\(>0:\) if \(\mathbb{N} F O=i\), the \(Q R\) algorithm failed to com -
pute all the eigenvalues, and no eigenvectors or condition num bers have been com puted; elem ents 1:ITO-1 and i+ \(1 \mathbb{N}\) ofW R and W I contain eigenvalues w hich have converged.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgegs - routine is deprecated and has been replaced by routine SG G ES

\section*{SYNOPSIS}
```

SU BROUTINEDGEGS (JOBVSL,NOBVSR,N,A,LDA,B,LDB,ALPHAR,ALPHA I,
BETA,VSL,LDVSL,VSR,LDVSR,W ORK,LDW ORK,INFO)
CHARACTER * 1 JOBVSL, JOBVSR
\mathbb{NTEGERN,LDA,LDB,LDVSL,LDVSR,LDW ORK,INFO}
DOUBLE PRECISION A (LDA,*),B (LDB,*), ALPHAR (*), ALPHAI(*),
BETA (*),VSL (LDVSL,*),VSR (LDVSR,*),W ORK (*)
SU BROUT\mathbb{NE DGEGS_64 (OD BV SL,JO BV SR ,N,A ,LDA,B ,LD B ,A LPHAR,}
ALPHAI,BETA,VSL,LDVSL,VSR,LDVSR,WORK,LDW ORK,INFO)
CHARACTER * 1 JobvSL, JOBVSR
$\mathbb{N} T E G E R * 8 N, L D A, L D B, L D V S L, L D V S R, L D W O R K, \mathbb{N} F O$
D OUBLE PRECISIONA (LDA, $), B(L D B, \star)$, ALPHAR ( $\left.{ }^{\star}\right)$, ALPHAI $(\star)$, BETA ( $)$, VSL (LDVSL, $\left.{ }^{\star}\right)$,VSR (LDVSR , $), \mathrm{W}$ ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GEGS (JO BVSL, JO BVSR, \(\mathbb{N}], A,[L D A], B,[L D B], A L P H A R\), A LPHA I, BETA,VSL, [LDVSL],VSR, [LDVSR], \(\mathbb{W}\) ORK], [LDW ORK], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1)::JOBVSL, JOBVSR
\(\mathbb{N} T E G E R:: N, L D A, L D B, L D V S L, L D V S R, L D W O R K, \mathbb{N F O}\)
REAL (8),D \(\mathbb{I}\) ENSION (:) ::A LPHAR,ALPHAI,BETA, W ORK
REAL (8), D IM ENSION (:,:) ::A,B,VSL,VSR
SU BROUT \(\mathbb{N} E\) GEGS_64 (0 BV SL, JOBV SR, \(\mathbb{N}], A,[L D A], B,[L D B], A L P H A R\), A LPHAI, BETA, VSL, [LDVSL],VSR, [LDVSR], [W ORK], [LDWORK], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1) :: JOBVSL, JOBVSR
\(\mathbb{N}\) TEGER (8) :: N , LDA , LD B , LDV SL , LDV SR , LDW ORK , \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::ALPHAR,ALPHAI,BETA, W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) :: A, B, VSL , V SR

\section*{C INTERFACE}
\#include <sunperfh>
void dgegs (char jobvsl, char jobvss, intn, double *a, int
lda, double *b, intldb, double *alphar, double
*alphai, double *beta, double *vsl, int ldvsl, double *vsr, int ldvsr, int*info);
void dgegs_64 (char jंbvsl, char jंbvsr, long n, double *a, long lda, double *b, long ldb, double *alphar, double *alphai, double *beta, double *vsl, long ldvsl, double *vsr, long ldvsr, long *info);

\section*{PURPOSE}
dgegs routine is deprecated and has been replaced by routine SG G ES .

SG EG S com putes for a pair of N -by N real nonsym \(m\) etric \(m\) atrices \(A, B\) : the generalized eigenvalues (alphar+/alphai*i, beta), the realSchur form ( \(\mathrm{A}, \mathrm{B}\) ), and optionally leftand/or rightS churvectors (VSL and V SR ).
(If only the generalized eigenvalues are needed, use the driverSG EGV instead.)

A generalized eigenvalue for a pair of \(m\) atriges ( \(A, B\) ) is, roughly speaking, a scalar w or a ratio alphaßbeta \(=\mathrm{w}\), such that \(A-w * B\) is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation forbeta \(=0\), and even forboth being zero. A good beginning reference is the book, "M atrix C om putations", by G.G olub \& C .van Loan (Johns H opkins U .Press)

The (generalized) Schur form of a pair of \(m\) atriges is the result of \(m\) ultiplying both \(m\) atrices on the leftby one orthogonalm atrix and both on the rightby another orthogonalm atrix, these tw o orthogonalm atrices being chosen so as to bring the pair ofm atrices into (real) Schur form .

A pair ofm atrices \(A, B\) is in generalized realSchur form if \(B\) is upper triangularw ith non-negative diagonal and \(A\) is block uppertriangularw ith 1 -by -1 and 2 -by -2 blocks. 1-by-1 blocks correspond to real generalized eigenvalues,
while 2-by-2 blocks ofA willbe "standardized" by making the corresponding elem ents ofB have the form :
[ a 0 ]
[ 0 b ]
and the pair of corresponding 2 -by -2 blocks in \(A\) and \(B\) w ill have a com plex conjugate pair of generalized eigenvalues.

The left and rightSchurvectors are the colum ns ofV SL and VSR, respectively, where VSL and VSR are the orthogonal \(m\) atrices \(w\) hich reduce \(A\) and \(B\) to Schur form :

Schurform of \((A, B)=((N S L) * * T A(N S R),(N S L) * * T B(N S R))\)

\section*{ARGUMENTS}

JO BV SL (input)
= N ': do notcom pute the leftSchurvectors;
\(=\mathrm{V}\) : com pute the leftSchurvectors.
\(J O B V S R\) (input)
\(=N\) ': do notcom pute the rightSchurvectors;
\(=\mathrm{V}\) : com pute the rightSchurvectors.
N (input) The order of the m atrices A , B , V SL, and V SR. N \(>=0\).

A (input/output)
O \(n\) entry, the first of the pair ofm atrioes whose generalized eigenvalues and (optionally) Schur vectors are to be com puted. On exit, the general ized Schur form of A. N ote: to avoid overflow, the Frobenius norm of the \(m\) atrix A should be less than the overflow threshold.

LD A (input)
The leading dim ension ofA . LD A \(>=\max (1, \mathbb{N})\).
B (input/output)
O \(n\) entry, the second of the pair ofm atrices w hose generalized eigenvalues and (optionally) Schur vectors are to be com puted. On exit, the generalized Schur form of B. N ote: to avoid overflow, the Frobenius norm of the matrix B should be less than the overflow threshold.

LD B (input)
The leading dim ension ofB . LD B \(>=m\) ax \((1, N)\).

ALPHAR (output)
On exit, (ALPHAR ( ) + ALPHAI ( 1 *i) BETA ( \(\boldsymbol{\lambda}\) ) ,于1,...,N, will be the generalized eigenvalues.
ALPHAR ( \() ~+~ A L P H A I(\beth * i, ~ j 1, \ldots, N\) and BETA ( 7\(), 1, \ldots, N\) are the diagonals of the com plex Schur form ( \(\mathrm{A}, \mathrm{B}\) ) thatw ould result if the 2-by-2 diagonal blocks of the realSchur form of ( \(A, B\) ) w ere further reduced to triangular form using 2 łoy -2 com plex unitary transform ations. If A LPHAI( \(\mathcal{j})\) is zero, then the \(j\) th eigenvalue is real; if positive, then the jth and (j+1)-st eigenvahues are a com plex conjugate pair, w ith A LPH A I(j+1) negative.

N ote: the quotients \(A \operatorname{LPHAR}\) ( 1 ) BETA ( \()\) ) and A LPHAI ( ) , BETA ( \()\) m ay easily over-orunderflow, and BETA (ㄱ) m ay even be zero. Thus, the user should avoid naively computing the ratio alpha/beta. H ow ever, A LPH AR and A LPH A I w ill be alw ays less than and usually com parable w ith norm (A) in magnitude, and BETA alw ays less than and usually com parable \(w\) ith norm (B).

\section*{A LPH A I (output)}

See the description forA LPH A R .

\section*{BETA (output)}

See the description forA LPH A R .

VSL (input)
If JOBVSL = V',VSL willcontain the left Schur vectors. (See "Punpose", above.) N ot referenced if \(J O B V S L=N^{\prime}\).

LDVSL (input)
The leading dim ension of the \(m\) atrix VSL. LDVSL \(>=1\), and if \(\mathrm{OBVSL}=\mathrm{V}^{\prime}, \mathrm{LDVSL}>=\mathrm{N}\).

VSR (input)
If OB BVSR \(=V\) ',VSR willcontain the right Schur
vectors. (See "Puppose", above.) N ot referenced if \(J O B V S R=N^{\prime}\) 。

LDVSR (input)
The leading dim ension of the \(m\) atrix \(V\) SR.LD VSR \(>=\) 1 , and if OB B \(S R=V^{\prime}\) ', LD \(V S R>=N\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al

LDW ORK.

LDW ORK (input)
The dim ension of the array \(W\) ORK. LDW ORK \(>=\) \(\max (1,4 \star N)\). For good perform ance, LDW ORK must generally be larger. To com pute the optim alvalue of LDW ORK, call ILAENV to getblocksizes (for SGEQRF, SORM QR, and SORGQR .) Then com pute: NB-
M AX of the blocksizes for SGEQRF, SORM QR, and SORGQR The optim alLDW ORK is \(2 * N+N * \mathbb{N B + 1 )}\).

IfLD W ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK aray, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LD W ORK is issued by XERBLA.
\(\mathbb{I N} F O\) (output)
= 0: successfiulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the i-th argum enthad an ille-
galvahue.
\(=1, \ldots, N\) : The \(\mathrm{Q} Z\) teration failed. ( \(\mathrm{A}, \mathrm{B}\) ) are
not in Schurform, butALPHAR ( \()\), ALPHAI( \()\), and
BETA ( 1 ) should be comect for \(\ddagger \mathbb{N} \mathrm{FO}+1, \ldots, N\). >
N : enors that usually indicate LA PA C K problem \(s\) :
\(=\mathrm{N}+1\) : error retum from SG GBAL
\(=\mathrm{N}+2\) : error retum from \(S G E Q R F\)
\(=\mathrm{N}+3\) : error retum from \(S O R M Q R\)
\(=\mathrm{N}+4\) : error retum from \(S O R G Q R\)
\(=\mathrm{N}+5\) : error retum from SG G H RD
\(=\mathrm{N}+6\) : emor retum from SHGEQZ (other than failed
iteration) \(=\mathrm{N}+7\) : enor retum from SG GBAK (com put-
ing V SL)
\(=\mathrm{N}+8\) : emor retum from SGGBAK (com puting \(V \mathrm{SR}\) )
\(=\mathrm{N}+9\) : error retum from SLA SCL (various places)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dgegv - routine is deprecated and has been replaced by routine SG G EV

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGEGV (JOBVL,NOBVR,N,A,LDA,B,LDB,ALPHAR,ALPHAI,}
BETA,VL,LDVL,VR,LDVR,W ORK,LDW ORK,INFO)
CHARACTER * 1 JOBVL,JOBVR
\mathbb{NTEGERN,LDA,LDB,LDVL,LDVR,LDW ORK, NNFO}
DOUBLE PRECISION A (LDA,*),B (LDB,*), ALPHAR (*), ALPHAI(*),
BETA (*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
SU BROUT\mathbb{NE DGEGV_64(JOBVL,NOBVR,N,A,LDA,B,LD B ,A LPHAR,A LPHA I,}
BETA,VL,LDVL,VR,LDVR,W ORK,LDW ORK,INFO)
CHARACTER * 1 JOBVL,JOBVR
NNTEGER*8N,LDA,LDB,LDVL,LDVR,LDW ORK,\mathbb{NFO}
DOUBLE PRECISION A (LDA,*),B (LDB,*), ALPHAR (*), ALPHAI(*),
BETA (*),VL (LDVL,*),VR (LDVR,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GEGV (JOBVL, \(\mathcal{J}\) BVR, \(\mathbb{N}], A,[L D A], B,[L D B], A L P H A R\), ALPHAI, BETA, VL, [LDVL],VR, [LDVR], [W ORK], [LDW ORK], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1)::JOBVL, JOBVR
\(\mathbb{N} T E G E R:: N, L D A, L D B, L D V L, L D V R, L D W O R K, \mathbb{N F O}\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::A LPHAR,ALPHAI,BETA,W ORK
REAL (8), D IM ENSION (:,:) ::A,B,VL,VR

SU BROU T \(\mathbb{N} E\) GEGV_64 (0 BVL, JO BVR, \(\mathbb{N}], A,[L D A], B,[L D B], A L P H A R\),

CHARACTER ( \(L E N=1\) ) :: JOBVL, OBBVR
\(\mathbb{N} T E G E R(8):: N, L D A, L D B, L D V L, L D V R, L D W O R K, \mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::ALPHAR,ALPHAI,BETA,W ORK REAL (8), D \(\mathbb{M}\) ENSION (:,:) :: A , B , VL, VR

\section*{C INTERFACE}
\#include <sunperfh>
void dgegv (char jobvl, char jobvr, int \(n\), double *a, int lda, double *b, intldb, double *alphar, double *alphai, double *beta, double *vl, int ldvl, double *Vr, int ldvr, int *info);
void dgegv_64 (char j̇bvl, char j̄jbvr, long n, double *a, long lda, double *b, long ldb, double *alphar, double *alphai, double *beta, double *vl, long ldvl, double *vr, long ldvr, long *info);

\section*{PURPOSE}
dgegv routine is deprecated and has been replaced by routine SG G EV .

SG EGV com putes for a pair of \(n-b y-n\) real nonsym \(m\) etric m atrices \(A\) and \(B\), the generalized eigenvalues (alphar \(+/-\) alphai* \(i\), beta), and optionally, the left and/or right generalized eigenvectors ( L and VR).

A generalized eigenvalue for a pair of \(m\) atrices ( \(A, B\) ) is, roughly speaking, a scalar w or a ratio alpha/beta \(=\mathrm{w}\), such that \(A-w * B\) is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable intenpretation forbeta \(=0\), and even forboth being zero. A good beginning reference is the book, "M atrix C om putations", by G.G olub \& C.van Loan (Johns H opkins U .Press)

A rightgeneralized eigenvector corresponding to a generalized eigenvalue \(w\) for pair ofm atrioes \((A, B)\) is a vector \(r\) such that ( \(A-w B\) ) \(r=0\). A left generalized eigenvector is a vectorlsuch thatl**H * (A -w B) \(=0\), where l**H is the conjugate-transpose of l.

N ote: this routine perform s "fullbalancing" on A and B see "FurtherD etails", below .

\section*{ARGUMENTS}

JO BVL (input)
\(=\mathrm{N}\) ': do notcom pute the leftgeneralized eigenvectors;
\(=\mathrm{V}\) ': com pute the leftgeneralized eigenvectors.

JO BVR (input)
\(=\mathrm{N}\) : : do not com pute the right generalized eigenvectors;
\(=\mathrm{V}\) : com pute the right generalized eigenvectors.

N (input) The order of the m atriges \(\mathrm{A}, \mathrm{B}, \mathrm{VL}\), and \(\mathrm{VR} . \mathrm{N}>=\) 0.

A (input/output)
O n entry, the first of the pair ofm atrices w hose generalized eigenvalues and (optionally) generalized eigenvectors are to be com puted. On exit, the contents w ill have been destroyed. Fora description of the contents of A on exit, see "FurtherD etails", below .)

LD A (input)
The leading dim ension ofA. LD A \(>=\max (1, N)\).

B (input/output)
O n entry, the second of the pair ofm atrices w hose generalized eigenvalues and (optionally) generalized eigenvectors are to be com puted. On exit, the contents will have been destroyed. Fora description of the contents of B on exit, see "FurtherD etails", below .)

LD B (input)
The leading dim ension ofB. LD \(B>=m a x(1, N)\).

\section*{ALPHAR (output)}

On exit, (ALPHAR ( ) + ALPHAI ( 1 *i) BETA ( 1 ), \(j 1, \ldots, N\), w ill be the generalized eigenvalues. IfA LPH A I( \(\mathfrak{j}\) ) is zero, then the \(j\) th eigenvalue is real; if positive, then the \(j\) th and ( \(j+1\) )-st eigenvahues are a com plex conjugate pair, w ith A LPH A I (j+1) negative.

Note: the quotients ALPHAR ( 7 ) BETA ( \()\) and A LPHAI( ) BETA ( \()\) m ay easily over-orunderflow, and BETA ( 7 ) m ay even be zero. Thus, the user should avoid naively computing the ratio alpha/beta. H ow ever, A LPH AR and A LPHAI w ill be
alw ays less than and usually com parable w ith norm (A) in magnitude, and BETA alw ays less than and usually com parable w ith norm (B).

\section*{A LPH A I (output)}

See the description of A LPH AR .

BETA (output)
See the description of A LPH AR .
VL (output)
If \(30 \mathrm{BVL}=\mathrm{V}\) ', the left generalized eigenvectors.
(See "Purpose", above.) Realeigenvectors take one colum n, com plex take tw o colum ns, the first for the realpart and the second for the im aginary part. Complex eigenvectors correspond to an eigenvalue w ith posilive im aginary part. Each eigenvectorw illbe scaled so the largest com ponent w ill have abs (realpart) + abs(im ag. part) \(=1\), *except* that for eigenvalues \(w\) ith alpha=beta=0, a zero vectorw illbe retumed as the comesponding eigenvector. N ot referenced if JOBVL = N '.

LDVL (input)
The leading dim ension of the \(m\) atrix \(\mathrm{V} \mathrm{L} . \mathrm{LD} \mathrm{VL}>=1\), and if \(\mathrm{OOBVL}=\mathrm{V}^{\prime}, \mathrm{LDVL}>=\mathrm{N}\) 。

\section*{VR (output)}

If \(\mathrm{OB} \mathrm{BR}=\mathrm{V}\) ', the right generalized eigenvectors. (See "Purpose", above.) Realeigenvectors take one colum n, com plex take two colum ns, the first for the real partand the second for the im aginary part. C om plex eigenvectors correspond to an eigenvalue \(w\) ith positive im aginary part. Each eigenvectorw illbe scaled so the largest com ponent w ill have abs(real part) + abs(im ag. part) \(=1\), *except* that for eigenvalues \(w\) ith alpha=beta=0, a zero vectorw illbe retumed as the corresponding eigenvector. N otreferenced if JOBVR = N'.

LDVR (input)
The leading dim ension of the \(m\) atrix \(V R\).LD \(V R>=1\), and if \(\mathrm{OBVR}=\mathrm{V}^{\prime}, \mathrm{LDVR}>=\mathrm{N}\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)

The dim ension of the amay W ORK. LDW ORK >= \(m\) ax \((1,8 \star N)\). For good perform ance, LDW ORK m ust generally be larger. To com pute the optim alvalue of LDW ORK, call HAENV to getblocksizes (for SGEQRF, SORMQR, and SORGQR .) Then com pute: NB -
\(M A X\) of the blocksizes for \(S G E Q R F\), \(S O R M Q R\), and SORGQR; The optim alLDW ORK is: \(2{ }^{*} \mathrm{~N}+\mathrm{MAX}\left(6^{*} \mathrm{~N}\right.\), \(N\) * (NB+1)).

IfLDW ORK \(=-1\), then a w ork.space query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LD W ORK is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the \(i-\) th argum ent had an illegalvalue.
\(=1, \ldots, N\) : The Q Z iteration failed. N o eigenvec-
tors have been calculated, but ALPHAR (1),
A LPHAI( \()\), and BETA ( 1 ) should be correct for
\(\dot{F} \mathbb{N}\) FO \(+1, \ldots, N .>N\) : errors thatusually indi-
cate LA PA CK problem s:
\(=\mathrm{N}+1\) : error retum from SGGBAL
\(=\mathrm{N}+2\) : error retum from SGEQRF
\(=N+3\) : error retum from \(S O R M Q R\)
\(=\mathrm{N}+4\) : error retum from \(S O R G Q R\)
\(=\mathrm{N}+5\) : error retum from SGGHRD
\(=\mathrm{N}+6\) : error retum from \(\mathrm{SH} G E Q Z\) (other than failed
iteration) \(=\mathrm{N}+7\) : error retum from STGEVC
\(=\mathrm{N}+8\) : enor retum from SGGBAK (com puting VL )
\(=\mathrm{N}+9\) : error retum from SGGBAK (com puting \(V R\) )
\(=\mathrm{N}+10\) : enor retum from SLA SC L (various calls)

\section*{FURTHER DETAILS}

Balancing

This driver calls SG G B A L to both perm ute and scale row s and colum ns of \(A\) and \(B\). The perm utations \(P L\) and \(P R\) are chosen so that \(\mathrm{PL}{ }^{*} \mathrm{~A} * P R\) and \(P L * B * R\) w illbe upper triangular except for the diagonal blocksA (i:ji:j) and B (i:ji:j) w ith i and jas close together as possible. The diagonal scaling \(m\) atrices DL and DR are chosen so that the pair \(D L * P L * A * P R * D R, D L * P L * B * P R * D R\) have elem ents close to one (except for the elem ents that start out zero.)

A fter the eigenvalues and eigenvectors of the balanced
\(m\) atrices have been com puted, SG G BA K transform s the eigenvectors back to what they w ould have been (in perfect arithm etic) ifthey had notbeen balanced.

Contents of \(A\) and \(B\) on Exit

If any eigenvectors are com puted (either \(J O B V L=V\) ' or JO BVR=V' or both), then on exit the arrays \(A\) and \(B\) w ill contain the realSchur form [ \({ }^{\star}\) ] of the "balanced" versions of A and B. If no eigenvectors are com puted, then only the diagonalblocks w illbe comect.
[*] See SH GEQ Z, SGEGS, or read the book "M atrix Com putations",
by G olub \& van Loan, pub. by Johns H opkins U .Press.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dgehrd - reduce a realgeneralm atrix A to upper H essenberg form \(H\) by an orthogonal sim ilarity transform ation

\section*{SYNOPSIS}


```

DOUBLE PRECISION A (LDA,*),TAU (*),W ORK NN (*)

```


```

DOUBLE PRECISION A (LDA,*),TAU (*),W ORK IN (*)

```

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E \operatorname{GEHRD}(\mathbb{N}], \mathbb{L O}, \mathbb{H} I, A,[L D A], T A U,[W O R K \mathbb{N}],[L W O R K \mathbb{N}]\), [ \(\mathbb{N}\) FO ])
\(\mathbb{N} T E G E R:: N, \mathbb{L} O, \mathbb{H} I, L D A, L W O R K \mathbb{N}, \mathbb{N} F O\)
REAL (8), D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK \(\mathbb{N}\)
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A
SU BROUTINE GEHRD_64 ( \(\mathbb{N}], \mathbb{I} O, \mathbb{H} I, A,[L D A], T A U,[W O R K \mathbb{N}],[L W O R K \mathbb{N}]\), [ \(\mathbb{N} F O\) ])
\(\mathbb{N}\) TEGER (8) :: N, \(\mathbb{N} O, \mathbb{H} I, L D A, L W O R K \mathbb{N}, \mathbb{N} F O\)
REAL (8), D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK \(\mathbb{N}\)
REAL (8),D IM ENSION (:,:) ::A
void dgehrd (intn, int ilo, int ini, double *a, int lda, double *tau, int *info);
void dgehrd_64 (long n, long 10 , long ihi, double *a, long lda, double *tau, long *info);

\section*{PURPOSE}
dgehrd reduces a realgeneralm atrix A to upper \(H\) essenberg form \(H\) by an orthogonalsim ilarity transform ation: Q '* A * \(\mathrm{Q}=\mathrm{H}\).

\section*{ARGUMENTS}

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
ㅍO (input)
It is assum ed that A is already upper triangular in row s and colum ns 1: \(\mathbb{T} 0-1\) and \(\mathbb{H}\) I+1:N. \(\mathbb{H} 0\) and IH I are norm ally setby a previous call to SG EBA L; otherw ise they should be set to 1 and \(N\) respectively. See FurtherD etails.

IH I (input)
See the description of IIO .
A (input/output)
O \(n\) entry, the N -by -N generalm atrix to be reduced.
O n exit, the upper triangle and the firstsubdiagonalofA are overw ritten \(w\) ith the upper H essenberg \(m\) atrix \(H\), and the elem ents below the first subdiagonal, w ith the array TAU, represent the orthogonal \(m\) atrix \(Q\) as a productof elem entary reflectors. See FurtherD etails.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, \mathbb{N})\).

TAU (output)
The scalar factors of the elem entary reflectors (see Further Details). Elem ents 1:THO-1 and IH IN -1 of TA U are setto zero.

W ORK \(\mathbb{N}\) (w orkspace)

On exit, if \(\mathbb{N F O}=0, W O R K \mathbb{N}\) (1) retums the optim allW ORK \(\mathbb{N}\).

LW ORK \(\mathbb{N}\) (input)
The length of the array \(W O R K \mathbb{N}\). LW ORK \(\mathbb{N}>=\) \(\max (1, \mathbb{N})\). For optim um perform ance LW ORK \(\mathbb{N}>=\) \(\mathrm{N} * \mathrm{NB}\), where NB is the optim alblocksize.

If LW ORK \(\mathbb{N}=-1\), then a workspace query is assum ed; the routine only calculates the optim al size of the \(W\) ORK \(\mathbb{N}\) array, retums this value as the firstentry of the \(W\) ORK \(\mathbb{N}\) array, and no error \(m\) essage related to LW ORK \(\mathbb{N}\) is issued by XERBLA.
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the i-th argum enthad an illegalvahue.

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a productof (hi-ib) ele\(m\) entary reflectors
\[
Q=H \text { ( } 0 \text { ) H (ib+1) . . H (ihi-1) . }
\]

Each H (i) has the form
\[
H(i)=I-\tan * V^{*} V^{\prime}
\]
where tau is a real scalar, and \(v\) is a realvectorw ith \(\mathrm{v}(1: i)=0, v(i+1)=1\) and \(v\) (init 1 n\()=0\); \(\mathrm{v}(\mathrm{i}+2\) : ihi) is stored on exitin A (i+2:ihi,i), and tau in TA U (i).

The contents of A are illustrated by the follow ing exam ple, w ith \(\mathrm{n}=7\), il = \(=2\) and ihi= 6:
```

on entry, on exit,

```
(a a a a a a a) (a a h h h h a) ( a a a a a a) ( a h h h ha) ( a a a a a a) ( \(h\) h h h h h ) ( a a a a a a) ( v2 h \(h \mathrm{~h} h \mathrm{~h})(\mathrm{a}\) a a a a a) ( v2 v3 h h h h ) ( a a a a a a) ( v2 v3 v4 h h h) (
a) (

> a)
\(w\) here a denotes an elem ent of the original \(m\) atrix \(A, h\) denotes a \(m\) odified elem ent of the upper \(H\) essenberg \(m\) atrix \(H\), and videnotes an elem ent of the vector defining \(H\) (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dgelgf-com pute an LQ factorization of a realM -by -N m atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGELQF(M,N,A,LDA,TAU,W ORK,LDW ORK,INFO)}
\mathbb{NTEGER M,N,LDA,LDW ORK,\mathbb{NFO}}\mathbf{N}\mathrm{ , L}
DOUBLE PRECISION A (LDA,*),TAU(*),W ORK (*)
SU BROUT\mathbb{NE DGELQF_64M,N,A,LDA,TAU,W ORK,LDW ORK, INFO )}
\mathbb{NTEGER*8M,N,LDA,LDW ORK, NNFO}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GELQF (M ], \(\mathbb{N}], A,[L D A], T A U, \mathbb{W} O R K],[L D W O R K],[\mathbb{N F O}])\)
\(\mathbb{N} T E G E R:: M, N, L D A, L D W\) ORK, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A
SU BROUTINE GELQF_64 ( \(\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{N} O R K],[L D W O R K],[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER (8) ::M ,N,LDA,LDW ORK, \(\mathbb{N}\) FO
REAL (8), D IM ENSION (:) ::TAU,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include < sunperfh>
void dgelqf(intm, intn, double *a, intlda, double *tau, int*info);
void dgelqf_64 (long m, long n, double *a, long lda, double *tau, long *info);

\section*{PURPOSE}
dgelqf com putes an \(L Q\) factorization of a realM -by N matrix \(A: A=L\) * \(Q\).

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the M -by -N m atrix A. On exit, the ele\(m\) ents on and below the diagonalof the array contain the \(m\)-by \(m\) in \((m, n)\) low er trapezoidalm atrix \(L\) ( \(L\) is lower triangular ifm <= n); the elem ents above the diagonal, with the array TAU, represent the orthogonalm atrix \(Q\) as a product of elem entary reflectors (see FurtherD etails).

LDA (input)
The leading dim ension of the array A. LDA >= max (1, M).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the amay \(W\) ORK. LDW ORK >= \(m a x(1, M)\). Foroptim um perform ance LDW ORK \(>=M * N B\), where NB is the optim alblocksize.

IfLDW ORK = -1 , then aw orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LDW ORK is issued by XERBLA.
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\(Q=H(k) \ldots H(2) H(1)\), where \(k=m\) in \((m, n)\).
Each \(H\) (i) has the form
H (i) \(=I-\tan * V^{*} V^{\prime}\)
w here tau is a real scalar, and v is a realvectorw ith \(v(1: i-1)=0\) and \(v(i)=1\); \(v(i+1 n)\) is stored on exit in A ( \(i, i+1 \mathrm{n})\), and tau in TAU (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgels -solve overdeterm ined or underdeterm ined real linear system \(s\) involving an \(M\) by -N matrix \(A\), or its transpose, using a QR orlQ factorization ofA

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DGELS (TRANSA,M ,N,NRHS,A,LDA,B,LDB,W ORK,LDW ORK,}
\mathbb{NFO)}
CHARACTER * 1 TRANSA
\mathbb{NTEGERM,N,NRHS,LDA,LDB,LDW ORK,INFO}
DOUBLE PRECISION A (LDA,*),B (LDB,*),W ORK (*)
SU BROUT\mathbb{NE DGELS_64 (TRANSA,M ,N,NRHS,A,LDA,B,LDB,W ORK,LDW ORK,}
INFO)

```
CHARACTER * 1 TRANSA
\(\mathbb{N} T E G E R * 8 M, N, N R H S, L D A, L D B, L D W O R K, \mathbb{N} F O\)
DOUBLE PRECISION A (LDA, \(\left.{ }^{\star}\right)\), B (LDB,\(\left.{ }^{\star}\right)\), W ORK ( \({ }^{*}\) )

\section*{F95 INTERFACE}

SU BROUTINE GELS ([TRANSA], \(\mathbb{M}], \mathbb{N}], \mathbb{N R H S}], A,[L D A], B,[L D B],[W\) ORK ], LDW ORK, [ \(\mathbb{N} F \mathrm{~F}\) ])

CHARACTER ( \(4 E N=1\) ) ::TRANSA
\(\mathbb{N}\) TEGER ::M,N,NRHS,LDA,LDB,LDW ORK, \(\mathbb{N}\) FO
REAL (8),D IM ENSION (:) ::W ORK
REAL (8), D IM ENSION (:,:) ::A,B

SU BROU T \(\mathbb{N} E\) GELS_64 ([TRANSA], \(\mathbb{M}], \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B]\), [ \(\mathbb{W}\) ORK ],LDW ORK, [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N}\) TEGER (8) ::M , N,NRHS,LDA,LDB,LDW ORK, \(\mathbb{N}\) FO
REAL (8),D IM ENSION (:) ::W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void dgels (char, int, int, int, double*, int, double*, int, int*);
void dgels_64 (char, long, long, long, double*, long, double^, long, long*);

\section*{PURPOSE}
dgels solves overdeterm ined or underdeterm ined real linear system \(s\) involving an \(M\) boy -N matrix \(A\), or its transpose, using a Q R orLQ factorization ofA. It is assum ed that A has full rank.

The follow ing options are provided:
1. IfTRAN \(S=N\) 'and \(m>=n\) : find the least squares solution of
an overdeterm ined system , i.e., solve the least squares problem
\[
m \text { inim ize }\|\mathrm{B}-\mathrm{A} * \mathrm{X}\| .
\]
2. IfTRAN \(S=N\) 'and \(m<n\) : find the \(m\) inim um norm solution of an underdeterm ined system \(A * X=B\).
3. IfTRANS = T'and \(m>=n\) : find them inim um norm solution of
an undeterm ined system \(A * * T * X=B\).
4. IfTRANS = \(T\) 'and \(m<n\) : find the least squares solution of
an overdeterm ined system , ie., solve the least squares problem
\[
\mathrm{m} \text { inim ize }\|\mathrm{B}-\mathrm{A} * * \mathrm{~T} * \mathrm{X}\| .
\]

Several right hand side vectors \(b\) and solution vectors \(x\) can be handled in a single call; they are stored as the colum ns of the M -by-NRHS righthand side m atrix B and the \(N\)-by-NRHS solution \(m\) atrix \(X\).

\section*{ARGUMENTS}

TRANSA (input)
\(=N^{\prime}\) : the linearsystem involves A;
\(=T\) ': the linear system involves \(A * * T\).

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

M (input) The num ber of row s of the matrix A. M >=0.

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the m atrioes B and X .NRH S \(>=0\).
A (input/output)
On entry, the M -by -N m atrix A. On exit, if \(\mathrm{M}>=\) \(\mathrm{N}, \mathrm{A}\) is overw ritten by details of its Q R factorization as retumed by SGEQRF; if \(M<N, A\) is overw rilten by details of its LQ factorization as retumed by SGELQF.

LDA (input)
The leading dim ension of the aray A. LDA >= \(\max (1, M)\).

B (input/output)
O n entry, the \(m\) atrix \(B\) of righthand side vectors,
stored colum nw ise; B is M byy-NRHS ifTRANSA = N',
orN boy-NRHS ifTRANSA = T'. On exit, B is
overw rilten by the solution vectors, stored colum nw ise: ifTRAN SA \(=\mathrm{N}\) 'and \(\mathrm{m}>=\mathrm{n}\), row s 1 to \(n\) ofB contain the least squares solution vectors; the residual.sum ofsquares for the solution in each colum \(n\) is given by the sum of squares of ele\(m\) ents \(N+1\) to \(M\) in thatcolumn; ifTRANSA \(=N\) 'and \(m<n\), row \(s 1\) to \(N\) ofB contain them inim um norm solution vectors; ifTRANSA \(=T\) 'and \(m>=n\), row \(s\) 1 to M ofB contain them inim um norm solution vectors; ifTRANSA \(=T\) 'and \(m<n\), row 1 to \(M\) of \(B\) contain the least squares solution vectors; the residual.sum of squares forthe solution in each colum \(n\) is given by the sum of squares of elem ents \(M+1\) to \(N\) in that colum \(n\).

LD B (input)
The leading dim ension of the aray \(\mathrm{B} . \mathrm{LD} \mathrm{B}>=\) M AX ( \(1, \mathrm{M}, N\) ) 。

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (output)
The dim ension of the amay W ORK. LDW ORK >= \(\max (\)
1, \(\mathrm{M} N+\max (\mathrm{M} N, \mathrm{NRHS})\) ). Foroptim alperfor
\(m\) ance, LDW ORK \(>=\max (1, M N+\max (M N, N R H S) * N B\)
). where \(M N=m\) in \(M N\) ) and \(N B\) is the optim um
block size.

IfLD W ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
theW ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LD W ORK is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an ille-
galvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dgelsd -com pute the m inim um -norm solution to a real linear least squares problem

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGELSD M,N,NRHS,A,LDA,B,LDB,S,RCOND,RANK,W ORK,}
LW ORK,IN ORK,\mathbb{NFO)}
\mathbb{NTEGERM,N,NRHS,LDA,LDB,RANK,LW ORK, INFO}
INTEGER IN ORK (*)
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA,*),B (LDB ,*),S (*),W ORK (*)
SUBROUT\mathbb{NEDGELSD_64M,N,NRHS,A,LDA,B,LDB,S,RCOND,RANK,}
W ORK,LW ORK,\mathbb{IN ORK,INFO)}
\mathbb{NTEGER*8M,N,NRHS,LDA,LDB,RANK,LW ORK,NNFO}
INTEGER*8 IN ORK (*)
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA,*),B (LDB,*),S (*),W ORK (*)

```

\section*{F95 INTERFACE}

SUBROUTINE GELSD (M) \(\mathbb{M}], \mathbb{N} R H S], A,[L D A], B,[L D B], S, R C O N D\), RANK, [W ORK ], [LW ORK ], [ \(\mathbb{W}\) ORK ], [ \(\mathbb{N} F \mathrm{O}\) ])
\(\mathbb{N}\) TEGER :: M , N,NRHS,LDA,LDB,RANK,LW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL (8) ::RCOND
REAL (8), D IM ENSION (:) ::S,W ORK
REAL (8), D IM ENSION (:,:) ::A,B

SUBROUTINE GELSD_64 ( \(\mathbb{M}], \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], S, R C O N D\), RANK, [W ORK ], [LW ORK ], [IW ORK ], [ \(\mathbb{N} F O\) ])
\(\mathbb{N} T E G E R(8):: M, N, N R H S, L D A, L D B, R A N K, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I W}\) ORK
REAL (8) :: RCOND
REAL (8),D IM ENSION (:) ::S,W ORK
REAL (8), D IM ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void dgelsd (intm , intn, intnrhs, double *a, int lda, double *b, int ldb, double *s, double rcond, int *rank, int*info);
void dgelsd_64 (long m, long n, long nrhs, double *a, long lda, double *b, long ldb, double *s, double roond, long *rank, long *info);

\section*{PURPOSE}
dgelsd com putes the \(m\) inim um -norm solution to a real linear least squares problem :
\(m\) inim ize 2 -nom (|b-A *x )
using the singularvalue decom position (SVD ) ofA.A is an M -by-N m atrix which \(m\) ay be rank-deficient.

Several righthand side vectors b and solution vectors \(x\) can be handled in a single call; they are stored as the colum ns of the M -by-NRH S righthand sidem atrix \(B\) and the \(N\) by-NRH S solution \(m\) atrix \(X\).

The problem is solved in three steps:
(1) Reduce the coefficientm atrix A to bidiagonal form w th H ouseholder transform ations, reducing the original problem into a "bidiagonal least squares problem " (BLS)
(2) Solve the BLS using a divide and conquer approach.
(3) A pply back all the \(H\) ouseholder tranform ations to solve the original least squares problem .

The effective rank of A is determ ined by treating as zero those singular values which are less than RCOND tim es the largestsingularvalue.

The divide and conquer algorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on \(m\) achines w ith a guard digit in add/subtract, or on those binary \(m\) achines \(w\) ithout guard digits w hich subtract like the \(C\) ray

X - M P , C ray Y M P , C ray C -90, orC ray-2. It could conceivably fail on hexadecim al or decim al machines \(w\) thout guard digits, butw e know of none.

\section*{ARGUMENTS}

M (input) The num ber of row sofA. \(\mathrm{M}>=0\).
N (input) The num ber of colum nsofA . \(\mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the \(m\) atrices \(B\) and X.NRHS \(>=0\).
A (input/output)
On entry, the M -by -N m atrixA. On exit, A has been destroyed.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, M)\).

B (input/output)
On entry, the M -by \(-N\) RH \(S\) righthand side \(m\) atrix \(B\).
On exit, B is overw ritten by the \(N\) by-NRHS solution \(m\) atrix \(X\). Ifm \(>=n\) and RANK \(=n\), the residual sum-of-squares for the solution in the \(i\)-th colum \(n\) is given by the sum of squares of elem ents \(\mathrm{n}+1 \mathrm{~m}\) in thatcolum n .

LD B (input)
The leading dim ension of the array \(\mathrm{B} . \mathrm{LD} \mathrm{B}>=\) \(\max (1, m a x(M, N))\).

S (output)
The singularvalues of A in decreasing order. The condition number of \(A\) in the 2 -norm \(=\) \(S(1) / S(m\) in \((m, n))\).

RCOND (input)
RCOND is used to determ ine the effective rank of A. Singularvalues \(S\) (i) <= RCOND *S (1) are treated as zero. IfRCOND \(<0, \mathrm{~m}\) achine precision is used instead.

RANK (output)
The effective rank of A, i.e., the num ber of singular values w hich are greater than RCO N D *S (1).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK.LW ORK \(>=1\). The exact \(m\) inim um am ount of w orkspace needed depends on M , N and NRHS. As long as LW ORK is at least \(12 * \mathrm{~N}+2 * \mathrm{~N} * \mathrm{SM} \mathrm{LSIZ}+8 * \mathrm{~N} * \mathrm{NLVL}+\mathrm{N} * \mathrm{NRHS}\) * (SMLSLZ +1 ) \({ }^{* *} 2\), ifM is greater than orequal to \(N\) or \(12 * \mathrm{M}+2 * \mathrm{M} * \mathrm{SMLSIZ}+8 * \mathrm{M} * \mathrm{NLVL}+\mathrm{M}\) *NRHS + (SM LSIZ + 1) **2, ifM is less than N , the code will execute correctly. SM LSIZ is retumed by ILA ENV and is equal to the \(m\) axim um size of the subproblems at the bottom of the com putation tree (usually about 25) , and \(\mathrm{NLVL}=\mathbb{N} T\left(\mathrm{LOG} \_2\right.\) ( \(\mathrm{M} \mathbb{N}(\mathrm{M}, \mathbb{N}\) )/(SM LSIZ+1) ) ) + 1 Forgood penform ance, LW ORK should generally be larger.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IW ORK (w orkspace)

\(M \mathbb{N} M N=M \mathbb{N}(M, N)\) 。

IN FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an illegalvahue.
\(>0\) : the algorithm forcom puting the SVD failed to converge; if \(\mathbb{N} F O=\) i, ioff-diagonalelem ents of an interm ediate bidiagonalform did not converge to zero.

\section*{FURTHER DETAILS}

B ased on contributions by
M ing Gu and Ren-C ang Li, Computer Science D ivision, U niversity of C alifomia atB erkeley, U SA

O sniM arques, LBNLNERSC , U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgelss - com pute them inim um norm solution to a real linear least squares problem

\section*{SYNOPSIS}

```

    W ORK,LDW ORK,INFO)
    INTEGERM,N,NRHS,LDA,LDB, \mathbb{RANK,LDW ORK, INFO}
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA,*),B (LDB,*),SING (*),W ORK (*)
SU BROUT\mathbb{NE DGELSS_64M,N,NRHS,A,LDA,B,LDB,SING,RCOND, \mathbb{RANK,}}\mathbf{N},\textrm{L}
W ORK,LDW ORK,\mathbb{NFO)}
\mathbb{NTEGER*8M,N,NRHS,LDA,LDB, \mathbb{RANK,LDW ORK,NNFO}}\mathbf{N},\mp@code{LN}
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA,*),B (LDB,*),SING (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GELSS (M) \(\mathbb{M}], \mathbb{N} R H S], A,[L D A], B,[L D B], S \mathbb{N} G, R C O N D\), \(\mathbb{R} A N K, \mathbb{W}\) ORK], [LDW ORK], [ \(\mathbb{N F O}])\)
\(\mathbb{N} T E G E R:: M, N, N R H S, L D A, L D B, \mathbb{R A N K}, L D W O R K, \mathbb{N} F O\)
REAL (8) :: RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::SING,W ORK
REAL (8),D IM ENSION (:,:) ::A ,B
SU BROUTINE GELSS_64 (M) \(\mathbb{M}], \mathbb{N} R H S], A,[L D A], B,[L D B], S \mathbb{N} G\),
RCOND, \(\mathbb{R A N K , [ W O R K ] , [ L D W ~ O R K ] , [ \mathbb { N F O } ] ) ~}\)
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{NRHS}, \mathrm{LD} A, L D B, \mathbb{R} A N K, L D W O R K, \mathbb{N} F O\)

REAL (8) ::RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::SNG,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include < sunperfh>
void dgelss(intm, intn, intnris, double *a, int lda, double *b, int ldb, double *sing, double rcond, int *irank, int*info);
void dgelss_64 (long m, long n, long nihs, double *a, long lda, double *b, long ldb, double *sing, double rcond, long *irank, long *info);

\section*{PURPOSE}
dgelss com putes the \(m\) inim um norm solution to a real linear least squares problem :

M inin ize 2 -norm ( \(\mid \mathrm{b}-\mathrm{A}\) * \(\mathrm{x} \mid\).
using the singularvalue decom position (SVD) ofA.A is an M -by-N m atrix which \(m\) ay be rank-deficient.

Several righthand side vectors \(b\) and solution vectors x can be handled in a single call; they are stored as the colum ns of the M -by-NRH S righthand side m atrix B and the N -by-NRH S solution \(m\) atrix \(X\).

The effective rank ofA is determ ined by treating as zero those singular values which are less than RCOND tim es the largest singularvalue.

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(A . M>=0\).
\(N\) (input) The num ber of colum ns of the m atrix \(A \cdot N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the \(m\) atrices \(B\) and \(X . N R H S>=0\).

A (input/output)
On entry, the \(M\) by -N m atrix A. On exit, the first
\(m\) in \((m, n)\) row sofA are overw ritten \(w\) ith its right
singularvectors, stored row w ise.
LD A (input)
The leading dim ension of the array A. LDA >= \(\mathrm{max}(1, \mathrm{M})\).

B (input/output)
On entry, the M -by-NRHS righthand sidem atrix B . On exit, B is overw rilten by the \(N\)-by NRHS solution \(m\) atrix \(X\). Ifm \(>=n\) and \(\mathbb{R} A N K=n\), the residual sum-of-squares for the solution in the \(i\)-th colum \(n\) is given by the sum of squares of elem ents \(\mathrm{n}+1 \mathrm{~m}\) in that colum n .
LD B (input)
The leading dim ension of the array \(B . L D B>=\) \(\max (1, m a x M, N)\) ).

SING (output)
The singularvalues of A in decreasing order. The condition number of \(A\) in the 2 -norm \(=\) \(S \mathbb{N} G(1) / S \mathbb{N} G(m\) in \((m, n))\).

RCOND (input)
RCOND is used to determ ine the effective rank of
A. Singular values \(S \mathbb{N} G\) (i) \(<=\) RCOND*SNI (1) are treated as zero. If RCOND \(<0, \mathrm{~m}\) achine precision is used instead.

RANK (output)
The effective rank of A, i.e., the num ber of singular values which are greater than RCOND*SING (1).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dim ension of the array \(W\) ORK.LDW ORK >=1, and also: LDW ORK \(>=3 \star_{m}\) in \(\left.M, N\right)+m a x(2 \star m\) in \(M, N)\), max (M,N),NRHS ) For good perform ance, LDW ORK should generally be larger.

IfLDW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
= 0: successfulexit
< 0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum enthad an illegalvalue.
>0: the algorithm forcom puting the SVD failed to converge; if \(\mathbb{N} F O=\) i, ioff-diagonalelem ents of an interm ediate bidiagonal form did not converge to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgelsx -routine is deprecated and has been replaced by routine SG ELSY

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGELSX M,N,NRHS,A,LDA,B,LDB,UPIVOT,RCOND,\mathbb{RANK,}}\mathbf{N},\textrm{L}
WORK,\mathbb{NFO)}
\mathbb{NTEGERM,N,NRHS,LDA,LDB,\mathbb{RANK,INFO}}\mathbf{N},\mp@code{N}
INTEGER JPIVOT(*)
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA,*),B (LDB,*),W ORK (*)
SUBROUT\mathbb{NEDGELSX_64M,N,NRHS,A,LDA,B,LDB,JPIVOT,RCOND,}
\mathbb{RANK,W ORK,INFO)}
\mathbb{NTEGER*8M,N,NRHS,LDA,LDB, RRANK,INFO}
INTEGER*8 \mathbb{PIVOT (*)}
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA,*),B (LDB,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E \operatorname{GELSX}(\mathbb{M}], \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], J P I V O T, R C O N D\), \(\mathbb{R A N K},[\mathbb{W}\) ORK], \([\mathbb{N} F O]\) )
\(\mathbb{N}\) TEGER :: M , N,NRHS,LDA,LDB, \(\mathbb{R} A N K, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{I}\) ENSION (:) :: JPIV OT
REAL (8) ::RCOND
REAL (8), D IM ENSION (:) ::W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B
SU BROUTINE GELSX_64 (M ], \(\mathbb{N}], \mathbb{N} R H S], A,[\operatorname{LD} A], B,[L D B], \mathbb{P} I V O T\),
\(\mathbb{N} T E G E R(8):: M, N, N R H S, L D A, L D B, \mathbb{R A N K}, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V O T}\)
REAL (8) :: RCOND
REAL (8), D \(\mathbb{I M}\) ENSION (:) ::W ORK
REAL (8), D \(\mathbb{M}\) ENSION (: : : : : A , B

\section*{C INTERFACE}
\#include <sunperfh>
void dgelsx (intm , intn, intnrhs, double *a, int lda, dou-
ble *b, int lalo, int * jpivot, double rcond, int
*irank, int *info);
void dgelsx_64 (long m, long n, long nrhs, double *a, long
lda, double *b, long ldb, long *jívot, double
rcond, long *irank, long *info);

\section*{PURPOSE}
dgelsx routine is deprecated and has been replaced by routine SGELSY .

SG ELSX com putes them inim um norm solution to a real linear
least squares problem :
\(m\) inim ize \(\|A * X-B\|\)
using a com plete orthogonal factorization of A. A is an \(M-\) by-N m atrix w hich \(m\) ay be rank-deficient.

Several right hand side vectors b and solution vectors x can be handled in a single call; they are stored as the colum ns of the M boy-N RHS righthand sidem atrix B and the N boy-N R H S solution \(m\) atrix X .

The routine first com putes a Q R factorization with colum \(n\) pivoting:
\[
A * P=Q *[R 11 R 12]
\]
[ 0 R22]
w ith R11 defined as the largest leading subm atrix whose estim ated condition num ber is less than \(1 \notin C O N D\). The order ofR 11, RANK, is the effective rank ofA.

Then, R 22 is considered to be negligible, and R 12 is annihilated by orthogonal transform ations from the right, arriving at the com plete orthogonal factorization:
\(A * P=Q *\left[\begin{array}{lll}11 & 0\end{array}\right] * Z\)
\(\left[\begin{array}{ll}0 & 0\end{array}\right]\)
Them inim um norm solution is then
\(\mathrm{X}=\mathrm{P} * \mathrm{Z}^{\prime}\left[\operatorname{inv}(\mathrm{T} 11) \star \mathrm{Q} 1^{*} \mathrm{~B}\right]\)
where Q 1 consists of the firstRA K colum ns of Q .

\section*{ARGUMENTS}

M (input) The num ber of row s of the matrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of collm ns of the m atrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns ofm atrices \(B\) and \(X\).NRH \(S>=0\). A (input/output)

On entry, the M -by -N m atrix A. On exit, A has been overw rilten by details of its complete orthogonal factorization.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, M)\).

B (input/output)
On entry, the M by -N RH S righthand side \(m\) atrix \(B\).
On exit, the N -by-N RH S solution \(m\) atrix X . Ifm >= \(n\) and \(\mathbb{R A N K}=\mathrm{n}\), the residual sum -of-squares for the solution in the \(i\)-th colum \(n\) is given by the sum of squares of elem ents \(N+1 \mathrm{M}\) in that colum \(n\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, M, N)\).

JPIVOT (input/output)
On entry, if \(\mathbb{P} \mathbb{I V O T}\) (i) ne.0, the i-th colum n of \(A\) is an initial colum \(n\), otherw ise it is a free colum \(n\). Before the \(Q R\) factorization of \(A\), all initial colum ns are perm uted to the leading positions; only the rem aining free colum ns are m oved as a result of colum \(n\) pivoting during the factorization. On exit, if JPIV OT \((i)=k\), then the \(i\)-th colum \(n\) of A *P was the \(k\)-th collm \(n\) of A.

RCOND (input)
RCOND is used to determ ine the effective rank of A, which is defined as the order of the largest leading triangular subm atrix R 11 in the \(Q R\) factorization w ith pivoting ofA , whose estim ated condition num ber \(<1\) RCOND.

IRANK (output)
The effective rank ofA, ie., the order of the subm atrix R11. This is the sam e as the order of the subm atrix T11 in the com plete orthogonal factorization of A.

W ORK (w orkspace)
\((m\) ax \((m\) in \(M, N)+3 * N, 2 * m\) in \((M, N)+N R H S))\), INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dgelsy -com pute the m inim um -norm solution to a real linear
least squares problem

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DGELSY M,N,NRHS,A,LDA,B,LDB,JPVT,RCOND,RANK,}
W ORK,LW ORK,\mathbb{NFO)}
\mathbb{NTEGERM,N,NRHS,LDA,LDB,RANK,LW ORK, INFO}
INTEGER JPVT (*)
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA,*),B (LDB,*),W ORK (*)
SUBROUTINEDGELSY_64M,N,NRHS,A,LDA,B,LDB,JPVT,RCOND,RANK,
W ORK,LW ORK,INFO)
\mathbb{NTEGER*8M,N,NRHS,LDA,LDB,RANK,LW ORK,NNFO}
INTEGER*8 JPVT (*)
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA,*),B (LDB,*),W ORK (*)

```

\section*{F95 INTERFACE}

SUBROUTINE GELSY (M ], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], J P V T, R C O N D\), RANK, [W ORK], [LW ORK], [ \(\mathbb{N F O}\) ])
\(\mathbb{N}\) TEGER :: M , N,NRHS,LDA,LDB,RANK,LW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: J P V T\)
REAL (8) ::RCOND
REAL (8), D IM ENSION (:) ::W ORK
REAL (8), D IM ENSION (:,:) ::A , B

SUBROUTINE GELSY_64 (M) \(\mathbb{M}], \mathbb{N} R H S], A,[L D A], B,[L D B], \mathbb{P V} T\), RCOND, RANK, [W ORK], [LW ORK ], [ \(\mathbb{N F O}]\) )
\(\mathbb{N} T E G E R(8):: M, N, N R H S, L D A, L D B, R A N K, L W O R K, \mathbb{N F O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P V} T\)
REAL (8) :: RCOND
REAL (8),D IM ENSION (:) ::W ORK
REAL (8), D IM ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void dgelsy (intm , intn, intnrhs, double *a, int lda, dou-
ble *b, int ldb, int * jpvt, double roond, int
*rank, int*info);
void dgelsy_64 (long m, long n, long nihs, double *a, long lda, double *b, long ldb, long * pivt, double rcond, long *rank, long *info);

\section*{PURPOSE}
dgelsy com putes the \(m\) in'm um -norm solution to a real linear least squares problem :
\(m\) inim ize \(\|A * X-B\|\)
using a com plete orthogonal factorization ofA. A is an M -by-N \(m\) atrix which \(m\) ay be rank-deficient.

Several righthand side vectors b and solution vectors \(x\) can be handled in a single call; they are stored as the colum ns of the M by-NRHS righthand side m atrix \(B\) and the \(N\) by-NRHS solution \(m\) atrix \(X\).

The routine firstcom putes \(a Q R\) factorization \(w\) th \(c o l u m n\) pivoting:
\(A * P=Q *[R 11 R 12]\)
[ 0 R22]
w ith R 11 defined as the largest leading subm atrix whose estim ated condition num ber is less than 1RCOND. The order ofR11,RANK, is the effective rank ofA.

Then, R 22 is considered to be negligible, and R 12 is anninilated by orthogonal transform ations from the right, aniving at the com plete orthogonal factorization:
\[
\begin{gathered}
A * P=Q *[T 110] * Z \\
{\left[\begin{array}{ll}
0 & 0
\end{array}\right]}
\end{gathered}
\]

Them inim um norm solution is then
\(\mathrm{X}=\mathrm{P} * \mathrm{Z}^{\prime}[\operatorname{inv}(\mathrm{T} 11) * \mathrm{Q} 1\) *B]
[ 0 ]
where Q 1 consists of the firstRANK colum ns of Q .

This routine is basically identical to the original xG ELSX except three differences:
o The call to the subroutine xGEQPF has been substituted by the
the call to the subroutine xG EQP3.This subroutine is a B las-3
version of the \(Q R\) factorization \(w\) ith colum \(n\) pivoting.
0 M atrix B (the righthand side) is updated w ith B las -3 . - The perm utation ofm atrix B (the right hand side) is fasterand m ore sim ple.

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of collm ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns ofm atrices B and X.NRHS \(>=0\).

A (input/output)
On entry, the M -by -N m atrix A. On exit, A has been overw rilten by details of its com plete orthogonal factorization.

LD A (input)
The leading dim ension of the array A. LDA >= max (1,M).

B (input/output)
On entry, the M \(-b y-N\) RH S righthand side m atrix B . On exit, the N -by-N RH S solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B. LD B >= max (1, M N \()\).

JPVT (input/output)
On entry, if JPV T (i) ne. 0, the i-th collm n of A is perm uted to the frontof \(P\), otherw ise colum \(n\) i is a free colum n. On exit, if JPV T (i) = k, then the \(i\)-th colum \(n\) of AP \(w\) as the \(k\)-th colum \(n\) of \(A\).

RCOND (input)
RCOND is used to determ ine the effective rank of A, which is defined as the order of the largest
leading triangularsubm atrix R 11 in the Q R factorization w ith pivoting ofA , w hose estim ated condition num ber<1/RCOND.

RANK (output)
The effective rank ofA, ie., the order of the subm atrix R11. This is the sam e as the order of the subm atrix T11 in the com plete orthogonal factorization of A.

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, \mathrm{~W}\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the aray W ORK. The unblocked strategy requires that: LW ORK \(>=\mathrm{MAX}(\mathrm{MN}+3 * \mathrm{~N}+1\), \(2 * M N+N R H S)\), where \(M N=m\) in ( \(M, N\) ). The block algorithm requires that: LWORK >= MAX ( \(\mathrm{M} N+2 * N+N B *(\mathbb{N}+1), 2 * \mathrm{MN}+\mathrm{NB} * \mathrm{NRHS}\) ), where NB is an upper bound on the blocksize retumed by HAENV for the routines SGEQP3, STZRZF, STZRQF, SORMQR, and SORM RZ .

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK aray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) If \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue.

\section*{FURTHER DETAILS}
```

B ased on contributions by
A .Petitet, C om puterS cience D ept., U niv . ofTenn ., K nox-
ville, U SA
E .Q uintana-O nti, D epto.de Inform atica, U niversidad Jaim e
I, Spain
G . Q uintana-O rti, D epto. de Inform atica, U niversidad Jaim e
I, Spain

```

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgem \(m\)-perform one of the \(m\) atrix-m atrix operations \(C:=\) alpha*op (A ) *op (B) + beta*C

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGEMM (TRANSA,TRANSB,M,N,K,ALPHA,A,LDA,B,LDB,}
BETA,C,LDC)
CHARACTER * 1 TRANSA,TRANSB
INTEGERM,N,K,LDA,LDB,LDC
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION A (LDA,*),B (LDB,*),C (LDC ,*)
SU BROUT INE DGEM M _64 (TRANSA,TRANSB ,M ,N ,K,ALPHA,A ,LDA,B,LD B,
BETA,C,LDC)

```
CHARACTER * 1 TRANSA, TRANSB
\(\mathbb{N}\) TEGER*8M,N,K,LDA,LDB,LDC
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION A (LDA, \(\left.{ }^{\star}\right), \mathrm{B}(\mathrm{LD} B, \star), \mathrm{C}(\mathrm{LDC}, \star)\)

\section*{F95 INTERFACE}

SU BROUTINE GEM M ([TRANSA], [TRANSB], \(\mathbb{M}], \mathbb{N}],[K], A L P H A, A,[L D A]\), B, [LD B],BETA, C , [LD C ])

CHARACTER (LEN=1) ::TRANSA,TRANSB
\(\mathbb{N}\) TEGER :: M , N, K, LDA, LD B, LD C
REAL (8) ::ALPHA,BETA
REAL (8),D IM ENSION (: : : : : A , B, C
SU BROUTINE GEMM _64 ([TRANSA], [TRANSB], \(\mathbb{M}], \mathbb{N}],[\mathbb{K}], A L P H A, A,[L D A]\), B, [LDB],BETA, C, [LDC])

CHARACTER (LEN=1) ::TRANSA, TRAN SB
\(\mathbb{N}\) TEGER (8) :: M , N , K , LDA , LD B , LD C
REAL (8) ::ALPHA,BETA
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A, B, C

\section*{C INTERFACE}
\#include <sunperfh>
void dgem \(m\) (chartransa, chartranso, intm, int \(n\), int \(k\), double alpha, double *a, intlda, double *b, int ldb, double beta, double * c, int ldc);
void dgem m _64 (chartransa, chartransb, long m, long n, long k , double alpha, double *a, long lda, double *b, long lalb, double beta, double * c , long ldc);

\section*{PURPOSE}
dgem \(m\) perform s one of the \(m\) atrix \(-m\) atrix operations \(C:=\) alpha*op (A )*op (B) + beta*C where op (X ) is one of
\[
o p(X)=X \quad \text { or } o p(X)=X^{\prime},
\]
alpha and beta are scalars, and \(A, B\) and \(C\) are \(m\) atrices, with op (A ) an m by kmatrix, op (B) a k by \(n m a t r i x\) and \(C\) an \(m\) by \(n m\) atrix.

\section*{ARGUMENTS}

TRANSA (input)
O n entry, TRANSA specifies the form of op (A ) to be used in the \(m\) atrix \(m\) ultiplication as follow \(s\) :

TRANSA \(=N\) 'or \(h^{\prime}, ~ o p(A)=A\).

TRANSA = T'ort', op (A ) = A'.

TRANSA = C 'or \(C^{\prime}\), op (A) \(=\) A '.

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

TRANSB (input)
O n entry, TRAN SB specifies the form ofop (B) to be used in the \(m\) atrix \(m\) ultiplication as follow \(s\) :

TRANSB = N 'or h', op (B) = B.
TRANSB \(=T\) 'or \(\mathrm{t}^{\prime}, \mathrm{op}(\mathrm{B})=\mathrm{B}\) '.
TRANSB = C'or \(\mathrm{t}^{\prime}, \mathrm{op}(\mathrm{B})=\mathrm{B}\) '.
U nchanged on exit.
TRANSB is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

M (input)
O n entry, M specifies the num ber of row s of the matrix op (A ) and of the matrix C.M m ust be at least zero. U nchanged on exit.
N (input)
O n entry, N specifies the num ber of colum ns of the \(m\) atrix \(o p(B)\) and the num ber of colum ns of them atrix C.N m ustbe at least zero. U nchanged on exit.

K (input)
On entry, \(K\) specifies the num berof colum ns of the \(m\) atrix \(o p\) ( \(A\) ) and the num berof row s of the \(m\) atrix op ( B ). K must be at least zero. U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
D OUBLE PRECISION amay ofD \(\mathbb{I M} E N S I O N(L D A, k a)\),
where ka isk when TRANSA = N 'or h', and is
\(m\) otherw ise. Before entry \(w\) ith TRANSA \(=N\) ' or
\(h\) ', the leading \(m\) by \(k\) partof the aray \(A\)
\(m\) ustcontain the \(m\) atrix \(A\), otherw ise the leading
\(k\) by \(m\) part of the array A mustcontain the \(m\) atrix A. U nchanged on exit.

LDA (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen TRANSA \(=N\) 'or \(h\) 'then LDA \(>=\max (1, m)\), otherw ise LDA \(>=\max (1, k)\). U nchanged on exit.

B (input)
DOUBLE PRECISION amay ofD \(\mathbb{I M}\) ENSION (LDB, kb ),
w here kb isn when TRANSB \(=\mathrm{N}\) 'or h ', and is
k otherw ise. Before entry w ith TRANSB \(=\mathrm{N}^{\prime}\) or
h ', the leading k by n part of the aray B \(m\) ust contain the \(m\) atrix \(B\), otherw ise the leading \(n\) by \(k\) partofthe array B mustcontain the \(m\) atrix \(B\). U nchanged on exit.

LD B (input)
On entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program. W hen TRANSB \(=N^{\prime}\) 'or h'then LD B \(>=m a x(1, k)\), otherw ise LD \(B>=m a x(1, n)\). U nchanged on exit.
BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then C need notbe set on input. U nchanged on exit.

C (input/output)
D OUBLE PREC ISION amay ofD \(\mathbb{I M}\) ENSION (LD \(\mathrm{C}, \mathrm{n}\) ).
Before entry, the leading \(m\) by \(n\) partof the array \(\mathrm{C} m\) ust contain the m atrix C , exœept when beta is zero, in which case \(C\) need notbe set on entry. On exit, the array \(C\) is overw rilten by the \(m\) by \(n m a t r i x ~(a l p h a * o p(A) * o p(B)+\) beta*C).

LD C (input)
O n entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program . LD C \(>=\max (1, m)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgem v -perform one of the m atrix-vectoroperations \(\mathrm{y}:=\) alpha*A *x + beta* \(y\) ory := alpha*A *x + beta* \(y\)

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NEDGEMV (TRANSA,M,N,ALPHA,A,LDA,X, INCX,BETA,Y, INCY)}
CHARACTER * 1 TRANSA
\mathbb{NTEGERM,N,LDA,INCX,INCY}
DOUBLE PRECISION ALPHA,BETA
D OUBLE PRECISION A (LDA,*),X (*),Y (*)
SU BROUTINE DGEM V_64 (TRANSA,M,N,ALPHA,A,LDA,X, INCX,BETA,Y,
\mathbb{NCY)}
CHARACTER * 1 TRANSA
\mathbb{NTEGER*8M,N,LDA, INCX,INCY}
DOUBLE PRECISION ALPHA,BETA
D OUBLE PRECISION A (LDA,*),X (*),Y (*)

```

\section*{F95 INTERFACE}
```

SUBROUTINE GEMV ([TRANSA], $\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A], X,[\mathbb{N C X}], B E T A$, Y, [ $\mathbb{N C Y}$ )
CHARACTER ( $4 E N=1$ ) ::TRANSA
$\mathbb{N} T E G E R:: M, N, L D A, \mathbb{N} C X, \mathbb{N} C Y$
REAL (8) ::A LPHA,BETA
REAL (8), D IM ENSION (:) :: X,Y
REAL (8), D $\mathbb{M}$ ENSION (: : : : : A
SUBROUTINE GEM V_64 ([TRANSA], $\mathbb{M}], \mathbb{N}], A \operatorname{LPHA}, A,[L D A], X,[\mathbb{N C X}]$, BETA, Y, [ $\mathbb{N C Y}$ ])

```

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{LD} A, \mathbb{N} C X, \mathbb{N} C Y\)
REAL (8) ::ALPHA,BETA
REAL (8), D \(\mathbb{M}\) ENSION (:) ::X,Y
REAL (8), D \(\mathbb{I M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dgem v (chartransa, intm, intn, double alpha, double
*a, int lda, double *x, intincx, double beta, double *y, int incy);
void dgem v_64 (chartransa, long m, long n, double alpha, double *a, long lda, double *x, long incx, double beta, double *y, long incy);

\section*{PURPOSE}
dgem v perform s one of the \(m\) atrix-vector operations \(y:=\) alpha*A *x + beta*y, ory \(:=\) alpha*A *x + beta*y, w here alpha and beta are scalars, \(x\) and \(y\) are vectors and \(A\) is an \(m\) by \(n\) \(m\) atrix .

\section*{ARGUMENTS}

TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=\mathrm{N}\) 'or h ' \(\mathrm{y}:=\) alpha*A * \(\mathrm{x}+\) beta* y .
TRANSA \(=\) ' 'or \(t^{\prime} y=a l p h a * A ~ * x+b e t a * y\).
TRANSA \(=\) C'ort' \(y:=a l p h a * A\) * \(x+\) beta* \(y\).

U nchanged on exit.
TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

M (input)
O \(n\) entry, M specifies the num ber of row s of the \(m\) atrix \(A . M>=0\). U nchanged on exit.

N (input)
O n entry, \(N\) specifies the num ber of colum ns of the \(m\) atrix A. \(N>=0\). U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha.
U nchanged on exit.

A (input)
B efore entry, the leading \(m\) by \(n\) part of the array
A must contain the \(m\) atrix of coefficients.
U nchanged on exit.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A >= \(\max (1, m)\). U nchanged on exit.

X (input)
\((1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C X))\) when TRANSA \(=\mathrm{N}\) 'or \(h^{\prime}\) and at least \((1+(m-1) * a b s(\mathbb{N} C X))\)
otherw ise. Before entry, the increm ented array \(X\) \(m\) ustcontain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X<>0\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then \(Y\) need notbe set on input. U nchanged on exit.

Y (input/output)
\((1+(m-1) \star \operatorname{abs}(\mathbb{N} C Y))\) when TRANSA \(=\mathrm{N}\) 'or
\(h^{\prime}\) and at least ( \(\left.1+(\mathrm{n}-1)^{\star} \operatorname{abs}(\mathbb{N} C Y)\right)\)
otherw ise. Before entry \(w\) ith BETA non-zero, the increm ented array \(Y\) m ust contain the vectory. On exit, \(Y\) is overw rilten by the updated vectory.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of. \(\mathbb{N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dgeqlf - com pute a Q L factorization of a realM -by-N m atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGEQLFM,N,A,LDA,TAU,W ORK,LDW ORK,INFO)}
\mathbb{N TEGER M ,N,LDA,LDW ORK, INFO}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
SU BROUT\mathbb{NE DGEQLF_64M,N,A,LDA,TAU,W ORK,LDW ORK, INFO )}
\mathbb{NTEGER*8M,N,LDA,LDW ORK, NNFO}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GEQLF (M ], \(\mathbb{N}], A,[L D A], T A U, \mathbb{W} O R K],[L D W O R K],[\mathbb{N F O}])\)
\(\mathbb{N} T E G E R:: M, N, L D A, L D W\) ORK, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A
SU BROUTINE GEQLF_64 ( \(\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{N} O R K],[L D W O R K],[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER (8) ::M ,N,LDA,LDW ORK, \(\mathbb{N}\) FO
REAL (8), D IM ENSION (:) ::TAU,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dgeqlf(intm, intn, double *a, int lda, double *tau, int*info);
void dgeqlif_64 (long m , long n, double *a, long lda, double *tau, long *info);

\section*{PURPOSE}
dgeqlf com putes a Q L factorization of a real M -by -N m atrix \(A: A=Q\) * .

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the M -by -N m atrix A. On exit, if \(m>=\) \(n\), the lower triangle of the subanay A ( \(m\) \(\mathrm{n}+1 \mathrm{~m}, 1 \mathrm{n}\) ) contains the N by N low er triangular \(m\) atrix \(L\); ifm <= \(n\), the elem ents on and below the ( \(n-m\) )-th superdiagonalcontain the M -by -N lower trapezoidalm atrix \(L\); the rem aining elem ents, w ith the array TAU, represent the orthogonal matrix \(Q\) as a product of elem entary reflectors (see Further D etails).

LD A (input)
The leading dim ension of the aray A. LDA >= max (1,M).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dim ension of the array \(W\) ORK. LDW ORK >= \(m\) ax \((1, N)\). Foroptim um perform ance LD \(W\) ORK \(>=N * N B\), where NB is the optim alblocksize.

IfLDW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N F O}\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{I N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(k) \ldots H(2) H(1), \text { where } k=m \text { in }(m, n) .
\]

Each \(H\) (i) has the form
\[
\mathrm{H}(\mathrm{i})=\mathrm{I}-\tan * \mathrm{~V}^{*} \mathrm{~V}^{\prime}
\]
where tau is a real scalar, and \(v\) is a realvectorw ith \(v(m-k+i+1 m)=0\) and \(v(m-k+i)=1\); \(v(1 m-k+i-1)\) is stored on exitin A ( \(1 \mathrm{~m}-\mathrm{k}+\mathrm{i}-1, \mathrm{n}-\mathrm{k}+\mathrm{i}\) ), and tau in TAU (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dgeqp3 - com pute a Q \(R\) factorization \(w\) ith colum n pivoting of a matrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGEQP3M,N,A,LDA,JPVT,TAU,W ORK,LW ORK,INFO)}
INTEGERM,N,LDA,LW ORK,INFO
INTEGER JPVT (*)
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK(*)
SU BROUT\mathbb{NE DGEQP3_64M ,N,A ,LDA ,JPVT,TAU,W ORK,LW ORK, INFO )}
\mathbb{NTEGER*8M,N,LDA,LW ORK,INFO}
INTEGER*8 \mathbb{PVT (})
DOUBLE PRECISION A (LDA,*),TAU (*),WORK(*)

```

\section*{F95 INTERFACE}

SU BROUTINE GEQP3 (M ], \(\mathbb{N}], A,[L D A], J P V T, T A U,[\mathbb{W}\) ORK ], [LW ORK ], [ \(\mathbb{N} F O\) ])
\(\mathbb{N}\) TEGER ::M,N,LDA,LW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: JPVT
REAL (8),D \(\mathbb{I}\) ENSION (:) ::TAU,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A
SU BROUTINE GEQP3_64 ( \(\mathbb{M}], \mathbb{N}], A,[L D A], \mathbb{P V} T, T A U,[W O R K],[L W\) ORK ], [ \(\mathbb{N}\) FO ])
\(\mathbb{N} T E G E R(8):: M, N, L D A, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) :: JPVT

\section*{C INTERFACE}
\#include <sunperfh>
void dgeqp3 (intm , intn, double *a, int lda, int * jpvt, double *tau, int *info);
void dgeqp3_64 (long m, long n, double *a, long lda, long
* jpvt, double *tau, long *info);

\section*{PURPOSE}
dgegp3 com putes a Q R factorization \(w\) th colum \(n\) pivoting of a \(m\) atrix \(A: A * P=Q * R\) using Level3 BLA \(S\).

\section*{ARGUMENTS}
\(M\) (input) The num ber of row s of the \(m\) atrix \(A . M>=0\).

N (input) The num ber of colum ns of them atrix \(A . N>=0\).

A (input/output)
O n entry, the M -by-N m atrix A. On exit, the upper triangle of the aray contains the \(m\) in \((M, N)\)-by \(-N\) upper trapezoidalm atrix R ; the elem ents below the diagonal, together w ith the array TA \(U\), represent the orthogonalm atrix \(Q\) as a product of \(m\) in \((M, N)\) elem entary reflectors.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, M)\).

JPVT (input/output)
On entry, if JPV T (J) ne. 0 , the J-th colum n of \(A\) is perm uted to the frontofA *P (a leading colum \(n\) ); if \(\operatorname{JPV} T(J)=0\), the \(J\) th column of \(A\) is a free column. On exit, if \(\mathbb{P V V T}(J)=K\), then the \(J\) th colum \(n\) of \(A\) * w as the the K -th colum \(n\) of \(A\).

TAU (output)
The scalar factors of the elem entary reflectors.

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al

LW ORK.

LW ORK (input)
The din ension of the anray \(W\) ORK. LW ORK \(>=3 * N+1\). For optim al perform ance LW ORK \(>=2 * N+(N+1) * N B\), w here NB is the optim alblocksize.

IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N F O}\) (output)
\(=0\) : successfulexit.
< 0 : if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue.

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(1) H(2) \ldots H(k), \text { where } k=m \text { in }(m, n) .
\]

Each H (i) has the form
H (i) \(=I-\tan * V^{*} V^{\prime}\)
w here tau is a real/com plex scalar, and \(v\) is a real/com plex vectorw th \(v(1: i-1)=0\) and \(v(i)=1 ; v(i+1 \mathrm{~m})\) is stored on exitin A ( \(+1+1 \mathrm{~m}, \mathrm{i})\), and tau in TAU (i).

B ased on contributions by
G. Q uintana-O rti, D epto. de Inform atica, U niversidad Jaim e I, Spain
X . Sun, C om puterS Science D ept., D uke U niversity, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dgeqpf - routine is deprecated and has been replaced by routine SGEQP3

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGEQPF M,N,A,LDA,JPIVOT,TAU,W ORK,\mathbb{NFO)}}\mathbf{N},\textrm{N},\textrm{N}
\mathbb{NTEGER M,N,LDA,INFO}
INTEGER JPIVOT (*)
DOUBLE PRECISION A (LDA,*),TAU(*),WORK(*)

```

```

\mathbb{NTEGER*8M,N,LDA,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
\mathbb{NTEGER*8 JPIVOT (*)}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GEQPF ( \(\mathbb{M}], \mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V O T}, T A U,[\mathbb{W}\) ORK ], [ \(\mathbb{N F O}]\) )
\(\mathbb{N} T E G E R:: M, N, L D A, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V} O T\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL (8),D IM ENSION (:,:) ::A
SUBROUTINE GEQPF_64 (M ], \(\mathbb{N}], A,[L D A], \mathbb{N} \mathbb{I V O T}, T A U,[\mathbb{W} O R K],[\mathbb{N} F O])\)
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, L D A, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL (8), D \(\mathbb{I M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include < sunperfh>
void dgeqpf(intm, intn, double *a, int lda, int * jpivot, double *tau, int *info);
void dgeqpf_64 (long m, long n, double *a, long lda, long
* jpivot, double *tau, long *info);

\section*{PURPOSE}
dgeqpf routine is deprecated and has been replaced by routine SGEQP3.

SG EQPF com putes a Q R factorization \(w\) ith colum \(n\) pivoting of a realM -by \(-\mathrm{N} m\) atrix \(A: A * P=Q * R\).

\section*{ARGUMENTS}

M (input) The num ber of row sof the \(m\) atrix \(A . M>=0\).
N (input) The num ber of colum ns of the m atrix \(\mathrm{A} \cdot \mathrm{N}>=0\)

A (input/output)
On entry, the M by -N m atrix A. On exit, the upper triangle of the aray contains the \(m\) in \((M, N)\)-by \(-N\) upper triangularm atrix \(R\); the elem ents below the diagonal, together \(w\) ith the array TA \(U\), represent the orthogonalm atrix \(Q\) as a product of \(m\) in \((m, n)\) elem entary reflectors.

LD A (input)
The leading dim ension of the array A. LD A >= max (1, M).

JPIVOT (input/output)
On entry, if \(\mathbb{P} \mathbb{I V O T}\) (i) ne.0, the i-th column of \(A\) is perm uted to the front ofA *P (a leading colum \(n\) ); if \(\mathbb{P} \mathbb{I V O T}\) ( \(i\) ) \(=0\), the \(i\) th colum \(n\) of A is a free column. On exit, if \(\operatorname{PPIVOT}(i)=k\), then the i-th column of A *P was the \(k\)-th colum \(n\) of A.

TAU (output)
The scalar factors of the elem entary reflectors.

W ORK (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{I N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{I N}\) FO \(=-i\), the -th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

Them atrix Q is represented as a product of elem entary reflectors
```

Q = H (1)H(2) ...H (n)

```

Each H (i) has the form \(\mathrm{H}=\mathrm{I}-\tan { }^{*} \mathrm{v}^{*} \mathrm{v}^{\prime}\)
\(w\) here tau is a realscalar, and \(v\) is a real vectorw ith \(\mathrm{v}(1: i-1)=0\) and \(\mathrm{v}(\mathrm{i})=1 ; \mathrm{v}(\mathrm{i}+1 \mathrm{~m})\) is stored on exit in A (i+1 \(m, ~ i)\).

Them atrix \(P\) is represented in jpvtas follow \(s\) : If jut( 7 ) \(=i\)
then the th colum \(n\) ofP is the ith canonicalunitvector.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dgeqrif-com pute a \(Q R\) factorization of a realM -by \(-N \mathrm{~m}\) atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGEQRF(M,N,A,LDA,TAU,W ORK,LDW ORK,INFO)}
\mathbb{NTEGER M,N,LDA,LDW ORK,\mathbb{NFO}}\mathbf{N}\mathrm{ , L}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
SU BROUT\mathbb{NEDGEQRF_64M,N,A,LDA,TAU,W ORK,LDW ORK,INFO )}
\mathbb{NTEGER*8M,N,LDA,LDW ORK, INFO}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)

```

\section*{F95 INTERFACE}
\(\operatorname{SUBROUT\mathbb {NE}GEQRF}(\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N F O}])\)
\(\mathbb{N} T E G E R:: M, N, L D A, L D W\) ORK, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL (8),D IM ENSION (:,:) ::A
SU BROUTINE GEQRF_64 ( \(\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER (8) ::M ,N,LDA,LDW ORK, \(\mathbb{N}\) FO
REAL (8), D IM ENSION (:) ::TAU,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include < sunperfh>
void dgeqrf(intm, intn, double *a, int lda, double *tau, int*info);
void dgeqrif_ 64 (long m , long n, double *a, long lda, double *tau, long *info);

\section*{PURPOSE}
dgeqrf com putes \(a \mathrm{QR}\) factorization of a real M -by-N matrix \(A: A=Q\) * .

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of colum ns of the matrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the M -by -N m atrix A. On exit, the ele\(m\) ents on and above the diagonal of the array contain the \(m\) in \((M, N)\)-by \(-N\) uppertrapezoidalm atrix \(R\) \((R\) is upper triangular if \(m>=n\) ); the elem ents below the diagonal, w ith the aray TAU, represent the orthogonal \(m\) atrix \(Q\) as a productofm in \((m, n\) ) elem entary reflectors (see FurtherD etails).

LDA (input)
The leading dim ension of the array A. LDA >= max (1, M).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the amay \(W\) ORK. LDW ORK >= \(m\) ax \((1, N)\). Foroptim um perform ance LDW ORK \(>=N * N B\), w here NB is the optim alblocksize.

If LD W ORK = -1 , then aw orkspace query is assum ed;
the routine only calculates the optim al size of
the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LDW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\(Q=H(1) H(2) \ldots H(k)\), where \(k=m\) in \((m, n)\).
Each \(H\) (i) has the form
H (i) \(=I-\tan * V^{*} V^{\prime}\)
where tau is a real scalar, and \(v\) is a realvectorw ith
\(\mathrm{v}(1: \mathrm{i}-1)=0\) and \(\mathrm{v}(\mathrm{i})=1\); \(\mathrm{v}(\mathrm{i}+1 \mathrm{~m})\) is stored on exit in A ( \(i+1 m, i)\), and tau in TAU (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dger-perform the rank 1 operation A: alpha*x*y'+A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGER M,N,ALPHA,X, INCX,Y, INCY,A,LDA)}

```

```

DOUBLE PRECISION ALPHA
DOUBLE PRECISION X (*),Y (*),A (LDA,*)

```

```

\mathbb{N}TEGER*8M,N,\mathbb{NCX,}\mathbb{N}CY,LDA
DOUBLE PRECISION ALPHA
DOUBLE PRECISION X (*),Y (*),A (LDA ,*)

```

\section*{F95 INTERFACE}

SUBROUTINE GER (M ], \(\mathbb{N}], A L P H A, X,[\mathbb{N} C X], Y,[\mathbb{N} C Y], A,[L D A])\)
\(\mathbb{N} T E G E R:: M, N, \mathbb{N C X}, \mathbb{N C Y}, L D A\)
REAL (8) ::A LPHA
REAL (8), D IM ENSION (:) :: X,Y
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A
SUBROUTINE GER_64 (M ], \(\mathbb{N}], A L P H A, X,[\mathbb{N C X}], Y,[\mathbb{N C Y}], A,[L D A])\)
\(\mathbb{N} T E G E R(8):: M, N, \mathbb{N C X}, \mathbb{N C Y}, L D A\)
REAL (8) ::ALPHA
REAL (8), D IM ENSION (:) ::X,Y
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A
void dger(intm, intn, double alpha, double *x, int incx, double *y, int incy, double *a, int lda);
void dger_64 (long m, long n, double alpha, double *x, long incx, double *y, long incy, double *a, long lda);

\section*{PURPOSE}
dgerperform s the rank 1 operation \(A:=a \prod h a^{\star} x^{\star} y^{\prime}+A\), \(w\) here alpha is a scalar, \(x\) is an \(m\) elem entvector, \(y\) is an \(n\) elem entvector and \(A\) is an \(m\) by \(n m\) atrix.

\section*{ARGUMENTS}

M (input)
On entry, M specifies the num ber of rows of the \(m\) atrix \(A . M>=0\). U nchanged on exit.

N (input)
O n entry, \(N\) specifies the num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\). U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
\((1+(m-1) \star a b s(\mathbb{N C X}))\). Before entry, the increm ented array \(X \mathrm{~m}\) ust contain them elem ent vectorx. U nchanged on exit.
\(\mathbb{N} C X\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X<>0\). U nchanged on exit.

Y (input)
\((1+(n-1) \star \operatorname{abs}(\mathbb{N} C Y))\). B efore entry, the increm ented array \(Y \mathrm{~m}\) ust contain the \(n\) elem ent vectory. U nchanged on exit.
\(\mathbb{N} C Y\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(. \mathbb{N} C Y<>0\). U nchanged on exit.

A (input/output)

Before entry, the leading \(m\) by \(n\) part of the anray
A must contain the matrix ofcoefficients. On exit, A is overw rilten by the updated \(m\) atrix.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program . LD A \(>=\) \(\max (1, m)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dgerfs -im prove the com puted solution to a system of linear
equations and provides emorbounds and backw ard enroresti-
m}\mathrm{ ates forthe solution

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGERFS (TRANSA,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,B,LDB,}}\mathbf{N},\textrm{L},\textrm{L}
X,LDX,FERR,BERR,W ORK,W ORK2,\mathbb{NFO)}

```
CHARACTER * 1 TRANSA
\(\mathbb{N}\) TEGER N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{P} \mathbb{V} O T\left({ }^{( }\right), W\) ORK \(2\left({ }^{( }\right)\)
D OU BLE PRECISION A (LDA , *), AF (LDAF, \(\left.{ }^{\star}\right)\), B (LDB,\(\left.\star\right), ~ X(L D X, \star)\),
FERR (*), BERR (*), W ORK (*)
SU BROUTINEDGERFS_64 (TRANSA,N,NRHS,A,LDA,AF,LDAF, IPIVOT,B,
    LD \(B, X, L D X, F E R R, B E R R, W\) ORK,W ORK2, \(\mathbb{N} F O\) )
CHARACTER * 1 TRANSA
\(\mathbb{N}\) TEGER*8 N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER * \(8 \mathbb{P} \mathbb{I V O T}(*), W\) ORK 2 ( )
D OUBLE PRECISION A (LDA, \(\left.{ }^{\star}\right)\), AF (LDAF, \(\left.{ }^{\star}\right)\), B (LDB,\(\left.\star\right), ~ X(L D X, \star)\),
\(\operatorname{FERR}\) (*), BERR (*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GERFS ([TRANSA], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F], \mathbb{P} \mathbb{I V O T}\),


CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V O T}, \mathrm{W} O R K 2\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK

SU BROUTINE GERFS_64 ([TRANSA], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F]\), \(\mathbb{P} \mathbb{V} O T, B,[L D B], X,[L D X], F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N F O}])\)

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N}\) TEGER ( 8 ) :: N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{V} O T, W\) ORK2
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (: : : : : A, AF, B, X

\section*{C INTERFACE}
\#include < sunperfh>
void dgenfs (chartransa, intn, int nrhs, double *a, int lda, double *af, int ldaf, int *ipivot, double *b, int ldb, double *x, int ldx, double *ferr, double *bers, int*info);
void dgenfs_64 (chartransa, long \(n\), long nihs, double *a, long lda, double *af, long ldaf, long *ipivot, double *b, long ldb, double *x, long ldx, double *ferr, double *berr, long *info);

\section*{PURPOSE}
dgerfs im proves the com puted solution to a system of linear equations and provides errorbounds and backw ard erroresti\(m\) ates for the solution.

\section*{ARGUMENTS}

TRANSA (input)
Specifies the form of the system of equations:
\(=N\) ': A * \(\mathrm{X}=\mathrm{B} \quad\) ( o transpose)
\(=T\) ': A ** T * \(\mathrm{X}=\mathrm{B}\) ( T ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose \(=\mathrm{T}\) ranspose)

TRANSA is defaulted to N 'forF95 \(\mathbb{I N}\) TERFACE.

N (input) The order of the matrix A. \(\mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the m atrices B and X. NRHS >=0.

A (input) The originalN -by N m atrix A.

\section*{LD A (input)}

The leading dim ension of the array A. LDA >= \(\max (1, N)\).

\section*{AF (input)}

The factors L and U from the factorization \(\mathrm{A}=\) \(\mathrm{P} * \mathrm{~L} * \mathrm{U}\) as com puted by SGETRF.

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, \mathbb{N})\).
\(\mathbb{P I V O T}\) (input)
The pivotindioes from SGETRF ; for \(1<=i<=N\), row \(i\) of the \(m\) atrix \(w\) as interchanged \(w\) th row \(\mathbb{P}\) IV OT (i).
\(B\) (input) The righthand side m atrix \(B\).
LD B (input)
The leading dim ension of the array \(B\). LD B >= \(\max (1, N)\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by SGETRS. On exit, the im proved solution \(m\) atrix \(X\).

\section*{LD X (input)}

The leading dim ension of the array X . LD X >= \(\max (1, \mathbb{N})\).

\section*{FERR (output)}

The estim ated forw ard errorbound for each solution vector \(X()\) (the \(j\) th column of the solution \(m\) atrix X). If XTRUE is the true solution comesponding to \(X(\mathcal{O})\), FERR ( \()\) ) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{\nu})-X\) TRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

BERR (output)
The com ponentw ise relative backw ard error of each solution vector X (j) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension \((3 * N)\)

W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dgerqf-com pute an RQ factorization of a realM -by -N m atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGERQF(M,N,A,LDA,TAU,W ORK,LDW ORK,INFO)}
\mathbb{NTEGER M,N,LDA,LDW ORK,\mathbb{NFO}}\mathbf{N}\mathrm{ , L}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
SUBROUT\mathbb{NEDGERQF_64M,N,A,LDA,TAU,W ORK,LDW ORK,INFO )}
\mathbb{NTEGER*8M,N,LDA,LDW ORK, INFO}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)

```

\section*{F95 INTERFACE}
\(\operatorname{SUBROUT\mathbb {NE}GERQF}(\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N F O}])\)
\(\mathbb{N} T E G E R:: M, N, L D A, L D W\) ORK, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL (8),D \(\mathbb{I}\) ENSION (:,:) ::A
SU BROUTINE GERQF_64 (M ], \(\mathbb{N}], A,[L D A], T A U,[\mathbb{N} O R K],[L D W O R K],[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER (8) ::M ,N,LDA,LDW ORK, \(\mathbb{N}\) FO
REAL (8), D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include < sunperfh>
void dgerqf(intm, intn, double *a, int lda, double *tau, int*info);
void dgerqf_ 64 (long m , long n, double *a, long lda, double *tau, long *info);

\section*{PURPOSE}
dgergf com putes an \(R Q\) factorization of a realM boy-N matrix \(A: A=R * Q\).

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of colum ns of the matrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the M -by -N m atrix A. On exit, if \(m<=\) \(n\), the upper triangle of the subarray \(A(1 m, n-\) \(m+1 \mathrm{n}\) ) contains the \(M\) by \(-M\) upper triangularm atrix \(R\); if \(m>=n\), the elem ents on and above the \(m\) n )-th subdiagonalcontain the M by -N upper trapezoidal \(m\) atrix \(R\); the rem aining elem ents, \(w\) ith the array TAU, represent the orthogonal \(m\) atrix \(Q\) as a product of \(m\) in \((m, n\) ) elem entary reflectors (see FurtherD etails).

LD A (input)
The leading dim ension of the anay A. LDA >= max (1, M).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the array W ORK. LDW ORK >= \(m \operatorname{ax}(1, M)\). Foroptim um perform ance LDW ORK \(>=M * N B\), where NB is the optim alblocksize.

IfLD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N N}\) FO = -i, the i-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\(Q=H(1) H(2) \ldots H(k)\), where \(k=m\) in \((m, n)\).
Each \(H\) (i) has the form

H (i) \(=I-\tan * V^{*} V^{\prime}\)
where tau is a real scalar, and \(v\) is a realvectorw ith \(v(n-k+i+1 m)=0\) and \(v(n-k+i)=1 ; v(1 m-k+i-1)\) is stored on exitin \(A(m-k+i, 1 n-k+i-1)\), and tau in TA U (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dgesdd - com pute the singularvalue decom position (SV D ) of a real M -by-N m atrix A, optionally com puting the leftand right singularvectors

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGESDD(JOBZ,M,N,A,LDA,S,U,LDU,VT,LDVT,W ORK,}
LW ORK,INORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ
\mathbb{NTEGERM,N,LDA,LDU,LDVT,LW ORK,NNFO}
INTEGER IV ORK (*)
DOUBLE PRECISION A (LDA,*), S(*), U (LDU,*), VT (LDVT,*),
W ORK (*)
SUBROUTINEDGESDD_64(0)BZ,M,N,A,LDA,S,U,LDU,VT,LDVT,W ORK,
LW ORK,\mathbb{N ORK,INFO)}
CHARACTER * 1 JOBZ
INTEGER*8M,N,LDA,LDU,LDVT,LW ORK, INFO
INTEGER*8 IN ORK (*)
DOUBLE PRECISION A (LDA,*), S (*), U (LDU,*), VT (LDVT,*),
W ORK (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE GESDD (JOBZ, $\mathbb{M}], \mathbb{N}], A,[L D A], S, U,[L D U], V T,[L D V T]$, [W ORK ], [LW ORK ], [IW ORK ], [ $\mathbb{N} F O$ ])
CHARACTER (LEN=1)::JOBZ
$\mathbb{N}$ TEGER :: M , N,LDA,LDU,LDVT,LW ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{I N}$ ORK

```

SUBROUTINE GESDD_64 (JOBZ, M ], \(\mathbb{N}], A,[L D A], S, U,[L D U], V T,[L D V T]\), [W ORK], [LW ORK ], [IW ORK], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1): : JOBZ
\(\mathbb{N}\) TEGER ( 8 ) :: M , N, LDA, LDU ,LDVT, LW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
REAL (8), D IM ENSION (:) ::S,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:)::A, U,VT

\section*{C INTERFACE}
\# include <sunperfh>
void dgesdd (char jobz, intm, intn, double *a, int lda, double *s, double *u, int ldu, double *vt, int ldvt, int *info);
void dgesdd_64 (char jobz, long \(m\), long \(n\), double *a, long lda, double *s, double *u, long ldu, double *vt, long ldvt, long *info);

\section*{PURPOSE}
dgesdd com putes the singular value decom position (SVD ) of a real M -by -N m atrix A , optionally com puting the left and right singular vectors. If singular vectors are desired, it uses a divide-and-conquer algorithm .

The SVD isw ritten
\(=U * S I G M A *\) transpose \((N)\)
where SIG M A is an \(M\) boy \(-\mathrm{N} m\) atrix which is zero except for its \(m\) in \((m, n)\) diagonal elem ents, \(U\) is an \(M\) boy \(-M\) orthogonal m atrix, and V is an N -by N orthogonalm atrix. The diagonal elem ents of SIGM A are the singular values ofA; they are realand non-negative, and are retumed in descending order. The firstm in ( \(m, n\) ) colum ns of \(U\) and \(V\) are the left and right singular vectors of \(A\).

N ote that the routine retums \(\mathrm{V} \mathrm{T}=\mathrm{V}\) ** T , not V .
The divide and conqueralgorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on \(m\) achines w ith a guard digit in add/subtract, or on those binary m achines w thout guard digits w hich subtract like the C ray X M P , C ray Y M P , C ray C-90, orC ray-2. Itcould conceivably fail on hexadecim al or decim al \(m\) achines \(w\) ithout guard digits, butw e know of none.

\section*{ARGUMENTS}

JOBZ (input)
Specifies options for com puting allorpart of the matrix U :
= A': allm colum ns of U and all N row sof \(\mathrm{V} * * \mathrm{~T}\) are retumed in the amays \(U\) and \(V T ;=S\) : the firstm in \((M, N)\) colum nsof \(U\) and the firstm in \(M, N\) ) row \(S\) of \(V * * T\) are retumed in the amays \(U\) and \(V T\); \(=\mathrm{O}^{\prime}\) : IfM \(>=\mathrm{N}\), the first N columns of U are overw ritten on the array \(A\) and all row sofV \(* * T\) are retumed in the anay VT; otherw ise, all colum ns of \(U\) are retumed in the array \(U\) and the firstM row sofV **T are overw rilten in the aray VT ; \(=\mathrm{N}^{\text {': }}\) no colum ns of U or row sof \(\mathrm{V} * *\) T are com puted.

M (input) The num ber of row sof the inputm atrix \(A . M>=0\).
N (input) The num ber of Colmm ns of the inputm atrix \(\mathrm{A} . \mathrm{N}>=\) 0.

A (input/output)
On entry, the M -by -N m atrix A. On exit, if \(\mathrm{JOBZ}=\) \(0^{\prime}\) ', A is overw ritten \(w\) ith the first N colum ns of U (the left singularvectors, stored colum nw ise) if \(M>=N\); \(A\) is overw ritten \(w\) ith the firstM row \(s\) of \(V * * T\) the right singularvectors, stored row wise) otherw ise. if JOBZ ne. 0 ', the contents ofA are destroyed.

LD A (input)
The leading dim ension of the anay A. LDA >= \(\max (1, M)\).

S (output)
The singularvalues of \(A\), sorted so that \(S\) (i) >= S (i+1).

U (output)
\(\mathrm{UCOL}=\mathrm{M}\) if \(\mathrm{JOBZ}=\mathrm{A}\) 'orJOBZ \(=0\) 'and \(\mathrm{M}<\mathrm{N}\);
\(\mathrm{UCOL}=\mathrm{m}\) in \((\mathrm{M}, N)\) if \(\mathrm{JOBZ}=\mathrm{S}^{\prime}\). If \(\mathrm{JOBZ}=\mathrm{A}\) 'or
JOBZ \(=0\) 'and \(\mathrm{M}<\mathrm{N}\), U contains the M -by -M orthogonalm atrix U ; if \(\mathrm{JOB} \mathrm{Z}=\mathrm{S}\) ', U contains the firstm in \((M, N)\) colum ns of \(U\) the left singular vectors, stored colum nw ise); if \(\mathrm{JOBZ}=0\) 'and M \(>=N\), or \(J O B Z=N\) ', U is not referenced.

LD U (input)
The leading dim ension of the amay \(\mathrm{U} . \mathrm{LDU}>=1\);
if \(\mathrm{OBZ}=\mathrm{S}^{\prime}\) or \(\mathrm{A}^{\prime}\) or \(\mathrm{OBZ}=\mathrm{O}^{\prime}\) and \(\mathrm{M}<\mathrm{N}, \mathrm{LDU}\)
\(>=M\).

VT (output)
If \(\mathrm{OBZ}=\mathrm{A}\) ' or \(\mathrm{OBBZ}=\mathrm{D}^{\prime}\) 'and \(\mathrm{M}>=\mathrm{N}, \mathrm{VT}\) contains the N -by N orthogonalm atrix V **T ; if \(\mathrm{OB} \mathrm{BZ}=\) S',V T contains the firstm in \((M, N)\) row \(s\) of \(V * * T\) (the right singularvectors, stored row w ise); if \(\mathrm{JOBZ}=\mathrm{D}^{\prime}\) 'and \(\mathrm{M}<\mathrm{N}\), or \(\mathrm{OBZ}=\mathrm{N}^{\prime}\), VT is not referenced.

LDVT (input)
The leading dim ension of the aray V T. LDV T >=1;
if \(\mathrm{JOBZ}=\mathrm{A}^{\prime}\) 'or \(\mathrm{OBZ}=\mathrm{O}^{\prime}\) 'and \(\mathrm{M}>=\mathrm{N}, \mathrm{LDVT>=N} \mathrm{;} \mathrm{;}\)
if \(J O B Z=S^{\prime}, L D V T>=m\) in \((M, N)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al
LW ORK;

LW ORK (input)
The dim ension of the anay \(W\) ORK.LW ORK >=1. If
JOBZ \(=N^{\prime}\) ', LWORK >= \(3 * m\) in \(M, N\) ) + \(\max \left(m \operatorname{ax}(\mathbb{M}, N), 6 \star_{m}\right.\) in \(\left.(M, N)\right)\). If OBBZ \(=0^{\prime}, \mathrm{LW} O R K>=\) \(3 \star_{m}\) in \((M, N) \star_{m}\) in \((M, N)+m \operatorname{ax}\left(m a x(M, N), 5 *_{m}\right.\) in \((M, N) *\)
 \(>=3 *_{m}\) in \((\mathbb{M}, N) \star_{m}\) in \((M, N)+m \operatorname{ax}\left(m \operatorname{ax}(M, N), 4 \star_{m}\right.\) in \((M, N) *\) \(m\) in \((M, N)+4 \star_{m}\) in \(\left.(M, N)\right)\). Forgood perform ance, LW ORK should generally be larger. If LW ORK \(<0\) but other input argum ents are legal, W ORK (1) retums optim alLW ORK .

IW ORK (w orkspace)
dim ension \((8 * \mathrm{M} \mathbb{N} \mathbb{M}, N))\)
\(\mathbb{N} F O\) (output)
= 0: successfiulexit.
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue.
\(>0\) : SBD SD C did not converge, updating process failed.

\section*{FURTHER DETAILS}

B ased on contributions by
M ing G u and H uan Ren, C om puterScience D ivision, U niversity of

C alifomia atB erkeley, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgesv - com pute the solution to a real system of linear equations \(A * X=B\),

\section*{SYNOPSIS}

```

\mathbb{NTEGERN,NRHS,LDA,LDB,INFO}
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION A (LDA,*),B (LDB,*)

```

```

\mathbb{N}TEGER*8N,NRHS,LDA,LDB,\mathbb{NFO}
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION A (LDA,*),B(LDB,*)

```

\section*{F95 INTERFACE}

SUBROUTINEGESV ( \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I} O T, B,[L D B],[\mathbb{N F O}])\)
\(\mathbb{I N}\) TEGER ::N,NRHS,LDA,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V O T}\)
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A , B
SUBROUTINE GESV_64 ( \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}],[\mathbb{N} F \mathrm{O}])\)
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDA,LDB, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8), D IM ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void dgesv (intn, intnrhs, double *a, int lda, int *ipivot, double *b, int ldb, int *info);
void dgesv_64 long n, long nrhs, double *a, long lda, long *ipivot, double *b, long ldb, long *info);

\section*{PURPOSE}
dgesv com putes the solution to a real system of linearequations
\(A\) * \(X=B\), where \(A\) is an \(N\) boy \(N \mathrm{~m}\) atrix and \(X\) and \(B\) are N -by-N RH S m atrices.

The LU decomposition with partial pivoting and row interchanges is used to factorA as
\(A=P * L * U\),
\(w\) here \(P\) is a perm utation \(m\) atrix, \(L\) is unit low er triangular, and \(U\) is upper triangular. The factored form of \(A\) is then used to solve the system ofequations \(A * X=B\).

\section*{ARGUMENTS}

N (input) The num ber of linear equations, i.e., the order of them atrix A. N >=0.

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of them atrix B. NRHS \(>=0\).

A (input/output)
On entry, the N -by -N coefficient m atrix A . On exit, the factors \(L\) and \(U\) from the factorization \(A\) \(=\mathrm{P} * \mathrm{~L} * \mathrm{U}\); the unitdiagonalelem ents of L are not stored.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

IPIVOT (output)
The pivot indices that define the perm utation \(m\) atrix \(P\); row i of the \(m\) atrix was interchanged w ith row \(\mathbb{P I V O T}\) (i).

B (input/output)
On entry, the N -by-N RH S m atrix of righthand side
\(m\) atrix \(B\). On exit, if \(\mathbb{N} F O=0\), the \(N\) boy-NRHS solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray \(B\). LD B \(>=\) \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO = -i, the i-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{U}(i, i)\) is exactly zero. The factorization has been com pleted, but the factor \(U\) is exactly singular, so the solution could not be com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dgesvd -com pute the singularvalue decom position (SV D ) ofa
real M -by-N m atrix A, optionally com puting the left and/or
right singularvectors

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGESVD (JOBU,NOBVT,M,N,A,LDA,SNNG,U,LDU,VT,LDVT,}
W ORK,LDW ORK,INFO)
CHARACTER * 1 JOBU,JOBVT
INTEGERM,N,LDA,LDU,LDVT,LDW ORK,\mathbb{NFO}
DOUBLE PRECISION A (LDA,*),SING (*),U (LDU,*),VT (LDVT,*),
W ORK (*)
SU BROUT\mathbb{NEDGESVD_64(JOBU,NOBVT,M,N,A,LDA,SING,U,LDU,VT,}
LDVT,W ORK,LDW ORK,NNFO)
CHARACTER * 1 JOBU,JOBVT
\mathbb{NTEGER*8M,N,LDA,LDU,LDVT,LDW ORK, INFO}
DOUBLE PRECISION A (LDA,*),SING (*), U (LDU,*),VT (LDVT,*),
W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GESVD (JOBU, JOBVT, \(\mathbb{M}], \mathbb{N}], A,[L D A], S \mathbb{N} G, U,[L D U], V T\), [LDVT], \(\mathbb{W}\) ORK], [LDW ORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1)::JOBU, JO BV T
\(\mathbb{N} T E G E R:: M, N, L D A, L D U, L D V T, L D W O R K, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::SING,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A, U,VT
SU BROUTINE GESVD_64 (OBBU, OOBVT, M ], \(\mathbb{N}], A,[L D A], S \mathbb{N} G, U,[L D U]\),

VT, [LDVT], [W ORK], [LDW ORK], [NFO])

CHARACTER (LEN=1) :: \(0 \mathrm{OBU}, \mathrm{JOBV}\) T
\(\mathbb{N}\) TEGER (8) ::M,N,LDA,LDU,LDVT,LDW ORK, \(\mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::SNG,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A , U ,VT

\section*{C INTERFACE}
\#include <sunperfh>
void dgesvd (char jंbu, char jobvt, intm, intn, double *a, int lda, double *sing, double *u, int ldu, double *vt, int ldvt, int *info);
void dgesvd_64 (char jobu, char jobvt, long m, long \(n\), double
*a, long lda, double *sing, double *u, long ldu, double *vt, long ldvt, long *info);

\section*{PURPOSE}
dgesvd com putes the singular value decom position (SVD ) of a real M -by -N m atrix A , optionally com puting the leftand/or right singularvectors. The SVD isw rilten \(=U * S \mathbb{G} M A *\) transpose \((N)\)
where SIG M A is an M -by \(-\mathrm{N} m\) atrix which is zero except for its \(m\) in \((m, n\) ) diagonal elem ents, \(U\) is an \(M\) by \(M\) orthogonal \(m\) atrix, and \(V\) is an \(N\) boy \(-N\) orthogonalm atrix. The diagonal elem ents of SIGM A are the singular values of A ; they are real and non-negative, and are retumed in descending order. The firstm in ( \(m, n\) ) colum ns of \(U\) and \(V\) are the left and right singularvectors ofA.

N ote that the routine retums \(\mathrm{V} * * \mathrm{~T}\), not V .

\section*{ARGUMENTS}
\(J 0 \mathrm{BU}\) (input)
Specifies options for com puting allor part of the \(m\) atrix U :
= \(A\) ': all M colum ns of U are retumed in array U :
\(=S^{\prime}\) : the firstm in \((m, n)\) colum ns of \(U\) (the left singular vectors) are retumed in the array \(U\); \(=\) \(O^{\prime}\) : the firstm in \((m, n)\) colum ns of \(U\) the left singular vectors) are overw ritten on the array A; \(=\mathrm{N}^{\prime}\) : no colum ns of U (no left singularvectors) are com puted.

JOBVT (input)
Specifies options for com puting allor part of the m atrix V **T :
= A : alln rowsofV**T are retumed in the aray VT;
\(=S^{\prime}\) : the firstm in \((m, n)\) row sofV \(* * T\) the right singular vectors) are retumed in the array VT ; \(=\) 0 ': the firstm in \((m, n)\) row s ofV **T the right singular vectors) are overw ritten on the array A; \(=\mathrm{N}^{\prime}\) : no row sofV \({ }^{* *} \mathrm{~T}\) (no right singular vectors) are com puted.

JO BVT and JOBU cannotboth be \(\mathrm{D}^{\prime}\).
\(M\) (input) The num ber of row s of the inputm atrix \(A . M>=0\).
N (input) The num ber of \(\propto\) lum ns of the inputm atrix \(\mathrm{A} . \mathrm{N}>=\) 0.

A (input/output)
On entry, the M -by -N m atrix A. On exit, if \(\mathrm{OBC}=\) \(O^{\prime}\) ' A is overw rilten w ith the firstm in \((m, n)\)
colum ns of (the left singular vectors, stored colum nw ise); if JOBVT = O',A is overw ritten \(w\) th the firstm in \((m, n)\) row sof \(V * * T\) the right singularvectors, stored row w ise); if JO BU ne. 0 'and JOBVT ne. O', the contents of A are destroyed.

LD A (input)
The leading dim ension of the anay A. LDA >= max (1,M).

SING (output)
The singularvalues ofA, sorted so that SING (i) \(>=S \mathbb{N} G(i+1)\).

U (input) ( \(L D U, M\) ) if \(J 0 B U=A\) 'or \((L D U, m\) in \(M, N)\) ) if \(J O B U=\)
\(S^{\prime}\). If \(\mathrm{OBU}=\mathrm{A}\) ', U contains the M -by -M orthogonalm atrix \(U\); if \(J O B U=S\) ', \(U\) contains the first \(m\) in \((m, n)\) colum ns of \(U\) (the left singular vectors, stored colum nw ise); if \(\mathrm{JOBU}=\mathrm{N}\) 'or \(\mathrm{O}^{\prime}\) ' U is not referenced.

LD U (input)
The leading dim ension of the array \(U . L D U \quad>=1\); if \(J 0 B U=S\) 'or \(A\) ', LD \(U>=M\).

VT (input)
If \(\mathrm{OOBVT}=\mathrm{A}\) ', VT contains the N -by-N orthogonal \(m\) atrix \(V * * T\); if \(J O B V T=S^{\prime}, V T\) contains the first
\(m\) in \((m, n)\) row sof \(V * * T\) (the right singularvectors, stored row wise); if JOBVT = N 'or \(\mathrm{D}^{\prime}\), VT is not referenced.

LDVT (input)
The leading din ension of the amray VT. LD V T >=1; if \(\mathrm{JOBVT}=\mathrm{A}, \mathrm{LDVT}>=\mathrm{N}\); if \(\mathrm{if} \mathrm{OBVT}=\mathrm{S}\) ',LDVT >= \(m\) in \(M, N\) ).
W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, \mathrm{~W}\) ORK (1) retums the optim al LDW ORK; if \(\mathbb{N} F O>0, W O R K(2 \mathbb{M} \mathbb{N} M, N)\) ) contains the unconverged superdiagonal elem ents of an upper bidiagonalm atrix B whose diagonal is in SIN G (not necessarily sorted). \(B\) satisfies \(A=U * B * V T\), so it has the same singular values as \(A\), and singular vectors related by \(U\) and V T.

LDW ORK (input)
The dim ension of the array \(W\) ORK. LDW ORK >= 1 . LDW ORK >= MAX ( \(3 \star \mathrm{M} \mathbb{N} M, N\) ) M AX \(M, N), 5 \star \mathrm{M} \mathbb{N} \mathbb{M}, \mathbb{N})\) ). For good perform ance, LD W ORK should generally be larger.

IfLDW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit.
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue.
\(>0\) : if SBD SQR did not converge, \(\mathbb{N} F O\) specifies how \(m\) any superdiagonals of an interm ediate bidiagonalform B did not converge to zero. See the description ofW ORK above fordetails.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgesvx -use the LU factorization to com pute the solution to a realsystem of linear equations \(A * X=B\),

\section*{SYNOPSIS}

```

    EQUED,R,C,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,
    W ORK2,\mathbb{NFO)}
    CHARACTER * 1FACT,TRANSA,EQUED
\mathbb{NTEGER N,NRHS,LDA,LDAF,LDB,LDX, INFO}
\mathbb{NTEGER \mathbb{PIVOT (*),W ORK2(*)}}\mathbf{(*)}
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA,*),AF (LDAF,*),R (*),C (*),B (LDB,*),
X (LDX , *),FERR (*), BERR (*),W ORK (*)
SU BROUTINEDGESVX_64(FACT,TRANSA,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,}
EQUED,R,C,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,
W ORK2,\mathbb{NFO)}
CHARACTER * 1 FACT,TRANSA,EQUED
INTEGER*8N,NRHS,LDA,LDAF,LDB,LDX,}\mathbb{N}F
INTEGER*8 P\mathbb{IVOT (*),W ORK2 (*)}
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA,*),AF (LDAF,*),R (*),C (*),B (LDB ,*),
X (LDX,*),FERR (*),BERR (*),WORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GESVX (FACT, [TRANSA], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F]\), IPIVOT,EQUED,R,C,B,[LDB],X, [LDX],RCOND,FERR, BERR, [W ORK], [W ORK2], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::FACT,TRANSA,EQUED
\(\mathbb{N}\) TEGER :: N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}, \mathrm{W}\) ORK2
REAL (8) :: RCOND
REAL (8),D \(\mathbb{I}\) ENSION (:) ::R,C,FERR,BERR,W ORK
REAL (8),D IM ENSION (:,:) ::A,AF,B,X
SUBROUTINE GESVX_64 (FACT, [TRANSA], \(\mathbb{N}], \mathbb{N R H S}], A,[L D A], A F,[\operatorname{LDAF}]\), IPIVOT,EQUED,R,C,B,[LDB],X,[LDX],RCOND,FERR, BERR, [W ORK], [W ORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::FACT,TRANSA,EQUED
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENS} \operatorname{ION}(:):: \mathbb{P} \mathbb{V} O T, W\) ORK2
REAL (8) ::RCOND
REAL (8),D IM ENSION (:) ::R,C,FERR,BERR,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,AF,B,X

\section*{C INTERFACE}
\#include <sunperfh>
void dgesvx (char fact, chartransa, intn, intnrhs, double
*a, int lda, double *af, int ldaf, int *ipivot, charequed, double *r, double *c, double *b, int ldb, double *x, int ldx, double *roond, double * ferr, double *berr, int *info);
void dgesvx_64 (char fact, chartransa, long n, long nrhs, double *a, long lda, double *af, long ldaf, long *ịívot, char equed, double *r, double *c, double *b, long ldb, double *x, long ldx, double *rcond, double *ferr, double *berr, long *info);

\section*{PURPOSE}
dgesvx uses the LU factorization to com pute the solution to a realsystem of linear equations
\(A * X=B\), where \(A\) is an \(N\) by \(-N m\) atrix and \(X\) and \(B\) are N -by-N R H S m atrices.

Errorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT = E', real scaling factors are com puted to equilibrate
the system:
TRANS \(=\mathrm{N}^{\prime}: \operatorname{diag}(\mathbb{R}) * \mathrm{~A} * \operatorname{diag}(\mathrm{C}) \quad * \operatorname{inv}(\operatorname{diag}(\mathrm{C})) * \mathrm{X}=\) \(\operatorname{diag}(\mathbb{R}) * B\)

TRANS \(=T:(\operatorname{diag}(\mathbb{R}) \star A * \operatorname{diag}(C)) * * T * \operatorname{inv}(\operatorname{diag}(\mathbb{R})) * X=\) diag (C)*B

TRANS \(=C\) ': \((\operatorname{diag}(\mathbb{R}) * A * \operatorname{diag}(C)) * * H * \operatorname{inv}(\operatorname{diag}(\mathbb{R})) * X=\) diag (C)*B

W hether or not the system w illbe equilibrated depends on the
scaling of the m atrix A , but ifequilibration is used, A is
overw rilten by diag \((\mathbb{R}) \star A\) *diag \((C)\) and \(B\) by diag \((\mathbb{R}) \star B\) (if TRANS = N ) ordiag (C)*B (ifTRANS = T'or C).
2. IfFACT = N 'or E', the LU decomposition is used to factor the
\(m\) atrix A (afterequilibration ifFACT = E ) as
\[
A=P * L * U
\]
\(w\) here \(P\) is a perm utation \(m\) atrix, \(L\) is a unit low er triangular
\(m\) atrix, and \(U\) is upper triangular.
3. If som e \(U(i, i)=0\), so that \(U\) is exactly singular, then the routine
retums w ith \(\mathbb{N} F O=\) i. O therw ise, the factored form of \(A\) is used
to estim ate the condition num ber of the \(m\) atrix A. If the reciprocal of the condition num ber is less than m achine precision,
\(\mathbb{N} F O=N+1\) is retumed as a waming, but the routine stillgoes on
to solve for \(X\) and compute error bounds as described below.
4. The system ofequations is solved forX using the factored form
of A.
5. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for it.
6. If equilibration \(w\) as used, the \(m\) atrix \(X\) is prem ultiplied by
diag (C) (ifTRANS = N ) ordiag \((\mathbb{R})\) (ifTRANS \(=T^{\prime}\) or C) so
that it solves the originalsystem before equilibration.

\section*{ARGUMENTS}

FACT (input)
Specifies w hether ornotthe factored form of the \(m\) atrix \(A\) is supplied on entry, and ifnot, w hether them atrix A should be equilibrated before it is factored. = F': On entry, AF and IPIV OT contain the factored form of \(A\). IfEQUED is not \(N^{\prime}\), the \(m\) atrix A has been equilibrated \(w\) ith scaling factors given by R and \(\mathrm{C} . \mathrm{A}, \mathrm{AF}\), and \(\mathbb{P}\) IV OT are not m odified. \(=\mathrm{N}\) : Them atrix A w illbe copied to A F and factored.
\(=\mathrm{E}\) : The matrix A will be equilibrated if necessary, then copied to AF and factored.

TRANSA (input)
Specifies the form of the system of equations:
\(=N^{\prime}: A * X=B \quad\) N o transpose)
\(=T\) ': \(A * * T * X=B \quad\) ( ranspose)
\(=C: A * * H * X=B \quad\) (Transpose)

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE. N (input) The num ber of linearequations, ie., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atriges B and X. NRHS \(>=0\).

A (input/output)
O n entry, the N -by -N m atrix A . IfFA CT \(=\mathrm{F}^{\prime}\) and EQUED is not \(N\) ', then A m usthave been equilibrated by the scaling factors in R and/orC. A is not modified if \(\mathrm{FACT}=\mathrm{F}^{\prime}\) or \(\mathrm{N}^{\prime}\), or if \(\mathrm{FACT}=\) E'and EQU ED = N 'on exit.

On exit, ifEQ UED ne. \(N\) ', A is scaled as follow s: \(\operatorname{EQUED}=\mathrm{R}: A:=\operatorname{diag}(\mathbb{R}) * A\)
\(E Q U E D=C\) ': A \(=A * \operatorname{diag}(C)\)
\(E Q U E D=B \prime A:=\operatorname{diag}(R) * A * \operatorname{diag}(C)\).

LDA (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

AF (input/output)
If FACT \(=F^{\prime}\), then \(A F\) is an inputargum entand on entry contains the factors \(L\) and \(U\) from the factorization \(A=P * L * U\) as com puted by SGETRF. If EQUED ne. \(N^{\prime}\) ', then \(A F\) is the factored form of the equilibrated \(m\) atrix A.

IfFACT \(=N\) ', then AF is an output argum ent and on exit retums the factors \(L\) and \(U\) from the factorization \(A=P * L * U\) of the originalm atrix \(A\).

If \(F A C T=E\) ', then \(A F\) is an output argum ent and on exit retums the factors \(L\) and \(U\) from the factorization \(\mathrm{A}=\mathrm{P} * \mathrm{~L} * \mathrm{U}\) of the equilibrated m atrix A (see the description of \(A\) for the form of the equilibrated \(m\) atrix) .

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, N)\).

PIVOT (inputoroutput)
IfFACT = \(\mathrm{F}^{\prime}\), then \(\mathbb{P I V O T}\) is an input argum ent and on entry contains the pivot indioes from the factorization \(A=P * L * U\) as com puted by SGETRF ; row \(i\) of the matrix was interchanged with row \(\mathbb{P I V O T}\) (i).

IfFACT = N', then \(\mathbb{P I V O T}\) is an output argum ent and on exit contains the pivot indices from the factorization \(A=P * L * U\) of the originalm atrix \(A\).

IfFACT = E', then \(\mathbb{P I V O T}\) is an output argum ent and on exit contains the pivot indices from the factorization \(\mathrm{A}=\mathrm{P} * \mathrm{~L} * \mathrm{U}\) of the equilibrated m atrix A.

EQUED (input/output)
Specifies the form of equilibration thatw as done. \(=\mathrm{N}\) ': No equilibration (alw ays true iffACT = \(\mathrm{N})\).
\(=R\) ': Row equilibration, ie., A has been prem ultiplied by diag \((R)\). = C ': C olum n equilibration, ie., A has been postm ultiplied by diag (C ). = B': B oth row and colum n equilibration, ie., A has been replaced by diag \((\mathbb{R})\) * A * diag (C). EQUED is an inputargum entifFACT= F '; otherw ise, it is an output argum ent.

R (input/output)
The row scale factors for \(A\). IfEQUED \(=R^{\prime}\) or B', A is multiplied on the left.by diag \((\mathbb{R})\); if EQUED = N 'or C', R is notaccessed. \(R\) is an input argum ent ifFACT = \(F\) '; otherw ise, \(R\) is an output argum ent. IfFACT = F'andEQUED = R'or \(B\) ',each elem entofR m ustbe positive.

C (input/output)
The colum n scale factors for \(A\). IfEQ UED = C 'or B', A ismultiplied on the rightby diag (C ) ; if EQUED \(=N\) 'or \(R\) ', \(C\) is notaccessed. \(C\) is an input argum ent ifFACT \(=F\) '; otherw ise, \(C\) is an outputargum ent. IfFACT = F'and EQUED = C'or \(B\) ', each elem entofC \(m\) ust.be positive.

B (input/output)
On entry, the N -by-NRHS righthand side m atrix \(B\). On exit, if EQUED = \(N\) ', \(B\) is notm odified; if TRANSA \(=N^{\prime}\) and EQUED \(=R^{\prime}\) or \(B^{\prime}, B\) is overw rilten by diag \((R) * B\); if TRANSA \(=T\) 'or \(C^{\prime}\) and EQUED \(=C^{\prime}\) or \(B^{\prime}, B\) is overw ritten by diag (C) *B.

LD B (input)
The leading dim ension of the aray \(B\). LD B \(>=\) \(\max (1, N)\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=N+1\), the \(N\)-by-NRH \(S\) solution
\(m\) atrix \(X\) to the original system ofequations.
\(N\) ote that \(A\) and \(B\) are \(m\) odified on exit if EQUED
ne. N ', and the solution to the equilibrated
system is inv (diag (C))*X ifTRANSA = N 'and EQUED
\(=C\) 'or \(B\) ', orinv (diag \((R)) * X\) ifTRANSA \(=T\) 'or \(C^{\prime}\) 'and \(E Q U E D=R\) 'or \(B\) '.

LD X (input)
The leading dim ension of the aray X . LD X >= \(\max (1, N)\).

RCOND (output)
The estim ate of the reciprocal condition num berof the matrix A after equilibration (if done). If
RCOND is less than the \(m\) achine precision (in particular, ifRCOND \(=0\) ), them atrix is singular to w orking precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO \(>0\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X()\) ) the \(j\) th colum n of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{H}), \operatorname{FERR}(1)\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{\nu})-X\) TRU \(E\) ) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost
alw ays a slight overestim ate of the true error.

BERR (output)
The com ponentw ise relative backw ard error of each solution vectorX ( \(j\) ) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exact solution).

W ORK (w orkspace)
dim ension ( \(4 * \mathrm{~N}\) ) On exit, W ORK (1) contains the reciprocal pivot grow th factornom (A)/norm (U). The "m ax absolute elem ent" norm is used. If W ORK (1) is m uch less than 1 , then the stability of the LU factorization of the (equilibrated) matrix A could be poor. This also \(m\) eans that the solution X, condition estim atorRCOND, and forw ard error bound FERR could be unreliable. If factorization fails w ith \(0<\mathbb{N} F O<=N\), then \(W\) ORK (1) contains the reciprocal pivot grow th factor forthe leading \(\mathbb{N} F O\) colum nsofA.

W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{N}\) FO \(=-\)-i, the \(i\)-th argum enthad an illegalvalue
> 0 : if \(\mathbb{N} F O=i\), and \(i\) is
\(<=\mathrm{N}: \mathrm{U}(i, i)\) is exactly zero. The factorization has been completed, but the factor \(U\) is exactly singular, so the solution and error bounds could not be com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1: \mathrm{U}\) is nonsingular, butRCOND is less than machine precision, \(m\) eaning that the \(m\) atrix is singular to w orking precision. N evertheless, the solution and error bounds are com puted because there are a num berof situations w here the com puted solution can bem ore accurate than the value ofRC O ND w ould suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgetf2 -com pute an LU factorization of a general \(m\)-by-n \(m\) atrix A using partialpivoting \(w\) th row interchanges

\section*{SYNOPSIS}

```

\mathbb{NTEGERM,N,LDA,INFO}
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(*)}
DOUBLE PRECISION A (LDA,*)

```

```

INTEGER*8M,N,LDA,INFO
\mathbb{NTEGER*8 \mathbb{PNV (*)}}\mathbf{*}\mathrm{ ( }
DOUBLE PRECISION A (LDA,*)
F95 INTERFACE

```

```

    \mathbb{NTEGER ::M,N,LDA,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
    INTEGER,D IM ENSION (:) :: \mathbb{PIV}
    REAL (8),D IM ENSION (:,:) ::A
    ```

```

    \mathbb{NTEGER (8)::M ,N,LDA,NNFO}
    INTEGER (8),D IM ENSION (:) ::\mathbb{PIV}
    REAL (8),D IM ENSION (:,:) ::A
    C INTERFACE
\#include <sunperfh>

```
void dgetf2 (intm, intn, double *a, int lda, int *ipiv, int *info);
void dgetf2_64 (long m, long n, double *a, long lda, long *ị̀iv, long *info);

\section*{PURPOSE}
dgetf2 com putes an LU factorization of a general \(m\) boy-n \(m\) atrix A using partialpivoting \(w\) ith row interchanges.

The factorization has the form
\[
A=P * L * U
\]
\(w\) here \(P\) is a perm utation \(m\) atrix, \(L\) is low ertriangular \(w\) ith unit diagonal elem ents (low ertrapezoidalifm >n), and U is uppertriangular (uppertrapezoidalifm < n).

This is the right-looking Level2 B LA S version of the algorithm.

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, them by \(n m\) atrix to be factored. On
exit, the factors \(L\) and \(U\) from the factorization \(A\)
\(=P * L * U\); the unitdiagonalelem ents of \(L\) are not stored.

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, M)\).
\(\mathbb{P} \mathbb{I V}\) (output)
The pivotindices; for \(1<=i<=m\) in \((M N)\), row i of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P} \mathbb{I V}\) (i).
\(\mathbb{I N}\) FO (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\mathrm{k}\), the \(k\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=k, U(k, k)\) is exactly zero. The factorization has been com pleted, but the factor \(U\) is
exactly singular, and division by zero will occur
if it is used to solve a system of equations.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgetrf-com pute an LU factorization of a general M -by -N \(m\) atrix A using partialpivoting \(w\) th row interchanges

\section*{SYNOPSIS}

\(\mathbb{N}\) TEGER \(M, N, L D A, \mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{P} \mathbb{I} O T\left({ }^{( }\right)\)
DOUBLE PRECISION A (LDA,*)

SU BROUTINEDGETRF_64 \(M, N, A, L D A, \mathbb{P} \mathbb{I V O T}, \mathbb{N} F O\) )
\(\mathbb{N}\) TEGER*8 M , N,LDA, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{V} O T\left({ }^{*}\right)\)
DOUBLE PRECISION A (LDA,*)

\section*{F95 INTERFACE}
\(\operatorname{SUBROUTINE~GETRF}(\mathbb{M}], \mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I} O T,[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER :: M, N,LDA, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL (8),D \(\mathbb{I}\) ENSION (:,:) ::A
SUBROUTINE GETRF_64 (M ], \(\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V O T},[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER (8) ::M , N,LDA , \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V O T}\)
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dgetrf(intm, intn, double *a, int lda, int *ípivot, int*info);
void dgetrf_ 64 (long m, long n, double *a, long lda, long *ipivot, long *info);

\section*{PURPOSE}
dgetrf com putes an LU factorization of a general M -by -N \(m\) atrix A using partialpivoting \(w\) ith row interchanges.

The factorization has the form
\[
A=P \star L \star U
\]
\(w\) here \(P\) is a perm utation \(m\) atrix, \(L\) is low er triangular \(w\) ith unit diagonal elem ents (low ertrapezoidalifm > n), and U is upper triangular (uppertrapezoidalifm < n).

This is the right-looking Level3 B LA S version of the algorithm .

\section*{ARGUMENTS}
\(M\) (input) The num ber of row s of the \(m\) atrix \(A . M>=0\).

N (input) The num ber of colum ns of the m atrix \(A . N>=0\).

A (input/output)
On entry, the \(M\) by -N m atrix to be factored. On exit, the factors \(L\) and \(U\) from the factorization \(A\)
\(=\mathrm{P} * \mathrm{~L} * \mathrm{U}\); the unitdiagonalelem ents of L are not stored.

LD A (input)
The leading dim ension of the aray A. LD A >= \(\max (1, M)\).

PIVOT (output)
The pivotindioes; for \(1<=i<=m\) in \((M)\) ), row i
of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P I V O T}\) (i).

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the \(i\) th argum enthad an ille-
galvalue
\(>0:\) if \(\mathbb{N} F O=i, U(i, i)\) is exactly zero. The factorization has been com pleted, but the factor \(U\)
is exactly singular, and division by zero will occur if it is used to solve a system ofequations.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgetri-com pute the inverse of a m atrix using the LU factorization com puted by SG ETRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGETRIN,A,LDA, \mathbb{PNOT,W ORK,LDW ORK,INFO )}}\mathbf{N}=1

```
\(\mathbb{I N}\) TEGER N,LDA,LDW ORK, \(\mathbb{N}\) FO \(\mathbb{N} T E G E R \mathbb{P} \mathbb{I} O T\left({ }^{( }\right)\)
D OUBLE PRECISION A (LDA,*),W ORK (*)
SU BROUTINEDGETRI_64 \(\mathbb{N}, A, L D A, \mathbb{P} \mathbb{I} O T, W\) ORK,LDW ORK, \(\mathbb{N} F O\) )
\(\mathbb{N} T E G E R * 8 N, L D A, L D W\) ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{V} O T\left({ }^{*}\right)\)
DOUBLE PRECISION A (LDA ,*),W ORK (*)

\section*{F95 INTERFACE}

SUBROUTINE GETRI( \(\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V O T}, \mathbb{W}\) ORK \(]\), [LDW ORK], [ \(\mathbb{N F O}])\)
\(\mathbb{N}\) TEGER ::N,LDA,LDW ORK, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8),D IM ENSION (:) ::W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A
SU BROUTINE GETRI_64 (N ],A, [LDA], \(\mathbb{P} \mathbb{V} O T,[\mathbb{N} O R K],[L D W\) ORK \(],[\mathbb{N F O}])\)
\(\mathbb{N}\) TEGER (8) ::N,LDA,LDW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8), D IM ENSION (:) ::W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dgetri(intn, double *a, int lda, int *ipivot, int *info);
void dgetri_64 (long n, double *a, long lda, long *ịívot, long *info);

\section*{PURPOSE}
dgetricom putes the inverse of am atrix using the LU factorization com puted by SGETRF .

Thism ethod inverts \(U\) and then com putes inv (A) by solving the system inv (A) \({ }^{(A L}=\operatorname{inv}(\mathbb{U})\) for inv (A).

\section*{ARGUMENTS}

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the factors \(L\) and \(U\) from the factoriza-
tion \(A=P * L * U\) as com puted by SGETRF. On exit, if \(\mathbb{N} F O=0\), the inverse of the originalm atrix \(A\).

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, N)\).

IPIVOT (input)
The pivotindiges from SGETRF; for \(1<=i<=N\), row \(i\) of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P}\) IV OT (i).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0\), then \(W\) ORK (1) retums the optim alLDW ORK.

LDW ORK (input)
The dimension of the array W ORK. LDW ORK >= \(\max (1, N)\). Foroptim alperform ance LDW ORK \(>=N * N B\), where NB is the optim al blocksize retumed by แAENV.

If LD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK aray, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i, U(i, i)\) is exactly zero; the \(m\) atrix is singular and its inverse could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dgetrs -solve a system of linearequations A * X = B orA'

```
* \(\mathrm{X}=\mathrm{B}\) w th a generall -by-N m atrix A using the LU factori- zation com puted by SG ETRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DGETRS(TRANSA,N,NRHS,A,LDA,\mathbb{PIVOT,B,LDB,NNFO)}}\mathbf{N}\mathrm{ (NA,}
CHARACTER * 1 TRANSA
\mathbb{NTEGER N,NRHS,LDA,LDB,}\mathbb{N}FO
\mathbb{NTEGER \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION A (LDA,*),B(LDB,*)
SUBROUT\mathbb{NE DGETRS_64 (TRANSA,N,NRHS,A ,LDA, PIVOT,B,LDB,INFO)}
CHARACTER * 1 TRANSA
INTEGER*8N,NRHS,LDA,LDB,INFO
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION A (LDA,*),B(LDB,*)

```

\section*{F95 INTERFACE}

SU BROUTINE GETRS ([TRANSA], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I V O T}, \mathrm{B},[\operatorname{LD} B]\), [ \(\mathbb{N}\) FO ])

CHARACTER ( \(4 E N=1\) ) ::TRANSA
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V} O T\)
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A , B

SU BROUTINE GETRS_64 ([TRANSA], \(\mathbb{N}], \mathbb{N R H S}], A,[L D A], \mathbb{P} \mathbb{I} O T, B,[L D B]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDA ,LDB, \(\mathbb{N}\) FO
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A, B

\section*{C INTERFACE}
\#include <sunperfh>
void dgetrs (chartransa, intn, int nrhs, double *a, int lda, int *íivot, double *b, int ldb, int *info);
void dgetrs_64 (chartransa, long \(n\), long nrhs, double *a, long lda, long *ipivot, double *b, long ldb, long *info);

\section*{PURPOSE}
dgetrs solves a system of linear equations
\(A * X=B\) or \(A^{\prime *} X=B\) w ith a generalN boy \(N \mathrm{~N}\) matrix \(A\) using the LU factorization com puted by SGETRF .

\section*{ARGUMENTS}

TRANSA (input)
Specifies the form of the system of equations:
\(=N^{\prime}: A * X=B \quad\) N o transpose)
\(=T\) ': A * \(\mathrm{X}=\mathrm{B} \quad\) (Transpose)
\(=C\) : \(A * X=B\) (C onjugate transpose \(=\) Transpose)

TRANSA is defaulted to \(N\) 'forF95 \(\mathbb{I N T E R F A C E .}\)

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS \(>=0\).

A (input) The factors \(L\) and \(U\) from the factorization \(A=\) \(\mathrm{P} * \mathrm{~L} * \mathrm{U}\) as com puted by SG ETRF .

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, N)\).

IPIVOT (input)
The pivotindiaes from \(\operatorname{SGETRF}\); for \(1<=i<=N\), row \(i\)
of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P}\) IV OT (i).

B (input/output)
On entry, the right hand side \(m\) atrix \(B\). On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B . LD B >= \(\max (1, N)\).

IN FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N N}\) FO \(=-i\), the \(i\) th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dggbak - form the rightor lefteigenvectors of a real gen-
eralized eigenvalue problem \(A * x=\operatorname{lam}\) bda*B*x, by backw ard transform ation on the com puted eigenvectors of the balanced pair ofm atrices outputby SG G BA L

\section*{SYNOPSIS}
```

SUBROUTINEDGGBAK(JOB,S\mathbb{DE,N,}\mathbb{NO,IHI,LSCALE,RSCALE,M,V,LDV,}
\mathbb{NFO)}
CHARACTER * 1 JOB,SIDE
\mathbb{NTEGERN,}\mathbb{NO},\mathbb{H}\textrm{I},\textrm{M},LDV,\mathbb{NFO}
DOUBLE PRECISION LSCALE (*),RSCALE (*),V (LDV ,*)
SU BROUTINE DGGBAK_64(ND B,S\mathbb{DE,N,\mathbb{O},\mathbb{H}I,LSCALE,RSCALE,M,V,}
LDV,\mathbb{NFO)}
CHARACTER * 1 JOB,SDE
\mathbb{NTEGER*8N, IOO,\mathbb{H I,M ,LDV,INFO}}\mathbf{N}=1
DOUBLE PRECISION LSCALE (*),RSCALE (*),V (LDV,*)

```

\section*{F95 INTERFACE}

SU BROUTINE GGBAK (JOB,SDE, \(\mathbb{N}], \mathbb{L} O, \mathbb{H} I, L S C A L E, R S C A L E, \mathbb{M}], V\), [LDV], [ \(\mathbb{N} F O]\) )
```

CHARACTER (LEN=1)::JOB,SDE

```

```

REAL (8),D IM ENSION (:) ::LSCALE,RSCALE
REAL (8),D IM ENSION (:,:) ::V

```
SU BROUTINE GGBAK_64 (JOB,SIDE, \(\mathbb{N}], \mathbb{L} O, \mathbb{H} I, L S C A L E, R S C A L E, ~ M ~], V\),

CHARACTER (LEN=1) :: JOB,SDE
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{ILO}, \mathbb{H} \mathrm{I}, \mathrm{M}, \mathrm{LDV}, \mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::LSCALE,RSCALE
REAL (8), D \(\mathbb{I}\) EN SION (:,:) :: V

\section*{C INTERFACE}
\#include <sunperfh>
void dggbak (char job, char side, intn, int ilo, int ini, double *lscale, double *rscale, intm , double *v, intldv, int*info);
void dggbak_64 (char job, charside, long n, long ilo, long ini, double *lscale, double *rscale, long m, double *v, long ldv, long *info);

\section*{PURPOSE}
dggbak form s the right or lefteigenvectors of a real generalized eigenvalue problem \(A{ }^{*} x=\) lam bda*B *x, by backw ard transform ation on the com puted eigenvectors of the balanced pair ofm atriges outputby SG G BA L .

\section*{ARGUMENTS}

JO B (input)
Specifies the type of backw ard transform ation
required:
\(=\mathrm{N}^{\prime}\) : do nothing, retum im m ediately;
\(=\mathrm{P}^{\prime}:\) do backw ard transform ation forperm utation
only;
= S ': do backw ard transform ation for scaling
only;
\(=\mathrm{B}^{\prime}\) : do backw ard transform ations forboth per\(m\) utation and scaling. OB B ustbe the sam e as the argum ent O B supplied to SG G BA L .

SIDE (input)
\(=R\) ': V contains righteigenvectors;
\(=\mathrm{L} \cdot: \mathrm{V}\) contains lefteigenvectors.

N (input) The num ber of row s of the m atrix \(\mathrm{V} . \mathrm{N}>=0\).

IIO (input)
The integers \(\mathbb{H O}\) and \(\mathbb{H}\) I determ ined by SGGBAL. 1
\(<=\mathbb{H} O<=\mathbb{H} I<=N\), if \(N>0\); \(\mathbb{H} O=1\) and \(\mathbb{H} \mathrm{I}=0\), if \(\mathrm{N}=0\).

IH I (input)
See the description for \(\mathbb{I} \mathrm{O}\).
LSCALE (input)
D etails of the perm utations and/or scaling factors applied to the left side of \(A\) and \(B\), as retumed by SG GBAL .

RSCALE (input)
D etails of the perm utations and/or scaling factors applied to the right side of \(A\) and \(B\), as retumed by SG GBAL.

M (input) The num ber of colum ns of the m atrix \(\mathrm{V} . \mathrm{M}>=0\).

V (input/output)
O \(n\) entry, the \(m\) atrix of rightor lefteigenvectors to be transform ed, as retumed by STGEVC. On exit, \(V\) is overw ritten by the transform ed eigenvectors.

LD V (input)
The leading din ension of the \(m\) atrix \(V\). LD \(V>=\) \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0: successfulexit.
<0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvahue.

\section*{FURTHER DETAILS}

See R C.W ard, B alancing the generalized eigenvalue problem, SIAM J.Sci.Stat.C omp. 2 (1981),141-152.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
```

dggbal-balance a pairofgeneral realm atrices (A,B)

```

\section*{SYNOPSIS}
```

SU BROUTINEDGGBAL (JOB,N,A,LDA,B,LDB, ILO, $\mathbb{H} I, L S C A L E, R S C A L E$,
W ORK, $\mathbb{N} F O$ )

```
CHARACTER * 1 Job
\(\mathbb{N}\) TEGER N,LDA,LDB, \(\mathbb{L}, \mathbb{H} \mathrm{I}, \mathbb{N} F O\)
D OU BLE PREC ISION A (LDA,*), B (LDB,*), LSCALE (*), RSCALE (*),
W ORK (*)
SUBROUTINEDGGBAL_64 (OB,N,A,LDA,B,LDB, ILO, IHI,LSCALE,
    RSCALE, W ORK, \(\mathbb{N} F O\) )
CHARACTER * 1 Job
\(\mathbb{N} T E G E R * 8 N, L D A, L D B, \mathbb{L}, \mathbb{H} I, \mathbb{N} F O\)
D OUBLE PRECISION A (LDA, \(\left.{ }^{\star}\right)\), B (LDB,\(\left.^{\star}\right)\), LSCALE (*), RSCALE (*),
W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GGBAL (JOB, \(\mathbb{N}\) ], A, [LDA ], B, [LD B], \(\mathbb{I} O, \mathbb{H} I, L S C A L E\), RSCALE, [W ORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1): : JOB
\(\mathbb{N} T E G E R:: N, L D A, L D B, \mathbb{H}, \mathbb{H} I, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::LSCALE,RSCALE,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B
SU BROUTINE GGBAL_64 (JOB, \(\mathbb{N}], A,[L D A], B,[L D B], \mathbb{L O}, \mathbb{H} I, L S C A L E\), RSCALE, [W ORK], [ \(\mathbb{N} F O]\) )

CHARACTER ( \(L E N=1\) ) : : JOB
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} A, L D B, \Pi O, \mathbb{H} \mathrm{I}, \mathbb{N}\) FO
REAL (8), D \(\mathbb{M}\) ENSION (:) ::LSCALE,RSCALE,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (: : : ) :: A, B

\section*{C INTERFACE}
\#include <sunperfh>
void dggbal(char job, intn, double *a, int lda, double *b, int ldly, int *ilo, int *ini, double *lscale, double *rscale, int*info);
void dggbal 64 (char j’.b, long n, double *a, long lda, double
*b, long ldb, long *ilo, long *ihi, double
*lscale, double *rscale, long *info);

\section*{PURPOSE}
dggbalbalances a pair of general realm atrices ( \(A, B\) ). This involves, first, perm uting \(A\) and \(B\) by sim ilarity transform \(a-\) tions to isolate eigenvalues in the first1 to \(\Pi 0 \$-\$ 1\) and last IH I+1 to N elem ents on the diagonal; and second, applying a diagonalsim ilarity transform ation to row s and colum ns IIO to IH I to \(m\) ake the row s and colum ns as close in norm as possible. B oth steps are optional.

B alancing \(m\) ay reduce the 1-norm of the \(m\) atrices, and im prove the accuracy of the com puted eigenvalues and/oreigenvectors in the generalized eigenvalue problem \(A{ }^{*} x=1 a m . b d a * B{ }^{*} x\).

\section*{ARGUMENTS}

JO B (input)
Specifies the operations to be perform ed on A and
B :
\(=\mathrm{N}^{\prime}\) : none: simply set \(\mathbb{H} 0=1, \mathbb{H} I=\mathrm{N}\), LSCALE (I) = 1.0 and RSCALE (I) = 1.0 for \(i=\)
\(1, \ldots, N .=P\) ': perm ute only;
\(=S\) ': scale only;
\(=B\) ': both perm ute and scale.

N (input) The order of the m atriges A and \(\mathrm{B} . \mathrm{N}>=0\).

A (input/output)
On entry, the inputm atrix A. On exit, A is
overw ritten by the balanced \(m\) atrix. If \(\mathrm{OB}=\mathrm{N}^{\prime}\),

A is not referenced.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, \mathbb{N})\).
\(B\) (input) On entry, the inputm atrix \(B\). On exit, \(B\) is overw rilten by the balanced \(m\) atrix. If \(J \mathrm{~B}=\mathrm{N}^{\prime}\), \(B\) is not referenced.

LD B (input)
The leading dim ension of the array \(B . L D B>=\) \(\max (1, \mathbb{N})\).

ШО (output)
IO O and \(\mathbb{H}\) Iare set to integers such thaton exit
\(A(i, j)=0\) and \(B(i, 1)=0\) if \(i>\) jand \(j=\) \(1, \ldots\), ILO O -1 or \(i=\mathbb{H}\) I \(+1, \ldots, N\). If \(J O B=N\) ' or \(\mathrm{S}^{\prime}, \mathrm{HO}=1\) and \(\mathbb{H} \mathrm{I}=\mathrm{N}\).

IH I (output)
See the description for ILO .
LSCALE (input)
D etails of the perm utations and scaling factors applied to the left side of A and B. IfP \((\mathcal{J})\) is the index of the row interchanged w ith row \(j\) and D ( \(j\) ) is the scaling factor applied to row \(j\) then LSCALE ( \()=\mathrm{P}(\mathrm{j})\) for \(\mathrm{J}=1, \ldots\), ILO-1 \(=\mathrm{D}(\boldsymbol{j})\) for \(J=\mathbb{L O}, \ldots, \mathbb{H} I=P(j) \quad\) for \(J=\mathbb{H} I+1, \ldots, N\). The order in w hich the interchanges are m ade is N to \(\mathbb{H} \mathrm{I}+1\), then 1 to \(\mathbb{L O} \mathrm{O}\).

RSCALE (input)
D etails of the perm utations and scaling factors applied to the right side of \(A\) and \(B\). IfP \((\mathcal{F})\) is the index of the colum \(n\) interchanged with column \(j\) and \(D(j)\) is the scaling factorapplied to column \(j\) then LSCALE \((\mathcal{j})=P(\mathcal{i})\) for \(J=\) \(1, \ldots, \mathbb{H} O-1=D() \quad\) for \(J=\mathbb{H}, \ldots, \mathbb{H} I=P( \rangle)\) for \(J=\mathbb{H} I+1, \ldots, N\). The order in which the interchanges are m ade is N to \(\mathrm{IH} \mathrm{I}+1\), then 1 to ㅍㅇㅇ․

W ORK (w orkspace)
dim ension ( \(6 * \mathrm{~N}\) )
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent had an ille-
galvahue.

\section*{FURTHER DETAILS}

See R . . .W A RD , B alancing the generalized eigenvalue problem, SIAM J.Sci.Stat.Comp. 2 (1981),141-152.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgges - com pute for a pair of \(N\)-by -N real nonsym \(m\) etric
\(m\) atrices \((A, B)\),

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NEDGGES (JOBVSL,JOBVSR,SORT,DELCTG,N,A,LDA,B,LDB,}
SD IM,ALPHAR,ALPHAI,BETA,VSL,LDVSL,VSR,LDVSR,W ORK,LW ORK, BW ORK, $\mathbb{N} F O$ )

```

CHARACTER * 1 JOBVSL, JOBVSR, SORT
\(\mathbb{N}\) TEGER N,LDA,LDB,SD \(\mathbb{I M}, L D V S L, L D V S R, L W O R K, \mathbb{N} F O\) LOGICALDELCTG
LOG ICAL BW ORK (*)
D OUBLE PRECISION A (LDA ,*), B (LDB,*), ALPHAR (*), ALPHAI(*), BETA (*),VSL (LDVSL,*),VSR (LDVSR,*),W ORK (*)

SU BROUTINE DGGES_64 (OO BV SL, JO BV SR , SORT,DELCTG ,N,A, LD A , B ,LD B, SD \(\mathbb{I}\), ALPHAR,ALPHAI,BETA,VSL,LDVSL,VSR,LDVSR,W ORK,LW ORK, BW ORK, \(\mathbb{N} F O\) )

CHARACTER * 1 JOBVSL, JO BVSR,SORT
\(\mathbb{N}\) TEGER* \(8 \mathrm{~N}, \mathrm{LD} A, L D B, S D \mathbb{M}, L D V S L, L D V S R, L W O R K, \mathbb{N} F O\)
LOG ICAL*8DELCTG
LOG ICAL*8 BW ORK (*)
D OUBLE PRECISION A (LDA, *), B (LDB,*), ALPHAR (*), ALPHAI(*),


\section*{F95 INTERFACE}

SU BROUTINE GGES (JOBVSL, JOBVSR,SORT, DELCTG], \(\mathbb{N}], A,[L D A], B,[L D B]\), SD \(\mathbb{M}, A L P H A R, A L P H A I, B E T A, V S L,[L D V S L], V S R,[L D V S R],[W O R K]\), [LW ORK], BW ORK ], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1): : JOBVSL, JOBVSR, SORT
\(\mathbb{N}\) TEGER :: N,LDA,LDB,SD \(\mathbb{I M}, L D V S L, L D V S R, L W O R K, \mathbb{N} F O\)
LOG ICAL ::DELCTG
LOG ICAL,D IM ENSION (:) ::BW ORK
REAL (8),D \(\mathbb{I}\) ENSION (:) ::ALPHAR,ALPHAI,BETA,W ORK
REAL (8), D IM ENSION (:,:) ::A,B,VSL,VSR
SU BROUTINE G GES_64 (JOBVSL, JO BV SR , SORT, DELCTG ], \(\mathbb{N}], A,[L D A], B\), [LD B],SD \(\mathbb{I M}, A L P H A R, A L P H A I, B E T A, V S L,[L D V S L], V S R,[L D V S R]\), [W ORK], [LW ORK], [BW ORK], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1) :: JO BVSL, JO BV SR , SO RT
\(\mathbb{N}\) TEGER (8) ::N,LDA,LDB,SD \(\mathbb{I M}, L D V S L, L D V S R, L W O R K, \mathbb{N F O}\)
LOG ICAL (8) ::DELCTG
LO G ICAL (8),D IM ENSION (:) ::BW ORK
REAL (8),D \(\mathbb{I M}\) ENSION (:) ::A LPHAR,ALPHAI,BETA,W ORK
REAL (8), D IM ENSION (:,:) ::A,B,VSL,VSR

\section*{C INTERFACE}
\#include <sunperfh>
void dgges(char jobvsl, char jobvsr, char sort, int(*delctg) (double,double,double), intn, double *a, int lda, double *b, int ldb, int *sdim, double *alphar, double *alphai, double *beta, double *vsl, int ldvsl, double *vss, int ldvsr, int *info);
void dgges_64 (char jobvsl, char jobvsi, char sort, long (*delctg) (double,double,double), long n, double *a, long lda, double *b, long ldo, long *sdim , double *alphar, double *alphai, double *beta, double *vsl, long ldvsl, double *vsr, long ldvsr, long *info);

\section*{PURPOSE}
dgges com putes for a pair of \(N\)-by -N real nonsym \(m\) etric \(m\) atrices ( \(A, B\) ), the generalized eigenvalues, the generalized realSchur form ( \(\mathrm{S}, \mathrm{T}\) ), optionally, the left and/or right \(m\) atrices of Schurvectors (VSL and V SR ). This gives the generalized Schur factorization
\[
(A, B)=(N S L) * S *(N S R) * * T,(N S L) * T *(N S R) * * T)
\]

Optionally, it also orders the eigenvalues so that a selected chuster of eigenvalues appears in the leading diagonalblocks of the upperquasi-triangularm atrix \(S\) and the upper triangularm atrix \(T\).The leading colum ns of V SL and V SR then form an orthonorm albasis for the comesponding left
and righteigenspaces (deflating subspaces).
(If only the generalized eigenvalues are needed, use the driverSG G EV instead, which is faster.)

A generalized eigenvalue for pairofm atrices \((A, B)\) is a scalar \(w\) or a ratio alpha/beta \(=w\), such that \(A-w\) *B is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation for beta=0 orboth being zero.

A pairofm atrices ( \(\mathrm{S}, \mathrm{T}\) ) is in generalized real Schur form if \(T\) is upper triangularw ith non-negative diagonal and \(S\) is block upper triangularw ith 1 -by-1 and 2 -by-2 blocks. 1-by-1 blocks correspond to real generalized eigenvalues, while 2 -by- 2 blocks ofS w illbe "standardized" by making the corresponding elem ents of \(T\) have the form :
[ a 0 ]
[ 0 b ]
and the pair of corresponding 2 -by- 2 blocks in \(S\) and \(T\) will have a com plex conjugate pairof generalized eigenvalues.

\section*{ARGUMENTS}
\(J 0\) BV SL (input)
\(=N^{\prime}:\) do notcom pute the left:Schurvectors;
\(=\mathrm{V}\) : com pute the left:Schurvectors.
JO BV SR (input)
\(=\mathrm{N}\) ': do notcom pute the rightSchurvectors;
= V ': com pute the rightSchurvectors.
SORT (input)
Specifies w hether ornot to order the eigenvalues on the diagonal of the generalized Schur form . = N ': Eigenvalues are notordered;
= S': Eigenvalues are ordered (see D ELC TG );
DELCTG (input)
DELCTG m ustbe declared EX TERNAL in the calling subroutine. If SORT = N',DELCTG is notreferenced. If SORT = S',DELCTG is used to select eigenvalues to sort to the top leftof the Schur
form. A n eigenvalue (ALPHAR ( \()+\) A LPHA I ( \()\) ) BETA ( \()\) is selected ifD ELCTG (A LPHAR ( 7 ) A LPHAI ( \()\), BETA ( \(\mathbf{j}\) ) is true; ie. if either one of a com plex conjugate pair of eigenvalues is selected, then both com plex eigenvalues are selected.

N ote that in the ill-conditioned case, a selected complex eigenvalue may no longer satisfy DELCTG (ALPHAR ( ) \()\) ALPHAI \((\boldsymbol{\jmath})\), BETA ( \(\boldsymbol{j})=\operatorname{TRUE}\). after ordering. \(\mathbb{N} F O\) is to be set to \(N+2\) in this case.

N (input) The order of the m atrices A, B, VSL, and VSR. N \(>=0\).

A (input/output)
O \(n\) entry, the firstof the pair of \(m\) atrices. On
exit, A has been overw ritten by its generalized Schur form \(S\).
LDA (input)
The leading dim ension ofA. LD A \(>=\max (1, N)\).

B (input/output)
O \(n\) entry, the second of the pairofm atrices. On exit, B has been overw ritten by its generalized Schur form T.

LD B (input)
The leading dim ension ofB. LD B \(>=m\) ax \((1, N)\).

SD \(\mathbb{M}\) (output)
If \(S O R T=N^{\prime}, S D \mathbb{M}=0\). IfSORT \(=S^{\prime}, S D \mathbb{M}=\)
num ber of eigenvalues (aftersorting) forw hich
DELCTG is true. (Complex conjugate pairs for which DELCTG is true foreithereigenvalue count as 2.)

ALPHAR (output)
On exil, (ALPHAR ( ) + ALPHAI ( ) *i) BETA ( ) , \(\dot{j} 1, \ldots, N\), w ill be the generalized eigenvalues.
 are the diagonals of the com plex Schur form ( \(\mathrm{S}, \mathrm{T}\) ) that w ould result if the 2 -by-2 diagonalblocks of the realSchur form of ( \(A, B\) ) w ere further reduced to triangular form using 2 -by -2 complex unitary transform ations. If A LPHA \(I()\) is zero, then the \(j\) th eigenvalue is real; if positive, then the \(j\) th and ( \(j+1\) )-steigenvalues are a com plex con \(j u-\) gate pair, w ith A LPHA I (j+1) negative.

Note: the quotients ALPHAR ( \()\) ) BETA ( \()\) and A LPHAI( ) BETA ( ) may easily over-or underflow, andBETA ( \(\mathcal{O}\) ) may even be zero. Thus, the user should avoid naively com puting the ratio. H ow ever, A LPH AR and A LPH A Iw illbe alw ays less than
and usually com parable \(w\) ith norm (A) in \(m\) agnitude, and BETA alw ays less than and usually com parable w th norm (B).

\section*{A LPH A I (output)}

See the description forA LPHAR.
BETA (output)
See the description forA LPHAR .
VSL (input)
If \(J 0 B V S L=V\) ',VSL willcontain the left Schur vectors. N ot referenced if \(\mathrm{JO} \mathrm{BVSL}=\mathrm{N}\) '.

\section*{LD V SL (input)}

The leading dim ension of the \(m\) atrix VSL. LDVSL
\(>=1\), and if JOBV SL = V', LD V SL >= N .
VSR (input)
If Jo BV SR = V', V SR w illcontain the right Schur vectors. N ot referenced if \(\mathrm{JO} \mathrm{BV} \mathrm{SR}=\mathrm{N}\) '.

LD V SR (input)
The leading dim ension of the \(m\) atrix \(V\) SR .LD \(V\) SR >= 1 , and if \(J O B V S R=V\) ', LD VSR \(>=N\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. LW ORK \(>=8 * N+16\).
IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LW ORK is issued by X ERBLA.
BW ORK (w orkspace)
dim ension \((\mathbb{N}) \mathrm{N}\) ot referenced if SORT \(=\mathrm{N}^{\prime}\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue.
\(=1, \ldots, N\) : The Z Z teration failed. ( \(\mathrm{A}, \mathrm{B}\) ) are not in Schurform, butALPHAR ( \(\mathcal{j})\), ALPHAI \((\mathcal{j})\), and
BETA ( \()\) ) should be comect for \(\mathcal{j} \mathbb{N} F O+1, \ldots, N . \quad>\)
\(\mathrm{N}:=\mathrm{N}+1\) : other than Q Z iteration failed in
SHGEQZ.
\(=\mathrm{N}+2\) : after reordering, roundoff changed values of some com plex eigenvalues so that leading eigenvalues in the Generalized Schur form no longer satisfy DELCTG=TRUE. This could also be caused due to scaling. \(=\mathrm{N}+3\) : reordering failed in STG SEN .

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dggesx - com pute fora pair of \(N\)-by N real nonsym \(m\) etric
\(m\) atrices \((A, B)\), the generalized eigenvalues, the realSchur form ( \(\mathrm{S}, \mathrm{T}\) ), and,

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE DGGESX (JO BV SL,JOBVSR,SORT,DELCTG ,SENSE,N,A,LDA,B,}
LDB,SD IM,ALPHAR,A LPHA I,BETA,VSL,LDVSL,VSR,LDVSR,RCONDE,
RCONDV,W ORK,LW ORK,IW ORK,LIN ORK,BW ORK,INFO)

```

CHARACTER * 1 JOBVSL, JOBVSR,SORT,SENSE
\(\mathbb{N}\) TEGER \(N, L D A, L D B, S D \mathbb{M}, L D V S L, L D V S R, L W O R K, L \mathbb{I N} O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{I N}\) ORK (*)
LOG ICAL DELCTG
LOG ICAL BW ORK (*)
D OUBLE PREC ISION A (LDA, \(\left.{ }^{\star}\right)\), B (LDB,\(\left.^{\star}\right)\), ALPHAR ( \(\left.{ }^{\star}\right)\), ALPHAI( \(\left.{ }^{\star}\right)\),
 W ORK (*)

SU BROUTINE DGGESX_64 (OOBVSL, JOBVSR, SORT,DELCTG, SENSE,N,A,LDA, \(B, L D B, S D \mathbb{I}, A L P H A R, A L P H A I, B E T A, V S L, L D V S L, V S R, L D V S R\), RCONDE,RCONDV,W ORK,LW ORK, \(\mathbb{I W} O R K, L \mathbb{I V} O R K, B W O R K, \mathbb{N} F O\) )

CHARACTER * 1 JOBVSL, JOBVSR, SORT, SEN SE
\(\mathbb{N} T E G E R * 8 N, L D A, L D B, S D \mathbb{M}, L D V S L, L D V S R, L W O R K, L \mathbb{I N} O R K\), \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{I N}\) ORK (*)
LOG ICAL*8DELCTG
LO G ICAL*8BWORK (*)
D OUBLE PREC ISION A (LDA, *), B (LDB, \(\left.{ }^{\star}\right)\), ALPHAR (*), ALPHAI(*), BETA (*), VSL (LDVSL, \(\left.{ }^{\star}\right), \operatorname{VSR}(\mathbb{L D V S R}, \star), \operatorname{RCONDE}\left({ }^{\star}\right)\), RCONDV (*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE G GESX (JOBV SL, JOBVSR, SORT, [DELCTG ], SENSE, \(\mathbb{N}]\) ], A, [LDA ], B, [LDB],SD \(\mathbb{M}, A L P H A R, A L P H A I, B E T A, V S L,[L D V S L], V S R,[L D V S R]\), RCONDE,RCONDV, [WORK], [LW ORK], [IW ORK], [LIW ORK], [BW ORK], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) :: JOBVSL, JOBVSR,SORT,SENSE
\(\mathbb{N} T E G E R:: N, L D A, L D B, S D \mathbb{I}, L D V S L, L D V S R, L W O R K, L \mathbb{N} O R K\),
\(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
LOG ICAL ::DELCTG
LOG ICAL,D IM ENSION (:) ::BW ORK
REAL (8), D \(\mathbb{M}\) ENSION (:) :: ALPHAR, ALPHAI, BETA, RCONDE,
RCONDV,WORK
REAL (8), D IM ENSION (:,:) :: A, B, VSL, V SR
SUBROUTINE GGESX_64 (JOBVSL, JOBVSR,SORT, DELCTG], SENSE, \(\mathbb{N}], A,[L D A]\), B, [LDB],SD \(\mathbb{M}, A L P H A R, A L P H A I, B E T A, V S L,[L D V S L], V S R,[L D V S R]\), RCONDE,RCONDV, [W ORK], [LW ORK], [IW ORK], [LIW ORK], [BW ORK], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) :: JOBVSL, JOBVSR,SORT,SENSE
\(\mathbb{N} T E G E R(8):: N, L D A, L D B, S D \mathbb{M}, L D V S L, L D V S R, L W O R K\),
LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{I N}\) ORK
LO G ICAL (8) ::DELCTG
LOG ICAL (8), D IM ENSION (:) ::BW ORK
REAL (8), D \(\mathbb{M}\) ENSION (:) :: ALPHAR, ALPHAI, BETA, RCONDE, RCONDV,WORK
REAL (8), D IM ENSION (:,:) ::A, B,VSL,VSR

\section*{C INTERFACE}
\#include < sunperfh>
void dggesx (char jobvsl, char jobvsr, char sort, int(*delctg) (double,double,double), char sense, intn, double *a, int lda, double *b, int ldb, int *sdim, double *alphar, double *alphai, double *beta, double *vsl, int ldvsl, double *vss, int ldvsr, double *rconde, double *rcondv, int *info);
void dggesx_64 (char jobvsl, char jobvss, char sort, long (*delctg) (double,double,double), char sense, long n, double *a, long lda, double *b, long ldb, long *sdim , double *alphar, double *alphai, double *beta, double *vsl, long ldvsl, double *vss, long ldvsr, double *roonde, double *roondv, long *info);

\section*{PURPOSE}
dggesx com putes for a pair of N -by N real nonsym \(m\) etric \(m\) atrices \((A, B)\), the generalized eigenvalues, the realSchur form ( \(\mathrm{S}, \mathrm{T}\) ), and, optionally, the leftand/or right \(m\) atrices of Schurvectors (V SL and V SR ). This gives the generalized Schur factorization
A, B\()=(\mathrm{NSL}) \mathrm{S}(\mathrm{NSR}) * * T,(\mathrm{NSL}) \mathrm{T}(\mathrm{NSR}) * * T\) )
Optionally, it also orders the eigenvalues so that a selected cluster of eigenvalues appears in the leading diagonalblocks of the upperquasi-triangularm atrix \(S\) and the upper triangular \(m\) atrix \(T\); com putes a reciprocalcondition num ber for the average of the selected eigenvalues (RCONDE); and com putes a reciprocal condition num ber for the right and left deflating subspaces corresponding to the selected eigenvalues (RCONDV). The leading colum ns ofVSL and VSR then form an orthonorm albasis for the corresponding left and righteigenspaces (deflating subspaces).

A generalized eigenvalue for a pairofm atrices \((A, B)\) is a scalar w or a ratio alphaゐbeta \(=\mathrm{w}\), such that \(A-w * B\) is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation for beta=0 or forboth being zero.

A pairofm atrices ( \(S, T\) ) is in generalized real Schur form if \(T\) is upper triangularw th non-negative diagonal and \(S\) is block uppertriangularw ith 1-by-1 and 2 -by-2 blocks. 1-by-1 blocks comespond to real generalized eigenvalues, while 2-by-2 blocks of S w illbe "standardized" by making the comesponding elem ents of \(T\) have the form :
[ a 0 ]
[ 0 b ]
and the pairof corresponding 2 -by- 2 blocks in \(S\) and \(T\) will have a com plex conjugate pairof generalized eigenvalues.

\section*{ARGUMENTS}

JO BV SL (input)
= N : : do notcom pute the leftSchurvectors;
= V ': com pute the leftSchurvectors.
JO BV SR (input)
\(=\mathrm{N}^{\prime}\) : do not com pute the rightSchurvectors;
= V ': com pute the rightSchurvectors.

\section*{SORT (input)}

Specifies w hether ornot to order the eigenvalues on the diagonal of the generalized Schur form . = N ': Eigenvalues are not ordered;
= S': Eigenvahues are ordered (see D ELC TG ).
DELCTG (input)
DELCTG mustbe declaredEXTERNAL in the calling
subroutine. If \(S O R T=N\) ',DELCTG is not refer-
enced. IfSORT = S',DELCTG is used to select
eigenvalues to sort to the top leftof the Schur
form. A n eigenvalue (A LPHAR ( \(\mathfrak{j}+\mathrm{A}\) LPHA I ( \()\) ) BETA ( 1 )
is selected ifD ELCTG (ALPHAR ( \()\),ALPHAI( ) , BETA ( \(\mathfrak{j}\) )
is true; ie. if either one of a com plex conjugate
pair of eigenvalues is selected, then both com plex
eigenvalues are selected. \(N\) ote that a selected
complex eigenvalue may no longer satisfy
DELCTG (ALPHAR ( \(\mathcal{j}\), ALPHAI \((\mathcal{\jmath})\),BETA \((\mathcal{j})=\) TRUE. after
ordering, since ordering \(m\) ay change the value of
com plex eigenvalues (especially if the eigenvalue
is ill-conditioned), in this case \(\mathbb{N}\) FO is set to
\(\mathrm{N}+3\).

SENSE (input)
D eterm ines which reciprocal condition num bers are com puted. = N ': N one are com puted;
\(=E ': C\) om puted for average of selected eigenvalues only;
= V':C om puted forselected deflating subspaces
only;
= B ':Com puted forboth. IfSENSE = E', V',
or B',SORT m ustequal S'.

N (input) The order of the m atrioes A, B , V SL, and V SR. N \(>=0\).

A (input/output)
O \(n\) entry, the first of the pairof \(m\) atrices. On
exit, A has been overw ritten by its generalized Schur form \(S\).

LD A (input)
The leading dim ension of A . LD \(\mathrm{A}>=\mathrm{max}(1, \mathrm{~N})\).

B (input/output)
O \(n\) entry, the second of the pair ofm atrices. On exit, B has been overw rilten by its generalized Schurform T.

The leading dim ension ofB. LD B \(>=m\) ax \((1, N)\).

SD \(\mathbb{M}\) (output)
If \(S O R T=N^{\prime}, S D \mathbb{I M}=0\). IfSORT \(=S^{\prime}, S D \mathbb{M}=\) num ber of eigenvalues (aftersorting) forw hich DELCTG is true. (Complex conjugate pairs for which DELCTG is true foreithereigenvalue count as 2.)

ALPHAR (output)
On exit, (ALPHAR ( ) + ALPHAI ( ) *i) BETA ( \(\mathcal{7}\), \(\dot{j} 1, \ldots, N\), w ill be the generalized eigenvalues.
 the diagonals of the com plex Schur form \((S, T)\) that w ould result if the 2 -by- 2 diagonalblocks of the real Schur form of ( \(A, B\) ) w ere further reduced to triangular form using 2 -by -2 complex unitary transform ations. If A LPHA \(I()\) is zero, then the \(j\) th eigenvalue is real; if positive, then the \(j\) th and ( \(j+1\) )-steigenvahues are a com plex con \(j u-\) gate pair, w ith A LPHA I(j+1) negative.

Note: the quotients ALPHAR ( \(\mathcal{(}) B E T A(\mathcal{)}\) and ALPHAI ( ) BETA ( ) m ay easily over-or underflow, andBETA ( \(\mathcal{O}\) ) may even be zero. Thus, the user should avoid naively com puting the ratio. How ever, A LPH AR and A LPHA Iw illbe alw ays less than and usually com parable with norm (A) in m agnitude, and BETA alw ays less than and usually com parable w th norm (B).

\section*{A LPH A I (output)}

See the description forA LPHAR.

\section*{BETA (output)}

See the description forA LPHAR .
VSL (input)
If 30 BV SL = V',VSL willcontain the left Schur vectors. N ot referenced if \(\mathrm{JOBVSL}=\mathrm{N}\) '.

LD V SL (input)
The leading dim ension of the \(m\) atrix VSL. LDVSL \(>=1\), and if JO BVSL \(=\mathrm{V}^{\prime}, \mathrm{LDVSL}>=\mathrm{N}\).

VSR (input)
If \(J 0 B V S R=V\), VSR willcontain the right Schur
vectors. N ot referenced if \(\mathrm{JO} \mathrm{BV} \mathrm{SR}=\mathrm{N}\) '.

The leading dim ension of the \(m\) atrix \(V\) SR .LD V SR \(>=\) 1 , and if \(\mathrm{OBVSR}=V^{\prime}, \mathrm{LD} V \mathrm{SR}>=\mathrm{N}\).

RCONDE (output)
IfSENSE = E'or B', RCONDE (1) and RCONDE (2)
contain the reciprocalcondition num bers for the average of the selected eigenvalues. N ot referenced if SEN SE \(=N^{\prime}\) or V'.

RCONDV (output)
If SENSE = V 'or B', RCONDV (1) and RCONDV (2)
contain the reciprocalcondition num bers for the selected deflating subspaces. N ot referenced if SENSE = N'or E'.

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the amay W ORK. LW ORK \(>=\) 8* \((\mathbb{N}+1)+16\). IfSEN SE = E', V', or B',LW ORK >= \(\left.\operatorname{MAX}\left(8^{*} \mathbb{N}+1\right)+16,2 * S D \mathbb{M} *(\mathbb{N}-S D \mathbb{M})\right)\).

IV ORK (w orkspace)
N ot referenced ifSEN SE = N '.

LIW ORK (input)
The dim ension of the anay W ORK. LIW ORK \(>=\mathrm{N}+6\).

BW ORK (w orkspace)
dim ension \((\mathbb{N}) N\) ot referenced if \(S O R T=N^{\prime}\).
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-\) i, the i-th argum enthad an illegalvalue.
\(=1, \ldots, N\) : The \(\mathrm{Q} Z\) teration failed. ( \(\mathrm{A}, \mathrm{B}\) ) are not in Schur form, butA LPHAR ( \(\mathcal{j}\) ) , A LPHAI ( \(\mathcal{j}\) ), and BETA ( \(\mathcal{j}\) ) should be correct for \(\mathcal{F} \mathbb{N F O}+1, \ldots, N .>\)
\(\mathrm{N}:=\mathrm{N}+1\) : other than Q Z iteration failed in SH G EQ Z \(=\mathrm{N}+2\) : after reordering, roundoff changed values of som e complex eigenvalues so thatleading eigenvalues in the Generalized Schur form no longer satisfy DELCTG=.TRUE. This could also be caused due to scaling. \(=\mathrm{N}+3\) : reordering failed in STG SEN .

Further details \(==============\)
A \(n\) approxim ate (asym ptotic) bound on the average
absolute error of the selected eigenvalues is

EPS * norm ( \(\mathrm{A}, \mathrm{B})\) )/RCONDE(1).

A \(n\) approxim ate (asym ptotic) bound on the \(m\) axim um angular error in the com puted deflating subspaces is

EPS * norm ( \((\mathrm{A}, \mathrm{B})) / \mathrm{RCONDV}(2)\).
See LAPACK U sers Guide, section 4.11 for more inform ation.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dggev - com pute for a pair of \(N\)-by -N real nonsym \(m\) etric \(m\) atrices \((A, B)\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGGEV (JOBVL,JOBVR,N,A,LDA,B,LDB,ALPHAR,ALPHAI,}
BETA,VL,LDVL,VR,LDVR,W ORK,LW ORK,\mathbb{NFO)}
CHARACTER * 1 JOBVL,JOBVR
\mathbb{NTEGERN,LDA,LDB,LDVL,LDVR,LW ORK, INFO}
D OUBLE PRECISION A (LDA,*),B (LDB,*), ALPHAR (*), ALPHAI(*),
BETA (*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
SU BROUT\mathbb{NE DGGEV_64(JO BVL,NO BVR,N,A ,LDA,B,LDB,ALPHAR,ALPHA I,}
BETA,VL,LDVL,VR,LDVR,W ORK,LW ORK,\mathbb{NFO)}

```
CHARACTER * 1 JOBVL, 10 BVR
\(\mathbb{N}\) TEGER*8N,LDA,LDB,LDVL,LDVR,LW ORK, \(\mathbb{N} F O\)
D OUBLE PRECISIONA (LDA, \()^{*}\), B (LDB,\(\left.\star\right)\), ALPHAR ( \(\left.{ }^{\star}\right)\), ALPHAI \((\star)\),
BETA (*), VL (LDVL,*), VR (LDVR,*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GGEV (JOBVL, \(\mathcal{J} 0 \mathrm{BVR}, \mathbb{N}], A,[L D A], B,[L D B], A L P H A R\), A LPHA I, BETA,VL, [LDVL],VR, [LDVR], [W ORK ], [LW ORK ], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1): JOBVL, JOBVR
\(\mathbb{N}\) TEGER ::N,LDA,LDB,LDVL,LDVR,LW ORK, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::ALPHAR,ALPHAI,BETA, W ORK
REAL (8), D IM ENSION (:,:) ::A, B,VL,VR
SU BROUTINE GGEV_64 (OOBVL, JOBVR, \(\mathbb{N}], A,[L D A], B,[L D B], A L P H A R\), A LPHAI, BETA, VL, [LDVL],VR, [LDVR], [WORK], [LW ORK], [ \(\mathbb{N F O}])\)

CHARACTER ( \(L E N=1\) ) :: JOBVL, OOBVR
\(\mathbb{N} \operatorname{TEGER}(8):: N, L D A, L D B, L D V L, L D V R, L W O R K, \mathbb{N} F O\)
REAL (8), D \(\mathbb{M} \operatorname{ENSION(:)::ALPHAR,ALPHAI,BETA,WORK~}\)
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A , B, VL , VR

\section*{C INTERFACE}
\#include <sunperfh>
void dggev (char jobvl, char jobvr, int \(n\), double *a, int
lda, double *b, intldb, double *alphar, double
*alphai, double *beta, double *vl, int ldvl, double *Vr, int ldvr, int *info);
void dggev_64 (char jobvl, char jobvr, long n, double *a, long lda, double *b, long ldb, double *alphar, double *alphai, double *beta, double *vl, long ldvl, double *vr, long ldvr, long *info);

\section*{PURPOSE}
dggev com putes for a pair of N -by -N real nonsym \(m\) etric \(m\) atrices ( \(A, B\) ) the generalized eigenvalues, and optionally, the leftand/or right generalized eigenvectors.

A generalized eigenvalue for a pair ofm atrices \((A, B)\) is a scalar lam bda or a ratio alpha/beta = lam bda, such thatA lam bda*B is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation for beta=0, and even forboth being zero.

The righteigenvectorv (i) corresponding to the eigenvalue lam bda ( \()\) ) of ( \(A, B\) ) satisfies
\[
A * v(\mathcal{j})=\operatorname{lam} \operatorname{bda}(\mathcal{j}) * B * v(\mathcal{I})
\]

The lefteigenvectoru ( \(\mathcal{I}\) ) corresponding to the eigenvalue lam boda ( \(\mathcal{I}\) ) of \((A, B)\) satisfies
\[
u(j) \star * H * A=\operatorname{lam} \operatorname{bda}(j) * u(j) \star * H * B .
\]
where \(u(\mathfrak{\jmath}) * * H\) is the conjugate-transpose ofu ( \()\).

\section*{ARGUMENTS}

\section*{JOBVL (input)}
\(=\mathrm{N}\) : : do notcom pute the left generalized eigenvectors;
= \(\mathrm{V}^{\prime}\) : com pute the left generalized eigenvectors.
JO BVR (input)
\(=\mathrm{N}^{\prime}\) : do not com pute the right generalized eigenvectors;
\(=\mathrm{V}\) ': com pute the right generalized eigenvectors.

N (input) The order of the m atrices \(\mathrm{A}, \mathrm{B}, \mathrm{VL}\), and V R. N >= 0.

A (input/output)
On entry, them atrix \(A\) in the pair \((A, B)\). On exit, A has been overw rilten.

\section*{LD A (input)}

The leading dim ension ofA. LD A \(>=\max (1, N)\).
B (input/output)
On entry, them atrix \(B\) in the pair \((A, B)\). On
exit, B has been overw rilten.
LD B (input)
The leading dim ension ofB. LD B \(>=m\) ax \((1, N)\).

\section*{ALPHAR (output)}

On exit, (ALPHAR ( ) + ALPHAI ( ) *i) BETA ( \(\mathcal{7}\), \(\dot{j} 1, \ldots, N, w i l l\) be the generalized eigenvalues. If A LPHAI \((\mathcal{J})\) is zero, then the \(j\) th eigenvalue is real; if positive, then the \(j\) th and (j+1)-st eigenvalues are a com plex conjugate pair, w ith A LPH A I(j+1 ) negative.

Note: the quotients ALPHAR ( ) BETA ( ) \()\) and A LPHAI ( ) BETA ( ) m ay easily over-or underflow, and BETA ( \()\) may even be zero. Thus, the user should avoid naively computing the ratio alpha/beta. H ow ever, A LPHAR and A LPHA I w ill be alw ays less than and usually com parable w ith norm (A) in \(m\) agnitude, and BETA alw ays less than and usually com parable w ith norm (B).

\section*{A LPHA I (output)}

See the description forA LPHAR.

\section*{BETA (output)}

See the description forA LPHAR.
VL (input)
If \(\mathrm{JOBVL}=\mathrm{V}\) ', the left eigenvectors \(\mathrm{u}(\mathcal{)}\) are
stored one after another in the colum ns of \(V L\) ，in the sam e order as theireigenvalues．If the jth eigenvalue is real，then \(u(\mathcal{I})=V L(:, \mathcal{J})\) ，the \(j\) th colum \(n\) ofV L．If the \(j\) th and（ \(j+1\) ）－th eigenvalues form a complex conjugate pair，then \(u(1)=\) \(V L(:\rceil+,i^{\star} V L(:, j+1)\) and \(u(j+1)=V L(:, j)-\) i＊V L（：ュj＋1）．Each eigenvectorw illbe scaled so the largest component have abs（real part）\(+a b s\)（＇m ag．part）\(=1\) ．N otreferenced if 30 BVL \(=\mathrm{N}^{\prime}\) ．

LDVL（input）
The leading dim ension of the \(m\) atrix \(\mathrm{V} \mathrm{L} . \mathrm{LD} V \mathrm{~L}>=1\) ， and if \(\mathrm{JOBVL}=\mathrm{V}^{\prime}, \mathrm{LDVL}>=\mathrm{N}\) 。

VR（input）
If \(\mathrm{OBVR}=\mathrm{V}\)＇，the right eigenvectors \(\mathrm{V}(\mathcal{I})\) are stored one after another in the colum ns of VR，in the sam e order as theireigenvalues．If the jth eigenvalue is real，then \(v(\mathcal{I})=\mathrm{VR}(:, 7)\) ，the \(j\) th colum \(n\) ofVR．If the \(j\) th and（ \(j+1\) ）－th eigenvalues form a complex conjugate pair，then \(v(1)=\) \(\operatorname{VR}(:, j)+i \star V R(:, j+1)\) and \(v(j+1)=\operatorname{VR}(:, \jmath)-\) i＊VR（：,\(j+1\) ）．Each eigenvectorw illbe scaled so the largest component have abs（real part）+ abs（im ag．part）\(=1\) ．N otreferenced if O BVR \(=\mathrm{N}^{\prime}\) ．

LDVR（input）
The leading dim ension of the \(m\) atrix \(V R . L D V R>=1\) ， and if \(\mathrm{JOBVR}=\mathrm{V}\)＇，LDVR \(>=\mathrm{N}\) 。

W ORK（w orkspace）
On exit，if \(\mathbb{N F O}=0, W\) ORK（1）retums the optim al
LW ORK．

\section*{LW ORK（input）}

The dim ension of the amay W ORK．LW ORK＞＝ \(\max (1,8 * N)\) ．Forgood perform ance，LW O RK m ustgen－ erally be larger．

If LW ORK \(=-1\) ，then a w orkspace query is assum ed；
the routine only calculates the optim alsize of the W ORK array，retums this value as the first entry of the W ORK array，and no errorm essage related to LW ORK is issued by XERBLA．
\(\mathbb{N}\) FO（output）
＝0：successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\) ，the \(i\) th argum enthad an ille－
galvalue.
\(=1, \ldots, N\) : The Q Z iteration failed. N o eigenvectors have been calculated, but ALPHAR ( \(\mathcal{I}\), A LPHAI( \()\), and BETA ( 1 ) should be comect for \(\dot{F} \mathrm{NFO}+1, \ldots, N .>N:=\mathrm{N}+1\) : other than Q Z iteration failed in SH G EQ Z .
\(=\mathrm{N}+2\) : error retum from STGEVC.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dggevx - com pute fora pair of \(N\)-by -N real nonsym \(m\) etric \(m\) atrices ( \(A, B\) )

\section*{SYNOPSIS}
```

SUBROUTINE DGGEVX (BALANC,JOBVL,JOBVR,SENSE,N,A,LDA,B,LDB,
A LPHAR,ALPHAI,BETA,VL,LDVL,VR,LDVR, IOO,\#HI,LSCALE,
RSCALE,ABNRM,BBNRM,RCONDE,RCONDV,W ORK,LW ORK,IN ORK,BW ORK,
\mathbb{NFO )}

```
CHARACTER * 1 BALANC, JOBVL, JOBVR, SEN SE
\(\mathbb{N}\) TEGER N,LDA,LDB,LDVL,LDVR, \(\mathbb{L O}, \mathbb{H} I, L W O R K, \mathbb{N} F O\)
\(\mathbb{N T E G E R} \mathbb{I N}\) ORK (*)
LOG ICAL BW ORK (*)
DOUBLE PRECISION ABNRM,BBNRM
D OUBLE PRECISION A (LDA, *), B (LDB,*), ALPHAR (*), ALPHAI(*),
BETA (*), VL (LDVL,*), VR (LDVR,*), LSCALE (*), RSCALE (*),
RCONDE (*), RCONDV (*), WORK (*)
SU BROUTINEDGGEVX_64 BALANC, JOBVL, JOBVR,SENSE,N, A, LDA, B, LD B,
    A LPHAR, ALPHAI, BETA, VL, LDVL, VR, LDVR, \(\mathbb{I} O, \mathbb{H} I, L S C A L E\),
    RSCALE,ABNRM,BBNRM,RCONDE,RCONDV,W ORK,LW ORK, IN ORK,BW ORK,
    \(\mathbb{N} F O\) )

CHARACTER * 1 BALANC, JOBVL, JOBVR,SENSE
\(\mathbb{I N} T E G E R * 8 N, L D A, L D B, L D V L, L D V R, \mathbb{L} O, \mathbb{H} I, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{I N}\) ORK (*)
LO GICAL*8 BW ORK (*)
DOUBLE PRECISION ABNRM,BBNRM
DOUBLE PRECISION A (LDA \(\left.{ }^{\star}\right), \mathrm{B}(\mathrm{LDB}, \star), \operatorname{ALPHAR}(\star), \operatorname{ALPHAI}(\star)\),


\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N E}\) G GEVX BALANC, JOBVL, JOBVR, SENSE, \(\mathbb{N}], A,[L D A], B,[L D B]\), A LPHAR, ALPHA \(\mathrm{I}, \mathrm{BETA}, \mathrm{VL},[\operatorname{LDVL}], \mathrm{VR},[\operatorname{LDVR}], \mathbb{I} O, \mathbb{H} \mathrm{I}, \mathrm{LSCALE}\), RSCALE,ABNRM,BBNRM,RCONDE,RCONDV, [W ORK], [LW ORK], [IN ORK], [BW ORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::BALANC, JOBVL, JOBVR, SEN SE
\(\mathbb{N} T E G E R:: N, L D A, L D B, L D V L, L D V R, \mathbb{L}, \mathbb{H} I, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
LOG ICAL,D IM ENSION (:) ::BW ORK
REAL (8) ::ABNRM,BBNRM
REAL (8), D \(\mathbb{M}\) ENSION (:) :: ALPHAR, ALPHAI, BETA, LSCALE,
RSCALE,RCONDE,RCONDV,W ORK
REAL (8), D \(\mathbb{I M}\) ENSION (:,:) :: A, B, VL, VR
SU BROUTINE GGEVX_64 (BALANC, JOBVL, JOBVR,SENSE, \(\mathbb{N}\) ], A, [LDA],B,
\([\mathrm{LD} B], A \operatorname{LPHAR}, \mathrm{~A} P \mathrm{H} A \mathrm{I}, \mathrm{BETA}, \mathrm{VL},[\mathrm{LDVL}], \mathrm{VR},[\mathrm{LDVR}], \mathbb{I} \mathrm{O}, \mathbb{H} \mathrm{I}\), LSCALE,RSCALE,ABNRM,BBNRM,RCONDE,RCONDV, [W ORK], [LW ORK], [ \(\mathbb{I W}\) ORK], \([B W\) ORK], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::BALANC, JOBVL, JOBVR,SENSE
\(\mathbb{N}\) TEGER (8) :: \(N\),LDA,LDB,LDVL,LDVR, \(\mathbb{L}, \mathbb{H} I, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{I W}\) ORK
LOG ICAL (8),D IM ENSION (:) ::BW ORK
REAL (8) ::ABNRM,BBNRM
REAL (8), D \(\mathbb{M}\) ENSION (:) :: ALPHAR, ALPHAI, BETA, LSCALE,
RSCALE,RCONDE,RCONDV,W ORK
REAL (8), D IM ENSION (:,:) ::A, B, VL,VR

\section*{C INTERFACE}
\#include <sunperfh>
void dggevx (charbalanc, char jobvl, char jobvr, char sense, int \(n\), double *a, int lda, double *b, intldb, double *alphar, double *alphai, double *beta, double *vl, int ldvl, double *vr, int ldvr, int *ilo, int*ihi, double *lscale, double *rscale, double *abnrm, double *bbnm, double *rconde, double *rcondv, int*info);
void dggevx_64 (charbalanc, char jobvl, char jobvr, char sense, long \(n\), double *a, long lda, double *b, long ldb, double *alphar, double *alphai, double *beta, double *vl, long ldvl, double *vr, long ldvr, long *ilo, long *ihi, double *lscale, double *rscale, double *abnm, double *bbnım, double *rconde, double *rcondv, long *info);

\section*{PURPOSE}
dggevx com putes for a pair of N boy -N real nonsymm etric \(m\) atrices ( \(A, B\) ) the generalized eigenvalues, and optionally, the left and/or rightgeneralized eigenvectors.

O ptionally also, itcom putes a balancing transform ation to im prove the conditioning of the eigenvalues and eigenvectors (ILO , IH I, LSCALE, RSCALE, ABNRM , and BBNRM ), reciprocal condition num bers for the eigenvalues \((\mathbb{R C O N D E )}\), and reciprocal condition num bers for the righteigenvectors (RCONDV).

A generalized eigenvalue for a pair ofm atrices \((A, B)\) is a scalar lam boda or a ratio alphaßeta = lam bda, such thatA lam loda*B is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation for beta=0, and even forboth being zero. The righteigenvectorv (i) corresponding to the eigenvalue lam bda ( \()\) ) of ( \(\mathrm{A}, \mathrm{B}\) ) satisfies
\[
A \star v(\mathcal{J})=\operatorname{lam} \operatorname{bda}(\mathcal{j}) \star B * v(\mathcal{I}) .
\]

The lefteigenvectoru ( 7 ) comesponding to the eigenvalue lam bda ( \()\) ) of ( \(A, B\) ) satisfies
\(u(\mathcal{j}) * * H * A=\operatorname{lam} \operatorname{bda}(\mathfrak{j}) * u(\mathfrak{j}) * * H * B\).
where \(u(\mathfrak{j}) * * H\) is the conjugate-transpose of \(u(7)\).

\section*{ARGUMENTS}

BALANC (input)
Specifies the balance option to be perform ed. =
N ': do notdiagonally scale orperm ute;
= \(\mathrm{P}^{\prime}\) : perm ute only;
= S ': scale only;
\(=B^{\prime}\) : both perm ute and scale. C om puted reciprocal condition num bers \(w\) ill be forthem atrices afterperm uting and/orbalancing. Perm uting does not change condition num bers (in exactarith\(m\) etic), butbalancing does.

JO BVL (input)
\(=\mathrm{N}\) : do notcom pute the leftgeneralized eigen-
vectors;
\(=\mathrm{V}\) ': com pute the left generalized eigenvectors.
\(=N^{\prime}\) : do not com pute the right generalized eigenvectors;
\(=\mathrm{V}\) : com pute the right generalized eigenvec-
tors.

SENSE (input)
D eterm ines which reciprocal condition num bers are
com puted. = N ': none are com puted;
\(=\mathrm{E}\) ': com puted for eigenvalues only;
\(=\mathrm{V}\) ': com puted foreigenvectors only;
\(=B\) ': com puted foreigenvalues and eigenvectors.
N (input) The order of the m atrices \(\mathrm{A}, \mathrm{B}, \mathrm{VL}\), and \(\mathrm{VR} . \mathrm{N}>=\) 0.

A (input/output)
On entry, them atrix A in the pair ( \(A, B\) ). On
exit, A has been overw rilten. If \(\mathrm{JOBVL}=\mathrm{V}\) ' or
\(J O B V R=V\) 'orboth, then A contains the first part
of the realSchur form of the "balanced" versions
of the input \(A\) and \(B\).

LDA (input)
The leading dim ension ofA. LD A \(>=\max (1, N)\).
B (input/output)
On entry, them atrix \(B\) in the pair \((A, B)\). On exit, \(B\) has been overw ritten. If \(J O B V L=V\) 'or \(J 0 B V R=V\) 'orboth, then \(B\) contains the second part of the realSchur form of the "balanced" versions of the input \(A\) and \(B\).

LD B (input)
The leading dim ension ofB. LD B \(>=m\) ax \((1, N)\).

\section*{ALPHAR (output)}
 \(\dot{于} 1, \ldots, \mathrm{~N}, \mathrm{w}\) ill be the generalized eigenvalues. IfA LPHA \(I(\mathcal{I})\) is zero, then the \(j\) th eigenvalue is real; if positive, then the \(j\) th and (j+1)-st eigenvalues are a com plex conjugate pair, w ith A LPHA I(j+1) negative.

Note: the quotients ALPHAR ( ) BETA ( ) and
A LPHAI ( \()\) BETA ( \()\) m ay easily over-orunderflow, andBETA \((\rightarrow)\) may even be zero. Thus, the user should avoid naively computing the ratio
A LPHA BETA. H ow ever, A LPHAR and ALPHAI w ill be alw ays less than and usually com parable w ith norm (A) in \(m\) agnitude, and BETA alw ays less than and usually com parable w ith nom (B).

A LPH A I (output)
See the description of A LPH A R .

BETA (output)
See the description of LPHAR.

VL (output)
If \(\mathrm{JOBVL}=\mathrm{V}\) ', the left eigenvectors \(\mathrm{u}(\mathrm{y})\) are stored one afteranother in the colum ns of V L, in the sam e order as theireigenvalues. If the \(j\) th eigenvalue is real, then \(u(j)=V L(:, ~), ~ t h e ~ j t h ~\) colum n ofVL. If the \(j\) th and ( \(j+1\) )-th eigenvalues form a complex conjugate pair, then \(u(\mathcal{l})=\) \(V L(:, \gamma+i \star V L(:, j+1)\) and \(u(j+1)=V L(:, \gamma)-\) i*VL (: \(\because\) jł1). Each eigenvectorw ill be scaled so the largest com ponent have abs (real part) + abs(mag. part) \(=1 . \operatorname{Not}\) referenced if \(J 0 \mathrm{BVL}=\) N '.

LDVL (input)
The leading dim ension of the \(m\) atrix \(V \mathrm{~L} . \mathrm{LD} \operatorname{VL}>=1\), and if \(\mathrm{JOBVL}=\mathrm{V}\) ', LDVL \(>=\mathrm{N}\).

VR (output)
If \(\mathrm{JOBVR}=\mathrm{V}\) ', the right eigenvectors \(\mathrm{v}(\mathrm{J})\) are stored one after another in the colum ns of VR, in the sam e order as theireigenvalues. If the jth eigenvalue is real, then \(v(j)=\operatorname{VR}(:, r)\), the \(j\) th colum \(n\) ofVR. If the \(j\) th and ( \(j+1\) )-th eigenvalues form a complex conjugate pair, then \(\mathrm{v}(\mathcal{1})=\) \(\operatorname{VR}(:, 1)+i \star V R(:, j+1)\) and \(V(j+1)=V R(:, j)-\) \(i \star V R(:, j+1)\). Each eigenvectorw ill be scaled so the largest com ponent have abs(real part) + abs(mag. part) \(=1\). N ot referenced if JO BVR \(=\) N '.

LDVR (input)
The leading dim ension of the \(m\) atrix V R .LD VR \(>=1\), and if \(J O B V R=V{ }^{\prime}, L D V R>=N\).

ㅍO (output)
IO and \(\mathbb{H}\) I are integer values such that on exit
\(A(i, j)=0\) and \(B(i, 7)=0\) if \(i>j\) jand \(j=\) \(1, \ldots, \mathbb{I} O-1\) or \(i=\mathbb{H} I+1, \ldots, N\). IfBALANC \(=N^{\prime}\) or \(\mathrm{S}^{\prime}, \mathrm{HO}=1\) and \(\mathbb{H} \mathrm{I}=\mathrm{N}\).

IH I (output)
See the description of ILO .

\section*{LSCALE (output)}

D etails of the perm utations and scaling factors applied to the left side of A and B. IfPL ( \()\) ) is the index of the row interchanged w ith row \(j\) and D L ( ) is the scaling factor applied to row \(j\) then LSCALE ( \(\mathfrak{j}\) ) = PL ( \(\mathfrak{j}\) ) for \(j=1, \ldots\), ILO-1 = DL ( \()\) for \(j=\mathbb{L} O, \ldots, \mathbb{H} I=P L(j)\) for \(j=\mathbb{H} I+1, \ldots, N\). The order in which the interchanges are m ade is N to \(\mathbb{H} \mathrm{I}+1\), then 1 to \(\mathbb{L} O-1\).

\section*{RSCALE (output)}

D etails of the perm utations and scaling factors applied to the right side of \(A\) and \(B\). IfPR () is the index of the colum \(n\) interchanged \(w\) th \(c o l m m\) \(j\) and \(\operatorname{DR}(j)\) is the scaling factor applied to column \(j\) then RSCALE ( \(\mathcal{j})=\operatorname{PR}(\mathcal{j})\) for \(j=\)
 for \(j=\mathbb{H} I+1, \ldots, N\) The order in which the interchanges are \(m\) ade is \(N\) to \(\mathbb{H} \mathrm{I}+1\), then 1 to \(\mathbb{I L O}-1\).

ABNRM (output)
The one-norm of the balanced m atrix A .

BBNRM (output)
The one-norm of the balanced \(m\) atrix B .

RCONDE (output)
If SENSE = E'or B', the reciprocal condition
num bers of the selected eigenvalues, stored in consecutive elem ents of the array. For a com plex conjugate pair of eigenvahues tw o consecutive ele\(m\) ents ofRCONDE are set to the sam e value. Thus RCONDE ( \(j\) ), RCONDV \((\boldsymbol{j})\), and the \(j\) th colum ns of \(L\) and VR allcorrespond to the sam e eigenpair but not in general the jth eigenpair, unless all eigenpairs are selected). If SEN SE = V', RCONDE is not referenced.

RCONDV (output)
If SEN SE = V 'or B ', the estim ated reciprocal condition num bers of the selected eigenvectors, stored in consecutive elem ents of the array. For a com plex eigenvector tw o consecutive elem ents of RCONDV are set to the sam e value. If the eigenvalues cannot be reordered to com pute RCONDV( \(\mathcal{j}\) ), RCONDV ( \(\mathcal{j}\) ) is set to 0 ; this can only occur when the true value would be very sm allanyw ay. If SENSE = E',RCONDV is notreferenced.

W ORK (w orkspace)

On exit, if \(\mathbb{N} F O=0, W\) ORK ( 1 ) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the amay W ORK. LW ORK >= \(\max (1,6 * N)\). If \(S E N S E=E\) ', LW ORK \(>=12 * N\). If SEN SE \(=\) V 'or B',LW ORK \(>=2 \star \mathrm{~N} * N+12 * N+16\).

IfLW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK amay, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

IW ORK (w orkspace)
dim ension \((\mathbb{N}+6)\) IfSEN \(S E=E ', \mathbb{I W}\) ORK is notreferenced.

BW O RK (w orkspace)
dim ension \((\mathbb{N})\) If \(S E N S E=N^{\prime}, B W\) ORK is not referenced.
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvahue.
\(=1, \ldots, N\) : The Q Z iteration failed. N o eigenvec-
tors have been calculated, but ALPHAR ( \(\mathcal{I}\),
A LPHAI( 1 ), and BETA ( 1 ) should be conrect for \(\dot{F} \mathbb{N} F O+1, \ldots, N .>N:=N+1:\) other than Q Z iteration failed in SH G EQ Z .
\(=N+2\) : error retum from STGEVC.

\section*{FURTHER DETAILS}
\(B\) alancing a m atrix pair ( \(A, B\) ) includes, first, perm uting row \(s\) and colum ns to isolate eigenvalues, second, applying diagonal sim ilarity transform ation to the row s and colum ns to \(m\) ake the row sand colum ns as close in norm as possible. The com puted reciprocal condition num bers comespond to the balanced \(m\) atrix. Perm uting row \(s\) and colum ns w ill notchange the condition num bers (in exact arithm etic) but diagonal scaling w ill. For further explanation ofbalancing, see section 4.11 .12 of LA PA CK U sers'G uide.

A \(n\) approxim ate errorbound on the chordal distance betw een the i-th computed generalized eigenvalue \(w\) and the corresponding exacteigenvalue lam bda is hord (w, lam bda) <= EPS * norm (A BNRM ,BBNRM) /RCONDE (I) A \(n\) approxim ate errorbound forthe angle betw een the \(i\)-th
com puted eigenvectorV L (i) orVR (i) is given by PS * norm (ABNRM , BBNRM) /D \(\mathbb{F}\) (i).

For furtherexplanation of the reciprocalcondition num bers
RCONDE and RCONDV , see section 4.11 ofLAPACK U sers G uide.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgggm -solve a general G auss -M arkov linear model (G LM ) problem

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGGGLM N,M,P,A,LDA,B,LDB,D,X,Y,W ORK,LDW ORK,}
\mathbb{NFO)}
\mathbb{NTEGERN,M,P,LDA,LDB,LDW ORK, INFO}
DOUBLE PRECISION A (LDA,*), B (LDB,*), D (*), X (*), Y (*),
W ORK (*)
SUBROUT\mathbb{NEDGGGLM_64 N,M,P,A,LDA,B,LDB,D,X,Y,W ORK,LDW ORK,}
\mathbb{NFO)}

```

```

DOUBLE PRECISION A (LDA,*), B (LDB,*), D (*), X (*), Y (*),
W ORK (*)

```

\section*{F95 INTERFACE}

SUBROUT \(\mathbb{N E}\) GGGLM ( \(\mathbb{N}], \mathbb{M}], \mathbb{P}], A,[L D A], B,[L D B], D, X, Y,\left[\begin{array}{l}\text { ORK }],\end{array}\right.\) [LDW ORK], [ \(\mathbb{N} F O]\) )
\(\mathbb{N} T E G E R:: N, M, P, L D A, L D B, L D W\) ORK, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D , X, Y, W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B
SU BROUTINE G G G LM _64 ( \(\mathbb{N}], \mathbb{M}],[P], A,[L D A], B,[L D B], D, X, Y,[\mathbb{N} O R K]\),
[LDW ORK], [ \(\mathbb{N F O}])\)
\(\mathbb{N} T E G E R(8):: N, M, P, L D A, L D B, L D W O R K, \mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D , X,Y,W ORK

REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void dggglm (intn, intm , intp, double *a, int lda, double
*b, int lda, double *d, double *x, double *y, int *info);
void dggglm _64 (long n, long m , long p, double *a, long lda, double *b, long ldb, double *d, double *x, double *y, long *info);

\section*{PURPOSE}
dggglm solves a general G auss M arkov linear model (G LM ) problem :
\(m\) inim ize \(\|y\| 2\) subject to \(d=A * x+B * y\)
x
where \(A\) is an \(N\) by \(M M\) atrix, \(B\) is an \(N\) by \(P m\) atrix, and \(d\) is a given N -vector. It is assum ed that \(\mathrm{M}<=\mathrm{N}<=\mathrm{M}+\mathrm{P}\), and
\[
\operatorname{rank}(A)=M \quad \text { and } \quad \operatorname{rank}(A B)=N .
\]

U nder these assum ptions, the constrained equation is alw ays consistent, and there is a unique solution x and a m inim al 2-norm solution \(y\), which is obtained using a generalized \(Q R\) factorization of \(A\) and \(B\).

In particular, ifm atrix \(B\) is square nonsingular, then the problem G LM is equivalent to the follow ing w eighted linear least squares problem
\(m\) iṅm ize \(\left\|\operatorname{inv}(B)^{\star}\left(d-A *_{x}\right)\right\| 2\)
x
where inv ( \(B\) ) denotes the inverse of \(B\).

\section*{ARGUMENTS}

N (input) The num ber of row sof the m atrioes A and \(\mathrm{B} . \mathrm{N}>=\) 0.

M (input) The num ber of colum ns of the m atrix \(\mathrm{A} .0<=\mathrm{M}<=\) N .
\(P\) (input) The num ber of colum ns of the \(m\) atrix \(B . P>=N-M\).

A (input/output)
O n entry, the \(N\) boy \(-M m\) atrix A. On exit, A is des troyed.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

B (input/output)
On entry, the \(N\) boy P m atrix B. On exit, B is destroyed.

LD B (input)
The leading dim ension of the aray \(\mathrm{B} . \mathrm{LD} \mathrm{B}>=\) \(\max (1, N)\).

D (input/output)
O \(n\) entry, \(D\) is the lefthand side of the G LM equation. On exit, D is destroyed.

X (output)
On exit, \(X\) and \(Y\) are the solutions of the GLM problem.

Y (output)
See the description ofX .

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK ( 1 ) retums the optim al
LDW ORK.

LDW ORK (input)
The dimension of the array W ORK. LDW ORK \(>=\) \(m\) ax \((1, N+M+P)\). Foroptim um perform ance, LD W ORK \(>=\) \(M+m\) in \((\mathbb{N}, P)+m\) ax \((\mathbb{N}, P) \star N B\), where \(N B\) is an upperbound for the optim al blocksizes forSGEQRF, SGERQF, SORM QR and SORMRQ .

If LD W ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim al size of
the W ORK aray, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LD W ORK is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit.
\(<0\) : if \(\mathbb{N F O}=-i\), the \(i-\) th argum enthad an ille-
galvalue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dgghrd -reduce a pair of realm atrices ( \(A, B\) ) to generalized upper H essenberg form using orthogonal transform ations, \(w\) here \(A\) is a generalm atrix and \(B\) is upper triangular

\section*{SYNOPSIS}

```

    Z,LDZ,INFO)
    CHARACTER * 1 COMPQ,COMPZ
\mathbb{NTEGERN, ILO,\mathbb{H I,LDA,LDB,LDQ,LD Z,INFO}}\mathbf{L},L
DOUBLE PRECISION A (LDA,*),B (LDB,*),Q (LDQ ,*),Z (LD Z ,*)

```

```

    LDQ,Z,LD Z,INFO)
    CHARACTER * 1 COMPQ,COMPZ
\mathbb{NTEGER*8N,\mathbb{LO},\mathbb{H}I,LDA,LDB,LDQ,LD Z,INFO}
DOUBLE PRECISION A (LDA *),B (LDB,*),Q (LDQ ,*),Z (LD Z,*)

```

\section*{F95 INTERFACE}

SU BROUTINE GGHRD (COMPQ,COMPZ, \(\mathbb{N}], \mathbb{H}, \mathbb{H} I, A,[L D A], B,[L D B], Q\), [ LD Q\(], \mathrm{Z},[\mathrm{LD} \mathrm{Z}],[\mathbb{N} F \mathrm{O}])\)

CHARACTER (LEN=1): :COM PQ,COM PZ
\(\mathbb{N} T E G E R:: N, \mathbb{N}, \mathbb{H} I, L D A, L D B, L D Q, L D Z, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A, B, \(Q, Z\)
SU BROUTINE GGHRD_64 (COMPQ,COMPZ, \(\mathbb{N}], \mathbb{H O}, \mathbb{H} I, A,[L D A], B,[L D B]\), \(Q,[L D Q], Z,[L D Z],[\mathbb{N F O}])\)

CHARACTER (LEN=1) ::COMPQ,COM PZ
\(\mathbb{N}\) TEGER (8) :: N , \(\mathbb{L O}, \mathbb{H} \mathrm{I}, \mathrm{LD} A, L D B, L D Q, L D Z, \mathbb{N} F O\) REAL (8), D \(\mathbb{M}\) ENSION (: : : : ::A, B, \(Q, Z\)

\section*{C INTERFACE}
\#include < sunperfh>
void dgghrd (charcom pq, charcom pz, intn, int ilo, int ini, double *a, int lda, double *b, int ldb, double *q, int ldq, double *z, int ldz, int *info);
void dgghrd_64 (charcom pq, char com pz, long n, long ilo, long ihi, double *a, long lda, double *b, long ldb, double *q, long ldq, double *z, long ldz, long *info);

\section*{PURPOSE}
dgghrd reduces a pair of realm atrioes \((A, B)\) to generalized upper Hessenberg form using orthogonal transform ations, \(w\) here \(A\) is a generalm atrix and \(B\) is uppertriangular: \(Q\) '* \(A * Z=H\) and \(Q ' * B * Z=T\), where \(H\) is upper \(H\) essenberg, \(T\) is uppertriangular, and \(Q\) and \(Z\) are orthogonal, and ' \(m\) eans transpose.

The orthogonalm atrices \(Q\) and \(Z\) are determ ined as products of \(G\) ivens rotations. They \(m\) ay eitherbe form ed explicitly, or they \(m\) ay be postm ultiplied into inputm atrices Q 1 and \(\mathrm{Z1}\), so that
\(1 * A * Z 1{ }^{\prime}=\left(Q 1^{*}\right)^{*} H^{*}(Z 1 * Z)^{\prime}\)

\section*{ARGUMENTS}

COMPQ (input)
\(=\mathrm{N}\) ': do not com pute Q ;
\(=I^{\prime}: \mathrm{Q}\) is initialized to the unit m atrix, and the orthogonalm atrix \(Q\) is retumed; \(=V\) : \(: Q \mathrm{~m}\) ust contain an orthogonalm atrix Q 1 on entry, and the product \(\mathrm{Q} \mathrm{I}^{*} \mathrm{Q}\) is retumed.

COMPZ (input)
\(=\mathrm{N}\) ': do notcom pute Z ;
\(=I^{\prime}: Z\) is initialized to the unit \(m\) atrix, and the orthogonalm atrix Z is retumed; = \(\mathrm{V}^{\prime}: \mathrm{Z} \mathrm{m}\) ust contain an orthogonalm atrix Z1 on entry, and the product Z1*Z is retumed.

N (input) The order of the m atrices A and \(\mathrm{B} . \mathrm{N}>=0\).

ㅍO (input)
It is assum ed thatA is already upper triangular
in row sand colum ns \(1: \mathbb{H O}-1\) and \(\mathbb{H} \mathrm{I}+1 \mathbb{N} . \mathbb{I} \mathrm{O}\) and
Hi I are norm ally setby a previous call to SG G BA L; otherw ise they should be setto 1 and \(N\) respec-
tively. \(1<=\mathbb{H O}<=\mathbb{H} I<=N\), if \(N>0 ; \mathbb{H O}=1\) and \(\mathbb{H} \mathrm{I}=0\), \(\dot{\text { if }} \mathrm{N}=0\).

IH I (input)
See the description of IIO .
A (input/output)
On entry, the N -by -N generalm atrix to be reduced.
On exit, the upper triangle and the first subdiagonalofA are overw ritten w th the upper \(H\) essenberg \(m\) atrix \(H\), and the rest is set to zero.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

B (input/output)
On entry, the \(\mathrm{N}-\) by -N upper triangular \(m\) atrix B .
On exit, the upper triangularm atrix \(T=Q\) ' Z . The elem ents below the diagonal are set to zero.

LD B (input)
The leading dim ension of the array \(B\). LD B \(>=\) \(\max (1, N)\).

Q (input/output)
If \(C O M P Q=N\) : \(Q\) is not referenced.
If COM PQ = 'I': on entry, Q need notbe set, and on exitit contains the orthogonalm atrix \(Q\), where \(Q\) ' is the product of the G ivens transform ations which are applied to \(A\) and \(B\) on the left. IfCOM \(P Q=V\) ': on entry, Q m ust contain an orthogonalm atrix Q 1 , and on exit this is overw rilten by \(Q 1 * Q\).

LD Q (input)
The leading dim ension of the array \(Q . L D Q>=N\) if \(\mathrm{COMPQ}=\mathrm{V}\) 'or I '; LD Q >= 1 otherw ise.

Z (input/output)
If \(C O M P Z=N^{\prime}: Z\) is not referenced.
If COM PZ= I': on entry, Z need notbe set, and on exititcontains the orthogonalm atrix \(Z\), which is the product of the G ivens transform ations which
are applied to \(A\) and \(B\) on the right. If \(\mathrm{COMPZ=V}\) : on entry, Z m ustcontain an orthogonal \(m\) atrix Z1, and on exit this is overw rilten by Z1*Z.

LD \(Z\) (input)
The leading dim ension of the array \(Z\). LD \(Z>=N\) if \(C O M P Z=V\) 'or I'; LD Z >= 1 otherw ise.
\(\mathbb{N F O}\) (output)
= 0: successfulexit.
<0: if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvahue.

\section*{FURTHER DETAILS}

This routine reduces \(A\) to \(H\) essenberg and \(B\) to triangular form by an unblocked reduction, as described in _M atrix_C om putations_, by G olub and V an Loan (Johns H opkins Press.)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgglse - solve the linearequality-constrained least squares (LSE) problem

\section*{SYNOPSIS}
```

SU BROUTINEDGGLSE M ,N,P,A,LDA,B,LDB,C,D,X,W ORK,LDW ORK,
\mathbb{NFO)}
INTEGERM,N,P,LDA,LDB,LDW ORK,\mathbb{NFO}
DOUBLE PRECISION A (LDA,*), B (LDB,*), C (*), D (*), X (*),
W ORK (*)
SUBROUT\mathbb{NEDGGLSE_64M,N,P,A,LDA,B,LDB,C,D,X,W ORK,LDW ORK,}
INFO )
\mathbb{NTEGER*8M,N,P,LDA,LDB,LDW ORK, NNFO}
DOUBLE PRECISION A (LDA,*), B (LDB,*), C (*), D (*), X (*),
W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GGLSE (M) \(\mathbb{M} \mathbb{N}],[\mathbb{P}], A,[L D A], B,[L D B], C, D, X,[\mathbb{O} O R K]\), [LDW ORK], [ \(\mathbb{N} F O]\) )
\(\mathbb{N}\) TEGER :: M , N, P,LDA,LDB,LDW ORK, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::C,D , X,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B
SU BROUTINE GGLSE_64 ( \(\mathbb{M}], \mathbb{N}], \mathbb{P}], A,[L D A], B,[L D B], C, D, X,[\mathbb{O}\) ORK ],
[LDW ORK], [ \(\mathbb{N} F O]\) )
\(\mathbb{N} T E G E R(8):: M, N, P, L D A, L D B, L D W O R K, \mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) :: C,D , X,W ORK

REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A, B

\section*{C INTERFACE}
\#include <sunperfh>
void dgglse (intm , intn, intp, double *a, intlda, double
*b, int ldlo, double * c , double *d, double *x, int *info);
void dgglse_64 (long m , long n, long p, double *a, long lda, double *b, long ldb, double *c, double *d, double *x, long *info);

\section*{PURPOSE}
dgglse solves the linear equality-constrained least squares (LSE ) problem :
\(m\) inim ize \(\left\|C-A *_{x}\right\| 2\) subject to \(B *_{x}=d\)
where \(A\) is an \(M\) boy \(N m\) atrix, \(B\) is a \(P\) boy \(N m\) atrix, \(c\) is a given M -vector, and d is a given P -vector. 化 is assum ed that
\(\mathrm{P}<=\mathrm{N}<=\mathrm{M}+\mathrm{P}\), and
\(\operatorname{rank}(B)=P\) and \(\operatorname{rank}((A))=N\).
( (B ) )

These conditions ensure that the LSE problem has a unique solution, which is obtained using a G RQ factorization of the \(m\) atrices \(B\) and A.

\section*{ARGUMENTS}

M (input) The num ber of row s of the matrix \(A . M>=0\).

N (input) The num ber of colum ns of the \(m\) atrices \(A\) and \(B . N\) \(>=0\).

P (input) The num ber of row s of them atrix B. \(0<=\mathrm{P}<=\mathrm{N}<=\) \(\mathrm{M}+\mathrm{P}\) 。

A (input/output)
On entry, the M boy-N m atrix A. On exit, A is destroyed.

LD A (input)

The leading dimension of the array A. LD A >= \(\max (1, M)\).

B (input/output)
On entry, the \(P\)-by \(-N\) m atrix B. On exit, \(B\) is destroyed.

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, P)\).
\(C\) (input/output)
On entry, \(C\) contains the right hand side vector for the leastsquares part of the LSE problem. On exit, the residual sum of squares for the solution is given by the sum of squares of elem ents \(\mathrm{N}+\mathrm{P}+1\) to \(M\) ofvector \(C\).

D (input/output)
On entry, D contains the right hand side vector for the constrained equation. On exit, \(D\) is destroyed.

X (output)
On exit, \(X\) is the solution of the LSE problem .

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the array W ORK. LDW ORK >= \(m a x(1, M+N+P)\). For optim um perform ance LDW ORK >= \(P+m\) in \((M, N)+m\) ax \(M, N) * N B\), where \(N B\) is an upperbound for the optim al blocksizes forSGEQRF, SGERQF, SORM QR and SORMRQ.

IfLDW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit.
<0: if \(\mathbb{N N}\) FO = -i, the i-th argum ent had an illegalvahue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
```

dggqrf-com pute a generalized Q R factorization of an N -by -M

```
\(m\) atrix \(A\) and an \(N\) by \(P m\) atrix \(B\).

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGGQRFN,M,P,A,LDA,TAUA,B,LDB,TAUB,W ORK,LW ORK,}
\mathbb{NFO)}

```
\(\mathbb{N}\) TEGER \(N, M, P, L D A, L D B, L W O R K, \mathbb{N} F O\)
DOUBLE PRECISION A (LDA, \(\left.{ }^{\star}\right)\), TAUA (*), B (LDB, \(\left.{ }^{\star}\right)\), TAUB (*),
W ORK (*)
SU BROUTINEDGGQRF_64 \(\mathbb{N}, \mathrm{M}, \mathrm{P}, \mathrm{A}, \mathrm{LDA}, \mathrm{TAUA}, \mathrm{B}, \mathrm{LDB}, \mathrm{TA} \mathrm{UB}, \mathrm{W} O R K\),
    LW ORK, \(\mathbb{N} F O\) )
\(\mathbb{N}\) TEGER*8N,M,P,LDA,LDB,LW ORK, \(\mathbb{N}\) FO
DOUBLE PRECISION A (LDA, \(\left.{ }^{\star}\right)\), TAUA (*), B (LDB, \(\left.{ }^{\star}\right)\), TAUB (*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N E}\) GGQRF ( \(\mathbb{N}], \mathbb{M}], \mathbb{P}], A,[L D A], T A U A, B,[L D B], T A U B,[\mathbb{O} O R K]\), [LW ORK ], [ \(\mathbb{N F O}\) ])
\(\mathbb{N} T E G E R:: N, M, P, L D A, L D B, L W O R K, \mathbb{N} F O\)
REAL (8),D \(\mathbb{I}\) ENSION (:) ::TAUA,TAUB,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (: : : : : A, B

SU BROUTINE GGQRF_64 ( \(\mathbb{N}], \mathbb{M}],[\mathbb{P}], A,[L D A], T A U A, B,[L D B], T A U B\), \(\mathbb{W}\) ORK ], [LW ORK ], [ \(\mathbb{N} F \mathrm{~F}\) ])
\(\mathbb{N}\) TEGER (8) ::N,M,P,LDA,LDB,LW ORK, \(\mathbb{N}\) FO

REAL (8), D \(\mathbb{M}\) ENSION (:) ::TAUA,TAUB,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void dggqnf(intn, intm, intp, double *a, int lda, double *taua, double *b, int lalb, double *taub, int *info);
void dggqnf_ 64 (long \(n\), long \(m\), long p, double *a, long lda, double *taua, double *b, long lalo, double *taub, long *info);

\section*{PURPOSE}
dggquf com putes a generalized Q \(R\) factorization of an \(N\) toy \(M\) \(m\) atrix \(A\) and an \(N\) boy \(P\) m atrix B :
\[
\mathrm{A}=\mathrm{Q} * \mathrm{R}, \quad \mathrm{~B}=\mathrm{Q} * \mathrm{~T} * \mathrm{Z},
\]
where \(Q\) is an \(N\) boy -N orthogonal matrix, \(Z\) is a \(P-b y-P\) orthogonalm atrix, and \(R\) and \(T\) assum e one of the form S :
```

ifN >= M , R = (R11)M , orifN < M , R = (R11 R12

``` ) N ,
( 0 ) N M
N \(\mathrm{M}-\mathrm{N}\)
M
w here R11 is upper triangular, and
if \(\mathrm{N}<=\mathrm{P}, \mathrm{T}=(0 \mathrm{~T} 12) \mathrm{N}\), orif \(\mathrm{N}>\mathrm{P}, \mathrm{T}=\) ( T 11 )
\(\mathrm{N} P\),
\[
\mathrm{P}-\mathrm{N} N \quad \text { (T21)P }
\]

P
w here T12 orT21 is upper triangular.

In particular, if \(B\) is square and nonsingular, the GQR factorization ofA and \(B\) im plicitly gives the \(Q R\) factorization of inv ( \(B\) ) *A:
\[
\operatorname{inv}(B) \star A=Z \star(\operatorname{inv}(T) \star R)
\]
where inv ( \(B\) ) denotes the inverse of the \(m\) atrix \(B\), and \(Z^{\prime}\) denotes the transpose of the \(m\) atrix \(Z\).

\section*{ARGUMENTS}

N (input) The num ber of row s of the m atrioes A and B. N >=
0.

M (input) The num ber of collm ns of the \(m\) atrix \(A . M>=0\).
\(P\) (input) The num ber of colum ns of the \(m\) atrix \(B . P>=0\).
A (input/output)
On entry, the \(N-b y-M m\) atrix A. On exit, the ele\(m\) ents on and above the diagonalof the aray contain the \(m\) in \((\mathbb{N}, M)-b y-M\) uppertrapezoidalm atrix \(R\) \((R\) is upper triangular if \(N>=M\) ); the elem ents below the diagonal, w ith the array TA U A , represent the orthogonal matrix \(Q\) as a productofm in \((\mathbb{N}, M\) ) elem entary reflectors (see FurtherD etails).

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

TAUA (output)
The scalar factors of the elem entary reflectors which represent the orthogonal matrix \(Q\) (see FurtherD etails).

B (input/output)
On entry, the \(N\) by P m atrix B. On exit, if \(N\) <= \(P\), the upper triangle of the subarray \(B(1 \mathbb{N}, P-\)
\(\mathrm{N}+1 \mathrm{P}\) ) contains the N -by N upper triangularm atrix
T ; if \(\mathrm{N}>\mathrm{P}\), the elem ents on and above the \((\mathbb{N}-\mathrm{P})\) th subdiagonal contain the N -by P upper trapezoidal \(m\) atrix \(T\); the rem aining elem ents, \(w\) ith the array TAUB, represent the orthogonalm atrix Z as a productofelem entary reflectors (see Further D etails).

LD B (input)
The leading dim ension of the array \(\mathrm{B} . \mathrm{LD} \mathrm{B}>=\) \(\max (1, N)\).

TAUB (output)
The scalar factors of the elem entary reflectors which represent the orthogonal matrix Z (see FurtherD etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, \mathrm{~W}\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the array W ORK. LW ORK >= \(\max (1, N, M, \mathbb{P})\). For optim um perform ance LW ORK \(>=\) \(\max (\mathbb{N}, \mathrm{M}, \mathbb{P}) * \max (\mathbb{N} 1, N B 2, N B 3)\), where \(N B 1\) is the optim al blocksize forthe \(Q R\) factorization of an N -by -M m atrix, NB2 is the optim al blocksize for the RQ factorization of an \(N\)-by \(P m\) atrix, and NB3 is the optim alblocksize for a callof \(S O R M Q R\).

IfLW O RK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue.

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(1) H(2) \ldots H(k), \text { where } k=m \text { in }(n, m) .
\]

Each H (i) has the form
H (i) \(=I-\operatorname{tana} * V^{*} V^{\prime}\)
where taua is a real scalar, and \(v\) is a realvectorw ith \(v(1: i-1)=0\) and \(v(i)=1\); \(v(i+1 n)\) is stored on exit in A (i+1 \(n, i)\), and taua in TAUA (i).
To form \(Q\) explicitly, use LAPACK subroutine \(S O R G Q R\). To use \(Q\) to update another matrix, use LAPACK subroutine SORMQR.

Them atrix \(Z\) is represented as a product of elem entary reflectors
\[
Z=H(1) H(2) \ldots H(k), w h e r e k=m \text { in }(n, p) .
\]

Each H (i) has the form
\[
H(i)=I-\operatorname{tanb} * V^{*} V^{\prime}
\]
where taub is a real scalar, and \(v\) is a realvectorw ith \(v(p-k+i+1: p)=0\) and \(v(p-k+i)=1 ; v(1: p-k+i-1)\) is stored on
exitin \(B(n-k+i, 1 p-k+i-1)\), and taub in TA UB (i).
To form Z explicitly, use LAPACK subroutine SORGRQ.
To use \(Z\) to update another \(m\) atrix, use LAPACK subroutine SORMRQ.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dggrof - com pute a generalized \(R Q\) factorization of an \(M\)-by \(-\mathbb{N}\) \(m\) atrix \(A\) and \(a-b y-N\) m atrix B

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGGRQFM,P,N,A,LDA,TAUA,B,LDB,TAUB,W ORK,LW ORK,}
\mathbb{NFO)}

```
\(\mathbb{N}\) TEGERM, \(\mathrm{P}, \mathrm{N}\), LDA, LD \(B, L W O R K, \mathbb{N} F O\)
DOUBLE PRECISION A (LDA, \(\left.{ }^{\star}\right)\), TAUA (*), B (LDB, \(\left.{ }^{\star}\right)\), TAUB (*),
W ORK (*)
SUBROUTINEDGGRQF_64M, P,N,A,LDA,TAUA,B,LDB,TAUB,WORK,
    LW ORK, \(\mathbb{N} F O\) )
\(\mathbb{N}\) TEGER*8M, \(\mathrm{P}, \mathrm{N}, \mathrm{LDA}, \mathrm{LD} B, L W\) ORK, \(\mathbb{N}\) FO
DOUBLE PRECISION A (LDA, *), TAUA (*), B (LDB,*), TAUB (*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E \operatorname{GGRQF}(\mathbb{M}],[\mathbb{P}], \mathbb{N}], A,[L D A], T A U A, B,[L D B], T A U B,[W O R K]\), [LW ORK], [ \(\mathbb{N F O}\) ])
\(\mathbb{N} T E G E R:: M, P, N, L D A, L D B, L W O R K, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::TAUA,TAUB,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A , B

SU BROUTINE GGRQF_64 (M) \(\mathbb{M}, \mathbb{P}], \mathbb{N}], A,[L D A], T A U A, B,[L D B], T A U B\), \(\mathbb{W}\) ORK ], [LW ORK ], [ \(\mathbb{N F O}]\) )
\(\mathbb{N}\) TEGER (8) ::M,P,N,LDA,LDB,LW ORK, \(\mathbb{N}\) FO

REAL (8), D \(\mathbb{M}\) ENSION (:) ::TAUA,TAUB,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void dggrqf(intm, intp, intn, double *a, int lda, double
*taua, double *b, int ldlo, double *taub, int
*info);
void dggrqf_64 (long m, long p, long n, double *a, long lda, double *taua, double *b, long lalo, double *taub, long *info);

\section*{PURPOSE}
dggrof com putes a generalized RQ factorization of an \(M\) boy -N \(m\) atrix \(A\) and a \(P-b y-N m\) atrix \(B\) :
\[
\mathrm{A}=\mathrm{R} * \mathrm{Q}, \quad \mathrm{~B}=\mathrm{Z} * \mathrm{~T} * \mathrm{Q},
\]
where \(Q\) is an \(N\) boy -N orthogonal matrix, \(Z\) is a \(P-b y-P\) orthogonalm atrix, and \(R\) and \(T\) assum e one of the form \(s\) :
```

ifM <=N, R = (O R12)M, orifM > N, R = ( R11 )
M -N ,
N-M M (R21)N
N

```
w here R12 orR21 is uppertriangular, and
```

ifP >= N, T=(T11)N , orifP < N, T = (T11 T12

```
) P,
\((0) P-N\)
\(N\)\(\quad P \quad N P\)
w here T11 is upper triangular.

In particular, ifB is square and nonsingular, the GRQ factorization ofA and \(B\) im plicitly gives the \(R Q\) factorization of \(A\) *inv ( \(B\) ):
\[
A * \operatorname{inv}(B)=(R * \operatorname{inv}(T)) * Z^{\prime}
\]
where inv \((B)\) denotes the inverse of the \(m\) atrix \(B\), and \(Z^{\prime}\) denotes the transpose of the \(m\) atrix \(Z\).

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
\(P\) (input) The num ber of row sof the \(m\) atrix \(B . P>=0\).
N (input) The num ber of colum ns of the \(m\) atrices \(A\) and \(B . N\) \(>=0\).

A (input/output)
On entry, the \(\mathrm{M}-\mathrm{by}-\mathrm{N} \mathrm{m}\) atrix A. On exit, if \(\mathrm{M}<=\)
N , the upper triangle of the subarray A ( \(1 \mathrm{M}, \mathrm{N}\) -
\(\mathrm{M}+1 \mathrm{~N}\) ) contains the M -by -M uppertriangularm atrix
\(R\); if \(M>N\), the elem ents on and above the ( \(M-N\) )th subdiagonal contain the M -by -N upper trapezoidal \(m\) atrix \(R\); the rem aining elem ents, \(w\) ith the array TAUA, represent the orthogonalm atrix \(Q\) as a product ofelem entary reflectors (see Further D etails).

LD A (input)
The leading dim ension of the array A. LDA >= \(m a x(1, M)\).

TAUA (output)
The scalar factors of the elem entary reflectors which represent the orthogonal matrix \(Q\) (see FurtherD etails).

B (input/output)
On entry, the \(P-\) by -N m atrix B. On exit, the ele\(m\) ents on and above the diagonalof the array contain the \(m\) in \((P, N)-b y-N\) uppertrapezoidalm atrix \(T\) ( \(T\) is upper triangular if \(P>=N\) ); the elem ents below the diagonal, w ith the amay TA UB, represent the orthogonalm atrix Z as a productof elem entary reflectors (see FurtherD etails).

LD B (input)
The leading dim ension of the array \(B . L D B>=\) \(\max (1, P)\).

TAUB (output)
The scalar factors of the elem entary reflectors which represent the orthogonal matrix Z (see FurtherD etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, \mathrm{~W}\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the array W ORK. LW ORK >= \(\max (1, N, M, P)\). For optim um perform ance LW ORK \(>=\) \(\max (\mathbb{N}, \mathbb{M}, \mathbb{P}){ }^{\star} \max (\mathbb{N} 1, N B 2, N B 3)\), where NB1 is the optim al blocksize forthe \(R Q\) factorization of an \(M\) by \(-N m\) atrix, NB2 is the optim al blocksize for the \(Q R\) factorization of \(a P-b y-N m\) atrix, and NB3 is the optim alblocksize for a callof \(S O R M R Q\).

IfLW O RK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{N} F 0=-\) i, the \(i\)-th argum enthad an illegalvalue.

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(1) H(2) \ldots H(k), \text { where } k=m \text { in }(m, n) .
\]

Each H (i) has the form
H (i) \(=I-\operatorname{tana} * V^{*} V^{\prime}\)
w here taua is a real scalar, and \(v\) is a realvectorw ith \(v(n-k+i+1 m)=0\) and \(v(n-k+i)=1 ; v(1 n-k+i-1)\) is stored on exitin \(A(m-k+i, 1 m-k+i-1)\), and taua in TAUA (i).
To form \(Q\) explicitly, use LAPACK subroutine SORGRQ. To use \(Q\) to update another \(m\) atrix, use LAPACK subroutine SORMRQ.

Them atrix Z is represented as a product of elem entary reflectors
\[
Z=H(1) H(2) \ldots H(k), w \text { here } k=m \text { in }(p, n) .
\]

Each H (i) has the form
\[
H(i)=I-\operatorname{tanb} * V^{*} V^{\prime}
\]
where taub is a real scalar, and \(v\) is a realvectorw ith \(\mathrm{v}(1: i-1)=0\) and \(\mathrm{v}(\mathrm{i})=1\); \(\mathrm{v}(\mathrm{i}+1 \mathrm{p})\) is stored on exit in

B (i+1 p,i), and taub in TA UB (i).
To form Z explicitly, use LAPACK subroutine \(\operatorname{SORGQR\text {.}}\)
To use \(Z\) to update another \(m\) atrix, use LAPACK subroutine SORMQR.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dggsvd -com pute the generalized singular value decom position ( \(G\) SVD) of an \(M\) by \(-N\) real \(m\) atrix \(A\) and \(P-b y-N\) real \(m\) atrix B

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE D G G SVD (OOBU,JOBV,NOBQ,M ,N,P,K,L,A,LDA ,B,LD B,}
A LPHA,BETA,U,LDU,V,LDV,Q,LDQ,W ORK,IN ORK 3, INFO)

```
CHARACTER * 1 JOBU, JOBV , JOBQ
\(\mathbb{N}\) TEGER \(M, N, P, K, L, L D A, L D B, L D U, L D V, L D Q, \mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{I}\) ORK 3 (*)
DOUBLE PRECISION A (LDA,*), B (LDB,*), ALPHA (*), BETA (*),
U (LDU,*), V (LDV,*), Q (LDQ ,*),W ORK (*)
SU BROUTINEDGGSVD_64(JOBU, \(\mathrm{JOBV}, \mathcal{O B Q}, \mathrm{M}, \mathrm{N}, \mathrm{P}, \mathrm{K}, \mathrm{L}, \mathrm{A}, \mathrm{LD} A, B, L D B\),
    A LPHA, BETA, U, LDU, \(V, L D V, Q, L D Q, W O R K, \mathbb{I N} O R K 3, \mathbb{N} F O)\)
CHARACTER * 1 JOBU, 0 BV , JOBQ
\(\mathbb{N}\) TEGER*8 M , N, P, K, L, LD A ,LD B, LD U ,LDV ,LD Q , \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R * 8 \mathbb{I N}\) ORK 3 (*)
DOUBLE PRECISION A (LDA,*), B (LDB,*), ALPHA (*), BETA (*),
\(\mathrm{U}(\mathrm{LD} \mathrm{U}, \star), \mathrm{V}(\mathrm{LDV}, \star), \mathrm{Q}(\mathrm{LD} \mathrm{Q}, \star), \mathrm{W} O R K(\star)\)

\section*{F95 INTERFACE}
 [LD B],A LPHA, BETA, U , [LDU ], V , [LDV], Q, [LD Q], [W ORK], IV ORK 3, [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1)::JOBU ,JOBV ,JOBQ
\(\mathbb{N} T E G E R:: M, N, P, K, L, L D A, L D B, L D U, L D V, L D Q, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{I N}\) ORK 3

REAL (8), D \(\mathbb{M} E N S I O N(:):: A L P H A, B E T A, W\) ORK
REAL (8), D \(\mathbb{M}\) ENSION (: : : : : : A , B , U , V , Q

SU BROUTINE G G SVD_64 (JO BU , JO BV , JO BQ, \(\mathbb{M}], \mathbb{N}],[P], K, L, A,[L D A]\), \(B,[L D B], A L P H A, B E T A, U,[L D U], V,[L D V], Q,[L D Q],[W O R K]\), \(\mathbb{I} W\) ORK \(3,[\mathbb{N} F O]\) )

CHARACTER (LEN=1) :: JOBU, OBV, OBQ
\(\mathbb{N}\) TEGER (8) :: M , N , P , K , L, LD A , LD B , LD U , LD V , LD Q , \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{I N} O R K 3\)
REAL (8), D \(\mathbb{I}\) ENSION (:) ::ALPHA,BETA, WORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A \(, \mathrm{B}, \mathrm{U}, \mathrm{V}, \mathrm{Q}\)

\section*{C INTERFACE}
\#include <sunperfh>
void dggsvd (char jंbu, char jंbv, char jobq, intm, int n, int \(p\), int *k, int * \(l\), double *a, int lda, double *b, int ldb, double *alpha, double *beta, double
\({ }^{*} u\), int ldu, double \({ }^{*}\), int ldv, double * \(q\), int ldq, int *iw ork3, int *info);
void dggsvd_64 (char jंbu, char jंbv, char jंbq, long m, long n, long p, long *k, long *l, double *a, long lda, double *b, long ldb, double *alpha, double *beta, double *u, long ldu, double *v, long ldv, double *q, long ldq, long *íw ork3, long *info);

\section*{PURPOSE}
dggsvd com putes the generalized singular value decom position (G SV D ) of an \(M\) boy \(-N\) realm atrix \(A\) and \(P\) boy \(-N\) realm atrix \(B\) :
\[
U{ }^{*} A * Q=D 1^{*}(0 R), \quad V * B * Q=D 2^{*}(0 R)
\]
where \(\mathrm{U}, \mathrm{V}\) and Q are orthogonal m atriges, and Z ' is the transpose of \(Z\). LetK \(+L=\) the effective num erical rank of them atrix ( \(A\) ', \(B)^{\prime}\) ', then \(R\) is a \(K+L\) boy \(K+L\) nonsingular upper triangularm atrix, \(D 1\) and \(D 2\) are \(M-b y-(K+L)\) and \(P-b y-\) \((\mathbb{K}+\mathrm{L})\) "diagonal" \(m\) atrices and of the follow ing structures, respectively:

IfM \(-\mathrm{K}-\mathrm{C}=0\),
\begin{tabular}{|c|}
\hline \multirow[t]{4}{*}{} \\
\hline \\
\hline \\
\hline \\
\hline
\end{tabular}

K L
```

    D2 = L (0 S )
    P-工 (0 0)
        N-K\dashv K L
    (0R ) = K (0 R11 R12 )
L(0 0 R22)

```
where
\[
\begin{aligned}
& C=\operatorname{diag}(A L P H A(K+1), \ldots, \text { A LPH }(\mathbb{K}+L)), \\
& S=\operatorname{diag}(\operatorname{BETA}(\mathbb{K}+1), \ldots, \operatorname{BETA}(\mathbb{K}+L)), \\
& C \star * 2+S * * 2=I .
\end{aligned}
\]
\(R\) is stored in \(A(1: K+L, N-K-1 \mathbb{N})\) on exit.

IfM \(\mathrm{K}-\mathrm{L}<0\),
\[
\begin{aligned}
& \text { K M K K + L } \mathrm{M} \\
& D 1=K\left(\begin{array}{ll}
I & 0
\end{array}\right) \\
& M K\left(\begin{array}{lll}
0 & C
\end{array}\right) \\
& \text { K M K K + L }-\mathrm{M} \\
& D 2=M K(0 S 0) \\
& \mathrm{K}+\mathrm{L} \mathrm{M} \text { ( } 0 \text { O I ) } \\
& \text { P }-\left(\begin{array}{lll}
0 & 0
\end{array}\right)
\end{aligned}
\]
```

                N-K\dashv K M K K +L M
    (0R ) = K (0 R11 R12 R13 )
M K (0 0 R22 R23 )
K+L-M (0 0 0 R33)

```
where
```

$C=\operatorname{diag}(A L P H A(K+1), \ldots, A L P H A(M))$,
$S=\operatorname{diag}(B E T A(K+1), \ldots, B E T A(M))$,
$C * * 2+S * * 2=I$.

```
(R11 R12 R13) is stored in A ( \(1 \mathrm{M}, \mathrm{N} \mathrm{K}-\mathrm{L}+1 \mathbb{N}\) ), and R 33 is stored
(0 R 22 R 23 )
in \(B(M-1: L, N+M-K+1 \mathbb{N})\) on exit.

The routine com putes \(C, S, R\), and optionally the orthogonal transform ation \(m\) atrices \(U, V\) and \(Q\).

In particular, if B is an N boy N nonsingular m atrix, then the G SVD ofA and B im plicitly gives the SVD ofA *inv \((B)\) :
\[
A * \operatorname{inv}(B)=U *(D 1 * \operatorname{inv}(D 2)) * V{ }^{\prime} .
\]

If ( \(A\) 'B )'has orthonorm alcolum ns, then the G SVD of A and \(B\) is also equal to the \(C S\) decom position of \(A\) and B.Further-
m ore, the G SVD can be used to derive the solution of the eigenvalue problem :

A *A \(\mathrm{x}=\operatorname{lam}\) bda* B *B x .
In som e literature, the G SVD of \(A\) and \(B\) is presented in the form

U *A * \(\mathrm{X}=(0 \mathrm{D} 1), \quad \mathrm{V}{ }^{*} \mathrm{~B} * \mathrm{X}=(0 \mathrm{D} 2)\)
\(w\) here \(U\) and \(V\) are orthogonal and \(X\) is nonsingular, \(D 1\) and \(D 2\) are "diagonal". The form erG SVD form can be converted to the latter form by taking the nonsingularm atrix X as
\[
\begin{aligned}
X= & Q *\left(\begin{array}{ll}
I & 0
\end{array}\right) \\
& (0 \operatorname{inv}(\mathbb{R})) .
\end{aligned}
\]

\section*{ARGUMENTS}
\(J O B U\) (input)
\(=\mathrm{U}:\) : O rthogonalm atrix U is com puted;
\(=\mathrm{N}\) : U is notcom puted.
JO BV (input)
\(=\mathrm{V}:\) : O rthogonalm atrix V is com puted;
\(=\mathrm{N}: \mathrm{V}\) is not com puted.
\(J O B Q\) (input)
= Q ': Orthogonalm atrix \(Q\) is com puted;
\(=\mathrm{N}\) ': Q is notcom puted.

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrioes A and B. N \(>=0\).
\(P\) (input) The num ber of row sof the \(m\) atrix \(B . P>=0\).

K (output)
On exit, \(K\) and L specify the dim ension of the subblocks described in the Pupose section. \(\mathrm{K}+\mathrm{L}=\) effective num erical rank of (A 'B )'.

L (output)
See the description of K .

A (input/output)
On entry, the M -by-N m atrix A. On exit, A contains the triangularm atrix \(R\), orpartofR. See
Purpose fordetails.

LD A (input)
The leading dim ension of the array A. LDA >= max (1, M).

B (input/output)
On entry, the \(P-b y-N\) m atrix B. On exit, B contains the triangularm atrix \(R\) if \(M K-4<0\). See Purpose fordetails.

LD B (input)
The leading dim ension of the array B. LD A >= \(\max (1, P)\).
ALPHA (output)
On exi, ALPHA and BETA contain the generalized singularvahue pairs of \(A\) and ; A LPHA \((1: K)=1\),
\(\operatorname{BETA}(1: K)=0\), and ifM \(K-\Psi=0\), ALPHA \((\mathbb{K}+1 \mathbb{K}+L)\)
= C,
BETA \((K+1 K+L)=S\), or if \(M-K-0\),
A LPHA \((\mathbb{K}+1 \mathrm{M})=\mathrm{C}, \mathrm{A}\) LPHA \((\mathrm{M}+1: \mathrm{K}+\mathrm{L})=0\)
\(\operatorname{BETA}(K+1 M)=S, \quad B E T A(M+1 \pi+L)=1\) and
A LPHA \((\mathbb{K}+\mathrm{L}+1 \mathbb{N})=0\)
\(\operatorname{BETA}(\mathbb{K}+\mathrm{L}+1 \mathbb{N})=0\)

\section*{BETA (output)}

See the description ofA LPH A .
U (output)
If \(\mathrm{JOBU}=\mathrm{U}\) ', U contains the M -by M orthogonal
\(m\) atrix \(U\). If \(J O B U=N ', U\) is not referenced.

LD U (input)
The leading dim ension of the array \(U\). LD \(U\) >= \(m a x(1, M)\) if \(J 0 B U=U ' ; L D U>=1\) otherw ise.

V (output)
If \(\mathrm{JOBV}=\mathrm{V}\) ', V contains the P -by P orthogonal \(m\) atrix \(V\). If \(J O B V=N ', V\) is not referenced.

LDV (input)
The leading dim ension of the array V . LDV >= \(\max (1, \mathrm{P})\) if \(\mathrm{JOBV}=\mathrm{V} ; \mathrm{LDV}>=1\) otherw ise.

Q (output)
If \(\mathrm{OOBQ}=\mathrm{Q}\) ', Q contains the N by-N orthogonal \(m\) atrix \(Q\). If \(J O B Q=N\) ', \(Q\) is not referenced.

LD Q (input)
The leading dimension of the array \(Q . L D Q>=\) \(\max (1, N)\) if \(J O B Q=Q ; L D Q>=1\) otherw ise.

W ORK (w orkspace)
dim ension \((\max (3 * N, M, P)+N)\)

IW ORK 3 (output)
dim ension \((\mathbb{N})\) On exit, \(\mathbb{I V}\) ORK 3 stores the sorting
inform ation. M ore precisely, the follow ing loop
w ill.sortALPHA for \(I=K+1\), \(m\) in \((M, K+L) ~ s w a p\)
A LPHA (I) and ALPHA ( \(\mathbb{I N}\) ORK 3 (I)) endfor such that
\(A \operatorname{LPHA}(1)>=A \operatorname{LPHA}(2)>=\ldots>=A L P H A(N)\).
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvahue.
\(>0\) : if \(\mathbb{N F O}=1\), the Jacobi-type procedure failed to converge. For further details, see subroutine STG SJA.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dggsvp - com pute orthogonalm atrices \(\mathrm{U}, \mathrm{V}\) and Q such that \(\mathrm{N}-\mathrm{K}-\mathrm{K} \mathrm{L} \mathrm{U} * \mathrm{~A} * \mathrm{Q}=\mathrm{K}\) (0A12A13) if \(\mathrm{M} \mathrm{K}-\mathrm{L}>=0\)

\section*{SYNOPSIS}
```

SU BROUTINE D G G SVP (OOBU,NOBV,NOBQ,M ,P,N,A ,LDA,B,LD B,TOLA,
TOLB,K,L,U,LDU,V,LDV,Q,LDQ,INORK,TAU,W ORK,INFO)
CHARACTER * 1 JOBU,NOBV,NOBQ
\mathbb{NTEGERM,P,N,LDA,LDB,K,L,LDU,LDV,LDQ, INFO}
INTEGER IN ORK (*)
DOUBLE PRECISION TOLA,TOLB
DOUBLE PRECISION A (LDA,*), B (LDB,*), U (LDU,*),V (LDV,*),
Q (LDQ ,*),TAU (*),WORK (*)
SU BROUT\mathbb{NEDGGSVP_64 (JOBU , OOBV,JOBQ,M ,P,N,A,LDA,B,LD B,TOLA,}
TOLB,K,L,U,LDU,V,LDV,Q,LDQ,IN ORK,TAU,W ORK,INFO)

```
CHARACTER * 1 JOBU, JOBV , JOBQ
\(\mathbb{N}\) TEGER*8M, \(\mathrm{P}, \mathrm{N}, \mathrm{LD} A, L D B, K, L, L D U, L D V, L D Q, \mathbb{N} F O\)
\(\mathbb{I N} T E G E R * 8 \mathbb{I V}\) ORK (*)
DOUBLE PRECISION TOLA, TOLB
DOUBLE PRECISION A (LDA,*), B (LDB,\(\left.^{\star}\right)\), U (LDU,\(\left.^{\star}\right), V(L D V, \star)\),
Q (LDQ,\(\left.^{\star}\right), T A U(*), W O R K(*)\)

\section*{F95 INTERFACE}

SU BROUTINE G GSVP (JOBU, \(\mathrm{JOBV}, \mathcal{O B Q}, \mathbb{M}],[\mathbb{P}], \mathbb{N}], A,[L D A], B,[L D B]\), TO LA, TOLB,K,L,U, [LDU],V, [LDV], Q, [LD Q], [IW ORK], [TAU], [W ORK], [ \(\mathbb{N} F O\) ])

\(\mathbb{N}\) TEGER ::M, \(\mathrm{P}, \mathrm{N}, \mathrm{LD} A, L D B, K, L, L D U, L D V, L D Q, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{N}\) ORK
REAL (8) ::TOLA,TOLB
REAL (8), D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A, \(B, U, V, Q\)
SU BROUTINE G GSVP_64 (JOBU, \(\mathcal{J O B V}, \mathcal{J O B Q}, \mathbb{M}], \mathbb{P}], \mathbb{N}], A,[L D A], B\), [LDB],TOLA,TOLB,K,L, U, [LDU], V, [LDV], \(\mathrm{Q},[\mathrm{LDQ}],[\mathbb{F} \mathrm{ORK}]\), [TAU], [W ORK], [NFO])

CHARACTER (LEN =1) :: JO BU , JOBV , 0
\(\mathbb{N}\) TEGER (8) ::M, \(\mathrm{P}, \mathrm{N}, \mathrm{LDA}, \mathrm{LD} B, \mathrm{~K}, \mathrm{~L}, \mathrm{LD} \mathrm{U}, \mathrm{LDV}, \mathrm{LD} Q, \mathbb{N} F \mathrm{O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{I N}\) ORK
REAL (8) ::TOLA,TOLB
REAL (8), D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL (8),D IM ENSION (:,:): : A, B, U, V,Q

\section*{C INTERFACE}
\#include <sunperfh>
void dggsvp (char jobu, char jobv, char jobg, intm, int p, int \(n\), double *a, int lda, double *b, int ldb, double tola, double tolb, int *k, int *l, double *u, int ldu, double *v, int ldv, double * \(q\), int ldq, int*info);
void dggsvp_64 (char jंbu, char j̀.bv, char jंbq, long m, long p, long \(n\), double *a, long lda, double *b, long ldlb, double tola, double tolb, long *k, long *l, double *u, long ldu, double *v, long ldv, double *q, long ldq, long *info);

\section*{PURPOSE}
dggsvp com putes orthogonalm atrices \(\mathrm{U}, \mathrm{V}\) and Q such that
L (0 0 A 23)
\(M\) K- ( \(0 \quad 0 \quad 0 \quad\) )

N-K L K L
\(\left.=\begin{array}{cc}K\left(\begin{array}{ll}0 & A 12 \\ M-K & \text { A } 13\end{array}\right) \text { ifM } K-L<0 ; ~ \\ 0 & 0 \\ \text { A } 23\end{array}\right)\).

N-K K L
\(\mathrm{V}{ }^{*} \mathrm{~B} * \mathrm{Q}=\mathrm{L}\left(\begin{array}{lll}0 & 0 & \mathrm{~B} 13\end{array}\right)\)
\(P-\left(\begin{array}{lll}0 & 0 & 0\end{array}\right)\)
where the K -by \(\mathrm{K} m\) atrix A 12 and L -by- L m atrix B13 are non-
singularuppertriangular; A 23 is L-by- L upper triangular if \(\mathrm{M} \mathrm{K-L} \mathrm{>=0}\),otherw ise A 23 is M K ) -by -L upper trapezoidal. \(K+L=\) the effective num erical rank of the \(M+P)\)-by \(-N \mathrm{~N}\) atrix
(A ', B )'. Z 'denotes the transpose of Z .

This decom position is the preprocessing step for com puting the Generalized Singular V alue D ecom position (GSVD), see subroutine SG G SV D .

\section*{ARGUMENTS}
\(J 0 \mathrm{BU}\) (input)
\(=\mathrm{U}:\) : O rthogonalm atrix U is com puted;
\(=N^{\prime}: U\) is notcom puted.
JO BV (input)
\(=\mathrm{V}\) : O rthogonalm atrix V is com puted;
= N ': V is not com puted.
\(J O B Q\) (input)
\(=Q\) : O rthogonalm atrix Q is com puted;
\(=N^{\prime}: Q\) is notcom puted.
M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
\(P\) (input) The num ber of row sof the \(m\) atrix \(B . P>=0\).

N (input) The num ber of collm ns of the m atrioes A and B. N \(>=0\).

A (input/output)
On entry, the M by-N matrix A. On exit, A contains the triangular (or trapezoidal) matrix described in the Purpose section.

LDA (input)
The leading dim ension of the array A. LDA >= max (1,M).

B (input/output)
On entry, the \(\mathrm{P}-\) by-N m atrix B. On exit, B contains the triangularm atrix described in the Purpose section.

LD B (input)
The leading dim ension of the aray \(B . L D B>=\) \(\max (1, P)\).

TO LA (input)
TOLA and TOLB are the thresholds to determ ine the effective num erical rank ofm atrix B and a subblock ofA. Generally, they are set to TO LA =
\(\operatorname{MAXM}(\mathbb{N}) \star\) norm (A)*MACHEPS, TOLB =
MAX \((P, N) \star\) norm ( \(B\) )*M ACHEPS. The size of TOLA and TO LB \(m\) ay affect the size of backw ard emors of the decom position.

\section*{TO LB (input)}

See the description of TO LA .

K (output)
O n exit, \(K\) and \(L\) specify the dim ension of the subblocks described in Punpose. \(\mathrm{K}+\mathrm{L}=\) effective num ericalrank of (A 'B )'.

L (output)
See the description of .
U (input) If \(\mathrm{OB} \mathrm{BU}=\mathrm{U}\) ', U contains the orthogonalm atrix U . If \(\mathrm{JO} \mathrm{BU}=\mathrm{N}\) ', U is not referenced.

LD U (input)
The leading dim ension of the array \(U\). LD U >= \(\mathrm{max}(1, \mathrm{M})\) if \(\mathrm{JOBU}=\mathrm{U}\) '; LD U >= 1 otherw ise.

V (input) If \(\mathrm{O} \mathrm{BV}=\mathrm{V}\) ', V contains the orthogonalm atrix V . If \(\mathrm{JO} \mathrm{BV}=\mathrm{N}\) ', V is not referenced.

\section*{LD V (input)}

The leading dim ension of the array V . LDV >= \(\max (1, \mathrm{P})\) if \(\mathrm{OOBV}=\mathrm{V}\) '; LDV \(>=1\) otherw ise.
\(Q\) (input) If \(J O B Q=Q\) ', \(Q\) contains the orthogonalm atrix \(Q\). If \(\mathrm{JOBQ}=\mathrm{N}, \mathrm{Q}\) is not referenced.

LD Q (input)
The leading dim ension of the aray \(\mathrm{Q} . \mathrm{LD} \mathrm{Q}\) >= \(\max (1, N)\) if \(J O B Q=Q ; L D Q>=1\) otherw ise.

IN ORK (w orkspace)
dim ension \((\mathbb{N})\)

TAU (w orkspace)
dim ension (N)

W ORK (w orkspace)
dim ension MAX ( \(\left.3{ }^{\star} \mathrm{N}, \mathrm{M}, \mathrm{P}\right)\) )
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvahue.

\section*{FURTHER DETAILS}

The subroutine uses LA PACK subroutine SGEQPF for the QR factorization \(w\) ith \(c o l u m n\) pivoting to detect the effective num erical rank of the a \(m\) atrix. It \(m\) ay be replaced by a better rank determ ination strategy.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
dgssco -G eneral sparse solver condition num berestim ate.

\section*{SYNOPSIS}

SUBROUTINEDGSSCO (COND,HANDLE, \(\mathbb{E R}\) )
\(\mathbb{N} T E G E R \quad \mathbb{E R}\)
DOUBLE PRECISION COND
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

DGSSCO -C ondition num berestim ate.

\section*{PARAMETERS}

COND -DOUBLE PRECISION
On exit, an estim ate of the condition num berof the factored \(m\) atrix. M ustbe called after the num erical factorization subroutine, DGSSFA 0 ).

HANDLE (150) -D OUBLE PREC IS IO N amay
On entry, HANDLE ( \(*\) ) is an array containing inform ation needed by the solver, and \(m\) ustbe passed unchanged to each sparse solver subroutine.
M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Enrornum ber. If no errorencountered, unchanged on
exit. If emorencountered, it is set to a non-zero
integer. Errornum bers setby this subroutine:
-700 : Invalid calling sequence - need to calld G SSFA first.
-710 : C ondition num ber estim ate not available (notim plem ented for this H A N D LE sm atix type).

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
dgssda -D eallocate w orking storage for the general sparse solver.

\section*{SYNOPSIS}

SUBROUTINE ZGSSDA (HANDLE, \(\mathbb{E R}\) )
\(\mathbb{N}\) TEGER \(\quad \mathbb{E R}\)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

ZGSSDA -D eallocate dynam ically allocated w orking storage.

\section*{PARAMETERS}

HANDLE (150) -D OUBLE PREC IS IO N array
On entry, HANDLE (*) is an array containing inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine. M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
E rrornum ber. If no emrorencountered, unchanged on exit. If errorencountered, it is set to a non-zero integer. Errornum bers setby this subroutine:
none

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
dgssfa - \(G\) eneral sparse solvernum eric factorization.

\section*{SYNOPSIS}

SUBROUTINEDGSSFA (NEQNS,COLSTR,ROW \(\mathbb{N D}, V A L U E S, H A N D L E, \mathbb{E R}\) )
\(\mathbb{N} T E G E R \quad\) NEQNS, COLSTR (*),ROW \(\mathbb{N D}(*), \mathbb{E R}\)
DOUBLE PRECISION VALUES (*)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

DGSSFA -N um eric factorization of a sparse m atrix.

\section*{PARAMETERS}

NEQNS - \(\mathbb{N}\) TEGER
On entry, NEQNS specifies the num ber of equations in coefficientm atrix. U nchanged on exit.
\(\operatorname{COLSTR}\left(^{*}\right)-\mathbb{N}\) TEG ER array
On entry, \(\operatorname{COLSTR}\left(^{*}\right)\) is an array of size \((\mathbb{N} E \mathrm{~N}+1\) ), containing the pointers of them atrix structure.
U nchanged on exit.
ROWIND ( \(\left.{ }^{( }\right)-\mathbb{N}\) TEGER array
On entry, ROWIND ( \({ }^{*}\) ) is an array of size
CO LSTR \(\mathbb{N} E Q N S+1)-1\), containing the indices of the \(m\) atrix structure. U nchanged on exit.
\(\operatorname{VALUES}\left(^{*}\right)\)-D OUBLE PREC ISIO N amay
On entry, VALUES (*) is an array of size
CO LSTR \(\mathbb{N E Q N S + 1 ) - 1 , ~ c o n t a i n i n g ~ t h e ~ n u m ~ e r i c ~ v a l u e s ~ o f ~}\)
the sparse \(m\) atrix to be factored. U nchanged on exit.

HANDLE (150) -D OUBLE PRECISIO N array On entry, HANDLE (*) is an array containing inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine. M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Errornum ber. If no emrorencountered, unchanged on exit. If emorencountered, it is set to a non-zero integer. Enrornum bers set.by this subroutine:
-300 : Invalid calling sequence - need to calld G SSO R first.
-301 : Failure to dynam ically allocate \(m\) em ory. -666 : Intemalerror.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
dgssfs -G eneral sparse solver one call interface.

\section*{SYNOPSIS}
```

SUBROUTINEDGSSFS(MTXTYP,PIVOT,NEQNS,COLSTR,ROW IND,
VALUES,NRHS ,RHS ,LDRHS,ORDMTHD,
OUTUNT,MSGLVL,HANDLE,\mathbb{ER)}

```
CHARACTER*2 M TXTYP
CHARACTER*1 PIVOT
\(\mathbb{N}\) TEGER NEQNS,COLSTR (*),ROW \(\mathbb{N} D\left({ }^{*}\right)\),NRHS,LDRHS,
    OUTUNT,MSGLVL, \(\mathbb{E R}\)
CHARACTER*3 ORDMTHD
DOUBLE PRECISION VALUES (*),RHS (*)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

DGSSFS -G eneral sparse solver one call interface.

\section*{PARAMETERS}
```

MTXTYP -CHARACTER*2
On entry, M TX TY P specifies the coefficientm atrix type. Specifically, the valid options are:

```

Sp 'or SP '-sym m etric structure, positive-definite values
ss'or SS '-sym m etric structure, sym m etric values
su'or SU '-sym m etric structure, unsym \(m\) etric values
uu 'or UU '-unsym \(m\) etric structure, unsym \(m\) etric values
U nchanged on exit.

\section*{PIVOT -CHARACTER*1}

On entry, pivot specifies w hether ornotpivoting is used in the course of the num eric factorization.
The valid options are:
h'or N '-no pivoting is used
(Pivoting is not supported forthis release).
U nchanged on exit.

NEQNS - \(\mathbb{N}\) TEGER
On entry, NEQN S specifies the num ber ofequations in the coefficientm atrix. NEQ NS m ustbe at leastone. U nchanged on exit.
\(\operatorname{COLSTR}\left(^{\star}\right)-\mathbb{N}\) TEGER array
On entry, \(\operatorname{COLSTR}\) ( \({ }^{*}\) ) is an array of size \((\mathbb{N} E Q \mathrm{~N}+1\) ), containing the pointers of them atrix structure.
U nchanged on exit.
ROWIND ( \({ }^{\star}\) ) - \(\mathbb{N}\) TEG ER array
On entry, ROWIND ( \({ }^{*}\) ) is an array of size CO LSTR \(\mathbb{N E Q N S + 1 ) - 1 , ~ c o n t a i n i n g ~ t h e ~ i n d i c e s ~ o f ~ t h e ~}\) \(m\) atrix structure. U nchanged on exit.
\(\operatorname{VALUES}\) ( \(^{*}\) ) -D O UBLE PREC ISION aray
O \(n\) entry, VALUES (*) is an aray of size
CO LSTR \(\mathbb{N} E Q N S+1)-1\), containing the non-zero num eric values of the sparse \(m\) atrix to be factored. U nchanged on exit.

NRHS - \(\mathbb{N}\) TEGER
On entry, N RH S specifies the num ber of righthand sides to solve for. U nchanged on exit.

RHS (*) -DOUBLE PRECISION anay
On entry, RHS (LDRHS,NRHS) contains the NRHS right hand sides. On exit, itcontains the solutions.

\section*{LDRHS - \(\mathbb{N}\) TEGER}

On entry, LD RH S specifies the leading dim ension of the RH S array. U nchanged on exit.

ORDMTHD -CHARACTER*3
On entry, ORDM THD specifies the fill-reducing ordering to be used by the sparse solver. Specifically, the valid options are:
hat'or NAT'-naturalordering (no ordering) mmd'or M M D '-m ultiplem inim um degree gnd 'or GND '-generalnested dissection
uso 'or U SO '-user specified ordering (see D G SSU O )
U nchanged on exit.

OUTUNT - \(\mathbb{N}\) TEGER
O utputunit. U nchanged on exit.
M SGLVL - \(\mathbb{N}\) TEGER
M essage level.
0 -no output from solver.
N o m essages supported for this release.)
U nchanged on exit.

HANDLE (150) -D OUBLE PREC IS IO N array
On entry, HANDLE (*) is an array of containing inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine.
M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
E rrornum ber. If no emrorencountered, unchanged on
exit. If errorencountered, it is set to a non-zero
integer. Enrornum bers setby this subroutine:
-101 : Failure to dynam ically allocate m em ory.
-102 : Invalid m atrix type.
-103 : Invalid pivotoption.
-104 : N um berofnonzeros is less than N EQ N S .
-105: NEQNS < 1
-201: Failure to dynam ically allocate \(m\) em ory.
-301 : Failure to dynam ically allocate \(m\) em ory.
-401 : Failure to dynam ically allocate \(m\) em ory.
-402 : NRHS < 1
-403 :NEQNS > LDRHS
-666 : Intemalemor.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
dgssin - Initialize the general sparse solver.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGSSIN(M TXTYP,PIVOT,NEQNS,COLSTR,ROW IND,OUTUNT,}
M SGLVL,HANDLE,\mathbb{ER )}
CHARACTER*2 M TXTYP
CHARACTER*1 PIVOT
\mathbb{NTEGER NEQNS,COLSTR (*),ROW IND (*),OUTUNT,MSGLVL,\mathbb{ER}}\mathbf{N}\mathrm{ (*)}
DOUBLE PRECISION HANDLE (150)

```

\section*{PURPOSE}

DGSSIN -Initialize the sparse solver and input the \(m\) atrix
structure.

\section*{PARAMETERS}
```

MTXTYP -CHARACTER*2
On entry,M TX TY P specifies the coefficientm atrix
type. Specifically, the valid options are:
sp 'or SP '-symm etric structure, positive-definite values
ss'or SS'-symm etric structure, sym m etric values
su'or SU '-symm etric structure, unsym m etric values
uu 'or UU '-unsymm etric structure, unsym m etric values
U nchanged on exit.
PIVOT -CHARACTER*1
On entry,PIV OT specifies w hetherornotpivoting is
used in the course of the num eric factorization.

```

The valid options are:
h'or \(\mathrm{N}^{\text {' }}\)-no pivoting is used
(Pivoting is not supported for this release).
U nchanged on exit.

NEQNS - \(\mathbb{N}\) TEGER
On entry, NEQNS specifies the num ber of equations in the coefficientm atrix. NEQNS m ustibe at leastone. U nchanged on exit.
\(\operatorname{COLSTR}\) ( \(\left.^{( }\right)-\mathbb{N}\) TEGER aray
On entry, \(\operatorname{COLSTR}\left({ }^{*}\right)\) is an array of size \((\mathbb{N} E Q \mathrm{~N}+1)\), containing the pointers of them atrix structure. U nchanged on exit.

\section*{ROWIND (*) - \(\mathbb{N}\) TEG ER anay}

On entry, ROWIND ( \({ }^{*}\) ) is an array of size
CO LSTR \(\mathbb{N} E Q N S+1)-1\), containing the indices of the \(m\) atrix structure. Unchanged on exit.

HANDLE (150) -D OUBLE PREC IS IO N array
On entry, HANDLE (*) is an array containing
inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine.
M odified on exit.

OUTUNT - \(\mathbb{N} T E G E R\)
O utputunit. U nchanged on exit.
M SGLVL - \(\mathbb{N}\) TEGER
M essage level.
0 -no output from solver.
(N om essages supported for this release.)

U nchanged on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
E rrornum ber. If no emrorencountered, unchanged on exit. If emorencountered, it is set to a non-zero integer. Enrornum bers setby this subroutine:
-101 : Faihure to dynam ically allocate \(m\) em ory.
-102 : Invalid m atrix type.
-103 : Invalid pivotoption.
-104 :N um berofnonzeros less than N EQ N S .
-105: NEQNS < 1

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
dgssor-G eneral sparse solver ordering and sym bolic factorization.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGSSOR(ORDMTHD,HANDLE,\mathbb{ER})}\()
CHARACTER*3 ORDMTHD
\mathbb{NTEGER ER}
DOUBLE PRECISION HANDLE (150)

```

\section*{PURPOSE}

DGSSOR -O rders and sym bolically factors a sparse m atrix .

\section*{PARAMETERS}

ORDMTHD -CHARACTER*3
On entry, ORDM THD specifies the fill-reducing ordering to be used by the sparse solver. Specifically, the valid options are:
hat'or NAT '-naturalordering (no ordering)
mmd'or M M D '-m ultiplem inim um degree
gnd 'or GND '-generalnested dissection
uso 'or U SO '-user specified ordering (see D G SSU O )
U nchanged on exit.
HANDLE (150) -DOUBLE PRECISIO N aray
On entry, HANDLE (*) is an array containing
inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine.

M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Errornum ber. If no errorencountered, unchanged on
exit. If emrorencountered, it is set to a non-zero
integer. Enrornum bers set.by this subroutine:
-200 : Invalid calling sequence -need to calld G SS IN first.
-201 : Failure to dynam ically allocate \(m\) em ory.
-666 : Intemalerror.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
dgssps -Print general.sparse solverstatics.

\section*{SYNOPSIS}

SUBROUTINEDGSSPS (HANDLE, \(\mathbb{E R}\) )
\(\mathbb{N} T E G E R \quad \mathbb{E R}\)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

DGSSPS -Print solver statistics.

\section*{PARAMETERS}

HANDLE (150) -D OUBLE PREC IS IO N array
On entry, HANDLE ( \(*\) ) is an amay containing
inform ation needed by the solver, and \(m\) ust.be passed
unchanged to each sparse solver subroutine.
M odified on exit.

ER
- \(\mathbb{N}\) TEGER

Errornum ber. If no errorencountered, unchanged on
exit. If errorencountered, it is set to a non-zero
integer. Errornum bers set.by this subroutine:
-800 : Invalid calling sequence - need to calld G SSSL first.
-899: Printed solver statistics not supported this release.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
dgssp -R etum perm utation used by the general sparse solver.

\section*{SYNOPSIS}

SUBROUTINEDGSSRP (PERM, HANDLE, \(\mathbb{E R}\) )
\(\mathbb{N} T E G E R \quad\) PERM (*), \(\mathbb{E R}\)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

DGSSRP - Retums the perm utation used by the solver for the fill-reducing ordering.

\section*{PARAMETERS}

PERM \(\mathbb{N E Q N S}\) ) - \(\mathbb{N}\) TEGER amay
U ndefined on entry. PERM \(\mathbb{N E Q N S}\) ) is the perm utation array used by the sparse solver for the fillreducing ordering. M odified on exit.

HANDLE (150) -D OUBLE PRECISION array
On entry, HANDLE ( \(*\) ) is an array containing
inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine.
M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Errornum ber. If no errorencountered, unchanged on exit. If errorencountered, it is set to a non-zero
integer. Errornum bers set.by this subroutine:
-600 : Invalid calling sequence - need to callD G SSO R first.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
dgsssl-Solve routine for the general sparse solver.

\section*{SYNOPSIS}

SU BROUTINEDGSSSL (NRHS,RHS,LDRHS,HANDLE, ERR)
\(\mathbb{I N}\) TEGER NRHS,LDRHS, \(\mathbb{E R}\)
DOUBLE PRECISION RHS (LDRHSNRHS)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

DGSSSL -Triangular solve of a factored sparse m atrix.

\section*{PARAMETERS}

NRHS - \(\mathbb{N}\) TEGER
On entry, N RH S specifies the num ber of righthand
sides to solve for. U nchanged on exit.

RHS (LDRHS,*) -D OUBLE PRECISION array
On entry, RH S (LDRHS,NRHS) contains the NRHS right hand sides. On exit, it contains the solutions.

LDRHS - \(\mathbb{N}\) TEGER
On entry, LD RH S specifies the leading dim ension of the RH S array. U nchanged on exit.

HANDLE (150) -D OUBLE PREC ISIO N aray
O n entry, HANDLE ( \({ }^{\star}\) ) is an array containing
inform ation needed by the solver, and \(m\) ustbe passed
unchanged to each sparse solver subroutine.
M odified on exit.

ER
- \(\mathbb{N}\) TEGER

Errornum ber. If no errorencountered, unchanged on
exit. If errorencountered, it is set to a non-zero
integer. Errornum bers setby this subroutine:
-400 : Invalid calling sequence - need to callD G SSFA first.
-401 : Failure to dynam ically allocate \(m\) em ory.
-402 : NRHS < 1
-403 : NEQN S > LD RHS

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
dgssuo - U ser supplied perm utation for ordering used in the general sparse solver.

\section*{SYNOPSIS}

SUBROUTINEDGSSUO (PERM,HANDLE, \(\mathbb{E R}\) )
\(\mathbb{N}\) TEGER PERM (*), \(\mathbb{E R}\)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

DGSSUO -U ser supplied perm utation for ordering. M ust.be called afterDGSS IN () (sparse solver initialization) and before \(D G S S O R 0\) (sparse solver ordering).

\section*{PARAMETERS}

PERM \(\mathbb{N} E Q N S\) ) - \(\mathbb{N}\) TEGER array
On entry, PERM (NEQNS) is a perm utation array supplied by the user for the fill-reducing ordering.
U nchanged on exit.
HANDLE (150) -D OUBLE PREC IS IO N array
On entry, HANDLE (*) is an anay containing
inform ation needed by the solver, and \(m\) ust.be passed
unchanged to each sparse solver subroutine.
M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Errornum ber. If no errorencountered, unchanged on
exit. If emorencountered, it is set to a non-zero
integer. Errornum bers setby this subroutine:
-500 : Invalid calling sequence - need to callD G SS \(\mathbb{N}\) first.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgtoon -estim ate the reciprocal of the condition num ber of a real tridiagonal \(m\) atrix \(A\) using the \(L U\) factorization as com puted by SG TTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGTCON NORM,N,LOW ,DIAG,UP1,UP2, \mathbb{PIVOT,ANORM,RCOND,}}\mathbf{N},
W ORK,\mathbb{N ORK2,INFO)}
CHARACTER * 1 NORM
\mathbb{NTEGER N,}\mathbb{N}FO
\mathbb{NTEGER \mathbb{PIVOT (*), IN ORK2 (*)}}\mathbf{(})
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION LOW (*),D IAG (*),UP1 (*),UP2 (*),W ORK (*)
SUBROUT\mathbb{NEDGTCON_64 NORM,N,LOW,DIAG,UP1,UP2, \mathbb{PIVOT,ANORM,}}\mathbf{N}\mathrm{ , NOT}
RCOND,W ORK,IN ORK2,\mathbb{NFO)}
CHARACTER * 1 NORM
INTEGER*8 N, INFO
\mathbb{NTEGER*8 \mathbb{PIVOT (*), IN ORK 2 (*)}}\mathbf{(*)}
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION LOW (*),DIAG (*),UP1 (*),UP2 (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GTCON \(\mathbb{N} O R M, \mathbb{N}], L O W, D I A G, U P 1, U P 2, \mathbb{P} \mathbb{I V O T}, A N O R M\), RCOND, [W ORK ], [ \(\mathbb{W}\) ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::NORM
\(\mathbb{N}\) TEGER :: N, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V} O T, \mathbb{I}\) ORK2
REAL (8) ::ANORM,RCOND

REAL (8),D \(\mathbb{M}\) ENSION (:) ::LOW ,D \(\mathbb{A} G, U P 1, U P 2, W\) ORK
SUBROUTINE GTCON_64 NORM, \(\mathbb{N}], L O W, D \mathbb{I A G}, U P 1, U P 2, \mathbb{P} \mathbb{I} O T, A N O R M\), RCOND, \(\mathbb{W}\) ORK], [ \(\mathbb{W}\) ORK2], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::NORM
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathbb{I N F O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T, \mathbb{N}\) ORK 2
REAL (8) ::ANORM,RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::LOW ,D \(\mathbb{A} G, U P 1, U P 2, W\) ORK

\section*{C INTERFACE}
\#include <sunperfh>
void dgtcon (charnorm, intn, double *low, double *diag, double *up1, double *up2, int *ipinot, double anorm , double *rcond, int *info);
void dgtcon_64 (charnorm, long n, double *low , double *diag, double *up1, double *up2, long *ipívot, double anorm , double *rcond, long *info);

\section*{PURPOSE}
dgtoon estim ates the reciprocal of the condition num berofa real tridiagonalm atrix A using the LU factorization as com puted by SG TTRF.

A \(n\) estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND = \(1 /\) (ANORM * norm (inv (A))).

\section*{ARGUMENTS}

NORM (input)
Specifies w hether the 1-norm condition number or the infinity-norm condition num ber is required:
= 1 'or \(\mathrm{O}^{\prime}\) : 1 -nom ;
= I': Infinity-norm .
N (input) The order of the matrix \(A . N>=0\).
LOW (input)
The \((n-1) m\) ultipliers that define the \(m\) atrix \(L\) from the LU factorization of A as com puted by SG TTRF.

D IA G (input)

The n diagonalelem ents of the upper triangular \(m\) atrix \(U\) from the \(L U\) factorization ofA.

UP1 (input)
The ( \(n-1\) ) elem ents of the first superdiagonal of U.

UP2 (input)
The ( \(n-2\) ) elem ents of the second superdiagonal of U.
\(\mathbb{P I V O T}\) (input)
The pivotindioes; for \(1<=i<=n\), row \(i\) of the matrix was interchanged with row PIVOT (i). IPIVOT (i) will always be either \(i\) or i+1; PIVOT (i) = iindicates a row interchange \(w\) as not required.

ANORM (input)
IfNORM = I'orD', the 1-norm of the original \(m\) atrix \(A\). IfNORM = I', the infinity-nom of the originalm atrix A .

RCOND (output)
The reciprocal of the condition number of the \(m\) atrix \(A\), com puted as RCOND \(=1 /(A N O R M * A \mathbb{N} V N M)\), where \(A \mathbb{N} V N M\) is an estim ate of the 1 -nom of \(\operatorname{inv}(A)\) com puted in this routine.

W ORK (w orkspace)
dim ension \((2 * N)\)

IV ORK 2 (w orkspace)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N N}\) FO = -i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgthr-G athers specified elem ents from \(y\) into \(x\).

\section*{SYNOPSIS}
```

SUBROUTINEDGTHR NZ,Y,X,\mathbb{NDX)}
DOUBLE PRECISION Y (*),X (*)
INTEGER NZ
INTEGER \mathbb{NDX (*)}
SUBROUT\mathbb{NEDGTHR_64 NZ,Y,X,NNDX)}
DOUBLE PRECISION Y (*),X (*)
INTEGER*8NZ
INTEGER*8 \mathbb{NDX (*)}
F95 INTERFACE
SUBROUTINE GTHR(NZ],Y,X,\mathbb{NDX)}
REAL (8),D IM ENSION (:) ::Y,X
INTEGER ::NZ
\mathbb{NTEGER,D IM ENSION (:) :: \mathbb{NDX}}\mathbf{N}=\mp@code{N}
SUBROUT\mathbb{NE GTHR_64(NZ],Y,X,\mathbb{NDX)}}\mathbf{N}\mathrm{ (N}
REAL (8),D IM ENSION (:) ::Y,X
INTEGER (8)::NZ
\mathbb{NTEGER (8),D IM ENSION (:)::\mathbb{NDX}}\mathbf{N}=\mp@code{N}

```

\section*{PURPOSE}

D GTHR -G athers the specified elem ents from a vectory in fullstorage form into a vectorx in com pressed form. Only
the elem ents of \(y\) w hose indices are listed in indx are referenced.
do \(i=1, n\) \(x(i)=y(\) indx \((i))\)
enddo

\section*{ARGUMENTS}

N Z (input) - \(\mathbb{N}\) TEGER
\(N\) um ber of elem ents in the com pressed form .
U nchanged on exit.
\(Y\) (input)
V ectorin fullstorage form . U nchanged on exit.
X (output)
V ector in com pressed form. C ontains elem ents ofy
whose indices are listed in indx on exit.
\(\mathbb{I N D X}\) (input) - \(\mathbb{N}\) TEGER
\(V\) ector containing the indices of the com pressed
form. It is assum ed that the elem ents in \(\mathbb{N} D \mathrm{X}\) are
distinct and greater than zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgthrz -G ather and zero.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DGTHRZ NZ,Y,X,NNDX)}
DOUBLE PRECISION Y (*),X (*)
INTEGER NZ
INTEGER \mathbb{NDX (*)}
SUBROUT\mathbb{NEDGTHRZ_64 NZ,Y,X,INDX)}
DOUBLE PRECISION Y (*),X (*)
INTEGER*8NZ
INTEGER*8 \mathbb{NDX (*)}
F95 INTERFACE
SUBROUT\mathbb{NE GTHRZ (NZ],Y,X,NNDX)}
REAL (8),D IM ENSION (:) ::Y,X
\mathbb{NTEGER ::NZ}
\mathbb{NTEGER,D IM ENSION (:) :: \mathbb{NDX}}\mathbf{N}=\mp@code{N}
SUBROUT\mathbb{NEGTHRZ_64(NZ],Y,X,\mathbb{NDX)}}\mathbf{N}=(\mathbb{N}
REAL (8),D IM ENSION (:) ::Y,X
INTEGER (8)::NZ
\mathbb{NTEGER (8),D IM ENSION (:) :: \mathbb{NDX}}\mathbf{N}={

```

\section*{PURPOSE}

D G THRZ -G athers the specified elem ents from a vectory in fullstorage form into a vectorx in com pressed form. The
gathered elem ents ofy are set to zero. O nly the elem ents ofy w hose indices are listed in indx are referenced.
```

do i=1,n
x (i) = y (indx (i))
y(indx (i)) = 0
enddo

```

\section*{ARGUMENTS}

N Z (input) - \(\mathbb{N}\) TEGER
N um ber of elem ents in the com pressed form .
U nchanged on exit.

Y (input/output)
V ector in fullstorage form. G athered elem ents are setto zero.
X (output)
V ector in com pressed form. C ontains elem ents ofy w hose indices are listed in indx on exit.
\(\mathbb{N} D X\) (input) - \(\mathbb{N} T E G E R\)
V ector containing the indiges of the com pressed form. It is assum ed that the elem ents in \(\mathbb{N D} X\) are distinctand greater than zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgtrfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is tridiagonal, and provides errorbounds and backw ard error estim ates for the solution

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE DGTRFS (TRANSA,N,NRHS,LOW ,D IAG,UP,LOW F,D IAGF,UPF1,}
UPF2,\mathbb{PIVOT,B,LDB,X,LDX,FERR,BERR,W ORK,W ORK 2, INFO)}
CHARACTER * 1 TRANSA
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*),W ORK2(*)}
DOUBLE PRECISION LOW (*),D IAG (*),UP (*),LOW F (*), D IA GF (*),
UPF1 (*), UPF2 (*), B (LDB (*), X (LDX ,*), FERR (*), BERR (*),
W ORK (*)
SUBROUT\mathbb{NEDGTRFS_64 (TRANSA,N,NRHS,LOW ,DIAG,UP,LOW F,DIAGF,}
UPF1,UPF2,\mathbb{PIVOT,B,LDB,X,LDX,FERR,BERR,W ORK,W ORK2,}
INFO)
CHARACTER * 1 TRANSA
\mathbb{N}TEGER*8N,NRHS,LDB,LDX,\mathbb{NFO}
INTEGER*8 \mathbb{PIVOT (*),W ORK2 (*)}
D OUBLE PRECISION LOW (*),D IAG (*),UP (*),LOW F (*), D IA GF (*),
UPF1 (*), UPF2 (*), B (LDB ,*), X (LDX ,*), FERR (*), BERR (*),
W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GTRFS ([TRANSA], \(\mathbb{N}], \mathbb{N} R H S], L O W, D \mathbb{I A}, U P, L O W ~ F, D \mathbb{I} G F\), UPF1, UPF2, \(\mathbb{P} \mathbb{I V}\) OT,B, [LDB],X, [LDX],FERR,BERR, [W ORK], [W ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N}\) TEGER :: N, NRHS,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}, \mathrm{W} O R K 2\)
REAL (8), D \(\mathbb{I M} \operatorname{ENSION}\) (:) ::LOW ,D \(\mathbb{A} G, U P, L O W F, D \mathbb{A} G F, U P F 1\),
UPF2,FERR, BERR, W ORK
REAL (8), D IM ENSION (:,:) ::B,X

SU BROUTINE GTRFS_64 ([TRANSA ], \(\mathbb{N}], \mathbb{N} R H S], L O W, D I A G, U P, L O W F\), D \(\operatorname{IAGF}, \mathrm{UPF} 1, \mathrm{UPF} 2, \mathbb{P} \mathbb{I} O T, B,[L D B], X,[L D X], F E R R, B E R R,[\mathbb{O} O R K]\), [W ORK2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T\), W ORK 2
REAL (8), D \(\mathbb{M} E N S I O N(:):: L O W, D \mathbb{A G}, U P, L O W F, D I A G F, U P F 1\),
UPF2,FERR,BERR,WORK
REAL (8), D \(\mathbb{M}\) ENSION (: : : : : B , X

\section*{C INTERFACE}
\#include <sunperfh>
void dgtrfs (chartransa, intn, intnrhs, double *low, double *diag, double *up, double *low f, double *diagf, double *upfl, double *upf2, int *ipivot, double *b, int ldb, double *x, intldx, double * ferr, double *berr, int *info);
void dgtrfs_64 (chartransa, long n, long nins, double *low , double *diag, double *up, double *low f, double *diagf, double *upfl, double *upf2, long *ipivot, double *b, long ldl, double *x, long ldx, double * ferrr, double *berr, long *info);

\section*{PURPOSE}
dgtrfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is tridiagonal, and provides enorbounds and backw ard enrorestim ates for the solution.

\section*{ARGUMENTS}

TRANSA (input)
Specifies the form of the system of equations:
\(=N\) : A * \(\mathrm{X}=\mathrm{B} \quad \mathrm{N} \circ\) transpose)
= \(T\) ': A ** \(T\) * \(\mathrm{X}=\mathrm{B}\) ( T ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose \(=\mathrm{T}\) ranspose)

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS \(>=0\).

LOW (input)
The \((n-1)\) subdiagonalelem ents of .

D IA G (input)
The diagonalelem ents of A.
UP (input)
The \((n-1)\) superdiagonalelem ents of \(A\).
LOW \(F\) (input)
The ( \(n-1\) ) multipliers that define the \(m\) atrix \(L\) from the LU factorization of \(A\) as com puted by SG TTRF.

D IA GF (input)
Then diagonalelem ents of the upper triangular \(m\) atrix \(U\) from the \(L U\) factorization ofA.

UPF1 (input)
The ( \(n-1\) ) elem ents of the first superdiagonal of U.

UPF2 (input)
The ( \(n-2\) ) elem ents of the second superdiagonal of U.
\(\mathbb{P I V O T}\) (input)
The pivot indices; for \(1<=\mathrm{i}<=\mathrm{n}\), row \(i\) of the \(m\) atrix \(w\) as interchanged \(w\) th row \(\mathbb{P I V O T}\) (i). PPIVOT (i) will alw ays be either \(i\) or i+1; PIV T (i) = iindicates a row interchange was not required.
\(B\) (input) The righthand side m atrix \(B\).
LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, \mathbb{N})\).

X (input/output)
O \(n\) entry, the solution \(m\) atrix \(X\), as com puted by SG TTRS. On exit, the im proved solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the anay X . LD X >= \(\max (1, N)\).
FERR (output)
The estim ated forw ard errorbound for each solution vector \(X()\) (the \(j\) th column of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{H})\), FERR ( \()\) ) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\underset{)}{ })-X\) TRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{j})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each
solution vector X (i) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension ( \(3 * \mathrm{~N}\) )

W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgtsv - solve the equation \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDGTSV N,NRHS,LOW ,D IAG,UP,B,LDB, INFO)}
\mathbb{NTEGER N,NRHS,LDB,INFO}
DOUBLE PRECISION LOW (*),DIAG (*),UP (*),B (LDB,*)
SU BROUT\mathbb{NE DGTSV_64 N ,NRHS,LOW,D IAG,UP,B,LDB, INFO )}
\mathbb{NTEGER*8N,NRHS,LDB,INFO}
DOUBLE PRECISION LOW (*),DIAG (*),UP (*),B (LDB,*)
F95 INTERFACE
SUBROUT\mathbb{NEGTSV (N ], NRHS],LOW ,D IAG,UP,B,[LDB],[NFO])}
INTEGER::N,NRHS,LDB,\mathbb{NFO}
REAL (8),D IM ENSION (:) ::LOW ,D IAG ,UP
REAL (8),D IM ENSION (:,:) ::B

```

```

    \mathbb{N TEGER (8) ::N N,NRHS,LD B,NNFO}
    REAL (8),D IM ENSION (:) ::LOW ,D IAG,UP
    REAL (8),D IM ENSION (:,:) ::B
    C INTERFACE
\#include <sunperfh>
void dgtsv (intn, intnrhs, double *low , double *diag, dou-
ble *up, double *b, int ldb, int *info);

```
void dgtsv_64 (long n, long nrhs, double *low , double *diag, double *up, double *b, long ldb, long *info);

\section*{PURPOSE}
dgtsv solves the equation
where \(A\) is an \(n\) by \(n\) tridiagonalm atrix, by \(G\) aussian elm ination \(w\) ith partialpivoting.

N ote that the equation \(A\) *X \(=\mathrm{B}\) m ay be solved by interchanging the order of the argum ents \(D U\) and \(D L\).

\section*{ARGUMENTS}

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
NRHS (input)
The num berof righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS >= 0 .

LOW (input/output)
On entry, LOW m ust contain the ( \(n-1\) ) sub-diagonal elem ents ofA.

On exit, LOW is overw ritten by the ( \(n-2\) ) elem ents of the second super-diagonal of the upper triangularm atrix \(U\) from the \(L U\) factorization of \(A\), in LOW (1), ..., LOW (n-2).

D IA G (input/output)
O n entry, D IA G m ustcontain the diagonal elem ents ofA.

On exit, D IA G is overw rilten by the \(n\) diagonal elem ents of \(U\).

UP (input/output)
O n entry, UP m ust contain the ( \(n-1\) ) super-diagonal elem ents of .

On exit, UP is overw ritten by the ( \(n-1\) ) elem ents of the first super-diagonal of \(U\).

B (input/output)
On entry, the N by NRH S m atrix of righthand side
matrix B. On exit, if \(\mathbb{N} F O=0\), the \(N\) by NRHS solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i, U(i, i)\) is exactly zero, and the solution has notbeen com puted. The factorization has notbeen com pleted unless \(i=N\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgtsvx - use the LU factorization to com pute the solution to a realsystem of linearequations \(A * X=B\) or \(A * T * X=B\),

\section*{SYNOPSIS}

SU BROUTINEDGTSVX EACT,TRANSA,N,NRHS,LOW,D IAG,UP,LOW F,D IAGF, UPF1, UPF2, \(\mathbb{P} \mathbb{I V O T}, \mathrm{B}, \mathrm{LD} \mathrm{B}, \mathrm{X}, \mathrm{LD} \mathrm{X}, \mathrm{RCOND}, \mathrm{FERR}, \mathrm{BERR}, \mathrm{W} O R K\), WORK2, \(\mathbb{N} F O\) )

CHARACTER * 1 FACT,TRANSA
\(\mathbb{N}\) TEGER \(N, N R H S, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{P} \mathbb{V} O T\left({ }^{( }\right), \mathrm{W} O R K 2\left({ }^{( }\right)\)
DOUBLE PRECISION RCOND
DOUBLE PRECISION LOW (*), D IA G (*), UP (*), LOW F (*), D IA GF (*),
 W ORK (*)

SU BROUTINEDGTSVX_64 EACT,TRANSA,N,NRHS,LOW,DIAG,UP,LOW F, D \(\mathbb{A} G F, U P F 1, U P F 2, \mathbb{P} \mathbb{I} O T, B, L D B, X, L D X, R C O N D, F E R R, B E R R\), W ORK, W ORK \(2, \mathbb{N} F O\) )

CHARACTER * 1 FACT,TRANSA
\(\mathbb{N} T E G E R * 8 N, N R H S, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER*8 \(\mathbb{P} \mathbb{I V O T}\) (*),WORK2(*)
DOUBLE PRECISION RCOND
D OUBLE PRECISION LOW (*), D IA G (*), UP (*), LOW F (*), D IA GF (*),
 WORK (*)

F95 INTERFACE
SUBROUTINE GTSVX \(\left.\left.\mathbb{E}^{\prime} A C T,[T R A N S A], \mathbb{N}\right], \mathbb{N} R H S\right], L O W, D I A G, U P, L O W F\), D \(\mathbb{A} G F, U P F 1, U P F 2, \mathbb{P} \mathbb{I V} O T, B,[L D B], X,[L D X], R C O N D, F E R R, B E R R\),
[W ORK], [W ORK 2], [ \(\mathbb{N F O}]\) )
```

CHARACTER (LEN=1) ::FACT,TRANSA
\mathbb{N TEGER ::N,NRHS,LDB,LDX, NNFO}

```

```

REAL (8) ::RCOND
REAL (8),D\mathbb{M ENSION (:) ::LOW ,DIAG ,UP,LOW F, DIAGF, UPF1,}
UPF2,FERR,BERR,W ORK
REAL (8),D IM ENSION (:,:) ::B,X

```
SU BROUTINE GTSVX_64 (FACT, [TRANSA], \(\mathbb{N}], \mathbb{N R H S}], L O W, D \mathbb{I} G, U P, L O W E\),
    D \(\mathbb{I A G F}, \mathrm{UPF} 1, \mathrm{UPF} 2, \mathbb{P} \mathbb{I} O T, B,[L D B], X,[L D X], R C O N D, F E R R, B E R R\),
    [ \(\mathbb{W}\) ORK], \([\mathbb{W}\) ORK2], \([\mathbb{N} F O]\) )
CHARACTER ( \(几 E N=1\) ) : : FACT,TRANSA
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I} O T, W O R K 2\)
REAL (8) :: RCOND
REAL (8), D \(\mathbb{M} E N S I O N(:):: L O W, D \mathbb{A G}, U P, L O W F, D \mathbb{A} G F, U P F 1\),
UPF2,FERR,BERR,WORK
REAL (8), D \(\mathbb{M}\) ENSION (: : : : : B, X

\section*{C INTERFACE}
\#include <sunperfh>
void dgtsvx (char fact, chartransa, intn, intnrhs, double
*low , double *diag, double *up, double *low f, double *diagf, double *upfl, double *upf2, int *ipivot, double *b, int ldb, double *x, int ldx, double *rcond, double *ferr, double *berr, int *info);
void dgtsvx_64 (char fact, chartransa, long n, long nrhs, double *low, double *diag, double *up, double * low f, double *diagf, double *upfl, double *upf2, long *ịíivot, double *b, long ldb, double *x, long ldx, double *rcond, double *ferr, double *benr, long *info);

\section*{PURPOSE}
dgtsvx uses the LU factorization to com pute the solution to a realsystem of linear equations \(A * X=B\) or \(A * T * X=B\), where \(A\) is a tridiagonalm atrix oforder \(N\) and \(X\) and \(B\) are N -by-N R H S m atrices.

Errorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT \(=N\) ', the LU decom position is used to factor the \(m\) atrix A
as \(A=L * U, w h e r e L\) is a product of perm utation and unitlow er
bidiagonal \(m\) atrices and \(U\) is upper triangular \(w\) ith nonzeros in
only the \(m\) ain diagonaland first tw o superdiagonals.
2. If som e \(U(i, i)=0\), so that \(U\) is exactly singular, then the routine
retums w ith \(\mathbb{N} F O=\) i. O therw ise, the factored form of \(A\) is used
to estim ate the condition num ber of the m atrix A. If the
reciprocal of the condition num ber is less than \(m\) achine precision,
\(\mathbb{N} F O=N+1\) is retumed as a waming, but the routine stillgoes on
to solve for X and com pute error bounds as described below .
3.The system ofequations is solved for X using the factored form of A.
4. Iterative refinem ent is applied to im prove the com puted solution
m atrix and calculate error bounds and backw ard enror estim ates for it.

\section*{ARGUMENTS}

FACT (input)
Specifies whether ornot the factored form of A has been supplied on entry.\(=F\) ': LOW F, D IA GF, UPF1, UPF2, and IPIVOT contain the factored form of \(A\); LOW, DIAG,UP,LOW F,D \(\mathbb{A} G F, U P F 1, U P F 2\) and \(\mathbb{P} \mathbb{V}\) O T w illnotbe m odified. \(=\mathrm{N}\) ': The m atrix w ill be copied to LOW F, D IA G F , and U PF1 and factored.

TRANSA (input)
Specifies the form of the system of equations:
\(=N: A * X=B \quad\) N \(\circ\) transpose)
\(=T\) ': A ** \(T\) * \(\mathrm{X}=\mathrm{B}\) ( T ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B}\) (C onjugate transpose = T ran-
spose)

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{N}\) TERFACE.

N (input) The order of them atrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the m atrix B. NRH S \(>=0\).

LOW (input)
The ( \(\mathrm{n}-1\) ) subdiagonalelem ents ofA.

D IA G (input)
The n diagonalelem ents of A.
UP (input/output)
The ( \(n-1\) ) superdiagonalelem ents ofA.

LOW F (input/output)
IfFACT \(=F^{\prime}\), then LOW \(F\) is an inputargum ent and on entry contains the \((\mathrm{n}-1) \mathrm{m} u\) ultipliers that define the \(m\) atrix \(L\) from the \(L U\) factorization of \(A\) as com puted by SG TTRF .

IfFACT \(=N^{\prime}\) ', then LOW \(F\) is an outputargum entand on exitcontains the \((n-1) m\) ultipliers that define them atrix \(L\) from the LU factorization of \(A\).

D IA G F (input/output)
If \(F A C T=F\) ', then \(D I A G F\) is an inputargum ent and on entry contains the \(n\) diagonalelem ents of the uppertriangularm atrix \(U\) from the LU factorization ofA.

IfFACT \(=\mathrm{N}^{\prime}\), then D IAGF is an output argum ent and on exit contains the \(n\) diagonalelem ents of the upper triangularm atrix \(U\) from the LU factorization ofA .

UPF1 (input/output)
IfFACT = F', then UPF1 is an inputargum ent and on entry contains the ( \(n-1\) ) elem ents of the first superdiagonal of \(U\).

IfFACT \(=N\) ', then UPF1 is an output argum entand on exit contains the ( \(n-1\) ) elem ents of the first superdiagonalof .

UPF2 (input/output)
IfFACT \(=F^{\prime}\), then \(U P F 2\) is an input argum ent and
on entry contains the ( \(n-2\) ) elem ents of the second superdiagonalofU .

IfFACT \(=\mathrm{N}\) ', then UPF2 is an outputargum entand on exitcontains the ( \(n-2\) ) elem ents of the second superdiagonalofU .

PIVOT (input/output)
If \(F A C T=F '\), then \(\mathbb{P I V O T}\) is an input argum ent and on entry contains the pivotindioes from the LU factorization of A as com puted by SG TTRF. IfFACT = \(N\) ', then \(\mathbb{P} \mathbb{I V O T}\) is an output argum ent and on exit contains the pivot indices from the LU factorization of \(A\); row iof the \(m\) atrix \(w\) as interchanged w ith row \(\mathbb{P} \mathbb{I V O T}\) (i). \(\mathbb{P} \mathbb{I V O T}\) (i) w illalw ays be eitherior i+1; \(\mathbb{P} \mathbb{I V}\) OT (i) = iindicates a row interchange w as not required.

B (input) The N -by -N RH S righthand side m atrix B .

LD B (input)
The leading dim ension of the aray B . LD B >= \(\max (1, N)\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=N+1\), the \(N\) by -N RH \(S\) solution
\(m\) atrix \(X\).

LD X (input)
The leading dim ension of the array \(\mathrm{X} . \mathrm{LD} \mathrm{X}>=\) \(\max (1, \mathbb{N})\).

RCOND (output)
The estim ate of the reciprocal condition num ber of the \(m\) atrix \(A\). IfRCOND is less than the \(m\) achine precision (in particular, if RCOND \(=0\) ), the \(m\) atrix is singular to working precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO > 0.

FERR (output)
The estim ated forw ard enrorbound for each solution vector \(X()\) ) the \(j\) th column of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{O})\) FERR ( \()\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(\mathrm{X}(\mathcal{j})\)-XTRUE) divided by the \(m\) agnitude of the largestelem entin \(\mathrm{X}(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector \(X\) (i) (i.e., the \(s m\) allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).
W ORK (w orkspace)
dim ension \((3 * N)\)

W ORK 2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{I N}\) FO = -i, the i-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N} F O=i\), and \(i\) is
\(<=\mathrm{N}: \mathrm{U}(i, i)\) is exactly zero. The factorization has notbeen com pleted unless \(i=N\), but the factorU is exactly singular, so the solution and error bounds could notbe com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1\) : U is nonsingular, but RCOND is less than \(m\) achine precision, meaning that the \(m\) atrix is singular to working precision. N evertheless, the solution and errorbounds are com puted because there are a num ber of sifuations where the com puted solution can be m ore accurate than the value of RCOND w ould suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dgturf-com pute an LU factorization ofa real tridiagonal \(m\) atrix \(A\) using elim ination \(w\) ith partial pivoting and row interchanges

\section*{SYNOPSIS}

```

\mathbb{NTEGER N,\mathbb{NFO}}\mathbf{N}\mathrm{ (})
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION LOW (*),D IAG (*),UP1 (*),UP2 (*)

```

```

\mathbb{NTEGER*8N,\mathbb{NFO}}\mathbf{N}\mathrm{ ( }
INTEGER*8\mathbb{PIVOT (*)}
DOUBLE PRECISION LOW (*),D IAG (*),UP1 (*),UP2 (*)
F95 INTERFACE

```

```

    \mathbb{NTEGER ::N,\mathbb{NFO}}0=0,
    ```

```

    REAL (8),D IM ENSION (:) ::LOW ,D IAG ,UP1,UP2
    ```

```

    \mathbb{NTEGER (8)::N,\mathbb{NFO}}\mathbf{N}=0
    ```

```

    REAL (8),D IM ENSION (:) ::LOW ,D IAG,UP1,UP2
    ```
#include <sunperfh>
```

void dgturf(intn, double *low , double *diag, double *up1,
double *up2, int *ipivivot, int*info);
void dgturf_64 (long n, double *low, double *diag, double
*up1, double *up2, long *ịíivot, long *info);

## PURPOSE

dgttrf com putes an LU factorization of a real tridiagonal $m$ atrix A using elim ination w ith partialpivoting and row interchanges.

The factorization has the form

$$
A=L * U
$$

where $L$ is a productof perm utation and unit low er bidiagonalm atrices and $U$ is uppertriangularw ith nonzeros in only them ain diagonal and first tw o superdiagonals.

## ARGUMENTS

N (input) The order of the m atrix A.

LOW (input/output)
On entry, LOW m ustcontain the ( $n-1$ ) sub-diagonal elem ents ofA.

On exit, LOW is overw ritten by the ( $n-1$ ) multipliers that define the $m$ atrix $L$ from the $L U$ factorization of A.

D IA G (input/output)
On entry, D IA G m ust contain the diagonal elem ents ofA.

On exit, D IA G is overw rilten by the $n$ diagonal elem ents of the upper triangularm atrix $U$ from the LU factorization ofA.

UP1 (input/output)
On entry, UP1 must contain the ( $n-1$ ) superdiagonalelem ents ofA.

On exit, UP1 is overw ritten by the ( $n-1$ ) elem ents of the first super-diagonalof $U$.

UP2 (output)
On exit, UP2 is overw rilten by the ( $n-2$ ) elem ents of the second super-diagonalofU.

IPIVOT (output)
The pivotindioes; for $1<=i<=n$, row $i$ of the matrix was interchanged w th row $\mathbb{P I V O T}$ (i). IPIVOT (i) will always be either $i$ or i+1; PIVOT (i) = iindicates a row interchange was not required.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0$ : if $\mathbb{N} F O=-k$, the $k$-th argum enthad an illegalvalue
> 0 : if $\mathbb{N} F O=k, U(k, k)$ is exactly zero. The factorization has been com pleted, but the factor $U$ is exactly singular, and division by zero will occur if it is used to solve a system ofequations.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dgttrs - solve one of the system sofequations $A * X=B$ or A ${ }^{*} X=B$,

## SYNOPSIS

```
SUBROUT\mathbb{NEDGTTRS (TRANSA,N,NRHS,LOW ,D IAG,UP1,UP2,\mathbb{PIVOT,B,}}\mathbf{N},\textrm{N},\textrm{N}
    LDB,INFO)
CHARACTER * 1 TRANSA
\mathbb{NTEGER N,NRHS,LDB,INFO}
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION LOW (*),DIAG (*),UP1 (*),UP2 (*),B (LDB,*)
```



```
    LDB, INFO)
CHARACTER * 1 TRANSA
INTEGER*8N,NRHS,LDB,INFO
\mathbb{NTEGER*8 P\mathbb{IVOT (*)}}\mathbf{(})
DOUBLE PRECISION LOW (*),DIAG (*),UP1 (*),UP2 (*),B (LDB,*)
```


## F95 INTERFACE

SUBROUTINE GTTRS ([TRANSA], $\mathbb{N}], \mathbb{N} R H S], L O W, D \mathbb{I A}, \operatorname{UP} 1, U P 2, \mathbb{P} \mathbb{I} O T$, B, [LDB], $[\mathbb{N} F O])$

CHARACTER (LEN=1) ::TRANSA
$\mathbb{N}$ TEGER ::N,NRHS,LDB, $\mathbb{N}$ FO
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T$
REAL (8), D $\mathbb{M}$ ENSION (:) ::LOW ,D IAG,UP1, UP2
REAL (8),D IM ENSIO N (:,:) ::B
SU BROUTINE GTTRS_64 ([TRANSA], $\mathbb{N}], \mathbb{N} R H S], L O W, D \mathbb{A} G, U P 1, U P 2$,
$\mathbb{P} \mathbb{V} O T, B,[\operatorname{LDB}],[\mathbb{N F O}])$

CHARACTER (LEN=1) ::TRANSA
$\mathbb{N}$ TEGER (8) :: N , NRHS,LD B, $\mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \mathrm{ENSION}(:):: \mathbb{P} \mathbb{I} O T$
REAL (8), D $\mathbb{I}$ ENSION (:) :: LOW ,D IA G , UP1, UP2
REAL (8), D $\mathbb{M}$ ENSION (: ::) ::B

## C INTERFACE

\#include <sunperfh>
void dgters (char transa, intn, intnrhs, double *low, double *diag, double *up1, double *up2, int *ipivot, double *b, int ldb, int *info);
void dgttrs_64 (chartransa, long n, long nrhs, double *low, double *diag, double *up1, double *up2, long *ípivot, double *b, long ldb, long *info);

## PURPOSE

dgttrs solves one of the system s of equations
$A * X=B$ or $A * X=B, w$ th a tridiagonalm atrix $A$ using the LU factorization com puted by SG TTRF.

## ARGUMENTS

TRANSA (input)
Specifies the form of the system ofequations. =
$\mathrm{N}^{\prime}: A * X=B \quad$ N o transpose)
$=T T^{\prime}: A \times N \quad$ (Transpose)
$=C$ : A * $\mathrm{X}=\mathrm{B}$ (C onjugate transpose $=$ Transpose)

TRANSA is defaulted to $N$ 'forF95 $\mathbb{N}$ TERFACE.

N (input) The order of the $m$ atrix $A$.

NRHS (input)
The num ber of right hand sides, ie., the num ber
of colum ns of the m atrix B. NRHS $>=0$.

LOW (input)
The $(n-1)$ m ultipliers that define the $m$ atrix $L$ from the LU factorization ofA.

D IA G (input)
The $n$ diagonalelem ents of the upper triangular
$m$ atrix $U$ from the $L U$ factorization ofA .

## UP1 (input)

The $(n-1)$ elem ents of the first super-diagonal of U .

UP2 (input)
The ( $n-2$ ) elem ents of the second super-diagonal of U.

IPIVOT (input)
The pivotindioes; for $1<=i<=n$, row $i$ of the $m$ atrix $w a s$ interchanged $w$ th row $\mathbb{P} \mathbb{I V O T}(i)$. IPIVOT (i) w ill always be either $i$ or i+1; IPIVOT (i) = iindicates a row interchange was not required.

B (input/output)
O n entry, the m atrix of righthand side vectors B . On exit, B is overw rilten by the solution vectors X .

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.

INFO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the i-th argum enthad an illegalvałue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dhgeqz -im plem ent a single-/double-shiftversion of the Q Z
$m$ ethod for finding the generalized eigenvalues $\mathrm{w}(\mathcal{j})=\left(\operatorname{ALPHAR}(\mathcal{j})+\mathrm{i}^{\star} \operatorname{LPHA}(\mathcal{j})\right.$ BETAR $(\mathcal{j})$ of the equation $\operatorname{det}(A-w(i) B)=0 \quad$ In addition, the pairA, B $m$ ay be reduced to generalized Schur form

## SYNOPSIS

```
SUBROUTINEDHGEQZ (ODB,COMPQ,COMPZ,N, HO, IHI,A,LDA,B,LDB,
    ALPHAR,ALPHAI,BETA,Q,LDQ,Z,LD Z,W ORK,LWORK,INFO)
```

CHARACTER * $1 \mathrm{JOB}, \mathrm{COMPQ}, \mathrm{COMPZ}$
$\mathbb{N} T E G E R N, \mathbb{L} O, \mathbb{H} I, L D A, L D B, L D Q, L D Z, L W O R K, \mathbb{N} F O$
DOUBLE PRECISION A (LDA, *), B (LDB,*), ALPHAR (*), ALPHAI(*),
BETA ( $\left.{ }^{*}\right), \mathrm{Q}(\mathrm{LD} \mathrm{Q}, \star), \mathrm{Z}(\mathrm{LD} \mathrm{Z}, \star), \mathrm{W} O R K(*)$
SU BROUTINE DHGEQZ_64 (J B , COM PQ, COM PZ, N, $\mathbb{H} O, \mathbb{H} I, A, L D A, B, L D B$,
A LPHAR,ALPHAI,BETA, $Q, L D Q, Z, L D Z, W O R K, L W O R K, \mathbb{N} F O)$
CHARACTER * $1 \mathrm{JOB}, \mathrm{COMPQ}, \mathrm{COMPZ}$
$\mathbb{N} T E G E R * 8 N, \mathbb{L} O, \mathbb{H} I, L D A, L D B, L D Q, L D Z, L W O R K, \mathbb{N F O}$
D OUBLE PRECISION A (LDA, $\left.{ }^{*}\right)$, B (LDB, $\left.{ }^{\star}\right)$, ALPHAR (*), ALPHAI(*), $\operatorname{BETA}(*), \mathrm{Q}(\mathrm{LD} Q, \star), \mathrm{Z}(\mathrm{LD} \mathrm{Z}, \star), \mathrm{W} O R K(*)$

## F95 INTERFACE

SU BROUTINE HGEQZ (JOB , COMPQ, COMPZ, $\mathbb{N}], \mathbb{L} O, \mathbb{H} I, A,[L D A], B,[L D B]$, ALPHAR,ALPHAI, BETA, $Q$, [LDQ ], $Z,[L D Z],[W O R K],[L W O R K],[\mathbb{N} F O])$

CHARACTER (LEN=1):: JOB,COMPQ,COMPZ
$\mathbb{N} T E G E R:: N, \mathbb{L} O, \mathbb{H} I, L D A, L D B, L D Q, L D Z, L W O R K, \mathbb{N F O}$
REAL (8), D $\mathbb{M}$ ENSION (:) ::ALPHAR,ALPHAI,BETA, W ORK

REAL (8), D IM ENSION (:,:) ::A, $B, Q, Z$

SU BROUTINE HGEQ Z_64 (JOB, COM PQ, COMPZ, $\mathbb{N}], \mathbb{H}, \mathbb{H} I, A,[L D A], B$, [LD B ], A LPHAR,A LPHA I, BETA , Q, [LD Q ], Z, [LD Z], [W ORK ], [LW ORK], [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1) :: JOB,COMPQ,COMPZ
$\mathbb{N} T E G E R(8):: N, \mathbb{L}, \mathbb{H} I, L D A, L D B, L D Q, L D Z, L W O R K, \mathbb{N F O}$
REAL (8),D $\mathbb{I M} E N S I O N(:):: A L P H A R, A L P H A I, B E T A, W$ ORK
REAL (8),D $\mathbb{M}$ ENSION (:,:)::A,B, $\mathrm{Q}, \mathrm{Z}$

## C INTERFACE

\#include <sunperfh>
void dhgeqz (char job, char com pq, char com pz, int n, int ilo, int ini, double *a, int lda, double *b, int ldb, double *alphar, double *alphai, double *beta, double *q, int ldq, double *z, int ldz, int *info);
void dhgeqz_64 (char job, charcom pq, char com pz, long n, long ilo, long ini, double *a, long lda, double *b, long ldb, double *alphar, double *alphai, double *beta, double *q, long ldq, double *z, long ldz, long *info);

## PURPOSE

dhgeqz im plem ents a single-/double-shift version of the Q Z $m$ ethod for finding the generalized eigenvalues $B$ is upper triangular, and A is block upper triangular, w here the diagonal blocks are either 1 -by-1 or 2 -by- 2 , the 2 -by- 2 blocks having com plex generalized eigenvalues (see the description of the argum entJo B.)

If $J 0 B=S$ ', then the pair $(A, B)$ is sim ultaneously reduced to Schur form by applying one orthogonal tranform ation (usually called Q ) on the left and another (usually called Z) on the right. The 2 -by -2 upper-triangular diagonalblocks ofB comesponding to 2 -by -2 blocks of A w ill be reduced to positive diagonal $m$ atrices. (Ie., if $A(j+1, j)$ is non-zero, then $B(j+1, j)=B(j j+1)=0$ and $B(j)$ and $B(j+1, j+1) w i l l$ be positive.)

If $\mathrm{JO} B=\mathrm{E}$ ', then ateach iteration, the sam e transform ations are com puted, but they are only applied to those parts of A and $B$ which are needed to com pute A LPHAR, A LPH A I, and BETAR .

If $J O B=S$ 'and $C O M P Q$ and COMPZ are $V$ ' or $I^{\prime}$ ', then the
orthogonal transform ations used to reduce ( $\mathrm{A}, \mathrm{B}$ ) are accum ulated into the arrays $Q$ and $Z$ s.t.:
(in) A (in) Z (in) ${ }^{\star}=\mathrm{Q}$ (out) A (out) Z (out)*

Ref: C B.M oler \& G W . Stew art, "A n A lgorithm for Generalized M atrixigenvalue Problem s", SIAM J. Num er. A nal, 10 (1973) „. 241-256.

## ARGUMENTS

JOB (input)
$=E$ ': com pute only A LPHAR, A LPH A I, and BETA. A and $B \mathrm{w}$ ill notnecessarily be putinto generalized
Schur form . = $S^{\prime}:$ putA and B into generalized
Schur form, as w ellas com puting A LPHAR,ALPHAI, and BETA .
COMPQ (input)
$=\mathrm{N}$ : do notm odify Q .
$=\mathrm{V}$ ':m ultiply the array Q on the right by the transpose of the orthogonaltranform ation that is applied to the left side of A and B to reduce them to Schur form . = 'I': like COM PQ = V ', except that Q w illbe initialized to the identily first.

COMPZ (input)
$=\mathrm{N}$ : do notm odify Z.
$=\mathrm{V}$ ': m ultiply the amray Z on the right by the orthogonal tranform ation that is applied to the right side of $A$ and $B$ to reduce them to Schur form . = I': like C OM PZ=V', exceptthatZ will be initialized to the identily first.

N (input) The order of the $m$ atrices $A, B, Q$, and $Z . N>=0$.

IIO (input)
It is assum ed thatA is already upper triangular in row s and colum ns $1: \Pi \mathrm{O}-1$ and $\mathrm{H} \mathrm{I}+1 \mathrm{~N} .1<=\Pi 0$ $<=\mathbb{H} I<=N$, if $N>0 ; \mathbb{H}=1$ and $\mathbb{H} I=0$, if $N=0$.

IH I (input)
See the description of IIO .

A (input) On entry, the $N$ boy- $N$ upper $H$ essenberg $m$ atrix $A$.
Elem ents below the subdiagonalm ustbe zero. If
$J O B=S$ ', then on exit $A$ and $B$ will have been
sim ultaneously reduced to generalized Schur form .
If $J O B=E$ ', then on exitA w ill have been des -
troyed. The diagonalblocks w illbe corect, but
the off-diagonalportion $w$ illbe $m$ eaningless.

LD A (input)
The leading din ension of the array A. LD A $>=\max$ ( $1, \mathrm{~N})$.
$B$ (input) On entry, the $N$ boy -N upper triangular $m$ atrix $B$. Elem ents below the diagonalm ustbe zero. 2 -by- 2 blocks in $B$ corresponding to 2 -by-2 blocks in A w illbe reduced to positive diagonal form . (I.e., if $(j+1, \gamma)$ is non-zero, then $B(j+1, j)=B(j j+1)=0$ and $B(j)$ ) and $B(j+1, j+1) w$ illlbe positive.) If JO $B=S$ ', then on exit $A$ and $B$ will have been sim ultaneously reduced to Schur form. If JO B=E', then on exith w illhave been destroyed. Elem ents corresponding to diagonal blocks of A w illlbe conect, but the off-diagonal portion w ill be m eaningless.

LD B (input)
The leading dim ension of the array $B . L D B>=m a x($ 1,N ).

ALPHAR (output)
A LPHAR ( $1 \mathbb{N}$ ) w illlbe set to realparts of the diagonalelem ents of $A$ thatw ould result from reducing $A$ and $B$ to Schur form and then further reducing them both to triangular form using unitary transform ations s.t.the diagonalof B was nonnegative real. Thus, if A ( $j, j$ is in a 1 by-1 block (i.e., $A(j+1, j)=A(j j+1)=0)$, then A LPHAR $(\mathcal{j})=A(j)$. N ote that the (realor com plex) values (ALPHAR ( $\mathcal{j}$ ) $+\mathrm{i}^{\star} A \operatorname{LPHAI}(\mathcal{j})$ )BETA ( $\mathcal{j}$, $j 1, \ldots, N$, are the generalized eigenvalues of the $m$ atrix pencill $-w B$.

## A LPHA I (output)

A LPH A I( $1: \mathbb{N}$ ) w illbe set to im aginary parts of the diagonal elem ents of A that would result from reducing $A$ and $B$ to Schur form and then further reducing them both to triangular form using unitary transform ations s.t. the diagonal of $B \mathrm{w}$ as non-negative real. Thus, ifA ( 7 ) is in a 1-by-1 block (i.e., $A(j+1, j)=A(j j+1)=0)$, then A LPHAR ( $\mathcal{j}=0$. N ote that the (real orcomplex)
 $\dot{F} 1, \ldots, N$, are the generalized eigenvalues of the $m$ atrix pencilA $-w B$.
$\operatorname{BETA}(1 \mathbb{N})$ w illbe set to the (real) diagonal ele$m$ ents of $B$ thatw ould result from reducing $A$ and $B$ to Schur form and then further reducing them both to triangular form using unitary transform ations s.t. the diagonal of $B$ was non-negative real. Thus, if A (j) is in a 1-by-1 block (ie., $A(j+1, j)=A(j j+1)=0)$, then BETA $(j=B(j)$. N ote that the (real or complex) values (A LPHAR ( $\mathcal{j}$ ) i*ALPHAI ( $\mathcal{I}$ ) ABETA ( $\mathcal{j}$, $\bar{j} 1, \ldots, N$, are the generalized eigenvalues of the $m$ atrix pencil $A-w B$. N ote thatBETA ( $1: \mathbb{N}$ ) w illalw ays be non-negative, and no BETA I is necessary .)

Q (input/output)
If $C O M P Q=N$ ', then $Q \mathrm{w}$ illnotbe referenced. If $C O M P Q=V^{\prime}$ or ' I ', then the transpose of the orthogonal transform ations w hich are applied to $A$ and $B$ on the leftw ill be applied to the array $Q$ on the right.

LD Q (input)
The leading dim ension of the array $\mathrm{Q} . \mathrm{LD} \mathrm{Q}>=1$. If $C O M P Q=V$ 'or $I$ ', then $L D Q>=N$.

Z (input/output)
If COM PZ=N', then Z w ill notbe referenced. If COM PZ=V' or 'I', then the orthogonal transform ations w hich are applied to $A$ and $B$ on the right w illbe applied to the array $Z$ on the right.

LD Z (input)
The leading dim ension of the array Z . LD $\mathrm{Z}>=1$. If $\mathrm{COM} \mathrm{MZ}=\mathrm{V}$ 'or I ', then LD $\mathrm{Z}>=\mathrm{N}$.

W ORK (w orkspace)
On exit, if $\mathbb{N}$ FO >= 0,W ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array W ORK. LW ORK >= $\max (1, N)$.

IfLW O RK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
$\mathbb{I N F O}$ (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N F O}=-$ i, the $i$-th argum enthad an illegalvalue
$=1, \ldots, N$ : the QZ iteration did not converge. ( $\mathrm{A}, \mathrm{B}$ ) is not in Schur form, but A LPHAR (i), A LPHAI(i), and BETA (i), $i=\mathbb{N} F O+1, \ldots, N$ should be correct. $=\mathrm{N}+1, \ldots, 2 \star \mathrm{~N}$ : the shiftcalculation failed. ( $A, B$ ) is not in Schur form, but ALPHAR (i), ALPHAI(i), and BETA (i), $i=\mathbb{N F O}-$ $\mathrm{N}+1, \ldots, \mathrm{~N}$ should be conect. > 2*N: various "im possible" errors.

## FURTHER DETAILS

Iteration counters:

JITER - counts iterations.
ITER - counts iterations run since ILAST w as last
changed. This is therefore resetonly when a 1-
by-1 or
2-by-2 block deflates off the bottom .

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dhsein-use inverse iteration to find specified right and/or lefteigenvectors of a realupper $H$ essenberg $m$ atrix $H$

## SYNOPSIS



```
    LDVL,VR,LDVR,MM,M,W ORK,\mathbb{FA}[L,\mathbb{FA}|R,\mathbb{NFO)}
CHARACTER * 1SDDE,EIGSRC, IN ITV
INTEGERN,LDH,LDVL,LDVR,MM,M, INFO
\mathbb{NTEGER \mathbb{FA}\mathbb{L}(*),\mathbb{FA}\mathbb{H}R(*)}\=(*)
LO G ICAL SELECT (*)
DOUBLE PRECISION H (LDH,*), W R (*), W I(*), VL (LDVL,*),
VR(LDVR,*),W ORK (*)
SU BROUT\mathbb{NEDHSEIN_64(S\mathbb{DE,EIG SRC, IN ITV ,SELECT,N,H,LDH,W R,W I,}}\mathbf{~},\textrm{N},\textrm{L}
    VL,LDVL,VR,LDVR,MM,M,W ORK,\mathbb{FA}\mathbb{L},\mathbb{FA}|\mathbb{H},\mathbb{N}FO)
CHARACTER * 1SDEE,EIGSRC, IN ITV
\mathbb{NTEGER*8N,LDH,LDVL,LDVR,MM,M,INFO}
\mathbb{N}TEGER*8 \mathbb{FA}\mathbb{LL}(*),\mathbb{FA}\mathbb{HR}(*)
LOG ICAL*8 SELECT (*)
DOUBLE PRECISION H (LDH,*), W R (*), W I(*), VL (LDVL,*),
VR(LDVR,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE HSEIN (SDE,EIGSRC, $\mathbb{N} \mathbb{I T V}, \operatorname{SELECT}, \mathbb{N}], H,[L D H], W R, W I$, VL, [LDVL], VR, [LDVR], MM, M, [W ORK], $\mathbb{F} A \mathbb{L}, \mathbb{F} A \mathbb{L} R,[\mathbb{N F O}])$

CHARACTER (LEN=1) ::SDE,EIGSRC, $\mathbb{N} \mathbb{I T V}$
$\mathbb{N} T E G E R:: N, L D H, L D V L, L D V R, M M, M, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:): \mathbb{F A} \mathbb{H} L, \mathbb{F} A \mathbb{H}$
LOGICAL, D $\mathbb{I M} E N S I O N(:):: S E L E C T$
REAL (8), D $\mathbb{M}$ ENSION (:) ::W R,W I, W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) :: H, VL,VR
 W I, VL, [LDVL], VR, [LDVR], MM, M, $\mathbb{W} O R K], \mathbb{F} A \mathbb{L}, \mathbb{F} A \mathbb{L} R,[\mathbb{N} F O])$

$\mathbb{N}$ TEGER (8) :: N, LD H,LDVL,LDVR, M M, M, $\mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{F} A \amalg L, \mathbb{F} A \mathbb{I}$
LOG ICAL (8), D IM ENSION (:) :: SELECT
REAL (8), D $\mathbb{M}$ ENSION (:) ::W R , W I, W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) :: H , VL, VR

## C INTERFACE

\#include <sunperfh>
void dhsein (char side, char eigsrc, char initv, int *select, intn, double *h, int ldh, double *W $r$, double *W i, double *Vl, int ldvl, double *Vr, int ldvr, int mm , int *m, int *ifaill, int *ifailr, int *info);
void dhsein_64 (charside, char eigsrc, char initv, long *select, long $n$, double *h, long ldh, double *W $r_{\text {, }}$ double *w i, double *vl, long ldvl, double *Vr, long ldvr, long mm, long *m, long *ifaill, long *ifailr, long *info);

## PURPOSE

dhsein uses inverse iteration to find specified rightand/or lefteigenvectors of a realupper H essenberg m atrix H .

The righteigenvectorx and the lefteigenvector $y$ of the $m$ atrix $H$ corresponding to an eigenvalue $w$ are defined by:

$$
\mathrm{H}^{\star} \mathrm{x}=\mathrm{w}^{\star} \mathrm{x}, \quad \mathrm{y}^{\star * \mathrm{~h}} \star \mathrm{H}=\mathrm{w}^{\star} \mathrm{y}^{\star *} \mathrm{~h}
$$

where $y^{\star *}$ h denotes the conjugate transpose of the vectory.

## ARGUMENTS

SID E (input)
$=R$ ': com pute righteigenvectors only;
$=\mathrm{L}$ ': com pute lefteigenvectors only;
$=\mathrm{B}$ ': com pute both right and lefteigenvectors.

E IG SRC (input)
Specifies the souroe of eigenvalues supplied in ( N R, W I):
= Q ': the eigenvalues were found using SHSEQR; thus, if H has zero subdiagonalelem ents, and so is block-triangular, then the jth eigenvalue can be assum ed to be an eigenvalue of the block containing the jth row /colum $n$. This property allow s SHSEIN to perform inverse iteration on justone diagonalblock. = N ': no assum ptions are $m$ ade on the correspondence betw een eigenvalues and diagonalblocks. In this case, SH SE $\mathbb{I N}$ m ustalw ays perform inverse tieration using the w hole $m$ atrix $H$.
$\mathbb{N} \mathbb{I T V}$ (input)
= N ': no initial vectors are supplied;
= U ': user-supplied initial vectors are stored in the arrays VL and/orVR.

## SELECT (input/output)

Specifies the eigenvectors to be com puted. To select the real eigenvector corresponding to a realeigenvalueW R ( $\mathcal{j}$ ), SELEC T ( 7 ) m ust be set to TRUE.. To select the complex eigenvector corresponding to a complex eigenvalue ( N R ( $\mathcal{j}$, N $\mathrm{I}(\mathcal{J})$ ), with complex conjugate

orboth m ustbe set to

N (input) The order of the matrix $\mathrm{H} . \mathrm{N}>=0$.

H (input) The upperH essenberg $m$ atrix H .

LD H (input)
The leading din ension of the amay H . LD H >= $\max (1, N)$.

W R (input/output)
On entry, the real and im aginary parts of the eigenvahues of H ; a complex conjugate pairof eigenvaluesm ustbe stored in consecutive elem ents
of W R and W I. On exit, W R m ay have been altered
since close eigenvalues are perturbed slightly in searching for independenteigenvectors.

W I (input)
See the description ofW R.

V L (input/output)
On entry, if $\mathbb{N} \operatorname{ITV}=\mathrm{U}$ 'and $S \mathbb{D E}=\mathrm{L}$ 'or $\mathrm{B}^{\prime}, \mathrm{VL}$
$m$ ust contain starting vectors for the inverse iteration for the lefteigenvectors; the starting vector for each eigenvectorm ustbe in the sam e colum $n(s)$ in which the eigenvectorw illbe stored. On exit, if $S \mathbb{D} E=$ L' or B', the lefteigenvectors specified by SELECT w ill.be stored consecutively in the colum ns ofV L, in the sam e order as their eigenvahues. A complex eigenvector corresponding to a com plex eigenvalue is stored in tw o consecutive colum ns , the first holding the real part and the second the im aginary part. If $S \mathbb{D E}=\mathrm{R}, \mathrm{VL}$ is notreferenced.

LDVL (input)
The leading dim ension of the array $\mathrm{VL} . \mathrm{LDVL}>=$ $\max (1, N)$ if $S \mathbb{D} E=L$ 'or $B^{\prime} ;$ LDVL $>=1$ other wise.

VR (input/output)
On entry, if $\mathbb{N}$ ITV $=U$ 'and $S \mathbb{D} E=R$ 'or $B^{\prime}, V R$ $m$ ust contain starting vectors for the inverse teration for the righteigenvectors; the starting vector for each eigenvectorm ustbe in the sam e colum $n(s)$ in $w$ hich the eigenvectorw ill.be stored. On exit, ifSIDE = R 'or B', the righteigenvectors specified by SELECT w ill.be stored consecutively in the colum ns ofVR, in the sam e order as their eigenvahues. A complex eigenvector corresponding to a com plex eigenvalue is stored in tw o conseculive colum ns , the first holding the real part and the second the im aginary part. If $S \mathbb{D E}=\mathbb{L}, \mathrm{VR}$ is notreferenced.

LDVR (input)
The leading dim ension of the array $V R$. LDVR >= $\max (1, N)$ if $S \mathbb{D} E=R$ 'or $B^{\prime} ; \operatorname{LDVR}>=1$ otherw ise.

M M (input)
The num ber of colum ns in the arrays $V L$ and/or $V R$. $M M>=M$.

M (output)
The num ber of colum $n s$ in the arrays $V L$ and/or VR required to store the eigenvectors; each selected realeigenvector occupies one column and each selected com plex eigenvector occupies tw o colum ns.

W ORK (w orkspace)
dim ension $(\mathbb{N}+2) * N)$

FAIIL (output)
 left eigenvector in the ith column of VL (comesponding to the eigenvalue w (J) failed to converge; $\mathbb{F A} \Pi L(i)=0$ ifthe eigenvectorconverged satisfactorily. If the i-th and (i+1)th colum ns of VL hold a com plex eigenvector, then FAIIL (i) and $\mathbb{F A} \Pi L$ (i+1) are set to the same value. If $S \mathbb{D} E=R$ ', $\mathbb{F} A \mathbb{I} L$ is notreferenced.

FAIIR (output)
If $S \mathbb{D} E=R$ 'or $B$ ', $\mathbb{F} A \mathbb{I} R(i)=j>0$ if the right eigenvector in the i-th colum $n$ of $V R$ (comesponding to the eigenvalue w (J) failed to converge; $\mathbb{F A} \Pi R(i)=0$ ifthe eigenvectorconverged satisfactorily. If the $i$ th and (i+1)th colum ns of VR hold a com plex eigenvector, then FAMR (i) and $\mathbb{F A} \Pi R(i+1)$ are set to the same value. IfS $\mathbb{E}=\mathbb{L}$ ', $\mathbb{F} A \mathbb{L}$ is notreferenced.
$\mathbb{I N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the i-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N} F O=i$, $i$ is the num ber of eigenvectors w hich failed to converge; see IFA IIL and IFA IIR for further details.

## FURTHER DETAILS

Each eigenvector is norm alized so that the elem ent of largest $m$ agnitude has $m$ agnitude 1 ; here the $m$ agnitude of a com plex num ber $(x, y)$ is taken to be $|x|+|y|$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dhseqr-com pute the eigenvalues of a real upper H essenberg $m$ atrix $H$ and, optionally, the $m$ atrices $T$ and $Z$ from the Schurdecom position $H=Z \mathrm{~T} \quad \mathrm{Z} * * \mathrm{~T}$, where T is an upper quasi-triangular $m$ atrix (the Schur form ), and $Z$ is the orthogonalm atrix of Schurvectors

## SYNOPSIS

```
SU BROUTINE DHSEQR(JOB,COMPZ,N,\mathbb{LO,HHI,H,LDH,W R,W I, Z,LD Z,}
    W ORK,LW ORK,NNFO)
CHARACTER * 1 J B ,COM PZ
\mathbb{NTEGERN, ILO,\mathbb{H I,LDH,LD Z,LW ORK,INFO}}\mathbf{N}\mathrm{ , LN}
DOUBLE PRECISION H (LDH,*),W R (*),W I(*),Z (LD Z ,*),W ORK (*)
SUBROUTINE DHSEQR_64(OOB,COM PZ,N,\mathbb{LO,HIN,H,LDH,W R,W I, Z,LD Z,}
    W ORK,LW ORK,INFO)
CHARACTER * 1 JOB,COMPZ
```



```
DOUBLE PRECISION H (LDH,*),W R (*),W I(*),Z (LD Z,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE HSEQR (JOB,COMPZ,N, $\mathbb{H} O, \mathbb{H} I, H,[L D H], W R, W I, Z,[L D Z]$, [ $\mathbb{N}$ ORK ], [LW ORK ], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1)::JOB,COMPZ
$\mathbb{N} T E G E R:: N, \mathbb{L} O, \mathbb{H} I, L D H, L D Z, L W O R K, \mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (:) ::WR,W I,W ORK
REAL (8),D $\mathbb{M}$ ENSION (:,:) :: H , Z
SU BROUTINE HSEQR_64 (OOB,COMPZ,N, $\mathbb{L} O, \mathbb{H} I, H,[L D H], W R, W I, Z$,
[LD Z], [W ORK ], [LW ORK ], [ $\mathbb{N} F \mathrm{FO}$ ])

CHARACTER (LEN=1) :: JOB,COM PZ
$\mathbb{N}$ TEGER (8) :: N, $\mathbb{H} O, \mathbb{H} I, L D H, L D Z, L W O R K, \mathbb{N} F O$ REAL (8),D $\mathbb{M}$ ENSION (:) ::W R,W I,W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) :: H , Z

## C INTERFACE

\#include <sunperfh>
void dhseqr(char job, char com pz, intn, int ilo, int ihi, double *h, int ldh, double *w r, double *w i, double *z, int ldz, int *info);
void dhseqr_64 (char jं.b, char com pz, long n, long ilo, long ini, double *h, long ldh, double *w r, double *w i, double *z, long ldz, long *info);

## PURPOSE

dhseqr com putes the eigenvalues of a real upper $H$ essenberg $m$ atrix $H$ and, optionally, the $m$ atrices $T$ and $Z$ from the Schurdecomposition $H=Z \mathrm{~T} \mathrm{Z**T}$, where T is an upper quasi-triangular $m$ atrix (the Schur form ), and $Z$ is the orthogonalm atrix of Schurvectors.

O ptionally Z m ay be postm ultiplied into an input orthogonal $m$ atrix $Q$, so that this routine can give the Schur factorization of a m atrix A which has been reduced to the $H$ essenberg form $H$ by the orthogonal $m$ atrix $Q: A=Q * H * Q * T=$ (Q Z) ${ }^{*} \mathrm{~T}^{*}(\mathrm{Q} Z){ }^{* *} \mathrm{~T}$.

## ARGUMENTS

JOB (input)
= E ': com pute eigenvalues only;
$=S^{\prime}$ : com pute eigenvalues and the Schur form T .

COMPZ (input)
$=\mathrm{N}$ ': no Schurvectors are com puted;
$=I^{\prime}: \mathrm{Z}$ is in inialized to the unit m atrix and
the m atrix Z of churvectors of H is retumed; $=$ V ': Z mustcontain an orthogonal matrix Q on entry, and the productQ *Z is retumed.

N (input) The order of the m atrix $\mathrm{H} . \mathrm{N}>=0$.

It is assum ed that H is already upper triangular in row s and colum ns 1: $\mathbb{T}-1$ and $\mathbb{H} \mathrm{I}+1 \mathrm{~N} . \mathbb{I} \mathrm{O}$ and $\mathbb{H}$ I are norm ally setby a previous call to SG EBA L, and then passed to SGEHRD w hen them atrix output by SG EBA $L$ is reduced to $H$ essenberg form. O therw ise HO and $\mathbb{H} I$ should be set to 1 and $N$ respectively. $1<=\mathbb{H O}<=\mathbb{H} \mathrm{I}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{H} \mathrm{O}=1$ and $\mathbb{H} \mathrm{I}=0$, if $\mathrm{N}=0$.

IH I (input)
See the description of IIO .
H (input/output)
On entry, the upper H essenberg $m$ atrix H . On exit, if $J O B=S ', H$ contains the upper quasitriangularm atrix $T$ from the Schur decom position (the Schur form); 2-by-2 diagonal blocks (comesponding to com plex conjugate pairs of eigenvalues) are retumed in standard form, w ith $\mathrm{H}(i, i)=H(i+1, i+1)$ and $H(i+1, i) * H(i, i+1)<0$. If $\mathrm{JOB}=\mathrm{E}$ ', the contents of H are unspecified on exit.

LD H (input)
The leading din ension of the array H. LD H >= $\max (1, N)$.

W R (output)
The real and im aginary parts, respectively, of the com puted eigenvalues. Iftw o eigenvalues are com puted as a com plex conjugate pair, they are stored in consecutive elem ents of $W R$ and $W I$, say the $i$-th and (i+1)th, with W I(i) > 0 and $W$ I(i+1) < 0 . If $J 0 B=S$ ', the eigenvalues are stored in the sam e order as on the diagonal of the Schur form retumed in $H$, w ith $W$ R (i) $=\mathrm{H}(i, i)$ and, if H ( $i: i+1, i: i+1$ ) is a 2 -by -2 diagonalblock, $W$ I(i) $=$ squt $(\mathrm{H}(i+1, i) * H(i, i+1))$ and $W I(i+1)=-W I(i)$.

W I (output)
See the description ofW R .

Z (input) IfCOM PZ = N ': Z is not referenced.
If COM PZ = I': on entry, Z need notbe set, and on exit, $Z$ contains the orthogonalm atrix $Z$ of the Schurvectors of H . IfCOM PZ = V ': on entry Z m ust contain an N -by -N m atrix Q , which is assum ed to be equal to the unitm atrix except for the sub$m$ atrix $Z$ ( $\mathbb{H} O: \mathbb{H} I, \mathbb{I} O: \mathbb{H} I$ ); on exitZ contains $Q$ *Z . N orm ally $Q$ is the orthogonalm atrix generated by SORGHR after the call to SGEHRD which form ed the

H essenberg $m$ atrix $H$.

LD Z (input)
The leading dim ension of the array Z . LD $\mathrm{Z}>=$ $\max (1, N)$ if COMPZ = I'orV';LD $Z>=1$ otherwise.
W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the array W ORK. LW ORK >= $\max (1, N)$.

IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
$\mathbb{N} F O$ (output)
= 0: successfulexit
<0: if $\mathbb{I N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue
> 0 : if $\mathbb{N} F O=i, S H$ SEQR failed to com pute allof the eigenvalues in a total of 30* ( $\mathbb{H} \mathrm{I}-\mathbb{H O}+1$ ) iterations; elem ents 1 :ilo- 1 and i+ 1 m of $\mathrm{W} R$ and W I contain those eigenvalues w hich have been successfully com puted.

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

## djadm m - Jagged diagonalm atrix-m atrix m ultiply (m odified

 Ellpack)
## SYNOPSIS

SUBROUTINEDJADMM (TRANSA, M, N, K,ALPHA,DESCRA,

* VAL, $\mathbb{N} D \mathrm{X}, \mathrm{PNTR}, \mathrm{MAXNZ}, \mathbb{P} E R M$,
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
$\mathbb{N} T E G E R$ TRANSA, M,N,K,DESCRA (5),MAXNZ,
* LDB,LDC,LW ORK
$\mathbb{N} T E G E R \quad \mathbb{N} D X(N N Z), P N T R(M A X N Z+1), \mathbb{P E R M}(M)$
DOUBLE PRECISION ALPHA,BETA
D OUBLE PRECISION VAL (NNZ), B (LDB,*), C (LDC, $\left.{ }^{*}\right), \mathrm{W}$ ORK (LW ORK)

SUBROUTINED JADMM_64(TRANSA, M,N,K,ALPHA,DESCRA,

* VAL, $\mathbb{N} D \mathrm{X}, \mathrm{PNTR}, \mathrm{MAXNZ}, \mathbb{P} E R M$,
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
$\mathbb{N} T E G E R * 8$ TRANSA, M,N,K,DESCRA (5),MAXNZ,
* LDB,LDC,LW ORK
$\mathbb{N} T E G E R * 8 \mathbb{N} D X(N N Z), P N T R(M A X Z+1), \mathbb{P E R M}(M)$
DOUBLE PRECISION ALPHA, BETA
DOUBLE PRECISION VAL (NNZ),B (LDB,*), C (LDC, $\left.{ }^{*}\right), W$ ORK (LW ORK)
where NN Z=PN TR M AXNZ+1)-PN TR (1)+1 is the num berofnon-zero elem ents


## F95 INTERFACE

SUBROUTINE JADMM (TRANSA, M, $\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N} D X$, * PNTR,MAXNZ, $\mathbb{P} E R M, B,[L D B], B E T A, C,[L D C],[W$ ORK], [LW ORK]) $\mathbb{N} T E G E R$ TRANSA, M, K, MAXNZ
$\mathbb{I}$ TEGER,D $\mathbb{M}$ ENSION (:) :: DESCRA, $\mathbb{N D D X , P N T R , ~ \mathbb { P E R M }}$
D OUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D IM ENSION (:) :: VAL

DOUBLE PRECISION ,D $\mathbb{M} E N S I O N(:,:):$ B, C

SUBROUTINE JADMM_64 (TRANSA, M, $\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N D X}$,

* PNTR,MAXNZ, $\mathbb{P} E R M, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])$
$\mathbb{N}$ TEGER*8 TRANSA, M, K, MAXNZ
$\mathbb{N}$ TEGER*8, D $\mathbb{M}$ ENSION (:) :: DESCRA, $\mathbb{N} D X, P N T R, ~ \mathbb{P E R M}$
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D $\mathbb{M} E N S I O N(:):: V A L$
DOUBLE PRECISION ,D IM ENSION (:, :) :: B , C


## DESCRIPTION

$$
C<- \text { alpha op (A) B + beta C }
$$

where A LPHA and BETA are scalar, $C$ and $B$ are dense $m$ atrioes, $A$ is a m atrix represented in jagged-diagonal form at and op (A) is one of $o p(A)=A$ or $o p(A)=A^{\prime}$ or $o p(A)=\operatorname{conjg}\left(A^{\prime}\right)$.
( 'indicates m atrix transpose)

## ARGUMENTS

TRANSA Indicates how to operate $w$ th the sparse $m$ atrix
0 : operate $w$ ith $m$ atrix
1 : operate $w$ ith transpose $m$ atrix
2 : operate $w$ th the conjugate transpose ofm atrix. 2 is equivalentto 1 ifm atrix is real.

M $\quad N$ um berof row $s$ in matrix A

N $\quad$ um berof colum ns in matrix C

K $\quad \mathrm{N}$ um berof colum ns in $m$ atrix $A$

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem ent integer anay
DESCRA (1) m atrix structure
0 : general
1 : symm etric ( $A=A$ )
2: Herm itian ( $A=C O N J(A)$ )
3 : Triangular
4 : Skew (A nti)-Symm etric ( $\mathrm{A}=-\mathrm{A}$ )
5 :D iagonal
6 : Skew Herm itian ( $A=-C O N J(A)$ )
D ESCRA (2) upper/low er triangular indicator
1 : low er

2 :upper
DESCRA (3) main diagonal type
0 :non-unit
1 :unit
DESCRA (4) A ray base $\mathbb{N} O T \mathbb{I}$ PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
D ESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 :no repeated indices

VAL () array of length NN Z consisting of entries of A. VA L can be view ed as a colum $n m$ ajorordering of a row perm utation of the Ellpack representation of , where the Ellpack representation is perm uted so that the row s are non-increasing in the num ber of nonzero entries. V alues added forpadding in E llpack are not included in the Jagged-D iagonal form at.
$\mathbb{I N D X}$ () array of length NNZ consisting of the colum n indices of the comesponding entries in VAL.
PN TR () array of length M AXNZ+1, where PNTR (I)-PN TR (1)+1 points to the location in VAL of the firstelem ent in the row -perm uted E llpack represenation of $A$.

MAXNZ max num berofnonzeros elem ents per row .
$\mathbb{P E R M} 0$ integer array of length $M$ such that $I=\mathbb{P E R M}$ ( $I$ ), where row I in the originalE llpack representation corresponds to row I' in the perm uted representation. If $\operatorname{PERM}(1)=0$, it is assum ed by convention that $\mathbb{P E R M}(\mathbb{I})=I . \mathbb{P E R M}$ is used to determ ine the order in which row s ofC are updated.

B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension of $B$
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension of $C$
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is notreferenced in the current version.

## SEE ALSO

N IST FO RTRA N Sparse B las U ser's G uide available at: http://m ath nistgov/n cso/Staff/k Rem ington/Ispblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

## Contents

- NAME
- SYNOPSIS

> - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

```
djadrp - rightperm utation of a jagged diagonalm atrix
```


## SYNOPSIS

SUBROUT INED JADRP (TRANSP, M, K, VAL, $\mathbb{N} D X$, PNTR, MAXNZ, * IPERM , W ORK, LW ORK)
$\mathbb{N}$ TEGER TRANSP, M, K, MAXNZ,LWORK
$\mathbb{N} T E G E R \quad \mathbb{N} D X(*)$, PNTR (MAXNZ+1), $\mathbb{P E R M}(\mathbb{K})$, W ORK (LW ORK) DOUBLE PRECISIONVAL (*)

SUBROUTINED JADRP_64 (TRANSP, M, K, VAL, INDX,PNTR,MAXNZ, * $\quad \mathbb{P E R M}$, $W$ ORK, LW ORK)
$\mathbb{N}$ TEGER*8 TRANSP, M, K, MAXNZ,LW ORK
$\mathbb{N T E G E R * 8} \mathbb{N} D X(*), \operatorname{PNTR}(M A X N Z+1), \mathbb{P E R M}(K), W$ ORK (LWORK)
DOUBLE PRECISION VAL (*)

## F95 INTERFACE

SUBROUTINE JADRP (TRANSP, M, K, VAL, $\mathbb{N} D X, P N T R, M A X N Z$, * $\quad \mathbb{P} E R M,[W O R K],[L W O R K])$
$\mathbb{N}$ TEGER TRANSP, M, K, MAXNZ
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{N} D X, P N T R, \mathbb{P E R M}$
D OUBLE PRECISION ,D IM ENSION (:) ::VAL

SUBROUTINE JADRP_64 (TRANSP, M, K, VAL, INDX,PNTR,MAXNZ,

* $\quad \mathbb{P E R M}$, [WORK], [LWORK])
$\mathbb{N}$ TEGER * 8 TRAN SP, M, K, MAXNZ
$\mathbb{N}$ TEGER*8,D $\mathbb{M}$ ENSION (:) :: $\mathbb{N} D \mathrm{X}$, PNTR, $\mathbb{P} E R M$
DOUBLE PRECISION ,D $\mathbb{I M} E N S I O N(:):: V A L$

DESCRIPTION

A $<-A P$
$A<-A P^{\prime}$
( 'indicates m atrix transpose)
$w$ here perm utation $P$ is represented by an integervector $\mathbb{P} E R M$, such that $\mathbb{P E R M}(I)$ is equal to the position of the only nonzero elem entin row Iofperm utation $m$ atrix $P$.

N O TE : In orderto get a sym etrically perm uted jagged diagonal $m$ atrix P A P', one can explicitly perm ute the colum ns P A by calling

SJADRP ( $0, M, M, V A L, \mathbb{N} D X, P N T R, M A X N Z, \mathbb{P} E R M, W$ ORK,LW ORK)
where param eters $V A L, \mathbb{N D X}, P N T R, M A X N Z, \mathbb{P E R M}$ are the representation of $A$ in the jagged diagonal form at. The operation $m$ akes sense if the originalm atrix $A$ is square.

## ARGUMENTS

TRAN SP Indicates how to operate $w$ ith the perm utation $m$ atrix
0 : operate w ith m atrix
1 : operate $w$ ith transpose $m$ atrix

M $\quad \mathrm{N}$ um berof row $s$ in matrix A

K $\quad \mathrm{N}$ um ber of colum ns in matrix A

VAL () amay of length PNTR MAXNZ+1)-PNTR (1) consisting of entries ofA. VA L can be view ed as a colum $n m$ ajor ordering of a row perm utation of the E llpack representation of A, w here the Ellpack representation is perm uted so that the row $s$ are non-increasing in the num ber of nonzero entries. $V$ alues added for padding in Ellpack are not included in the Jagged - D iagonal form at.

INDX () array of length PN TR MAXNZ+1)-PNTR (1) consisting of the colum $n$ indices of the corresponding entries in VAL.

PNTR () array of length M AXNZ+1, where PNTR (I) PNTR (1)+1 points to the location in VA L of the firstelem ent in the row -perm uted E lhpack represenation of .

M A X N Z max num ber ofnonzeros elem ents per row.
$\mathbb{P} E R M$ ( integeramay of length $K$ such that $I=\mathbb{P} E R M$ ( $I$ ).

A ray $\mathbb{P} E R M$ represents a perm utation $P$, such that $\mathbb{P E R M}$ ( I ) is equal to the position of the only nonzero elem ent in row Iofperm utation $m$ atrix $P$.
Forexam ple, if
|001|
$\mathrm{P}=\left|\begin{array}{lll}1 & 0 & 0\end{array}\right|$
|010|
then $\mathbb{P E R M}=(3,1,2)$.

W ORK () scratch array of length LW ORK. LW ORK should be at leastK.

LW ORK length ofW ORK aray

If LW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

## SEE ALSO

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/K Rem ington/tspblas/
"D ocum ent for the B asic $L$ inearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib org/utk/papers/sparse _ps

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

```
djadsm -Jagged-diagonal form at triangular solve
```


## SYNOPSIS

```
SUBROUTINEDJADSM(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,
* VAL,\mathbb{NDX,PNTR,MAXNZ,\mathbb{PERM,}}\mathbf{N},
* B,LDB,BETA,C,LDC,W ORK,LWORK)
INTEGER TRANSA,M,N,UNITD,DESCRA (5),MAXNZ,
* LDB,LDC,LW ORK
\mathbb{NTEGER INDX NNZ),PNTR MAXNZ+1),\mathbb{PERM M)}}\mathbf{M}\mathrm{ (N)}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION DV M),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
```

SU BROUTINED JAD SM _64 (TRANSA , M, N, UN ITD, DV ,A LPHA, DESCRA,

* VAL, $\mathbb{N} D \mathrm{X}, \mathrm{PNTR}, \mathrm{MAXNZ}, \mathbb{P} E R M$,
* B,LDB,BETA, C,LDC,W ORK,LW ORK )
$\mathbb{N} T E G E R * 8$ TRANSA, M,N,UNTID,DESCRA (5), MAXNZ,
* LDB,LDC,LWORK
$\mathbb{N} T E G E R * 8 \mathbb{N} D X(N N Z), P N T R(M A X N Z+1), \mathbb{P E R M}(M)$
DOUBLE PRECISION ALPHA, BETA
DOUBLE PRECISION DV M),VAL (NNZ),B (LDB,*), C (LDC,*),W ORK (LW ORK)
where NN Z=PN TR M A XNZ+1)-PN TR (1)+1 is the num berofnon-zero elem ents


## F95 INTERFACE

SUBROUTINE JADSM (TRANSA, M, $\mathbb{N}], U N \mathbb{I T D , D V , A L P H A , D E S C R A , V A L , ~} \mathbb{N} D X$,

* PNTR,MAXNZ, $\mathbb{P E R M}, \mathrm{B},[\mathrm{LD} \mathrm{B}], \mathrm{BETA}, \mathrm{C},[\mathrm{LD} \mathrm{C}],[W \mathrm{ORK}]$, [LW ORK ])
$\mathbb{N} T E G E R$ TRANSA, M, MAXNZ
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: D E S C R A, \mathbb{N D} X, P N T R, \mathbb{P E R M}$
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D $\mathbb{M}$ ENSION (:) :: VAL,DV
DOUBLE PRECISION ,D $\mathbb{M}$ ENSION (: : : :: B, C

SUBROUTINE JAD SM _64 (TRANSA, M, $\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X$,

* PNTR,MAXNZ, $\mathbb{P E R M}, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])$
$\mathbb{N} T E G E R * 8$ TRANSA, M, MAXNZ
$\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: D E S C R A, \mathbb{N} D X, P N T R, \mathbb{P} E R M$
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION ,D $\mathbb{M}$ ENSION (:) :: VAL, DV
DOUBLE PRECISION ,D $\mathbb{M}$ ENSION (:, :) :: B , C


## DESCRIPTION

$$
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { Op (A) B + BETA C } \\
& C<-A L P H A \text { Op (A)D B + BETA } C
\end{aligned}
$$

where A LPH A and BETA are scalar, $C$ and $B$ are $m$ by $n$ dense $m$ atrices, $D$ is a diagonalscaling $m$ atrix, $A$ is a unit, ornon-unit, upper or low ertriangularm atrix represented in jagged-diagonal form at and $o p(A)$ is one of $\operatorname{op}(A)=\operatorname{inv}(A)$ or op (A) $=\operatorname{inv}(A) \operatorname{or} \operatorname{op}(A)=\operatorname{inv}\left(\operatorname{conjg}\left(A^{\prime}\right)\right)$ (inv denotesm atrix inverse, 'indicates m atrix transpose)

## ARGUMENTS

TRAN SA Indicates how to operate $w$ th the sparse $m$ atrix 0 : operate $w$ ith $m$ atrix 1 : operate $w$ ith transpose $m$ atrix
2 : operate $w$ ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M $\quad N$ um berof row $s$ in $m$ atrix A

N $\quad N$ um berof colum ns in matrix $C$

UN ITD Type of scaling:
1 : Identity matrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)
4 : A utom atic row scaling (see section N O TES for further details)

DV () A rray of length M containing the diagonalentries of the scaling diagonalm atrix D .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay D ESCRA (1) m atrix structure

0 : general
1 : symm etric ( $A=A$ )
2 : Herm ( $\mathrm{A}=\mathrm{CONJG}$ (A))
3 :Triangular
4 : Skew (A nti)-Symm etric ( $A=-A$ )
5 :D iagonal
6 : Skew Herm itian ( $\mathrm{A}=-\mathrm{CON}$ J ( A ) )
N ote:For the routine, DESCRA (1)=3 is only supported.
DESCRA (2) upper/low er triangular indicator
1: low er
2 : upper
DESCRA (3) m ain diagonaltype
0 :non-unit
1 : unit
DESCRA (4) A ray base $\mathbb{N} O T \mathbb{M}$ PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? $\mathbb{N} O T \mathbb{I}$ PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL 0 array of length NNZ consisting of entries of A. VA L can be view ed as a colum $n m$ ajorordering of a row perm utation of the Ellpack representation of A, where the Ellpack representation is perm uted so that the row s are non-increasing in the num ber of nonzero entries. V alues added forpadding in E llpack are not included in the Jagged-D iagonal form at.
$\mathbb{N} D \mathrm{X} 0 \quad$ array of length NNZ consisting of the colum n indices of the comesponding entries in VAL.

PNTR 0) array of length M AXNZ +1 , where PNTR ( 1 ) PNTR (1) +1 points to the location in VA L of the firstelem ent in the row -perm uted $E$ lipack represenation of $A$.

MAXNZ max num berofnonzeros elem ents per row .
$\mathbb{P E R M}$ ) integer array of length M such that $\mathrm{I}=\mathbb{P} E R M$ ( I ), where row I in the originalE llpack representation corresponds to row I' in the perm uted representation. If $\operatorname{PERM}(\mathbb{1})=0$, its assum ed by convention that $\mathbb{P E R M}(\mathbb{I})=I . \mathbb{P E R M}$ is used to determ ine the order in which row s ofC are updated.

B 0 rectangular array w th first dim ension LD B.

LD B leading dim ension ofB
BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension of $C$

W ORK () scratch array of length LW ORK.
On exit, ifLW ORK $=-1, W$ ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK aray. LW ORK should be at least2*M.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on multiple processors, LW ORK $>=2 * \mathrm{M}$ *N_CPUS where N_CPUS is the maxim um num berof processors available to the program.

If LW ORK $=0$, the routine is to allocate $w$ orkspace needed.
IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

## SEE ALSO

N IST FORTRAN Sparse B las U ser's G uide available at:
http://m ath nistgov/m csd/Staff/k Rem ington/Ispblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

## NOTES/BUGS

1. No test for singularity ornear-singularity is included in this routine. Such tests $m$ ust.be perform ed before calling this routine.
2. If U N ITD $=4$, the routine scales the row s ofA such that their 2 -norm s are one. The scaling $m$ ay im prove the accuracy of the com puted solution. C orresponding entries ofV A L are changed only in the particular case. On retum DV m atrix stored as a vector contains the diagonalm atrix by w hich the row s have been scaled. U N ITD = 2 should be used for the next calls to the routine w ith overw ritten VAL and DV.

WORK (1)=0 on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here $i$ is the row num berw hich 2 -norm is exactly zero.
3. If $\operatorname{DESCRA}(3)=1$ and UN ITD < 4, the unitdiagonalelem ents $m$ ightorm ightnotbe referenced in the JA D representation of a sparse m atrix. They are notused anyw ay in these cases. ButifUN ITD=4, the unit diagonalelem ents M U ST be referenced in the $\sqrt{A} D$ representation.
4.The routine can be applied for solving triangular system $s$ w hen the upper or low er triangle of the general sparse m atrix A is used. H ow ever DESCRA (1) m ust.be equal to 3 in this case.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dlagtf-factorize the $m$ atrix ( I -lam bda夫 I ), where T is an n
by n tridiagonal m atrix and lam bda is a scalar, as T lam bda*I = PLU

## SYNOPSIS

```
SUBROUT\mathbb{NE DLAGTF N,A,LAM BDA,B,C,TOL,D , NN, NNFO )}
INTEGER N,\mathbb{NFO}
\mathbb{NTEGER \mathbb{N (*)}}\mathbf{(})
DOUBLE PRECISION LAM BDA,TOL
D OUBLE PRECISION A (*),B (*),C (*),D (*)
SUBROUT\mathbb{NEDLAGTF_64 N,A,LAM BDA,B,C,TOL,D,IN,INFO )}
\mathbb{NTEGER*8 N,\mathbb{NFO}}\mathbf{N}=0
\mathbb{NTEGER*8 \mathbb{N (*)}}\mp@subsup{}{(}{*})
DOUBLE PRECISION LAMBDA,TOL
DOUBLE PRECISION A (*),B (*),C (*),D (*)
```


## F95 INTERFACE

SU BROUTINE LAGTF ( $\mathbb{N}], A, L A M B D A, B, C, T O L, D, \mathbb{N},[\mathbb{N} F O])$
$\mathbb{N}$ TEGER ::N, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{N}$
REAL (8) :: LAM BDA,TOL
REAL (8),D $\mathbb{M}$ ENSION (:) ::A,B,C,D

SU BROUTINE LAGTF_64 ( $\mathbb{N}], A, L A M B D A, B, C, T O L, D, \mathbb{N},[\mathbb{N} F O])$
$\mathbb{N}$ TEGER ( 8 ) :: N, $\mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{N}$

REAL (8) :: LAM BDA, TOL
REAL (8),D IM ENSION (:) ::A,B,C,D

## C INTERFACE

\#include < sunperfh>
void dlagtf(intn, double *a, double lam bda, double *b, double *c, double tol, double *d, int *in, int *info);
void dlagtf_ 64 long $n$, double *a, double lam bda, double *b, double *c, double tol, double *d, long *in, long *info);

## PURPOSE

dlagtf factorizes the $m$ atrix ( $I-l a m$ bda* $I$ ), where $T$ is an $n$ by $n$ tridiagonalm atrix and lam bda is a scalar, as w here $P$ is a perm utation $m$ atrix, $L$ is a unit lower tridiagonal $m$ atrix $w$ ith atm ostone non-zero sub-diagonalelem ents per colum $n$ and $U$ is an uppertriangularm atrix $w$ th atm ost tw o non-zero super-diagonalelem ents per colum $n$.

The factorization is obtained by $G$ aussian elim ination $w$ ith partialpivoting and im plicit row scaling.

The param eterLAMBDA is included in the routine so that SLAGTF may be used, in conjunction w ith SLAGTS, to obtain eigenvectors of $T$ by inverse iteration.

## ARGUMENTS

N (input) The orderof the m atrix T .

A (input/output)
On entry, A m ust contain the diagonalelem ents of T.

On exit, A is overw ritten by the $n$ diagonal ele$m$ ents of the upper triangularm atrix $U$ of the factorization of $T$.

LAM BDA (input)
O n entry, the scalar lam bda.
B (input/output)
O $n$ entry, $B m$ ust contain the ( $n-1$ ) super-diagonal
elem ents of $T$.

On exit, B is overw rilten by the ( $\mathrm{n}-1$ ) superdiagonal elem ents of the $m$ atrix $U$ of the factorization of $T$.

C (input/output)
On entry, C m ustcontain the ( $\mathrm{n}-1$ ) sub-diagonal elem ents of $T$.

On exit, $C$ is overw rilten by the ( $n-1$ ) subdiagonal elem ents of the $m$ atrix $L$ of the factorization of $T$.
TOL (input/output)
On entry, a relative tolerance used to indicate whetherornot the $m$ atrix ( $T$-lam bda*I) is nearly singular. TO L should norm ally be chose as approxi$m$ ately the largest relative emror in the elem ents of T.For exam ple, if the elem ents of T are comect to about 4 significant figures, then TO L should be set to about $5 * 10 * *(-4)$. If TOL is supplied as less than eps, w here eps is the relative $m$ achine precision, then the value eps is used in place of TOL .

D (output)
Onexit, $D$ is overw ritten by the ( $n-2$ ) second super-diagonal elem ents of the $m$ atrix $U$ of the factorization of $T$.
$\mathbb{N}$ (output)
On exit, $\mathbb{N}$ contains details of the perm utation $m$ atrix $P$. If an interchange occumed at the $k$ th step of the elim ination, then $\mathbb{N}(\mathbb{K})=1$, otherw ise $\mathbb{N}(k)=0$.The elem ent $\mathbb{N}(n)$ retums the sm allest positive integer jsuch that
abs(u (خز) ) le.nom ( (T -lam bda*I) (ך) )*TO L,
where norm (A $(\mathcal{J})$ ) denotes the sum of the absolute values of the th row of the $m$ atrix A. If no such jexists then $\mathbb{N}(n)$ is retumed as zero. If $\mathbb{I N}(n)$ is retumed as positive, then a diagonalelem ent of U is sm all, indicating that ( $\mathrm{T}-\mathrm{lam}$ bda*I) is singularornearly singular,
$\mathbb{N}$ FO (output)
= 0 : successfulexit

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dlam rg -w ill create a perm utation listw hich willm erge the elem ents of (w hich is com posed of tw o independently sorted sets) into a single setw hich is sorted in ascending order

## SYNOPSIS



```
\mathbb{NTEGER N1,N2,TRD 1,TRD2}
INTEGER \mathbb{NDEX (*)}
DOUBLE PRECISION A (*)
SUBROUT\mathbb{NEDLAM RG_64(N 1,N 2,A,TRD 1,TRD 2,INDEX)}
INTEGER*8N1,N 2,TRD 1,TRD 2
INTEGER*8 \mathbb{NDEX (*)}
DOUBLE PRECISION A (*)
```


## F95 INTERFACE

SU BROUTINE LAM RG $\mathbb{N} 1, N 2, A, T R D 1, T R D 2, \mathbb{N D E X )}$
$\mathbb{N}$ TEGER :: N 1,N2, TRD 1, TRD 2
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: $\mathbb{N D} \operatorname{EX}$
REAL (8),D $\mathbb{M}$ ENSION (:) ::A
SU BROUTINE LAM RG_64 $\mathbb{N} 1, N 2, A, T R D 1, T R D 2, \mathbb{N} D E X)$
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N} 1, \mathrm{~N} 2, T R D 1, T R D 2$
$\mathbb{N} T E G E R(8), D \mathbb{I M}$ ENSION (:) :: $\mathbb{N} D E X$
REAL (8),D $\mathbb{M}$ ENSION (:) ::A
void dlam rg (intn1, intn2, double *a, inttral, int trd2, int*index);
void dlam rg_64 (long n1, long n2, double *a, long trod1, long trd2, long *index);

## PURPOSE

dlam rg w illcreate a perm utation listw hich will m erge the elem ents ofA (w hich is com posed of tw o independently sorted sets) into a single setw hich is sorted in ascending order.

## ARGUMENTS

N 1 (input)
Length of the first sequence to be m erged.

N 2 (input)
Length of the second sequence to be m erged.
A (input) On entry, the firstN 1 elem ents of $A$ contain a list of num bers w hich are sorted in either ascending ordescending order. Likew ise for the final
N 2 elem ents.

TRD 1 (input)
D escribes the stride to be taken through the array
A forthe firstN 1 elem ents.
$=-1$ subset is sorted in descending order.
= 1 subset is sorted in ascending order.
TRD 2 (input)
D escribes the stride to be taken through the anay
A for the firstN 1 elem ents.
$=-1$ subset is sorted in descending order.
= 1 subset is sorted in ascending order.
$\mathbb{I N D E X}$ (output)
On exitthis array w illcontain a perm utation such that if $\mathrm{B}(\mathrm{I})=\mathrm{A}(\mathbb{N} D \mathrm{EX}(\mathrm{I}))$ for $\mathrm{I}=1, \mathrm{~N} 1+\mathrm{N} 2$, then $B$ w illbe sorted in ascending order.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dlarz -applies a realelem entary reflector $H$ to a real M by $-\mathrm{N} m$ atrix C , from either the leftor the right

## SYNOPSIS

```
SUBROUT\mathbb{NE DLARZ (S\mathbb{DE,M,N,L,V,INCV,TAU,C,LDC,W ORK)}}\mathbf{M}\mathrm{ , (T,}
```

CHARACTER * 1 SDE
$\mathbb{N}$ TEGER $\mathrm{M}, \mathrm{N}, \mathrm{L}, \mathbb{I N C V}$,LDC
DOUBLE PRECISION TAU
DOUBLE PRECISION V (*), C (LDC, $\left.{ }^{\star}\right)$, W ORK (*)
SUBROUTINEDLARZ_64 (SDE, M,N,L,V, $\mathbb{N} C V, T A U, C, L D C, W$ ORK)
CHARACTER * 1 SDE
$\mathbb{N} T E G E R * 8 M, N, L, \mathbb{N} C V, L D C$
DOUBLE PRECISION TAU
DOUBLE PRECISIONV(*), C (LDC , *) ,W ORK (*)

## F95 INTERFACE

SU BROUTINE LARZ (SDE, $\mathbb{M}], \mathbb{N}], L, V,[\mathbb{N} C V], T A U, C,[L D C],[W O R K])$
CHARACTER (LEN=1): :SDE
$\mathbb{N} T E G E R:: M, N, L, \mathbb{N} C V, L D C$
REAL (8) ::TAU
REAL (8), D IM ENSION (:) ::V,W ORK
REAL (8),D IM ENSION (:,:) ::C
SU BROUTINE LARZ_64 (SDE, $\mathbb{M}], \mathbb{N}], L, V,[\mathbb{N} C V], T A U, C,[L D C],[W O R K])$

CHARACTER (LEN=1) ::SDE
$\mathbb{N} T E G E R(8):: M, N, L, \mathbb{N C V}, L D C$
REAL (8) :: TAU
REAL (8), D $\mathbb{M}$ ENSION (:) ::V,W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) ::C

## C INTERFACE

\#include <sunperfh>
void dlarz (charside, intm , intn, intl, double *v, int incv, double tau, double * C , int ldc);
void dlarz_64 (char side, long m, long n, long l, double *v, long incv, double tau, double *c, long ldc);

## PURPOSE

dlarz applies a realelem entary reflector $H$ to a realM foy-N $m$ atrix $C$, from either the left or the right. $H$ is represented in the form

$$
H=I-\tan ^{\star} \mathrm{v}^{\star} \mathrm{v}^{\prime}
$$

$w$ here tau is a realscalar and $v$ is a realvector.

If tau $=0$, then $H$ is taken to be the unitm atrix.
$H$ is a product of $k$ elem entary reflectors as retumed by STZRZF.

## ARGUMENTS

STDE (input)
$=\mathbb{L}$ : form $H * C$
$=R$ ': form $C * H$

M (input) The num ber of row sof the $m$ atrix $C$.

N (input) The num ber of colum ns of the m atrix C .

L (input) The num ber ofentries of the vector $V$ containing the $m$ eaningful part of the $H$ ouseholdervectors. If $S \mathbb{D} E=L ', M>=L>=0$, if $S D E=R \prime, N>=L$ $>=0$.

V (input) The vector v in the representation of H as retumed by $S T Z R Z F . V$ is notused ifTA $U=0$.
$\mathbb{N} C V$ (input)
The increm entbetw een elem ents ofv. $\mathbb{N} C V$ <> 0.

TAU (input)
The value tau in the representation of $H$.
C (input/output)
On entry, the M -by -N m atrix C. On exit, C is overw ritten by the $m$ atrix $H$ * $C$ if $S \mathbb{D} E=L$ ', or C * H if $S \mathbb{D} E=R$ '.
LDC (input)
The leading dim ension of the array C.LD C >= $m a x(1, M)$.

W ORK (w orkspace)
$(\mathbb{N})$ if $S \mathbb{D} E=L^{\prime}$ or $\left.M\right)$ if $S \mathbb{D} E=R^{\prime}$

## FURTHER DETAILS

B ased on contributions by
A. Petitet, C om puterScience D ept., U niv . of Tenn., K noxville, U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dlarzb -applies a realblock reflectort or its transpose $\mathrm{H} * * \mathrm{~T}$ to a real distributed M -by -N C from the leftor the right

## SYNOPSIS

```
SUBROUT\mathbb{NEDLARZB (S\mathbb{DE,TRANS,D RECT,STOREV,M,N,K,L,V,LDV,T,}}\mathbf{T},\textrm{L},\textrm{L}
    LDT,C,LDC,W ORK,LDW ORK)
CHARACTER * 1SDE,TRANS,D IRECT,STOREV
INTEGERM,N,K,L,LDV,LDT,LDC,LDW ORK
DOUBLE PRECISION V (LDV,*), T (LDT,*), C (LDC,*),
W ORK (LDW ORK,*)
SUBROUTINE DLARZB_64 (SDDE,TRANS,D RECT,STOREV ,M ,N,K,L,V,LDV,
    T,LDT,C,LDC,W ORK,LDW ORK)
CHARACTER * 1S\mathbb{DE,TRANS,D IRECT,STOREV}
\mathbb{N}TEGER*8M,N,K,L,LDV,LDT,LDC,LDW ORK
DOUBLE PRECISION V (LDV,*), T (LDT,*), C (LDC,*),
W ORK (LDW ORK,*)
```


## F95 INTERFACE

```
SU BROUTINE LARZB (SDE,TRANS,D \(\mathbb{R E C T}, \operatorname{STOREV}, \mathbb{M}], \mathbb{N}], K, L, V,[L D V]\), T, [LD T], C, [LDC], [W ORK ], [LDW ORK])
CHARACTER (LEN=1) ::SDE,TRANS,D RECT,STOREV
\(\mathbb{N}\) TEGER : : M , N, K, L, LDV,LD T,LD C,LD W ORK
REAL (8),D IM ENSION (:,:) ::V,T,C,W ORK
SU BROUTINE LARZB_64 (SDE,TRANS,D \(\mathbb{R E C T}, S T O R E V, \mathbb{M}], \mathbb{N}], K, L, V\),
```

[LDV ], T, [LD T], C, [LDC], [W ORK], [LDW ORK])

CHARACTER (LEN=1) ::SDE,TRANS,D $\mathbb{R E C T}, S T O R E V$
$\mathbb{N} T E G E R(8):: M, N, K, L, L D V, L D T, L D C, L D W O R K$
REAL (8),D IM ENSION (:,:) ::V,T,C,W ORK

## C INTERFACE

\#include <sunperfh>
void dlarzb (char side, chartrans, chardirect, char storev, int $m$, int $n$, int , int l, double * $v$, int ldv, double *t, int ldt, double *c, int ldc, int ldw ork);
void dlarzb_64 (char side, char trans, char direct, char storev, long $m$, long $n$, long $k$, long $l$, double *v, long ldv, double *t, long ldt, double *c, long ldc, long ldw ork);

## PURPOSE

dlarzb applies a realblock reflector H or its transpose $H * * T$ to a realdistributed $M$ boy -N C from the leftorthe right.

C unently, only STOREV = R'and D $\mathbb{R E C T}=\mathrm{B}$ 'are supported.

## ARGUMENTS

STDE (input)
= L': apply H orH 'from the Left
= R ': apply H orH 'from the Right

TRANS (input)
= N ': apply H N o transpose)
= C ': apply H ' (T ranspose)

D $\mathbb{R E C T}$ (input)
Indicates how $H$ is form ed from a product of ele-
$m$ entary reflectors = F':H = H (1) H (2) ...H (k)
(Forw ard, not supported yet)
$=B: H=H(k) \ldots H(2) H(1)(B a c k w a r d)$

STOREV (input)
Indicates how the vectors w hich define the elem en-
tary reflectors are stored:
= C : C olum nw ise
ported yet)
= R ': R ow w ise

M (input) The num ber of row s of the $m$ atrix $C$.

N (input) The num ber of colum ns of the $m$ atrix $C$.

K (input) The order of the m atrix $\mathrm{T} \Leftrightarrow$ the num ber of elem entary reflectors w hose product defines the block reflector).

L (input) The num ber of colum ns of the $m$ atrix V containing the $m$ eaningfulpart of the $H$ ouseholder reflectors. If $S \mathbb{D} E=L \prime, M>=L>=0$, if $S \mathbb{D} E=R \prime, N>=L$ $>=0$ 。

V (input) If $S T O R E V=C ', N V=K$; if $S T O R E V=R \prime, N V=L$.

LDV (input)
The leading dim ension of the array $V$. IfSTO REV $=$ C',LDV $>=\mathrm{L}$; ifSTOREV $=\mathrm{R}$ ', LDV $>=\mathrm{K}$.

T (input) The triangular K foy K m atrix T in the representation of the block reflector.

LD T (input)
The leading dim ension of the array $\mathrm{T} . \mathrm{LD} \mathrm{T}>=\mathrm{K}$.

C (input/output)
On entry, the $M$ boy N m atrix C . On exit, C is overw ritten by $\mathrm{H} * \mathrm{C}$ or $\mathrm{H}^{* *} \mathrm{C}$ or C H or C * H '.

LD C (input)
The leading dim ension of the array $C . \operatorname{LDC}>=$ $\max (1, M)$.

W ORK (w orkspace)
dim ension ( $\mathrm{M} A X(M, N), K)$

LDW ORK (input)
The leading dim ension of the anray W ORK. If S $\mathbb{D} E$
$=\mathbb{L}, L D W$ ORK >= max $(1, N)$; ifS $\mathbb{D} E=R \prime$ LDW ORK $>=\max (1, M)$.

## FURTHER DETAILS

B ased on contributions by
A. Petitet, C om puterS cience D ept., U niv . of Tenn ., K noxville, U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dlarzt-form the triangular factor $T$ of a real block reflectorH of order> $n$, which is defined as a product ofk elem entary reflectors

## SYNOPSIS

```
SUBROUT\mathbb{NE DLARZT (D RECT,STOREV,N,K,V,LDV,TAU,T,LDT)}
CHARACTER * 1D IRECT,STOREV
INTEGERN,K,LDV,LDT
DOUBLE PRECISION V (LDV,*),TAU(*),T (LDT,*)
SUBROUT\mathbb{NEDLARZT_64(D RECT,STOREV,N,K,V,LDV,TAU,T,LDT)}
CHARACTER * 1D RRECT,STOREV
INTEGER*8N,K,LDV,LDT
DOUBLE PRECISION V (LDV,\star),TAU(*),T (LDT,*)
```

F95 INTERFACE
SUBROUT $\mathbb{N} E$ LARZT (D $\mathbb{R E C T}, \operatorname{STOREV}, \mathrm{N}, \mathrm{K}, \mathrm{V},[\mathrm{LDV}], T A U, T,[L D T])$
CHARACTER (LEN=1)::D $\mathbb{R E C T}$,STOREV
$\mathbb{N} T E G E R:: N, K, L D V, L D T$
REAL (8),D IM ENSION (:) ::TAU
REAL (8), D $\mathbb{M}$ ENSIO N (:,:) ::V , T
SU BROUTINE LARZT_64 D $\mathbb{R E C T}, \operatorname{STOREV}, N, K, V,[L D V], T A U, T,[L D T])$
CHARACTER (LEN=1) ::D $\mathbb{R E C T}, S T O R E V$
$\mathbb{N}$ TEGER (8) :: $\mathrm{N}, \mathrm{K}, \mathrm{LDV}$,LDT
REAL (8), D IM ENSION (:) ::TAU

## C INTERFACE

\#include <sunperfh>
void dlarzt(char direct, char storev, intn, int $k$, double * $v$, int ldv, double *tau, double *t, int ldt);
void dlarzt_64 (chardirect, charstorev, long n, long k, double *v, long ldv, double *tau, double *t, long ldt);

## PURPOSE

dlarzt form s the triangular factor $T$ of a realblock reflector $H$ of order > $n$, which is defined as a productof $k$ elem entary reflectors.

IfD $\mathbb{R E C T}=\mathrm{F}^{\prime}, \mathrm{H}=\mathrm{H}(1) \mathrm{H}(2) \ldots \mathrm{H}(\mathrm{k})$ and T is upper triangular;

IfD $\mathbb{R E C T}=\mathrm{B}^{\prime}, \mathrm{H}=\mathrm{H}(\mathrm{k}) \ldots \mathrm{H}(2) \mathrm{H}(1)$ and T is lower triangular.

IfSTOREV = C', the vector which defines the elem entary reflector $H$ (i) is stored in the $i$-th colum $n$ of the array $V$, and

$$
\mathrm{H}=\mathrm{I}-\mathrm{V} * \mathrm{~T} * \mathrm{~V}^{\prime}
$$

IfSTOREV = R', the vector which defines the elem entary reflectort (i) is stored in the $i$-th row of the array $V$, and

$$
H=I-V^{\prime *} T * V
$$

C unently, only STOREV = R'and D $\mathbb{R E C T}=\mathrm{B}$ 'are supported.

## ARGUMENTS

D $\mathbb{R E C T}$ (input)
Specifies the order in which the elem entary
reflectors are multiplied to form the block
reflector:
$=F^{\prime}: H=H(1) H(2) \ldots H(k)$ Forw ard, notsup-
ported yet)
$=B^{\prime}: H=H(k) \ldots H(2) H(1)$ (Backw ard)

## STOREV (input)

Specifies how the vectors w hich define the elem entary reflectors are stored (see also Further
D etails):
= R ': row w ise
N (input) The order of the block reflector $\mathrm{H} . \mathrm{N}>=0$.
$K$ (input) The order of the triangular factor $T \vDash$ the num ber of elem entary reflectors). $\mathrm{K}>=1$.
$V$ (input) ( $L D V, K$ ) if $S T O R E V=C^{\prime}(L D V, N)$ if $S T O R E V=R^{\prime}$ Them atrix $V$. See furtherdetails.

LD V (input)
The leading dim ension of the array V . If $\operatorname{STOREV}=$ $C$ ', LDV $>=\max (1, N)$; ifSTOREV = $R$ ', LDV $>=K$.

TAU (input)
TAU (i) must contain the scalar factor of the elem entary reflectort (i).
$T$ (input) The $k$ by $k$ triangular factor $T$ of the block reflector. If $\mathrm{D} \mathbb{R E C T}=\mathrm{F}^{\prime}, \mathrm{T}$ is upper triangular; if $\mathrm{D} \mathbb{R E C T}=\mathrm{B}$ ', T is low er triangular. The restof the aray is notused.

LD T (input)
The leading dim ension of the array T.LD T >=K.

## FURTHER DETAILS

B ased on contributions by
A. Petitet, C om puter Science D ept., U niv. of Tenn., K noxville, U SA

The shape of the $m$ atrix $V$ and the storage of the vectors which define the $H$ (i) is bestillustrated by the follow ing exam ple w th $\mathrm{n}=5$ and $\mathrm{k}=3$. The elem ents equal to 1 are not stored; the comesponding aray elem ents arem odified but restored on exit. The restof the array is notused.

D $\mathbb{R E C T}=\mathrm{F}^{\prime}$ and STOREV $=\mathrm{C}^{\prime}: \quad \mathrm{D} \mathbb{R E C T}=\mathrm{F}^{\prime}$ and STOREV = R':

```
                                    V
```



```
(v1 v2 v3) (v1 v1 v1 v1 v1 \ldots..1
)
    V = (v1 v2 v3 ) (v2 v2 v2 v2 v2 .
```

```
..1 )
    (v1 v2 v3 ) (v3 v3 v3 v3 v3 .
    .1 )
        (v1 v2 v3 )
        . . .
        1..
            1.
                1
D\mathbb{RECT}=\mp@subsup{B}{}{\prime}\mathrm{ 'andSTOREV = C': D PRECT = B' and}
STOREV = R':
    1
```



```
    . 1
        (1 . . ..v1 v1 v1 v1 v1 )
        . . }
v2 v2 v2 )
    ...
v3 v3 v3 )
        •••
    (v1 v2 v3 )
    V = (v1 v2 v3)
        (v1 v2 v3 )
```


## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dlasit - the num bers in $D$ in increasing order (if $\mathbb{D}=\mathbb{I}$ ) or in decreasing order (if $\mathbb{D}=D^{\prime}$ )

## SYNOPSIS

```
SUBROUT\mathbb{NE DLASRT(\mathbb{D},N,D,\mathbb{NFO)}}\mathbf{N}=()
CHARACTER * 1 \mathbb{D}
INTEGERN, \mathbb{NFO}
DOUBLE PRECISION D (*)
```



```
CHARACTER * 1 \mathbb{D}
\mathbb{NTEGER*8N,INFO}
DOUBLE PRECISION D (*)
```

F95 INTERFACE
SUBROUTINE LASRT ( $\mathbb{D}, \mathbb{N}], D,[\mathbb{N} F O])$
CHARACTER (LEN=1) :: $\mathbb{D}$
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::D
SU BROUTINE LASRT_64 ( $\mathbb{D}, \mathbb{N}], D,[\mathbb{N} F O])$
CHARACTER (LEN=1) :: $\mathbb{D}$
$\mathbb{N} T E G E R(8):: N, \mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::D
C INTERFACE
\#include <sunperfh>
void dlastt(char id, intn, double *d, int *info);
void dlasit_64 (charid, long n, double *d, long *info);

## PURPOSE

dlastt the num bers in $D$ in increasing order (if $\mathbb{D}=I$ ) or in decreasing order (if $\mathbb{D}=D^{\prime}$ ).

U se Q uick Sort, reverting to Insertion sort on arrays of size $<=20$. D im ension of STA CK $\lim$ its N to about $2 * * 32$.

## ARGUMENTS

ID (input)
= I': sortD in increasing order;
$=D^{\prime}$ ': sortD in decreasing order.

N (input) The length of the aray D .

D (input/output)
O n entry, the array to be sorted. On exit, D has
been sorted into increasing order $(\mathbb{D}(1)<=\ldots<=$
D (N ) ) or into decreasing order (D (1) >= ... >=
$D(\mathbb{N})$ ), depending on $\mathbb{D}$.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dlatzm -routine is deprecated and has been replaced by routine SO RM RZ

## SYNOPSIS

```
SUBROUT\mathbb{NEDLATZM(SDE,M,N,V,INCV,TAU,C1,C2,LDC,W ORK)}
CHARACTER * 1SDE
INTEGERM,N,\mathbb{NCV,LDC}
DOUBLE PRECISION TAU
DOUBLE PRECISION V (*),C1 (LDC ,*),C2 (LDC ,*),W ORK (*)
```



```
CHARACTER * 1SDEE
INTEGER*8M,N,INCV,LDC
DOUBLE PRECISION TAU
DOUBLE PRECISION V (*),C1 (LDC,*),C2 (LDC,*),W ORK (*)
```


## F95 INTERFACE

SUBROUTINE LATZM (SDE, $\mathbb{M}], \mathbb{N}], V,[\mathbb{N} C V], T A U, C 1, C 2,[L D C],[\mathbb{O R K}])$

CHARACTER (LEN=1)::SDE
$\mathbb{N} T E G E R:: M, N, \mathbb{N C V}, L D C$
REAL (8) ::TAU
REAL (8),D $\mathbb{M}$ ENSION (:) ::V ,W ORK
REAL (8),D $\mathbb{M}$ ENSIO N (:,:) ::C1,C2
SU BROUTINE LATZM_64 (STDE, $\mathbb{M}], \mathbb{N}], V,[\mathbb{N} C V], T A U, C 1, C 2,[L D C]$, [ W ORK])

CHARACTER (LEN=1)::SDE
$\mathbb{N}$ TEGER (8) :: M , N , $\mathbb{N} C V$,LD C
REAL (8) ::TAU
REAL (8), D $\mathbb{M}$ ENSION (:) ::V ,W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) ::C1, C2

## C INTERFACE

\#include <sunperfh>
void dlatzm (charside, intm, intn, double ${ }^{*}$ v, int incv, double tau, double * c1, double *C2, int ldc);
void dlatzm _64 (char side, long m, long n, double *v, long incv, double tau, double *c1, double *c2, long ldc);

## PURPOSE

dlatzm routine is deprecated and has been replaced by routine SO RM RZ .

SLA TZM applies a H ouseholderm atrix generated by STZRQF to a $m$ atrix.

LetP $=I-\tan u^{*} u ', u=(1)$,
(v)
$w$ here $v$ is an ( $m-1$ ) vector if $S \mathbb{D} E=\mathbb{L}$ ', ora ( $n-1$ ) vector if $S \mathbb{D} E=R$.

IfS $\operatorname{DE}$ equals ${ }^{\text {L }}$ ', let
$C=[C 1] 1$
[C2]m-1
n
Then C is overw ritten by P * C .

If $S$ D E equals R', let $C=[C 1, C 2] m$

1 n-1
Then C is overw rilten by C *P.

## ARGUMENTS

```
SIDE (input)
    = L': form P * C
    = R': form C * P
```

M (input) The num ber of row s of the $m$ atrix $C$.

N (input) The num ber of colum ns of the $m$ atrix $C$.
$V$ (input) $(1+\mathbb{M}-1) * a b s(\mathbb{N C V}))$ if $S \mathbb{D} E=L^{\prime}(1+\mathbb{N}-$ 1)*abs ( $\mathbb{N} C V)$ ) if $S \mathbb{D} E=R$ 'The vectorv in the representation ofP. $V$ is notused if $T A U=0$.
$\mathbb{N} C V$ (input)
The increm entbetw een elem ents of $v . \mathbb{I N} C V<>0$

TAU (input)
The value tau in the representation ofP.

C1 (input/output)
$(L D C, N)$ if $S \mathbb{D} E=L^{\prime}(M, \mathbb{1})$ if $S \mathbb{D} E=R^{\prime} O n$ entry, the $n$-vector $C 1$ if $S \mathbb{D E}=\mathrm{L}$ ', or the $m-$ vectorC 1 if $S \mathbb{D} E=R$ '.

On exit, the first row ofP *C if $S \mathbb{D} E=$ ' ', or the first colum $n$ of $C * P$ if $S I D E=R$ '.

C2 (input/output)
$(\mathbb{L D} C, N)$ if $S \mathbb{D} E=\mathbb{L}^{\prime}(\mathrm{LD} C, N-1)$ if $S \mathbb{D} E=R^{\prime}$ On entry, the $(m-1) x n m$ atrix $C 2$ if $S \mathbb{D} E=L^{\prime}$, or them $x(n-1) m$ atrix $C 2$ if $S D E=R$.

Onexit, rows 2 m ofP*C if $S \mathbb{D} E=\mathrm{L}$ ', orcolum ns 2 m of $\mathrm{C} *$ if $S \mathbb{D} E=R$.

LD C (input)
The leading dim ension of the arrays C 1 and C 2.LD C $>=(1, \mathrm{M})$.

W ORK (w orkspace)
$(\mathbb{N})$ if $S \mathbb{D} E=L^{\prime}(M)$ if $S \mathbb{D} E=R^{\prime}$

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dnrm 2 -Retum the Euclidian norm of a vector.

## SYNOPSIS

$$
\text { DOUBLE PRECISION FUNCTION DNRM } 2 \mathbb{N}, \mathrm{X}, \mathbb{N} C X)
$$

$\mathbb{N} T E G E R N, \mathbb{N C X}$
D OUBLE PRECISION X (*)

DOUBLE PRECISION FUNCTION DNRM 2_64 $\mathbb{N}, \mathrm{X}, \mathbb{N} C X)$
$\mathbb{N}$ TEGER*8N, $\mathbb{N C X}$
DOUBLE PRECISION X (*)

## F95 INTERFACE

REAL (8) FUNCTION NRM 2 ( $\mathbb{N}], \mathrm{X},[\mathbb{N C X}]$ )
$\mathbb{N} T E G E R:: N, \mathbb{N} C X$
REAL (8),D $\mathbb{M}$ ENSION (:) ::X
REAL (8) FUNCTION NRM 2_64 ( $\mathbb{N}$ ], X, [ $\mathbb{N} C X]$ )
$\mathbb{N} T E G E R(8):: N, \mathbb{N C X}$
REAL (8),D $\mathbb{M}$ ENSION (:) ::X

## C INTERFACE

\#include <sunperfh>
double dnrm 2 (intn, double *x, intincx);
double dnım 2_64 (long n, double *x, long incx);

## PURPOSE

dnm 2 Retum the Euclidian norm of a vector $x$ where $x$ is an n-vector.

## ARGUMENTS

N (input)
O n entry, $N$ specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.
X (input)
$(1+(n-1) * a b s(\mathbb{N} C X))$. On entry, the increm ented array X m ust contain the vectorx. U nchanged on exit.
$\mathbb{N} C X$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of $\mathrm{X} . \mathbb{N} C X$ m ustbe positive. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dopgtr-generate a real orthogonal matrix $Q$ which is defined as the product ofn-1 elem entary reflectors $H$ (i) of ordern, as retumed by SSPTRD using packed storage

## SYNOPSIS

```
SUBROUTINEDOPGTR (UPLO,N,AP,TAU,Q,LDQ,W ORK,INFO)
CHARACTER * 1 UPLO
\mathbb{NTEGER N,LDQ,INFO}
DOUBLE PRECISION AP (*),TAU (*),Q (LDQ ,*),W ORK (*)
SUBROUT\mathbb{NE DOPGTR_64(UPLO,N,AP,TAU,Q,LDQ,W ORK,INFO )}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N,LDQ,INFO}
DOUBLE PRECISION AP (*),TAU (*),Q (LDQ,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE OPGTR (UPLO, $\mathbb{N}], A P, T A U, Q,[L D Q],[W O R K],[\mathbb{N} F O])$
CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R:: N, L D Q, \mathbb{N} F O$
REAL (8),D IM ENSION (:) ::AP,TAU,W ORK
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::Q

SU BROU T INE OPG TR_64 (UPLO, $\mathbb{N}], A P, T A U, Q,[L D Q],[\mathbb{N} O R K],[\mathbb{N} F O])$
CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER ( 8 ) :: $\mathrm{N}, \mathrm{LD} Q, \mathbb{N}$ FO
REAL (8),D $\mathbb{I}$ ENSION (:) ::AP,TAU,W ORK
REAL (8),D IM ENSION (:,:) ::Q

## C INTERFACE

\#include <sunperfh>
void dopgtr(char uplo, intn, double *ap, double *tau, double *q, int ldq, int *info);
void dopgtr_64 (charuplo, long n, double *ap, double *tau, double *q, long ldq, long *info);

## PURPOSE

dopgtrgenerates a realorthogonalm atrix Q which is defined as the productofn-1 elem entary reflectors $H$ (i) ofordern, as retumed by SSPTRD using packed storage:
if $U P L O=U ', Q=H(n-1) \ldots H(2) H(1)$,
if $U P L O=L^{\prime}, Q=H(1) H(2) \ldots H(n-1)$.

## ARGUMENTS

UPLO (input)
= U ':U ppertriangular packed storage used in previous call to SSPTRD; = L': Low ertriangular packed storage used in previous call to SSPTRD .

N (input) The order of the matrix $\mathrm{Q} . \mathrm{N}>=0$.
AP (input)
The vectors w hich define the elem entary reflectors, as retumed by SSPTRD .

TAU (input)
TAU (i) m ust contain the scalar factor of the elem entary reflectort (i), as retumed by SSPTRD.

Q (output)
The N -by -N orthogonalm atrix Q .
LD Q (input)
The leading dim ension of the array $\mathrm{Q} . \mathrm{LD} \mathrm{Q}>=$ $\max (1, N)$.

W ORK (w orkspace)
dim ension $(\mathbb{N}-1)$
$\mathbb{I N} F O$ (output)
= 0: successfulexit
< 0: if $\mathbb{N}$ FO $=-$ i, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dopm tr-overw rite the general real M -by -N m atrix $\mathrm{C} w$ ith $S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime}$ TRANS $=N^{\prime}$

## SYNOPSIS

```
SU BROUTINE DOPM TR (S\mathbb{DE,UPLO,TRANS,M,N,AP,TAU,C,LDC,W ORK,}
    \mathbb{NFO)}
CHARACTER * 1SIDE,UPLO,TRANS
INTEGERM,N,LDC,\mathbb{NFO}
DOUBLE PRECISION AP (*),TAU (*),C (LDC ,*),W ORK (*)
SUBROUTINE DOPM TR_64 (S\mathbb{DE,UPLO,TRANS,M,N,AP,TAU,C,LDC,W ORK,}
    \mathbb{NFO)}
```

CHARACTER * 1 SIDE, UPLO, TRANS
$\mathbb{N}$ TEGER*8 M , N, LD C , $\mathbb{N}$ FO
DOUBLE PRECISION AP (*), TAU (*), C (LDC , *), W ORK (*)

## F95 INTERFACE

SU BROUTINE OPM TR (SDE, UPLO, [TRANS], $\mathbb{M}], \mathbb{N}], A P, T A U, C,[L D C]$, [ W ORK], [ $\mathbb{N} F \mathrm{O}$ ])

CHARACTER (LEN=1) ::SDE,UPLO,TRANS
$\mathbb{N} T E G E R:: M, N, L D C, \mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::AP,TAU,W ORK
REAL (8),D $\mathbb{I M}$ ENSION (:,:) ::C
SU BROUTINE OPM TR_64 (SDE, UPLO, [TRANS], $\mathbb{M}], \mathbb{N}], A P, T A U, C,[L D C]$, [ $\mathrm{W} O \mathrm{RK}$ ], [ $\mathbb{N} F \mathrm{FO}$ )

CHARACTER (LEN=1)::SIDE,UPLO,TRANS
$\mathbb{N} T E G E R(8):: M, N, L D C, \mathbb{N} F O$
REAL (8),D $\mathbb{I}$ ENSION (:) ::AP,TAU ,W ORK
REAL (8),D IM ENSION (:,:) ::C

## C INTERFACE

\#include <sunperfh>
void dopm tre(charside, charuplo, chartrans, intm, int n, double *ap, double *tau, double *c, int ldc, int *info);
void dopm tr_64 (char side, char uplo, char trans, long m, long n, double *ap, double *tau, double *c, long ldc, long *info);

## PURPOSE

dopm troverw rites the general real M boy-N matrix C with TRANS = T: $\quad \mathrm{Q} * * \mathrm{~T} * \mathrm{C} \quad \mathrm{C} * \mathrm{Q} * * \mathrm{~T}$
where $Q$ is a realorthogonalm atrix of ordernq, $w$ ith $n q=m$ if $S \mathbb{D} E=L$ 'and $n q=n$ if $S \mathbb{D} E=R$ '. $Q$ is defined as the product of nq-1 elem entary reflectors, as retumed by SSPTRD using packed storage:
if $U P L O=U ', Q=H(n q-1) \ldots$ (2) $H(1) ;$
if UPLO = L', $\mathrm{Q}=\mathrm{H}(1) \mathrm{H}(2) \ldots \mathrm{H}(\mathrm{nq}-1)$.

## ARGUMENTS

STDE (input)
$=\mathrm{L}$ ': apply Q orQ ${ }^{* *}$ T from the Left;
= R ': apply Q orQ ${ }^{* * T}$ from the R ight.

UPLO (input)
= U ':U ppertriangular packed storage used in previous call to SSPTRD ; = L':Low ertriangular packed storage used in previous call to SSPTRD .

TRANS (input)
$=\mathrm{N}$ ': N o transpose, apply Q ;
$=T$ ': Transpose, apply $Q * * T$.
TRANS is defaulted to $N$ 'forF $95 \mathbb{N}$ TERFACE .

M (input) The num ber of row s of the $m$ atrix $\mathrm{C} . \mathrm{M}>=0$.

N (input) The num ber of colum ns of the m atrix $\mathrm{C} . \mathrm{N}>=0$.
AP (input)
$(M *(M+1) / 2)$ if $S D E=L(\mathbb{N} *(N+1) / 2)$ if $S D E=$
$R$ ' The vectors which define the elem entary
reflectors, as retumed by SSP TRD . A P ism odified
by the routine but restored on exit.
TAU (input)
or $(\mathbb{N}-1)$ if $S \mathbb{D} E=R^{\prime} T A U(i)$ must contain the scalar factor of the elem entary reflectorH (i), as retumed by SSPTRD.
C (input/output)
On entry, the M by -N m atrix C. On exit, C is overw ritten by Q * C or $\mathrm{Q} * \mathrm{~T}^{*} \mathrm{C}$ or $\mathrm{C}^{*} \mathrm{Q} * * \mathrm{~T}$ or $\mathrm{C}{ }^{\mathrm{Q}} \mathrm{Q}$.

LD C (input)
The leading dim ension of the array C.LD C >= $\max (1, \mathrm{M})$.

W ORK (w orkspace)
$(\mathbb{N})$ if $S \mathbb{D} E=L^{\prime}(M)$ if $S \mathbb{D} E=R^{\prime}$
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum enthad an ille-
galvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorg2l-generate an $m$ by $n$ realm atrix $Q$ with orthonorm al colum ns,

## SYNOPSIS

```
SUBROUT\mathbb{NEDORG2L M,N,K,A,LDA,TAU,W ORK,INFO)}
INTEGERM,N,K,LDA,\mathbb{NFO}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
SU BROUT\mathbb{NE DORG 2L_64 M ,N ,K,A ,LDA ,TAU,W ORK, NNFO)}
INTEGER*8M,N,K,LDA,INFO
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
F95 INTERFACE
SU BROUT\mathbb{NE ORG 2L (\mathbb{M ], N ], [K ],A , [LDA ],TAU, [W ORK ], [NFO ])}}\mathbf{N}\mathrm{ )}
\mathbb{NTEGER ::M,N,K,LDA,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REAL (8),D IM ENSION (:) ::TAU,W ORK
REAL (8),D IM ENSION (:,:) ::A
```



```
\mathbb{NTEGER (8)::M,N,K,LDA,\mathbb{NFO}}\mathbf{N},\textrm{N},
REAL (8),D IM ENSION (:) ::TAU ,W ORK
REAL (8),D IM ENSION (:,:) ::A
C INTERFACE
    #include <sunperfh>
    void dorg2l(intm, intn, intk, double *a, intlda, double
```

void dorg21 64 (long m, long n, long k, double *a, long lda, double *tau, long *info);

## PURPOSE

dorg2lL generates an $m$ by $n$ realm atrix $Q w i t h$ orthonorm al colum ns, which is defined as the lastn colum ns of a product ofk elem entary reflectors oforderm

$$
Q=H(k) \ldots H(2) H(1)
$$

as retumed by SG EQ LF .

## ARGUMENTS

M (input) The num ber of row s of the m atrix $\mathrm{Q} . \mathrm{M}>=0$.

N (input) The num ber of $\infty$ lum ns of the $m$ atrix $\mathrm{Q} . \mathrm{M}>=\mathrm{N} \quad>=$ 0.
$K$ (input) The num ber of elem entary reflectors $w$ hose product defines the $m$ atrix $Q . N>=K>=0$.

A (input/output)
On entry, the $(n-k+j)$-th columnmust contain the vector which defines the elem entary reflector H (i), for $i=1,2, \ldots, k$, as retumed by SGEQ LF in the last $k$ colum ns of its amay argum entA. On exit, them by $n m$ atrix $Q$.

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele$m$ entary reflectort (i), as retumed by SG EQ LF .

W ORK (w orkspace)
dim ension $\mathbb{N}$ )
$\mathbb{N} F O$ (output)
= 0 : successfiulexit
$<0:$ if $\mathbb{N} F O=-$ i, the $i$-th argum enthas an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorg 2 r -generate an m by n realm atrix Q w ith orthonorm al colum ns,

## SYNOPSIS

```
SU BROUT\mathbb{NE DORG2R M,N,K,A,LDA,TAU,W ORK, NNFO)}
INTEGERM,N,K,LDA,\mathbb{NFO}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
SUBROUT\mathbb{NE DORG2R_64M,N,K,A,LDA,TAU,W ORK,INFO)}
INTEGER*8M,N,K,LDA, INFO
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
F95 INTERFACE
```



```
\mathbb{NTEGER ::M,N,K,LDA,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REAL (8),D IM ENSION (:) ::TAU,W ORK
REAL (8),D IM ENSION (:,:) ::A
```



```
\mathbb{NTEGER (8)::M,N,K,LDA,\mathbb{NFO}}\mathbf{N},\textrm{N},
REAL (8),D IM ENSION (:) ::TAU ,W ORK
REAL (8),D IM ENSION (:,:) ::A
C INTERFACE
    #include <sunperfh>
    void dorg2r(intm, intn, intk, double *a, intlda, double
```

void dorg2r_64 (long m, long n, long k, double *a, long lda, double *tau, long *info);

## PURPOSE

dorg2rR generates an $m$ by $n$ realm atrix $Q$ with orthonorm al colum ns, which is defined as the firstn colum ns of a product of $k$ elem entary reflectors of orderm

$$
Q=H(1) H(2) \ldots H(k)
$$

as retumed by SG EQ RF .

## ARGUMENTS

M (input) The num ber of row s of the $m$ atrix $Q . M>=0$.

N (input) The num ber of colum ns of the matrix $\mathrm{Q} . \mathrm{M}>=\mathrm{N} \quad>=$ 0.
$K$ (input) The num ber of elem entary reflectors $w$ hose product defines the m atrix $\mathrm{Q} . \mathrm{N}>=\mathrm{K}>=0$.

A (input/output)
On entry, the $i$-th columnm ustcontain the vector which defines the elem entary reflectorH (i), for i
$=1,2, \ldots, k$, as retumed by SGEQRF in the first $k$
colum ns of its array argum entA. On exit, the $m$ by $-\mathrm{n} m$ atrix Q .

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele$m$ entary reflectort (i), as retumed by SGEQRF.

W ORK (w orkspace)
dim ension $\mathbb{N}$ )
$\mathbb{I N} F O$ (output)
= 0 : successfiulexit
$<0$ : if $\mathbb{N}$ FO $=-$ i, the i-th argum enthas an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorgbr-generate one of the realorthogonalm atrices $Q$ or $P * * T$ determ ined by SGEBRD when reducing a realm atrix $A$ to bidiagonal form

## SYNOPSIS

```
SUBROUT\mathbb{NEDORGBR NECT,M,N,K,A,LDA,TAU,W ORK,LW ORK,INFO)}
CHARACTER * 1 VECT
INTEGER M,N,K,LDA,LW ORK,\mathbb{NFO}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
SU BROUTINE DORGBR_64(NECT,M ,N,K,A,LDA,TAU,W ORK,LW ORK,\mathbb{NFO)}
CHARACTER * 1VECT
\mathbb{N}TEGER*8M,N,K,LDA,LW ORK,\mathbb{NFO}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE ORGBR $(N E C T, M, \mathbb{N}], K, A,[L D A], T A U,[W O R K],[L W$ ORK ], [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1)::VECT
$\mathbb{N}$ TEGER ::M,N,K,LDA,LW ORK, $\mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8),D IM ENSION (:,:) ::A
SUBROUTINE ORGBR_64 NECT,M, $\mathbb{N}], K, A,[L D A], T A U,[W O R K],[L W ~ O R K]$, [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::VECT
$\mathbb{N} T E G E R(8):: M, N, K, L D A, L W O R K, \mathbb{N} F O$

## C INTERFACE

\#include <sunperfh>
void dorgbr(charvect, intm, intn, int $k$, double *a, int lda, double *tau, int *info);
void dorgbr_64 (charvect, long m, long $n$, long $k$, double *a, long lda, double *tau, long *info);

## PURPOSE

dorgbrgenerates one of the realorthogonal matrices $Q$ or $\mathrm{P} * * \mathrm{~T}$ determ ined by $S G E B R D$ when reducing a realm atrix $A$ to bidiagonal form : $A=Q * B * P * * T . Q$ and $P * * T$ are defined as products of elem entary reflectors $H$ (i) orG (i) respectively.

IfVECT $=Q$ ', $A$ is assum ed to have been an $M$ boy $K m$ atrix, and $Q$ is of orderM :
ifm $>=k, Q=H(1) H(2) \ldots H(k)$ and $S O R G B R$ retums the firstn colum ns of $Q$, where $m>=n>=k$;
ifm $<k, Q=H(1) H(2) \ldots H(m-1)$ and $S O R G B R$ retums $Q$ as an $M$ boy $-M$ matrix.

IfVECT $=P$ ', A is assum ed to have been a K -by -N m atrix, and $P{ }^{* *} \mathrm{~T}$ is oforderN:
if $k<n, P * * T=G(k) \ldots G(2) G(1)$ and $S O R G B R$ retums the firstm row $\operatorname{sof} P * * T$, where $n>=m>=k$; ifk $>=n, P * * T=G(n-1) \ldots G(2) G(1)$ and $S O R G B R$ retums $P * * T$ as an $N$ boy $N$ m atrix.

## ARGUMENTS

VECT (input)
Specifies w hether the matrix $Q$ orthem atrix $P * * T$
is required, as defined in the transform ation
applied by SGEBRD :
= Q ': generate Q ;
$=\mathrm{P}^{\prime}:$ generate $\mathrm{P} * * \mathrm{~T}$.

M (input) The num ber of row s of the $m$ atrix $Q$ orP**T to be retumed. $\mathrm{M}>=0$.

N (input) The num ber of colum ns of the $m$ atrix Q or $\mathrm{P} * * T$ to
be retumed. $\mathrm{N}>=0$. IfVECT $=Q \mathrm{~V}^{\prime}, \mathrm{M}>=\mathrm{N}>=$ $m$ in $(M, K) ;$ ifVECT $=P^{\prime}, N>=M>=m$ in $(\mathbb{N}, K)$.
$K$ (input) IfVECT = Q ', the num berof colum ns in the original $M$ boy $K m$ atrix reduced by $S G E B R D$. IfVECT $=$ $P$ ', the num ber of row $s$ in the original $K$ boy $N$ m atrix reduced by SG EBRD . $\mathrm{K}>=0$.

A (input/output)
O n entry, the vectors w hich define the elem entary reflectors, as retumed by SGEBRD. On exit, the M łoy-N m atrix Q orP**T.

LD A (input)
The leading dim ension of the array $\mathrm{A} . \operatorname{LDA}>=$ $\max (1, M)$.

TAU (input)
$(m$ in $(\mathbb{M}, K))$ ifVECT = $Q^{\prime}(m$ in $(\mathbb{N}, K))$ ifVECT $=P^{\prime}$
TAU (i) m ustcontain the scalar factor of the ele$m$ entary reflectorH (i) orG (i), which determ ines Q or $\mathrm{P} * * \mathrm{~T}$, as retumed by $\mathrm{SG} E B R D$ in its array argu$m$ entTAUQ orTAUP.

W ORK (w orkspace)
On exit, if $\mathbb{N F O}=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the amay W ORK. LW ORK >= $m$ ax $(1, m$ in $M, N))$. Foroptim um perform ance LW ORK >= $m$ in $(M, N) * N B, w h e r e N B$ is the optim alblocksize.

IfLW ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N F O}=-i$, the $i$ th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorghr-generate a real orthogonal matrix Q which is defined as the productof $\mathbb{H}$ I-HO elem entary reflectors of orderN, as retumed by SG EH RD

## SYNOPSIS




```
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
```



```
\mathbb{N TEGER*8N,\mathbb{LO,\mathbb{H}I,LDA,LW ORK,INFO}}\mathbf{N}=1
DOUBLE PRECISION A (LDA,*),TAU(*),W ORK (*)
```

F95 INTERFACE
SU BROUTINE ORGHR ( $\mathbb{N}], \mathbb{I} O, \mathbb{H} I, A,[L D A], T A U,[\mathbb{W}$ ORK ], [LW ORK ], [ $\mathbb{N} F O])$
$\mathbb{N}$ TEGER :: $\mathrm{N}, \mathbb{L} \mathrm{O}, \mathbb{H} \mathrm{I}$, LDA, LW ORK, $\mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (:) ::TAU ,W ORK
REAL (8),D IM ENSION (:,:) ::A
SU BROUTINE ORGHR_64 ( $\mathbb{N}], \mathbb{I} O, \mathbb{H} I, A,[L D A], T A U,[W O R K],[L W ~ O R K]$,
[ $\mathbb{N} F O$ ])
$\mathbb{N}$ TEGER (8) :: N, $\mathbb{H} O, \mathbb{H} \mathrm{I}, \mathrm{LD} A, L W$ ORK, $\mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8),D IM ENSION (:,:) ::A
C INTERFACE
\#include <sunperfh>
void dorghr(intn, intilo, intihi, double *a, int lda, double *tau, int *info);
void dorghr_64 (long n, long 1 , long ini, double *a, long lda, double *tau, long *info);

## PURPOSE

dorghrgenerates a realorthogonalm atrix Q which is defined as the product of $\mathbb{H}$ I-ILO elem entary reflectors of orderN, as retumed by SG EH RD :
$Q=H$ ( 0 ) H ( i (م+1) . . H (ihi-1).

## ARGUMENTS

N (input) The order of the m atrix $\mathrm{Q} . \mathrm{N}>=0$.

IIO (input)
IIO and IH Im usthave the sam e values as in the previous call of SGEHRD.Q is equal to the unit $m$ atrix except in the subm atrix
Q (ilo+1: ihi, ilo+1: ihi). $1<=\mathbb{I} 0<=\mathbb{H} I<=N$, if $\mathrm{N}>0 ; ~ \Pi \mathrm{O}=1$ and $\mathrm{H} \mathrm{I}=0$, if $\mathrm{N}=0$.

IH I (input)
See the description of IIO .

A (input/output)
O n entry, the vectors w hich define the elem entary reflectors, as retumed by SGEHRD. On exit, the N boy -N orthogonalm atrix Q .

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, N)$.

TAU (input)
TAU (i) m ust contain the scalar factor of the ele$m$ entary reflectorH (i), as retumed by SGEHRD.

W ORK (w orkspace)
On exit, if $\mathbb{N F O}=0, W$ ORK (1) retums the optim al LW ORK.

The dim ension of the array $W$ ORK. LW ORK >= $\mathbb{H} I-\mathbb{H O}$. For optim um perform ance LW ORK $>=(\mathbb{H} I-\mathbb{H} O)^{*} N B$, where NB is the optim alblocksize.

IfLW ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim al size of theW ORK ancay, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
$\mathbb{N} F O$ (output)
= 0: successfulexit
<0: if $\mathbb{N F O}=-$ i, the $i$-th argum ent had an illegal value

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorg12 - generate an $m$ by $n$ realm atrix $Q$ with orthonorm al row S,

## SYNOPSIS

```
SUBROUT\mathbb{NEDORGL2M,N,K,A,LDA,TAU,W ORK,INFO)}
INTEGERM,N,K,LDA,\mathbb{NFO}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
SU BROUT\mathbb{NE DORGL2_64 M ,N ,K,A,LDA ,TAU,W ORK, NNFO)}
INTEGER*8M,N,K,LDA,INFO
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
F95 INTERFACE
    SU BROUT\mathbb{NE ORGL2 (\mathbb{M ], N ], [K ],A , [LDA ],TAU, [W ORK ], [NFO ])}}\mathbf{~}\mathrm{ )}
    \mathbb{NTEGER ::M,N,K,LDA,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
    REAL (8),D IM ENSION (:) ::TAU,W ORK
    REAL (8),D IM ENSION (:,:) ::A
```



```
    \mathbb{NTEGER (8)::M,N,K,LDA, NNFO}
    REAL (8),D IM ENSION (:) ::TAU ,W ORK
    REAL (8),D IM ENSION (:,:) ::A
C INTERFACE
    #include <sunperfh>
    void dorg12 (intm , intn, intk, double *a, intlda, double
```

void dorgl2_64 (long m , long n, long k, double *a, long lda, double *tau, long *info);

## PURPOSE

dorg12 generates an $m$ by $n$ realm atrix $Q w i t h$ orthonorm al row $s$, which is defined as the firstm row s of a product ofk elem entary reflectors of ordern

$$
Q=H(k) \ldots H(2) H(1)
$$

as retumed by SG ELQ F.

## ARGUMENTS

M (input) The num ber of row s of the $m$ atrix $Q . M>=0$.

N (input) The num ber of colum ns of the matrix $\mathrm{Q} . \mathrm{N}>=\mathrm{M}$.
$K$ (input) The num ber of elem entary reflectors w hose product defines the m atrix $\mathrm{Q} . \mathrm{M}>=\mathrm{K}>=0$.

A (input/output)
On entry, the $i$-th row must contain the vector which defines the elem entary reflector $H$ (i), for i $=1,2, \ldots, k$, as retumed by SGELQF in the first $k$ row sof its array argum entA. On exit, the $m$-by-n $m$ atrix $Q$.

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the elem entary reflectorH (i), as retumed by SGELQ F.

W ORK (w orkspace)
dim ension M )
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum enthas an illegalvahue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorglq - generate an $M$-by-N realm atrix $Q \mathrm{w}$ ith orthonorm al row S,

## SYNOPSIS



```
\mathbb{NTEGERM,N,K,LDA,LDW ORK,\mathbb{NFO}}\mathbf{N}=\mp@code{L}
DOUBLE PRECISION A (LDA,*),TAU (*),WORK (*)
```



```
\mathbb{NTEGER*8M,N,K,LDA,LDW ORK,INFO}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE ORGLQ $M, \mathbb{N}], \mathbb{K}], A,[L D A], T A U,[W O R K],[L D W O R K],[\mathbb{N} F O])$
$\mathbb{N} T E G E R:: M, N, K, L D A, L D W O R K, \mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::A

SU BROUTINE ORGLQ_64 $M, \mathbb{N}],[K], A,[L D A], T A U,[W O R K],[L D W O R K]$, [ $\mathbb{N}$ FO ])
$\mathbb{N}$ TEGER (8) ::M,N,K,LDA,LDW ORK, $\mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::A

## C INTERFACE

\#include <sunperfh>
void dorglq (intm , intn, int k, double *a, int lda, double
*tau, int *info);
void dorgle_64 (long m, long n, long k, double *a, long lda, double *tau, long *info);

## PURPOSE

dorglq generates an $M$ by N realm atrix Q with orthonorm al row $S$, which is defined as the firstM row $s$ of a product of $K$ elem entary reflectors of orderN

$$
Q=H(k) \ldots H(2) H(1)
$$

as retumed by SGELQF.

## ARGUMENTS

M (input) The num ber of row sof the m atrix $\mathrm{Q} . \mathrm{M}>=0$.

N (input) The num ber of colum ns of the $m$ atrix $\mathrm{Q} \cdot \mathrm{N}>=\mathrm{M}$.
$K$ (input) The num ber of elem entary reflectors $w$ hose product defines the $m$ atrix $Q . M>=K>=0$.

A (input/output)
On entry, the $i$-th row must contain the vector which defines the elem entary reflectorH (i), for i $=1,2, \ldots, k$, as retumed by SG ELQF in the first $k$ row sof its array argum entA. On exit, the $M$ by $-\mathbb{N}$ $m$ atrix $Q$.

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the elem entary reflectort (i), as retumed by SG ELQ F.

W ORK (w orkspace)
On exit, if $\mathbb{N F} F=0, W$ ORK (1) retums the optim al
LDW ORK.

LDW ORK (input)
The dimension of the amay W ORK. LDW ORK >= $\max (1, M)$. Foroptim um perform ance LDW ORK $>=M$ *NB,
w here N B is the optim al.blocksize.

IfLD W ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA .
$\mathbb{N}$ FO (output)
= 0: successfiulexit
$<0:$ if $\mathbb{I N F O}=-i$, the $i$ th argum enthas an illegalvałue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorgql-generate an M -by-N realm atrix Q with orthonorm al colum ns,

## SYNOPSIS



```
\mathbb{NTEGERM,N,K,LDA,LDW ORK,\mathbb{NFO}}\mathbf{N}=\mp@code{L}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
SU BROUTINEDORGQL_64M,N,K,A,LDA,TAU,W ORK,LDW ORK,\mathbb{NFO)}
\mathbb{NTEGER*8M,N,K,LDA,LDW ORK,INFO}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE ORGQLM, N ], [K],A, [LDA ],TAU, [W ORK ], [LDW ORK ], [ $\mathbb{N F O}]$ )
$\mathbb{N} T E G E R:: M, N, K, L D A, L D W O R K, \mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::A

SU BROUTINE ORGQL_64 M, $\mathbb{N}],[K], A,[L D A], T A U,[W O R K],[L D W O R K]$, [ $\mathbb{N} F O$ ])
$\mathbb{N}$ TEGER (8) ::M,N,K,LDA,LDW ORK, $\mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::A

## C INTERFACE

\#include < sunperfh>
void dorgql(intm , intn, int k, double *a, int lda, double
*tau, int *info);
void dorgql_64 (long m , long n, long k, double *a, long lda, double *tau, long *info);

## PURPOSE

dorgqlgenerates an $M$ by N realm atrix Q with orthonorm al colum ns, which is defined as the lastN colum ns of a product ofK elem entary reflectors of orderM
$\mathrm{Q}=\mathrm{H}(\mathrm{k}) \ldots \mathrm{H}(2) \mathrm{H}(1)$
as retumed by SG EQ LF .

## ARGUMENTS

M (input) The num ber of row s of the m atrix $\mathrm{Q} . \mathrm{M}>=0$.

N (input) The num ber of colum ns of the matrix $\mathrm{Q} . \mathrm{M}>=\mathrm{N} \quad>=$ 0.

K (input) The num ber of elem entary reflectors whose product defines the m atrix $\mathrm{Q} . \mathrm{N}>=\mathrm{K}>=0$.

A (input/output)
On entry, the $(n-k+i)$-th columnmust contain the vector which defines the elem entary reflector H (i), for $i=1,2, \ldots, k$, as retumed by SG EQ LF in the last k colum ns of its anay argum entA. On exit, the $M$-by $-\mathrm{N} m$ atrix $Q$.

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) must contain the scalar factor of the elem entary reflectort (i), as retumed by SG EQ LF .

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al
LDW ORK.
LDW ORK (input)
The dimension of the amay W ORK. LDW ORK >=
$m$ ax $(1, N)$. Foroptim um perform ance LD $W$ ORK $>=N * N B$, where NB is the optim alblocksize.

If LD W ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
< 0: if $\mathbb{N}$ FO $=-$ i, the $i$-th argum enthas an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorgqr-generate an $M$-by-N realm atrix $Q$ with orthonorm al colum ns,

## SYNOPSIS

```
SU BROUT\mathbb{NE DORGQRM,N,K,A,LDA,TAU,WORK,LDW ORK, INFO)}
\mathbb{NTEGERM,N,K,LDA,LDW ORK,\mathbb{NFO}}\mathbf{N}=\mp@code{L}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
SUBROUTINE DORGQR_64M,N,K,A,LDA,TAU,W ORK,LDW ORK,\mathbb{NFO)}
\mathbb{NTEGER*8M,N,K,LDA,LDW ORK,INFO}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE ORGQR $M, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N} F O])$
$\mathbb{N} T E G E R:: M, N, K, L D A, L D W O R K, \mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::A

SU BROUTINE ORGQR_64 $M, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{N}$ ORK ], [LDW ORK ], [ $\mathbb{N} F O$ ])
$\mathbb{N}$ TEGER (8) ::M,N,K,LDA,LDW ORK, $\mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8),D IM ENSION (:,:) ::A

## C INTERFACE

\#include < sunperfh>
void dorgqr(intm, intn, intk, double *a, int lda, double
*tau, int*info);
void dorgqr_ 64 (long m, long n, long k, double *a, long lda, double *tau, long *info);

## PURPOSE

dorgqrgenerates an M -by -N realm atrix Q w th orthonorm al colum ns, which is defined as the firstN colum ns of a productofK elem entary reflectors of orderM

$$
Q=H(1) H(2) \ldots H(k)
$$

as retumed by SG EQ RF.

## ARGUMENTS

M (input) The num ber of row sof the m atrix $\mathrm{Q} . \mathrm{M}>=0$.

N (input) The num ber of colum ns of the matrix $\mathrm{Q} . \mathrm{M}>=\mathrm{N} \quad>=$ 0.

K (input) The num ber of elem entary reflectors $w$ hose product defines the m atrix $\mathrm{Q} . \mathrm{N}>=\mathrm{K}>=0$.

A (input/output)
On entry, the $i$-th columnm ustcontain the vector which defines the elem entary reflector $H$ (i), for i
$=1,2, \ldots, k$, as retumed by SGEQRF in the first k colum ns of its array argum entA. On exit, the $M-$ by-N matrix $Q$.

LD A (input)
The first dim ension of the array A. LDA >= $m a x(1, M)$.

TAU (input)
TAU (i) $m$ ust contain the scalar factor of the ele$m$ entary reflectort (i), as retumed by SG EQRF.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK ( 1 ) retums the optim al
LDW ORK.
LDW ORK (input)
The dim ension of the amay W ORK. LDW ORK >=
$m$ ax $(1, N)$. Foroptim um perform ance LD $W$ ORK $>=N * N B$, where NB is the optim alblocksize.

If LD W ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
< 0: if $\mathbb{N}$ FO $=-$ i, the $i$-th argum enthas an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorgr2 - generate an $m$ by $n$ realm atrix $Q$ with orthonorm al row S,

## SYNOPSIS

```
SU BROUT\mathbb{NE DORGR2M,N,K,A,LDA,TAU,W ORK, NNFO)}
INTEGERM,N,K,LDA,\mathbb{NFO}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
SU BROUT\mathbb{NE DORGR2_64M,N,K,A,LDA,TAU,W ORK,INFO)}
INTEGER*8M,N,K,LDA, INFO
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
F95 INTERFACE
    SUBROUT\mathbb{NE ORGR2 (\mathbb{M ], N ], [K ],A, [LDA ],TAU , [W ORK ], [NFO ])}}\mathbf{N}\mathrm{ )}
    \mathbb{NTEGER ::M,N,K,LDA,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
    REAL (8),D IM ENSION (:) ::TAU,W ORK
    REAL (8),D IM ENSION (:,:) ::A
```



```
    \mathbb{NTEGER (8)::M,N,K,LDA, NNFO}
    REAL (8),D IM ENSION (:) ::TAU ,W ORK
    REAL (8),D IM ENSION (:,:) ::A
C INTERFACE
    #include <sunperfh>
    void dorgr2 (intm , intn, intk, double *a, int lda, double
```

void dorgr2_64 (long m, long n, long k, double *a, long lda, double *tau, long *info);

## PURPOSE

dorgr2 generates an $m$ by $n$ realm atrix $Q$ with orthonorm al row S , w hich is defined as the lastm row s of a productofk elem entary reflectors of ordern

$$
Q=H(1) H(2) \ldots H(k)
$$

as retumed by SGERQF.

## ARGUMENTS

M (input) The num ber of row s of the $m$ atrix $Q . M>=0$.

N (input) The num ber of colum ns of the matrix $\mathrm{Q} . \mathrm{N}>=\mathrm{M}$.
$K$ (input) The num ber of elem entary reflectors w hose product defines the $m$ atrix $\mathrm{Q} . \mathrm{M}>=\mathrm{K}>=0$.

A (input/output)
O $n$ entry, the ( $m-k+i$ )-th row $m$ ustcontain the vector which defines the elem entary reflectorH (i), fori= $1,2, \ldots, k$, as retumed by SGERQF in the lastk row sof its array argum entA. O n exit, the m by $n$ matrix $Q$.

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the elem entary reflectort (i), as retumed by SG ERQF.

W ORK (w orkspace)
dim ension M )
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum enthas an illegalvahue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorgrq - generate an $M$-by -N realm atrix Q w ith orthonorm al row S,

## SYNOPSIS

SU BROUTINE DORGRQ $M, N, K, A, L D A, T A U, W O R K, L D W O R K, \mathbb{N F O}$ )
$\mathbb{N}$ TEGER $M, N, K, L D A, L D W O R K, \mathbb{N} F O$
D OUBLE PRECISION A (LDA, $\left.{ }^{\star}\right)$, TAU ( ${ }^{*}$ ), WORK ( ${ }^{*}$ )

SU BROUTINE DORGRQ_64 M,N,K,A,LDA,TAU,W ORK,LDWORK, $\mathbb{N} F O$ )
$\mathbb{N} T E G E R * 8 M, N, K, L D A, L D W$ ORK, $\mathbb{N} F O$
DOUBLE PRECISION A (LDA, $\left.{ }^{*}\right)$,TAU (*),W ORK (*)

## F95 INTERFACE

SU BROUTINE ORGRQ $M, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N} F O])$
$\mathbb{N} T E G E R:: M, N, K, L D A, L D W$ ORK, $\mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::A
SU BROUTINE ORGRQ_64 $\mathbb{M}, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{N} O R K],[L D W$ ORK ], [ $\mathbb{N} F O$ ])
$\mathbb{N}$ TEGER (8) ::M,N,K,LDA,LDW ORK, $\mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::A

## C INTERFACE

\#include < sunperfh>
void dorgrq (intm, intn, intk, double *a, int lda, double
*tau, int *info);
void dorgra_64 (long m, long $n$, long $k$, double *a, long lda, double *tau, long *info);

## PURPOSE

dorgrq generates an $M$ boy $-N$ realm atrix $Q$ w th orthonorm al row $s$, which is defined as the lastM row s of a product of elem entary reflectors of orderN

$$
Q=H(1) H(2) \ldots H(k)
$$

as retumed by SG ERQF.

## ARGUMENTS

M (input) The num ber of row sof the m atrix $\mathrm{Q} . \mathrm{M}>=0$.

N (input) The num ber of colum ns of the $m$ atrix $Q . N>=M$.
$K$ (input) The num ber of elem entary reflectors w hose product defines the $m$ atrix $Q . M>=K>=0$.

A (input/output)
O $n$ entry, the ( $m-k+i$ )-th row $m$ ustcontain the vector which defines the elem entary reflectorH (i), fori=1,2,..,k, as retumed by SGERQF in the lastk row sof its array argum entA. O n exit, the M Hy -N m atrix Q .

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele$m$ entary reflectort (i), as retumed by SGERQF.

W ORK (w orkspace)
On exit, if $\mathbb{N F F O}=0, \mathrm{~W}$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the amay W ORK. LDW ORK >= $\max (1, M)$. Foroptim um perform ance LDW ORK $>=M * N B$,
w here N B is the optim al.blocksize.

IfLD W ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA .
$\mathbb{N}$ FO (output)
= 0: successfiulexit
$<0:$ if $\mathbb{I N F O}=-i$, the $i$ th argum enthas an illegalvałue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorgtr-generate a real orthogonal matrix $Q$ which is defined as the product of $n-1$ elem entary reflectors of order $N$, as retumed by SSY TRD

## SYNOPSIS

```
SUBROUT\mathbb{NEDORGTR(UPLO,N,A,LDA,TAU,W ORK,LW ORK,INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER N,LDA,LW ORK,\mathbb{NFO}}\mathbf{N}=\mp@code{L}
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
SUBROUT\mathbb{NE DORGTR_64(UPLO,N,A,LDA,TAU,W ORK,LW ORK,INFO )}
CHARACTER * 1 UPLO
NNTEGER*8N,LDA,LW ORK,NNFO
DOUBLE PRECISION A (LDA,*),TAU (*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE ORGTR (UPLO, $\mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L W O R K],[\mathbb{N} F O])$
CHARACTER (LEN=1)::UPLO
$\mathbb{N}$ TEGER ::N,LDA,LW ORK, $\mathbb{N}$ FO
REAL (8),D $\mathbb{I M}$ ENSION (:) ::TAU,W ORK
REAL (8),D IM ENSION (:,:) ::A
SU BROUTINE ORGTR_64 (UPLO, $\mathbb{N}], A,[L D A], T A U,[W$ ORK ], [LW ORK ], [ $\mathbb{N} F O$ ])
CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER (8) :: N, LDA, LW ORK, $\mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8),D IM ENSION (: : : : : A

## C INTERFACE

\#include <sunperfh>
void dorgtr(charuplo, intn, double *a, int lda, double
*tau, int *info);
void dorgtr_64 (charuplo, long n, double *a, long lda, double *tau, long *info);

## PURPOSE

dorgtrgenerates a realorthogonalm atrix Q which is defined as the productofn-1 elem entary reflectors of orderN, as retumed by SSY TRD :
if $U P L O=U$ ', $Q=H(n-1) \ldots$ (2) $H(1)$,
if $U P L O=L^{\prime}, Q=H(1) H(2) \ldots H(n-1)$.

## ARGUMENTS

UPLO (input)
= U ':U ppertriangle of A contains elem entary
reflectors from SSY TRD; = L': Low ertriangle of A
contains elem entary reflectors from SSY TRD.
N (input) The order of the $m$ atrix $\mathrm{Q} . \mathrm{N}>=0$.
A (input/output)
O $n$ entry, the vectors which define the elem entary reflectors, as retumed by SSY TRD. On exit, the N boy-N orthogonalm atrix Q .

LD A (input)
The leading din ension of the array A. LDA >= $\max (1, N)$.

TAU (input)
TAU (i) $m$ ust contain the scalar factor of the ele$m$ entary reflectort (i), as retumed by SSY TRD.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al
LW ORK.
LW ORK (input)
The dimension of the array W ORK. LW ORK >=
$\max (1, N-1)$. Foroptim um perform ance LW ORK $>=\mathbb{N}-$ 1) ${ }^{N} \mathrm{~N}$, where $N B$ is the optim alblocksize.

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{N F O}=-\mathrm{i}$, the i -th argum ent had an illegal value

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorm br-VECT = Q',SORM BR overw rites the general real M by $-\mathrm{N} m$ atrix C w th $S \mathbb{D} E=\mathrm{L} \mathrm{S}^{\prime} \mathrm{D} E=\mathrm{R}^{\prime} \mathrm{TRANS}=\mathrm{N}^{\prime}$

## SYNOPSIS

```
SUBROUTINE DORM BR NECT,SIDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,
    W ORK,LW ORK,INFO)
CHARACTER * 1VECT,SIDE,TRANS
\mathbb{NTEGER M,N,K,LDA,LDC,LW ORK,NNFO}
DOUBLE PRECISION A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
SUBROUTINE DORM BR_64 NECT,SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,
    W ORK,LW ORK,INFO)
```

CHARACTER * 1 VECT, SIDE,TRANS
$\mathbb{N}$ TEGER*8M,N,K,LDA,LDC,LW ORK, $\mathbb{N} F O$
DOUBLE PRECISION A (LDA,*),TAU (*), C (LDC ,*),W ORK (*)

## F95 INTERFACE

SU BROUTINE ORM BR $N E C T, S \mathbb{D} E,[T R A N S], \mathbb{M}], \mathbb{N}], K, A,[L D A], T A U, C$, [LDC], [W ORK], [LW ORK], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1) ::VECT,SDE,TRANS $\mathbb{N} T E G E R:: M, N, K, L D A, L D C, L W O R K, \mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8),D IM ENSION (:,:) ::A , C
SU BROUTINE ORM BR_64 NECT,SDE, [TRANS], $\mathbb{M}], \mathbb{N}], K, A,[L D A], T A U$, C , [LDC], [W ORK ], [LW ORK], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1) ::VECT,SDE,TRANS
$\mathbb{N}$ TEGER (8) :: M , N , K , LDA , LD C , LW ORK , $\mathbb{N}$ FO
REAL (8),D IM ENSION (:) ::TAU,W ORK
REAL (8), D $\mathbb{M}$ EN SION (:,:) ::A , C

## C INTERFACE

\#include <sunperfh>
void dorm br(charvect, charside, chartrans, intm, int n, intk, double *a, int lda, double *tau, double * C , int ldc, int *info);
void dorm br_64 (charvect, charside, char trans, long m, long $n$, long k, double *a, long lda, double *tau, double * c, long ldc, long *info);

## PURPOSE

dorm brVECT = Q', SO RM BR overw rites the general realM -by-N matrix C w ith

```
            \(S \mathbb{D E}=\mathbb{L}^{\prime} \quad S \mathbb{D E}=\mathrm{R}^{\prime} \operatorname{TRANS}=\mathrm{N}^{\prime}:\)
Q * C C * Q TRANS = T': Q**T *C C *
Q**T
```

IfVECT $=P$ ', SORM BR overw rites the general real M -by-N $m$ atrix C w ith

$$
S \mathbb{D} E=\mathbb{L}^{\prime} \quad S \mathbb{D} E=R^{\prime}
$$

TRANS $=N^{\prime}: \quad P * C \quad C * P$
TRANS = T': $\quad \mathrm{P} * * \mathrm{~T}$ * $\mathrm{C} \quad \mathrm{C}$ * P **T
$H$ ere $Q$ and $P * * T$ are the orthogonal $m$ atrices determ ined by SG EBRD when reducing a realm atrix A to bidiagonal form : A = $Q * B * P * * T . Q$ and $P * * T$ are defined asproducts of elem entary reflectors $H$ (i) and G (i) respectively.

Letnq $=m$ if $S \mathbb{D} E=L$ 'and $n q=n$ if $S \mathbb{D} E=R$ '. Thus nq is the order of the orthogonalm atrix Q orP**T that is applied.

IfVECT = Q',A is assum ed to have been an $N Q$ boy $K$ m atrix:
ifnq $>=k, Q=H(1) H(2) \ldots H(k)$;
ifng $<k, Q=H(1) H(2) \ldots H(n q-1)$.

IfVECT $=P$ ', A is assum ed to have been a K boy-NQ matrix:
if $k<n q, P=G(1) G(2) . . . G(k)$;
ifk $>=n q, P=G(1) G(2) \ldots G(n q-1)$.

## ARGUMENTS

VECT (input)
$=\mathrm{Q}$ ': apply Q orQ $\mathrm{A}^{*} \mathrm{~T}$;
= P ': apply P orP**T.

SID E (input)
$=\mathbb{L}$ ': apply $\mathrm{Q}, \mathrm{Q}$ **T, P orP**T from the Left;
$=R$ ': apply $Q, Q * * T, P$ or $P * * T$ from the $R$ ight.

TRANS (input)
$=\mathrm{N}: \mathrm{N}$ o transpose, apply Q orP;
$=T$ ': Transpose, apply $\mathrm{Q}^{* *} \mathrm{~T}$ or $\mathrm{P}^{* *} \mathrm{~T}$.

TRANS is defaulted to $N$ 'forF $95 \mathbb{N}$ TERFACE.

M (input) The num ber of row s of the m atrix $\mathrm{C} . \mathrm{M}>=0$.
N (input) The num ber of colum ns of the $m$ atrix $\mathrm{C} . \mathrm{N}>=0$.

K (input) IfVECT = Q', the num ber of colum ns in the original $m$ atrix reduced by $S G E B R D$. IfVECT $=P$ ', the num ber of row $S$ in the originalm atrix reduced by SGEBRD.K $>=0$.

A (input) (LDA,min (nq,K)) ifVECT = Q' (LDA,nq) if $\mathrm{VECT}=\mathrm{P}$ 'The vectors w hich define the elem entary reflectors H (i) and G (i) , w hose products determ ine the $m$ atrioes $Q$ and $P$, as retumed by SG EBRD.

LD A (input)
The leading dim ension of the array A. If VECT = Q', LDA $>=\max (1, n q)$; if $V E C T=P^{\prime}, L D A>=$ $m$ ax $(1, m$ in $(n q, K))$.

TAU (input)
TAU (i) m ustcontain the scalar factor of the ele$m$ entary reflectorH (i) orG (i) which determ ines Q orP, as retumed by SGEBRD in the array argum ent TAUQ orTAUP.

C (input/output)
On entry, the $M$ boy -N m atrix C . On exit, C is overw ritten by $\mathrm{Q} * \mathrm{C}$ or $\mathrm{Q} * \mathrm{~T} * \mathrm{C}$ or $\mathrm{C} \mathrm{Q}^{* *} \mathrm{~T}$ or C Q or $\mathrm{P} * \mathrm{C}$ or $\mathrm{P} * * \mathrm{~T} * \mathrm{C}$ or C * or $\mathrm{C} * \mathrm{P} * * \mathrm{~T}$.

LD C (input)
The leading dim ension of the aray C. LD C >= $\max (1, M)$.

W ORK (w orkspace)
On exit, if $\mathbb{N F O}=0, W$ ORK (1) retums the optim al

LW ORK .

LW ORK (input)
The dim ension of the array $W$ ORK. IfSDE $E L$ ',
LW ORK >= max (1,N); if $S \mathbb{D} E=R '$ LW ORK >=
max ( $1, \mathrm{M}$ ). Foroptim um perform ance LW ORK >= N *NB
if $S \mathbb{D} E=L^{\prime}$, and LW $O R K>=M * N B$ if $S \mathbb{D} E=R$ ',
where NB is the optim alblocksize.
IfLW ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LW ORK is issued by X ERBLA.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
< 0: if $\mathbb{N}$ FO $=-$ i, the i-th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorm hr-overw rite the general real M -by-N matrix C w ith $S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime}$ TRANS $=N^{\prime}$

## SYNOPSIS



```
    W ORK,LW ORK,INFO)
CHARACTER * 1SIDE,TRANS
```



```
DOUBLE PRECISION A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
```



```
    LDC,W ORK,LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
```



```
DOUBLE PRECISION A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE ORM HR (SDE, [TRANS], $\mathbb{M}], \mathbb{N}], \mathbb{H}, \mathbb{H} I, A,[L D A], T A U, C$, [LDC], [W ORK], [LW ORK], [NFO])

CHARACTER (LEN=1) ::SDE,TRANS
$\mathbb{N} T E G E R:: M, N, \mathbb{L O}, \mathbb{H} I, L D A, L D C, L W O R K, \mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8),D IM ENSION (:,:) ::A , C
SUBROUTINE ORM HR_64 (SDE, [TRANS], $\mathbb{M}], \mathbb{N}], \mathbb{H}, \mathbb{H} I, A,[L D A], T A U$, C, [LDC], [W ORK], [LW ORK], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1) ::SIDE,TRANS
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{ILO}, \mathbb{H} \mathrm{I}, \mathrm{LD} \mathrm{A}, \mathrm{LD} \mathrm{C}, \mathrm{LW} O R K, \mathbb{N} F O$ REAL (8), D $\mathbb{I M} E N S I O N(:):: T A U, W O R K$
REAL (8), D $\mathbb{M}$ ENSION (:,:) :: A , C

## C INTERFACE

\#include <sunperfh>
void dorm hr(charside, chartrans, intm , int n, int مlo, int ini, double *a, int lda, double *tau, double ${ }^{*} \mathrm{C}$, int $\mathrm{ld}_{\mathrm{l}}$, int *info);
void dorm hr 64 (charside, chartrans, long m, long n, long ilo, long ini, double *a, long lda, double *tau, double * c , long ldc, long *info);

## PURPOSE

dorm hroverw rites the general real $M-b y-N m a t r i x ~ C ~ w ~ i t h ~$ TRANS = T': $\mathrm{Q}^{* * T * C \quad C * Q * * T ~ T ~}$
where $Q$ is a real orthogonalm atrix of order nq, w ith nq = m if $S \mathbb{D} E=L$ 'and $n q=n$ if $S \mathbb{D} E=R$ '. Q is defined as the productof $\mathbb{H}$ I-HO elem entary reflectors, as retumed by SG EHRD :


## ARGUMENTS

SID E (input)
$=\mathrm{L}$ ': apply Q or $\mathrm{Q} * * \mathrm{~T}$ from the Left;
$=\mathrm{R}^{\prime}$ : apply Q orQ ${ }^{* *} \mathrm{~T}$ from the R ight.

TRANS (input)
$=\mathrm{N}^{\prime}: \mathrm{N}$ o transpose, apply Q ;
$=T$ ': Transpose, apply $\mathrm{Q} * * \mathrm{~T}$.

TRANS is defaulted to $N$ 'forF $95 \mathbb{N}$ TERFACE.

M (input) The num ber of row s of the m atrix $\mathrm{C} . \mathrm{M}>=0$.

N (input) The num ber of colum ns of the m atrix $\mathrm{C} . \mathrm{N}>=0$.

IIO (input)
IIO and IH Im usthave the sam evalues as in the
previous call of SGEHRD.Q is equal to the unit
$m$ atrix except in the subm atrix

Q (ilo+1:ihi,ilo+1: : ihi). IfS $\mathbb{D} E=4$ ', then $1<=$
$\mathbb{H O}<=\mathbb{H} I<=M$, if $M>0$, and $\mathbb{H O}=1$ and $\mathbb{H} I=$
0 , if $M=0$; if $S \mathbb{D} E=R$ ', then $1<=\mathbb{H O}<=\mathbb{H} I$
$<=\mathrm{N}$, if $\mathrm{N}>0$, and $\mathbb{H} \mathrm{O}=1$ and $\mathbb{H} \mathrm{I}=0$, if $\mathrm{N}=0$.

IH I (input)
See the description of IIO .

A (input) (LDA,M) if $S \mathbb{D} E=L^{\prime}(\mathbb{L D A} N)$ if $S \mathbb{D} E=R^{\prime}$ The vectors which define the elem entary reflectors, as retumed by SGEHRD.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, M)$ ifSDE $=L ; \operatorname{LDA}>=\max (1, N)$ if $S \mathbb{D} E=$ R'.

TAU (input)
$(M-1)$ if $S \mathbb{D} E=L^{\prime}(\mathbb{N}-1)$ if $S \mathbb{D E}=R^{\prime} T A U(i)$ $m$ ust contain the scalar factor of the elem entary reflectorH (i), as retumed by SG EH RD .

C (input/output)
On entry, the $M-b y-N$ matrix C. On exit, $C$ is overw ritten by Q * C or $\mathrm{Q} * * \mathrm{~T} * \mathrm{C}$ or C * $\mathrm{Q} * \mathrm{~T}$ or C * Q .

LD C (input)
The leading dim ension of the aray C. LD C >= max (1, M).

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al
LW ORK.

LW ORK (input)
The dim ension of the array $W$ ORK. IfSDE $=\mathrm{L}$ ', LW ORK >= max ( $1, N$ ); if $S \mathbb{D} E=R$ ', LW ORK $>=$ $\max (1, M)$. Foroptim um perform ance LW ORK $>=N * N B$ if $S \mathbb{D} E=L$ ', and LW ORK $>=M * N B$ if $S \mathbb{D} E=R$ ', w here NB is the optim alblocksize.

If LW ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum enthad an ille-
galvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorm lq-overw rite the general real M -by N matrix C with $S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R{ }^{\prime} T R A N S=N^{\prime}$

## SYNOPSIS

```
SUBROUT\mathbb{NEDORM LQ (SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}
    LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
\mathbb{N TEGER M,N,K,LDA,LDC,LW ORK, NNFO}
DOUBLE PRECISION A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
SUBROUT\mathbb{NE DORM LQ_64 (S\mathbb{DE,TRANS,M ,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{T},\textrm{L},\textrm{L}
    LW ORK,\mathbb{NFO)}
```

CHARACTER * 1 SIDE,TRANS
$\mathbb{N}$ TEGER*8M,N,K,LDA,LDC,LW ORK, $\mathbb{N} F O$
DOUBLE PRECISION A (LDA ,*),TAU (*), C (LDC , *), W ORK (*)

## F95 INTERFACE

SU BROUTINE ORM LQ (SDE, [TRANS], $\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C,[L D C]$, [W ORK ], [LW ORK ], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::SDE,TRANS
$\mathbb{N} T E G E R:: M, N, K, L D A, L D C, L W O R K, \mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8), D $\mathbb{M}$ ENSIO N (:,:) ::A , C
SUBROUTINE ORM LQ_64 (SDE, [TRANS], $\mathbb{M}], \mathbb{N}],[\mathbb{K}], A,[L D A], T A U, C$, [LDC], [W ORK ], [LW ORK ], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1) ::SDE,TRANS
$\mathbb{N}$ TEGER (8) :: $\mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathrm{LD} \mathrm{C}, \mathrm{LW}$ ORK, $\mathbb{N}$ FO
REAL (8),D IM ENSION (:) ::TAU,W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) ::A , C

## C INTERFACE

\#include <sunperfh>
void dorm lq (char side, chartrans, intm, intn, intk, double *a, int lda, double *tau, double *c, int ldc, int*info);
void dorm lq_64 (char side, chartrans, long m, long n, long k , double *a, long lda, double *tau, double *c, long ldc, long *info);

## PURPOSE

dorm lq overw rites the general real $M$ boy-N matrix $\mathrm{C} w$ ith TRANS = $\mathrm{T}: \quad \mathrm{Q} * * \mathrm{~T} * \mathrm{C} \quad \mathrm{C} * \mathrm{Q} * * \mathrm{~T}$
where $Q$ is a realorthogonalm atrix defined as the product ofk elem entary reflectors

$$
Q=H(k) \ldots H(2) H(1)
$$

as retumed by SGELQF.Q is oforderM if $S \mathbb{D} E=\mathbb{L}$ 'and of orderN ifSTDE = R'.

## ARGUMENTS

SID E (input)
$=\mathrm{L}$ ': apply Q or $\mathrm{Q}{ }^{* *} \mathrm{~T}$ from the Left;
$=R$ ': apply Q or $\mathrm{Q} * * \mathrm{~T}$ from the R ight.

TRANS (input)
$=\mathrm{N}$ : N o transpose, apply Q ;
$=T$ ': T ranspose, apply $Q * * T$.

TRANS is defaulted to $N$ 'forF $95 \mathbb{N}$ TERFACE.

M (input) The num ber of row s of the $m$ atrix $C . M>=0$.

N (input) The num ber of colum ns of the m atrix $\mathrm{C} . \mathrm{N}>=0$.

K (input) The num ber of elem entary reflectors w hose product defines them atrix $Q$. IfS $\mathbb{D} E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0$; if $S \mathbb{D} E=R \prime N>=K>=0$ 。
 $i$-th row m ustcontain the vectorw hich defines the elem entary reflector $H$ ( $i$ ), for $i=1,2, \ldots, k$, as retumed by $S G E L Q F$ in the firstk row sof its array argum entA. A ism odified by the routine butrestored on exit.

LDA (input)
The leading dim ension of the array A. LDA >= $\max (1, K)$.

TAU (input)
TAU (i) m ustcontain the scalar factor of the ele$m$ entary reflectorH (i), as retumed by SG ELQ F.

C (input/output)
On entry, the $M-b y-N m$ atrix $C$. On exit, $C$ is overw rilten by Q * C or $\mathrm{Q} * * \mathrm{~T} * \mathrm{C}$ or $\mathrm{C} * \mathrm{Q} * \mathrm{~T}$ or C Q .

LD C (input)
The leading dim ension of the array $C . \operatorname{LDC}>=$ $\max (1, M)$.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the aray W ORK. IfSIDE = L', LW ORK >= max ( $1, N$ ); if $S \mathbb{D} E=R$ ', LWORK $>=$ $m$ ax $(1, M)$. Foroptim um perform ance LW ORK $>=N * N B$ if $S \mathbb{D} E=L$ ', and LW ORK $>=M * N B$ ifS $\mathbb{D} E=R$ ', w here N B is the optim alblocksize.

If LW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N ~ F O ~}=-$ i, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorm ql-overw rite the general real M -by -N m atrix C w ith $S \mathbb{D E}=L^{\prime} S \mathbb{D} E=R^{\prime}$ TRANS $=N^{\prime}$

## SYNOPSIS

```
SUBROUT\mathbb{NEDORMQL (SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}
    LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
\mathbb{N TEGER M,N,K,LDA,LDC,LW ORK, NNFO}
DOUBLE PRECISION A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
SUBROUT\mathbb{NE DORMQL_64 (S\mathbb{DE,TRANS,M ,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{T}\mathrm{ , L,}
    LW ORK,\mathbb{NFO)}
```

CHARACTER * 1 SDE,TRANS
$\mathbb{N}$ TEGER*8M,N,K,LDA,LDC,LW ORK, $\mathbb{N} F O$
DOUBLE PRECISION A (LDA ,*),TAU (*), C (LDC , *), W ORK (*)

## F95 INTERFACE

SU BROUTINE ORM QL (SDE, [TRANS], M ], $\mathbb{N}],[K], A,[L D A], T A U, C,[L D C]$, [W ORK], [LW ORK ], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::SDE,TRANS
$\mathbb{N} T E G E R:: M, N, K, L D A, L D C, L W O R K, \mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8), D $\mathbb{M}$ ENSIO N (:,:) ::A , C
SUBROUTINE ORM QL_64 (SDE, [TRANS], $\mathbb{M}], \mathbb{N}],[\mathbb{K}], A,[L D A], T A U, C$, [LDC], [W ORK ], [LW ORK ], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1) ::SDE,TRANS
$\mathbb{N}$ TEGER (8) :: $\mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathrm{LD} \mathrm{C}, \mathrm{LW}$ ORK, $\mathbb{N}$ FO REAL (8),D IM ENSION (:) ::TAU,W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) ::A , C

## C INTERFACE

\#include <sunperfh>
void dorm ql(char side, chartrans, intm, intn, intk, double *a, int lda, double *tau, double *c, int ldc, int*info);
void dorm q1 64 (char side, char trans, long m, long n, long k , double *a, long lda, double *tau, double *c, long ldc, long *info);

## PURPOSE

dorm qloverw rites the general real $M$ boy- N matrix $\mathrm{C} w$ ith TRANS = $T^{\prime}: \quad Q^{* *} T * C \quad C * Q * * T$
where $Q$ is a realorthogonalm atrix defined as the product ofk elem entary reflectors

$$
Q=H(k) \ldots H(2) H(1)
$$

as retumed by SGEQLF. Q is oforderM if $S \mathbb{D} E=\mathrm{L}$ 'and of orderN ifSTDE = R'.

## ARGUMENTS

SID E (input)
$=\mathrm{L}$ ': apply Q or $\mathrm{Q}{ }^{* *} \mathrm{~T}$ from the Left;
$=R$ ': apply Q or $\mathrm{Q} * * \mathrm{~T}$ from the R ight.

TRANS (input)
$=\mathrm{N}$ : N o transpose, apply Q ;
$=T$ ': T ranspose, apply $Q$ **T .

TRANS is defaulted to $N$ 'forF $95 \mathbb{N}$ TERFACE.

M (input) The num ber of row s of the $m$ atrix $C . M>=0$.

N (input) The num ber of colum ns of the m atrix $\mathrm{C} . \mathrm{N}>=0$.

K (input) The num ber of elem entary reflectors w hose product defines them atrix $Q$. IfS $\mathbb{D} E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0$; if $S \mathbb{D} E=R \prime N>=K>=0$ 。

A (input) The i-th colum $n$ must contain the vector which defines the elem entary reflector H (i), for $i=$ $1,2, \ldots, k$, as retumed by SGEQ LF in the last $k$ colum ns of its aray argum entA. A ism odified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A . If $\mathrm{SDE}=$ L', LDA $>=m a x(1, M)$; if $S \mathbb{D E}=R \prime$,LDA $>=$ $\max (1, N)$.

TAU (input)
TAU (i) $m$ ust contain the scalar factor of the ele$m$ entary reflector H (i), as retumed by SGEQ LF .

C (input/output)
On entry, the $M$ by $-N$ matrix C. On exit, $C$ is overw ritten by Q * C or $\mathrm{Q} * * \mathrm{~T}$ * C or C Q Q * T or C * Q .

LD C (input)
The leading dim ension of the array C.LDC >= $\mathrm{max}(1, \mathrm{M})$.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array $W$ ORK. IfSDE $E L$ ', LW ORK >= max ( $1, N$ ); if $S \mathbb{D} E=R$ ', LW ORK >= $\max (1, M)$. Foroptim um perform ance LW ORK >= $N$ *NB if $S \mathbb{D} E=L '$, and $L W O R K>=M * N B$ if $S \mathbb{D} E=R$ ', where NB is the optim alblocksize.

IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK amray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{N}$ FO = -i, the i-th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorm qr-overw rite the general real M -by -N m atrix C w th $S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime}$ TRANS $=N^{\prime}$

## SYNOPSIS

```
SUBROUTINEDORMQR(SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,
    LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
\mathbb{N TEGERM,N,K,LDA,LDC,LW ORK, INFO}
DOUBLE PRECISION A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
SUBROUT\mathbb{NE DORM QR_64 (S\mathbb{DE,TRANS,M ,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{T},\textrm{T},\textrm{T}
    LW ORK,\mathbb{NFO)}
```

CHARACTER * 1 SDE,TRANS
$\mathbb{N}$ TEGER*8M,N,K,LDA,LDC,LW ORK, $\mathbb{N} F O$
DOUBLE PRECISION A (LDA ,*), TAU (*), C (LDC , *), W ORK (*)

## F95 INTERFACE

SU BROUTINE ORM QR (SDE, [TRANS], $\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C,[L D C]$, $[\mathbb{W}$ ORK ], [LW ORK ], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1) ::SDE,TRANS
$\mathbb{N} T E G E R:: M, N, K, L D A, L D C, L W O R K, \mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::A , C
SUBROUTINE ORM QR_64 (SDE, [TRANS], $\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C$, [LDC], [W ORK], [LW ORK], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1) ::SDE,TRANS
$\mathbb{N}$ TEGER (8) :: $\mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathrm{LDC}, \mathrm{LW}$ ORK, $\mathbb{N} F \mathrm{FO}$ REAL (8),D IM ENSION (:) ::TAU,W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) ::A , C

## C INTERFACE

\#include <sunperfh>
void dorm qr(char side, char trans, intm, intn, intk, double *a, int lda, double *tau, double *c, int ldc, int*info);
void dorm qr 64 (charside, chartrans, long m, long n, long k , double *a, long lda, double *tau, double *c, long ldc, long *info);

## PURPOSE

dorm qroverw rites the general real M boy N m atrix C w ith TRANS = T': $\mathrm{Q}^{* *} \mathrm{~T} * \mathrm{C} \quad \mathrm{C} * \mathrm{Q} * * \mathrm{~T}$
where $Q$ is a realorthogonalm atrix defined as the product ofk elem entary reflectors

$$
Q=H(1) H(2) \ldots H(k)
$$

as retumed by SGEQRF.Q is oforderM if $S \mathbb{D} E=L$ 'and of orderN ifSTDE = R'.

## ARGUMENTS

SDE (input)
$=\mathrm{L}$ ': apply Q or $\mathrm{Q}{ }^{* *} \mathrm{~T}$ from the Left;
$=R$ ': apply Q or $\mathrm{Q} * * \mathrm{~T}$ from the R ight.

TRANS (input)
$=\mathrm{N}$ : N o transpose, apply Q ;
$=T$ ': Transpose, apply $Q^{* *} T$.

TRANS is defaulted to $N$ 'forF $95 \mathbb{N}$ TERFACE.

M (input) The num ber of row s of the $m$ atrix $\mathrm{C} . \mathrm{M}>=0$.

N (input) The num ber of colum ns of the m atrix $\mathrm{C} . \mathrm{N}>=0$.

K (input) The num ber of elem entary reflectors w hose product defines them atrix $Q$. IfS $\mathbb{D} E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0$; if $S \mathbb{D} E=R \prime N>=K>=0$ 。

A (input) The i-th colum $n$ must contain the vector which defines the elem entary reflector $H$ (i), for $i=$ $1,2, \ldots, k$, as retumed by SGEQ RF in the first $k$ colum ns of its anay argum entA. A ism odified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A . If $\mathrm{SDE} \mathrm{E}=$ L', LDA >= max (1,M); if $S \mathbb{D E}=R$ ',LDA $>=$ $\max (1, N)$.

TAU (input)
TAU (i) $m$ ust contain the scalar factor of the ele$m$ entary reflector $H$ (i), as retumed by SG EQ RF .

C (input/output)
On entry, the $M$ by $-N$ matrix C. On exit, $C$ is overw ritten by Q * C or $\mathrm{Q} * \mathrm{~T}$ * C or C * $\mathrm{Q} * \mathrm{~T}$ or C * Q .

LD C (input)
The leading dim ension of the array C.LDC >= max (1, M).

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK ( 1 ) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array $W$ ORK. IfSDE $=\mathrm{L}$ ', LW ORK >= max ( $1, N$ ); if $S \mathbb{D} E=R$ ', LW ORK >= $\max (1, M)$. Foroptim um perform ance LW ORK $>=N * N B$ if $S \mathbb{D} E=L$ ', and LW ORK $>=M * N B$ if $S \mathbb{D} E=R$ ', where NB is the optim alblocksize.

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK ancay, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{N}$ FO $=-$ i, the i-th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorm rq -overw rite the general real M -by -N matrix C w ith $S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime}$ TRANS $=N^{\prime}$

## SYNOPSIS

```
SUBROUTINEDORM RQ (SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,
    LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
\mathbb{NTEGERM,N,K,LDA,LDC,LW ORK, INFO}
DOUBLE PRECISION A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
SUBROUT\mathbb{NE DORMRQ_64 (S\mathbb{DE,TRANS,M ,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{T},\textrm{L},\textrm{T}
    LW ORK,\mathbb{NFO)}
```

CHARACTER * 1 SDE,TRANS
$\mathbb{N}$ TEGER*8M,N,K,LDA,LDC,LW ORK, $\mathbb{N} F O$
DOUBLE PRECISION A (LDA,*),TAU (*), C (LDC , *),W ORK (*)

## F95 INTERFACE

SU BROUTINE ORMRQ (SDE, [TRANS], $\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C,[L D C]$, $[\mathbb{W}$ ORK ], [LW ORK ], [ $\mathbb{N F O}$ ])

CHARACTER ( $4 E N=1):: S \mathbb{D} E, T R A N S$
$\mathbb{N} T E G E R:: M, N, K, L D A, L D C, L W O R K, \mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::A , C
SU BROUTINE ORMRQ_64 (SDE, [TRANS], $\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C$, [LDC], [W ORK], [LW ORK], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1) ::SIDE,TRANS
$\mathbb{N}$ TEGER (8) :: $\mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathrm{LDC}, \mathrm{LW}$ ORK, $\mathbb{N} F \mathrm{FO}$ REAL (8),D IM ENSION (:) ::TAU,W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) ::A , C

## C INTERFACE

\#include <sunperfh>
void dorm rq (char side, chartrans, intm, intn, int k, double *a, intlda, double *tau, double *c, int ldc, int*info);
void dorm rq_64 (charside, chartrans, long m, long n, long k , double *a, long lda, double *tau, double *c, long ldc, long *info);

## PURPOSE

dorm rq overw rites the general real $M$ boy- N matrix $\mathrm{C} w$ ith TRANS = T': $\mathrm{Q}^{* *} \mathrm{~T} * \mathrm{C} \quad \mathrm{C} * \mathrm{Q} * * \mathrm{~T}$
where $Q$ is a realorthogonalm atrix defined as the product ofk elem entary reflectors

$$
Q=H(1) H(2) \ldots H(k)
$$

as retumed by SGERQF.Q is of orderM if $S \mathbb{D} E=L$ 'and of orderN ifSTDE = R'.

## ARGUMENTS

SDE (input)
$=\mathrm{L}$ ': apply Q or $\mathrm{Q}{ }^{* *} \mathrm{~T}$ from the Left;
$=R$ ': apply Q or $\mathrm{Q} * * \mathrm{~T}$ from the R ight.

TRANS (input)
$=\mathrm{N}$ : N o transpose, apply Q ;
$=T$ ': T ranspose, apply $Q$ **T .

TRANS is defaulted to $N$ 'forF $95 \mathbb{N}$ TERFACE.

M (input) The num ber of row s of the $m$ atrix $C . M>=0$.

N (input) The num ber of colum ns of the m atrix $\mathrm{C} . \mathrm{N}>=0$.
$K$ (input) The num ber of elem entary reflectors w hose product defines the $m$ atrix $Q$. IfS $\mathbb{D} E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0$; if $S \mathbb{D} E=R \prime N>=K>=0$ 。
 $i$-th row m ustcontain the vectorw hich defines the elem entary reflector $H$ ( $i$ ), for $i=1,2, \ldots, k$, as retumed by $S G E R Q F$ in the lastk row $s$ of its anay argum entA. A is modified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, K)$.

TAU (input)
TAU (i) m ust contain the scalar factor of the ele$m$ entary reflectorH (i), as retumed by SG ERQ F.

C (input/output)
On entry, the $M$ boy- $\mathrm{N} m$ atrix C . On exit, C is overw rilten by $\mathrm{Q} * \mathrm{C}$ or $\mathrm{Q} * * \mathrm{~T} * \mathrm{C}$ or C * $\mathrm{Q} * \mathrm{~T}$ or C Q .

LD C (input)
The leading dim ension of the array $C . \operatorname{LDC}>=$ $\max (1, M)$.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the amay W ORK. IfSIDE = L', LW ORK >= max ( $1, N$ ); if $S \mathbb{D} E=R$ ', LWORK >= $m$ ax $(1, M)$. Foroptim um perform ance LW ORK $>=N * N B$ if $S \mathbb{D E}=L^{\prime}$ ', and LW ORK $>=M * N B$ ifSDE $=R$ ', w here N B is the optim alblocksize.

If LW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N ~ F O ~}=-$ i, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dorm $r z$-overw rite the general real $M$ boy $-\mathrm{N} m$ atrix $C \mathrm{w}$ ith $S \mathbb{D} E=\mathbb{L} S \mathbb{D} E=R{ }^{\prime}$ TRANS $=N^{\prime}$

## SYNOPSIS

```
SUBROUT\mathbb{NE DORM RZ (S\mathbb{DE,TRANS,M ,N,K,L,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{T},\textrm{L},\textrm{L}
    LW ORK,\mathbb{NFO)}
CHARACTER * 1SDDE,TRANS
\mathbb{NTEGERM,N,K,L,LDA,LDC,LW ORK,INFO}
DOUBLE PRECISION A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
SUBROUTINE DORMRZ_64(SIDE,TRANS,M,N,K,L,A,LDA,TAU,C,LDC,
        W ORK,LW ORK,INFO)
```

CHARACTER * 1 SIDE,TRANS
$\mathbb{N}$ TEGER*8M,N,K,L,LDA,LDC,LWORK, $\mathbb{N}$ FO
DOUBLE PRECISION A (LDA ,*), TAU (*), C (LDC , *), W ORK (*)

## F95 INTERFACE

SU BROUTINE ORMRZ (SDE,TRANS, $\mathbb{M}], \mathbb{N}], K, L, A,[L D A], T A U, C,[L D C]$, [W ORK ], [LW ORK ], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::SDE,TRANS
$\mathbb{N}$ TEGER :: M , N, K, L,LDA,LDC,LW ORK, $\mathbb{N} F O$
REAL (8), D $\mathbb{I M}$ ENSION (:) ::TAU,W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) ::A, C
SUBROUTINE ORMRZ_64 (SDE,TRANS, $\mathbb{M}], \mathbb{N}], K, L, A,[L D A], T A U, C$,
[LDC], [W ORK], [LW ORK], [NFO])

CHARACTER (LEN=1) ::SIDE,TRANS
$\mathbb{N}$ TEGER (8) :: M , N , K , L, LDA , LD C , LW ORK , $\mathbb{N} F O$
REAL (8), D $\mathbb{I}$ ENSION (:) :: TAU ,W ORK
REAL (8), D $\mathbb{M}$ ENSION (: : : : : : A , C

## C INTERFACE

\#include <sunperfh>
void dorm rz (charside, chartrans, intm, intn, int $k$, int 1, double *a, int lda, double *tau, double *C, int ldc, int *info);
void dorm rz_64 (char side, char trans, long m, long n, long k, long l, double *a, long lda, double *tau, double *c, long ldc, long *info);

## PURPOSE

dorm $r z$ overw rites the general real $M$ boy N N matrix $\mathrm{C} w$ ith TRANS = T': $\quad Q^{* *} T * C \quad C * Q^{* *} T$
where $Q$ is a realorthogonalm atrix defined as the product ofk elem entary reflectors

$$
Q=H(1) H(2) \ldots H(k)
$$

as retumed by STZRZF.Q is oforderM if SID $E=L$ 'and of order $N$ if $S \mathbb{D} E=R$.

## ARGUMENTS

SIDE (input)
$=\mathrm{L}$ ': apply Q or $\mathrm{Q} * * \mathrm{~T}$ from the Left;
$=R$ ': apply Q or $\mathrm{Q} * * \mathrm{~T}$ from the R ight.

TRANS (input)
$=\mathrm{N}$ ': N o transpose, apply $Q$;
= T ': T ranspose, apply $Q$ **T .
$M$ (input) The num ber of row s of the $m$ atrix $C . M>=0$.

N (input) The num ber of colum ns of the m atrix $\mathrm{C} . \mathrm{N}>=0$.

K (input) The num ber of elem entary reflectors w hose product defines the $m$ atrix $Q$. IfS $\mathbb{D} E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0$;
ifS $\mathbb{D} E=R{ }^{\prime}, N>=K>=0$.

L (input) The num ber of colum ns of the $m$ atrix A containing
the $m$ eaningfulpart of the $H$ ouseholder reflectors.
If $S \mathbb{D} E=L \prime, M>=L>=0$, if $S \mathbb{D} E=R \prime N>=L$
$>=0$ 。

A (input) (LDA, M) if $S \mathbb{D} E=L \prime$ ( $L D A, N$ ) if $S \mathbb{D} E=R^{\prime}$ The $i$-th row $m$ ustcontain the vectorw hich defines the elem entary reflector $H$ ( $i$ ), for $i=1,2, \ldots, k$, as retumed by STZRZF in the lastk row s of its array argum entA. A is modified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array $\mathrm{A} . \mathrm{LDA}>=$ $\max (1, K)$.

TAU (input)
TAU (i) m ust contain the scalar factor of the ele$m$ entary reflector H (i), as retumed by STZRZF.

C (input/output)
On entry, the $M$ boy -N m atrix C . On exit, C is overw ritten by Q * C or $\mathrm{Q} * * \mathrm{H} * \mathrm{C}$ or $\mathrm{C} \mathrm{Q}^{*}{ }^{*} \mathrm{H}$ or C * Q .

LD C (input)
The leading dim ension of the aray C.LDC >= $\max (1, M)$.

W ORK (w orkspace)
On exit, if $\mathbb{N F O}=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array $W$ ORK. If $S \mathbb{D} E=\mathbb{L}$ ', LW ORK >= max ( $1, N$ ); if $S \mathbb{D} E=R \prime$ LW ORK >= $m$ ax $(1, M)$. Foroptim um perform ance LW ORK $>=N * N B$ if $S \mathbb{D} E=L '$, and LW ORK $>=M * N B$ if $S D E=R \prime$, w here N B is the optim alblocksize.

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

INFO (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue

## FURTHER DETAILS

B ased on contributions by
A. Petitet, C om puter Science D ept., U niv . of Tenn., K noxville, U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dorm tr-overw rite the general real M -by N matrix C w th $S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R{ }^{\prime} T R A N S=N^{\prime}$

## SYNOPSIS

```
SU BROUTINE DORM TR (SDE,UPLO,TRANS,M,N,A,LDA,TAU,C,LDC,W ORK,
    LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,UPLO,TRANS
INTEGERM,N,LDA,LDC,LW ORK, INFO
DOUBLE PRECISION A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
SU BROUT\mathbb{NE DORM TR_64 (S\mathbb{DE,UPLO,TRANS,M ,N,A,LDA,TAU,C,LDC,}}\mathbf{T},\textrm{T},\textrm{T}
    W ORK,LW ORK,\mathbb{NFO)}
CHARACTER * 1SDDE,UPLO,TRANS
INTEGER*8M,N,LDA,LDC,LW ORK,\mathbb{N FO}
DOUBLE PRECISION A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE ORM TR (SDE, UPLO, [TRANS], $\mathbb{M}], \mathbb{N}], A,[L D A], T A U, C$, [LDC], [W ORK], [LW ORK], [ $\mathbb{N F O}])$

CHARACTER (LEN=1) ::SDE,UPLO,TRANS
$\mathbb{N} T E G E R:: M, N, L D A, L D C, L W$ ORK, $\mathbb{N} F O$
REAL (8), D IM ENSION (:) ::TAU,W ORK
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::A,C
SU BROUTINE ORM TR_64 (SDE,UPLO, [TRANS], $\mathbb{M}], \mathbb{N}], A,[L D A], T A U, C$, [LDC ], [W ORK ], [LW ORK ], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1) ::SDE,UPLO,TRANS
$\mathbb{N}$ TEGER (8) :: M , N, LDA, LD C , LW ORK, $\mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) ::A , C

## C INTERFACE

\#include <sunperfh>
void dorm tr (char side, charuplo, chartrans, intm, int n, double *a, int lda, double *tau, double *c, int ldc, int*info);
void dorm tr_64 (char side, charuplo, char trans, long m, long $n$, double *a, long lda, double *tau, double
${ }^{*}$ c, long ldc, long *info);

## PURPOSE

dorm troverw rites the general real M boy -N matrix C w ith

where $Q$ is a real orthogonalm atrix of order nq, w ith nq = m if $S \mathbb{D} E=\mathbb{L}$ 'and $n q=n$ if $S \mathbb{D} E=R$ '. Q is defined as the product of nq-1 elem entary reflectors, as retumed by SSY TRD :
if $U P L O=U ', Q=H(n q-1) \ldots H(2) H(1) ;$
if $U P L O=L^{\prime}, Q=H(1) H(2) \ldots H(n q-1)$.

## ARGUMENTS

SIDE (input)

$=R$ ': apply Q or $\mathrm{Q} * * \mathrm{~T}$ from the R ight.

UPLO (input)
$=\mathrm{U}$ ': Uppertriangle of A contains elem entary
reflectors from SSY TRD ; = 'L ': Low er triangle of A
contains elem entary reflectors from SSY TRD .

TRANS (input)
$=N^{\prime}: \mathrm{N} \circ$ transpose, apply Q ;
$=T$ ': T ranspose, apply $Q * * T$.

TRANS is defaulted to $N$ 'forF $95 \mathbb{I N}$ TERFACE.

M (input) The num ber of row $s$ of the $m$ atrix $C . M>=0$.

N (input) The num ber of colum ns of the m atrix $\mathrm{C} . \mathrm{N}>=0$.
A (input) ( $L D A, M$ ) if $S \mathbb{D} E=L^{\prime}(L D A, N)$ if $S \mathbb{D} E=R^{\prime}$ The vectors w hich define the elem entary reflectors, as retumed by SSY TRD.

LD A (input)
The leading dim ension of the anay A. LDA >= $\max (1, M)$ ifS $\mathbb{D} E=L ; L D A>=m a x(1, N)$ if $S \mathbb{D} E=$ R.

TAU (input)
$(M-1)$ ifSTDE $=\mathbb{L}(N-1)$ if $S \mathbb{D E}=R^{\prime} T A U$ (i) $m$ ust contain the scalar factor of the elem entary reflectorH (i), as retumed by SSY TRD .

C (input/output)
On entry, the $M$ boy -N m atrix C . On exit, C is overw ritten by $Q * C$ or $Q * * T * C$ or $C * Q * *$ or $C * Q$.

LD C (input)
The leading dim ension of the aray C. LD C >= $m$ ax (1, M).

W ORK (w orkspace)
On exit, if $\mathbb{N F F O}=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array $W$ ORK. IfSDE $=\mathrm{L}$ ', LW ORK >= max ( $1, N$ ); if $S \mathbb{D} E=R$ ', LW ORK >= $m a x(1, M)$. Foroptim um perform ance LW ORK $>=N * N B$ if $S \mathbb{D} E=L '$, and $L W O R K>=M * N B$ if $S \mathbb{D} E=R$ ', w here NB is the optim alblocksize.

IfLW ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
$\mathbb{N F O}$ (output)
= 0 : successfulexit
<0: if $\mathbb{N N}$ FO = -i, the i-th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dpboon -estim ate the reciprocal of the condition num ber (in the 1 -norm ) of a real symmetric positive definite band
$m$ atrix using the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=$ L*L**T com puted by SPBTRF

## SYNOPSIS

```
SU BROUT\mathbb{NE DPBCON (UPLO,N,KD,A,LDA,ANORM,RCOND,W ORK,W ORK2,}
    \mathbb{NFO )}
CHARACTER * 1 UPLO
\mathbb{NTEGER N,KD,LDA,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
INTEGER W ORK2 (*)
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION A (LDA,*),W ORK (*)
SUBROUTINEDPBCON_64 (UPLO,N,KD,A,LDA,ANORM,RCOND,W ORK,
    W ORK2,\mathbb{NFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N,KD,LDA,}\mathbb{N}FO
INTEGER*8W ORK2 (*)
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION A (LDA,*),W ORK (*)
```


## F95 INTERFACE

```
SU BROUTINE PBCON \(\mathbb{U} P L O, \mathbb{N}], K D, A,[L D A], A N O R M, R C O N D,[W O R K]\), [W ORK2], [ \(\mathbb{N} F O\) ])
CHARACTER (LEN=1) ::UPLO
\(\mathbb{N} T E G E R:: N, K D, L D A, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION (:) ::W ORK 2
```

REAL (8) ::ANORM,RCOND
REAL (8), D $\mathbb{I M}$ ENSION (:) ::W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) :: A

SU BROUTINE PBCON_64 (UPLO, $\mathbb{N}], K D, A,[L D A], A N O R M, R C O N D,[W O R K]$, [ W ORK2], $[\mathbb{N} \mathrm{FO}])$

CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER (8) :: $\mathrm{N}, \mathrm{KD}, \mathrm{LD} A, \mathbb{N} F O$
$\mathbb{N}$ TEGER (8), D $\mathbb{I M} \operatorname{ENSION}(:):: W$ ORK2
REAL (8) :: ANORM ,RCOND
REAL (8), D $\mathbb{I M}$ ENSION (:) ::W ORK
REAL (8), D $\mathbb{I M}$ EN SION (:,:) ::A

## C INTERFACE

\#include <sunperfh>
void dpbcon (charuple, intn, int kd, double *a, int lda, double anorm, double *rcond, int *info);
void dpbcon_64 (charuplo, long n, long kd, double *a, long lda, double anorm , double *roond, long *info);

## PURPOSE

dpboon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a real symm etric positive definite band $m$ atrix using the Cholesky factorization $A=U \star * T * U$ or $A=$ L*L**T com puted by SPB TRF.

A $n$ estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND = 1 / (ANORM * norm (inv (A))).

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': U pper triangular factor stored in A ;
= L ': Low er triangular factor stored in A .

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

K D (input)
The num ber of superdiagonals of the $m$ atrix $A$ if $\mathrm{UPLO}=\mathrm{U}$ ', or the num berof subdiagonals ifUPLO $=\mathrm{L}^{\prime} . \mathrm{KD}>=0$ 。

A (input) The triangular factorU or $L$ from the Cholesky
factorization $A=U * * T * U$ orA $=\mathrm{L} * \mathrm{~L} * * T$ of the band
$m$ atrix $A$, stored in the first $K D+1$ row $s$ of the array. The $j$ th colum n ofU orL is stored in the $j$ th colum $n$ of the array A as follow s: if UPLO
$=U \prime A(k d+1+i-j)=U(i, j)$ for $\max (1, j$
$\mathrm{kd})<=\dot{i}<=\dot{j}$ ifUPLO $=\mathrm{L}$ ', $A(1+i-j)=L(i, j)$
for $\dot{j}=i<=m$ in $(n, j+k d)$.

LD A (input)
The leading dim ension of the aray A. LDA >= K D +1 。
ANORM (input)
The 1-norm (or infinity-norm ) of the symm etric band $m$ atrix A.

## RCOND (output)

The reciprocal of the condition num ber of the $m$ atrix $A$, com puted as RCOND $=1 /(A N O R M \star A \mathbb{N V N M ) \text { , }}$ $w$ here $A \mathbb{N} V N M$ is an estim ate of the 1 -norm of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension ( $3 * N$ )

W ORK2 (w orkspace)
dim ension $\mathbb{N}$ )
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N F O}=-i$, the i-th argum enthad an ille-
galvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dpbequ - com pute row and colum n scalings intended to equilibrate a sym $m$ etric positive definite band $m$ atrix $A$ and reduce its condition num ber (w ith respect to the tw o-norm )

## SYNOPSIS

```
SUBROUT\mathbb{NE DPBEQU (UPLO,N,KD,A,LDA,SCALE,SCOND,AMAX,INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER N,KD,LDA,NNFO}
DOUBLE PRECISION SCOND,AMAX
D OUBLE PRECISION A (LDA,*),SCALE (*)
SU BROUT\mathbb{NE DPBEQU_64 (UPLO,N,KD,A,LDA,SCALE,SCOND,AMAX,}
        \mathbb{NFO)}
CHARACTER * 1 UPLO
INTEGER*8N,KD,LDA,INFO
DOUBLE PRECISION SCOND,AMAX
D OU BLE PREC ISION A (LDA,*),SCALE (*)
```


## F95 INTERFACE

SU BROUTINE PBEQU (UPLO, $\mathbb{N}], K D, A,[L D A], S C A L E, S C O N D, A M A X$, [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R:: N, K D, L D A, \mathbb{N} F O$
REAL (8) :: SCOND,AMAX
REAL (8),D $\mathbb{I}$ ENSION (:) ::SCALE
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::A
SU BROUTINE PBEQU_64 (UPLO, $\mathbb{N}], K D, A,[L D A], S C A L E, S C O N D, A M A X$,
[ $\mathbb{N}$ FO ])

CHARACTER (LEN=1) :: UPLO
$\mathbb{N}$ TEGER (8) :: N , KD, LDA , $\mathbb{N} F O$
REAL (8) :: SCOND , AM AX
REAL (8), D $\mathbb{I M}$ ENSION (:) :: SCALE
REAL (8), D $\mathbb{I M}$ ENSION (:,:) ::A

## C INTERFACE

\#include <sunperfh>
void dpbequ (charuple, intn, int kd, double *a, int lda, double *scale, double *scond, double *am ax, int *info);
void dpbequ_64 (charuplo, long n, long kd, double *a, long lda, double *scale, double *scond, double *am ax, long *info);

## PURPOSE

dpbequ com putes row and colum n scalings intended to equilibrate a sym $m$ etric positive definite band $m$ atrix A and reduce its condition num ber (w ith respect to the tw o-norm ). S contains the scale factors, $S(i)=1 /$ squt (A (i,i)), chosen so that the scaled $m$ atrix $B \quad w$ ith elem ents $B(i, 1)=$ $S(i) \star A(i, j) * S(j)$ has ones on the diagonal. This choige of $S$ puts the condition num berofB $w$ ithin a factor $N$ of the sm allest possible condition num ber over allpossible diagonalscalings.

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': Upper triangular ofA is stored;
= L ': Low ertriangularofA is stored.

N (input) The order of them atrix A. N $>=0$.

K D (input)
The num ber of superdiagonals of the $m$ atrix $A$ if U PLO $=\mathrm{U}$ ', orthe num ber of subdiagonals if U PLO $=\mathbb{L}^{\prime} . \mathrm{KD}>=0$ 。

A (input) The upper or low er triangle of the sym m etric band $m$ atrix $A$, stored in the firstK $D+1$ row $s$ of the array. The $j$ th colum n of $A$ is stored in the $j$ th colum $n$ of the array A as follow s: if UPLO = U',

A $(k d+1+i-j, j)=A(i, 7)$ for $\max (1, j k d)<=i<=j$ if UPLO $=L^{\prime}, A(1+i-j)=A(i, 7)$ for $\dot{j}=i<=m$ in $(n, \dot{j}+k d)$.

LD A (input)
The leading dim ension of the array A. LDA >= KD+1.

SCALE (output)
If $\mathbb{N} F O=0, S C A L E$ contains the scale factors for A.

SCOND (output)
If $\mathbb{N} F O=0, S C A$ LE contains the ratio of the sm allest SCA LE (i) to the largestSCA LE (i). If SCOND $>=0.1$ and AMAX is neither too large nor too sm all, it is notw orth scaling by SCA LE .

AM AX (output)
A bsolute value of largestm atrix elem ent. IfA M A X
is very close to overflow orvery close to under-
flow , the m atrix should be scaled.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{N}$ FO = -i, the i-th argum ent had an illegalvahue.
$>0$ : if $\mathbb{N F O}=$ i, the $i$-th diagonal elem ent is nonpositive.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dpbrfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is sym $m$ etric positive definite and banded, and provides emrorbounds and backw ard errorestim ates for the solution

## SYNOPSIS

```
SU BROUT\mathbb{NE DPBRFS (UPLO,N,KD,NRHS,A,LDA,AF,LDAF,B,LDB,X,}
    LDX,FERR,BERR,W ORK,W ORK 2,INFO)
CHARACTER * 1 UPLO
INTEGERN,KD,NRHS,LDA,LDAF,LDB,LDX,INFO
INTEGER W ORK2 (*)
D OU BLE PRECISION A (LDA,*),AF (LDAF,*), B (LDB,*), X (LDX,*),
FERR (*),BERR (*),W ORK (*)
SU BROUT\mathbb{NE DPBRFS_64 (UPLO,N,KD,NRHS,A,LDA,AF,LDAF,B,LDB,}
    X,LDX,FERR,BERR,W ORK,W ORK 2, INFO)
CHARACTER * 1 UPLO
INTEGER*8N,KD,NRHS,LDA,LDAF,LDB,LDX,INFO
INTEGER*8W ORK2 (*)
DOUBLE PRECISION A (LDA,*),AF (LDAF,*), B (LDB,*), X (LDX,*),
FERR (*),BERR (*),W ORK (*)
```


## F95 INTERFACE

```
SU BROUTINE PBRFS (UPLO, \(\mathbb{N}], K D, \mathbb{N R H S}], A,[L D A], A F,[L D A F], B\), [LD B],X, [LDX ],FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N F O}\) ])
CHARACTER (LEN=1) ::UPLO
\(\mathbb{N} T E G E R:: N, K D, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: W\) ORK2
```

REAL (8),D $\mathbb{M}$ ENSION (:) ::FERR,BERR,W ORK
REAL (8), D $\mathbb{M}$ ENSION (: : : : : A , AF, B, X
SU BROUTINE PBRFS_64 (UPLO, $\mathbb{N}], K D, \mathbb{N} R H S], A,[L D A], A F,[L D A F]$, B, [LDB], X, [LDX],FERR,BERR, [W ORK], [W ORK2], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER (8) ::N,KD,NRHS,LDA,LDAF,LDB,LDX, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I M}$ ENSION (:) ::W ORK2
REAL (8),D IM ENSION (:) ::FERR,BERR,W ORK
REAL (8), D $\mathbb{M}$ ENSION (: : : : : A , AF, B, X

## C INTERFACE

\# include < sunperfh>
void dpbrfs (charuplo, intn, intkd, intnrhs, double *a, int lda, double *af, int ldaf, double *b, int ldb, double *x, int ldx , double *ferr, double *berr, int*info);
void dpbrfs_ 64 (char uplo, long n, long kd, long nris, double
*a, long lda, double *af, long ldaf, double *b, long ldb, double *x, long ldx, double *ferr, double *berr, long *info);

## PURPOSE

dpbrfs im proves the com puted solution to a system of linear equations $w$ hen the coefficientm atrix is sym $m$ etric positive definite and banded, and provides errorbounds and backw ard error estim ates for the solution.

## ARGUMENTS

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.
N (input) The order of the matrix $\mathrm{A} . \mathrm{N}>=0$.
KD (input)
The num ber of superdiagonals of the $m$ atrix A if UPLO $=\mathrm{U}$ ', or the num berof subdiagonals ifUPLO $=\mathbb{L}^{\prime} . \mathrm{KD}>=0$ 。

NRHS (input)
The num ber of righthand sides, i.e., the num ber
of collm ns of the m atrices $B$ and X. NRH S $>=0$.

A (input) The upper or low er triangle of the sym m etric band $m$ atrix $A$, stored in the firstKD +1 row s of the anay. The $j$ th colum $n$ ofA is stored in the $j$ th colum $n$ of the anray A as follow $s$ : if UPLO $=\mathrm{U}$ ', $A(k d+1+i-j, j)=A(i, j)$ for $m a x(1, j k d)<=i<=j$ if $\mathrm{UPLO}=\mathrm{L}$ ', $A(1+i-j, \bar{j})=A(i, j)$ for $j<=i<=m$ in $(n, j+k d)$.

LDA (input)
The leading dim ension of the anay A. LDA >= K D +1.
AF (input)
The triangular factor $U$ or $L$ from the Cholesky factorization $A=U * * T * U$ orA $=\mathrm{L} * \mathrm{~L} * * T$ of the band $m$ atrix A as computed by SPBTRF, in the same storage form atas A (see A).

LDAF (input)
The leading dim ension of the array AF. LDAF >= K D +1 .
$B$ (input) The righthand side $m$ atrix $B$.

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.

X (input/output)
On entry, the solution $m$ atrix $X$, as com puted by SPB TRS. On exit, the im proved solution $m$ atrix X .

LD X (input)
The leading dim ension of the array X. LD X >= $\max (1, N)$.

## FERR (output)

The estim ated forw ard enorbound for each solution vectorX ( 7 ) the $j$ th colum $n$ of the solution $m$ atrix X). If XTRUE is the true solution corresponding to $X(i), F E R R(i)$ is an estim ated upperbound forthe $m$ agnitude of the largest ele$m$ entin ( $X(\mathcal{J})-X$ TRUE) divided by the $m$ agninude of the largestelem entin X ( 7 ). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slightoverestim ate of the true error.

## BERR (output)

The com ponentw ise relative backw ard error of each
solution vectorX (i) (ie., the sm allest relative
change in any elem entofA orB thatm akes X ( 7 ) an exactsolution).

W ORK (w orkspace)
dim ension ( $3 * N$ )
W ORK 2 (w orkspace) dim ension $\mathbb{N}$ )
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0$ : if $\mathbb{N} F O=-i$, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dpbstf-com pute a splitC holesky factorization of a real
sym $m$ etric posilive definite band $m$ atrix A

## SYNOPSIS

```
SUBROUT\mathbb{NE DPBSTF (UPLO,N,KD,AB,LDAB, INFO)}
CHARACTER * 1 UPLO
INTEGERN,KD,LDAB,INFO
DOUBLE PRECISION AB (LDAB,*)
```



```
CHARACTER * 1 UPLO
INTEGER*8N,KD,LDAB,INFO
DOUBLE PRECISION AB (LDAB,*)
```


## F95 INTERFACE

```
SUBROUT\mathbb{NE PBSTF (UPLO, N ],KD,AB,[LDAB], [NNO ])}
CHARACTER (LEN=1) ::UPLO
\mathbb{NTEGER ::N,KD,LDAB,INFO}
REAL (8),D IM ENSION (:,:) ::AB
SU BROUT\mathbb{NE PBSTF_64 (UPLO , N ],KD ,AB, [LDAB ], [N FO ])}
CHARACTER (LEN=1) ::UPLO
\mathbb{NTEGER (8)::N,KD,LDAB,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REAL (8),D IM ENSION (:,:) ::AB
```

void dpbstf(charuple, intn, intkd, double *ab, int ldab, int*info);
void dpbstf_64 (charuplo, long n, long kd, double *ab, long ldab, long *info);

## PURPOSE

dpbstf com putes a split C holesky factorization of a real sym $m$ etric positive definite band $m$ atrix $A$.

This routine is designed to be used in conjunction w ith SSBG ST .
The factorization has the form $A=S * * T * S$ where $S$ is a band $m$ atrix of the sam ebandw idth as A and the follow ing structure:

$$
\begin{aligned}
& S=(U \quad) \\
& \text { (M L ) }
\end{aligned}
$$

w here U is upper triangular of orderm $=(\mathrm{n}+\mathrm{kd}) / 2$, and L is low er triangular of ordern-m .

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': U ppertriangle ofA is stored;
$=\mathrm{L}$ ': Low er triangle of A is stored.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

KD (input)
The num ber of superdiagonals of the $m$ atrix $A$ if
$\mathrm{UPLO}=\mathrm{U}$ ', orthe num ber of subdiagonals ifU PLO
$=\mathbb{L} . \mathrm{KD}>=0$ 。

A B (input/output)
O $n$ entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the first kd+1 row s of the array. The $j$ th column of A is stored in the $j$ th colum $n$ of the array A B as follow s: if $\mathrm{UPLO}=\mathrm{U}^{\prime}, \mathrm{AB}(\mathrm{kd}+1+i-j, j)=A(i, j)$ for $\mathrm{max}(1, j$ $\mathrm{kd})<=\dot{i}<=\dot{j}$ ifUPLO $=\mathrm{L}$ ', AB $(1+i-j, j=A(i, j)$ for $j=i<=m$ in $(n, j+k d)$.

On exit, if $\mathbb{N F O}=0$, the factors from the split Cholesky factorization $A=S * * T * S$. See Further D etails.

LDAB (input)
The leading dim ension of the array AB. LD A B >= K D +1 .
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the $i$-th argum enthad an illegalvalue $>0:$ if $\mathbb{N} F O=$ i, the factorization could not be com pleted, because the updated elem enta (i,i) w as negative; the $m$ atrix $A$ is notposilive definite.

## FURTHER DETAILS

The band storage schem e is illustrated by the follow ing exam ple, w hen $N=7, K D=2$ :

| $\mathrm{S}=(\mathrm{s} 11 \mathrm{~s} 12 \mathrm{~s} 13$ |  |
| :---: | :---: |
| ( | s22 s23 s24 ) |
| ( | s33 s34 ) |
| ( | s44 ) |
| ( | s53 s54 s55 |
| ( | s64 s65 s66 ) |
| ( | s75 s76 s77) |

If $\mathrm{U} P \mathrm{O}=\mathrm{U}$ ', the amay A B holds:
on entry: on exit:

*     * a13 a24 a35 a46 a57 * * s13 s24 s53
s64 s75
* a12 a23 a34 a45 a56 a67 * s12 s23 s34 s54
s65 s76 a11 a22 a33 a44 a55 a66 a77 s11 s22 s33
s44 s55 s66 s77

IfU PLO = L', the array AB holds:
on entry: on exit:

```
a11 a22 a33 a44 a55 a66 a77 s11 s22 s33 s44 s55
s66 s77 a21 a32 a43 a54 a65 a76 * s12 s23 s34
s54 s65 s76 * a31 a42 a53 a64 a64 * * s13
s24 s53 s64 s75 * *
```

A may elem entsm arked * are notused by the routine.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dpbsv - com pute the solution to a real system of linear
equations $A * X=B$,

## SYNOPSIS

```
SUBROUT\mathbb{NEDPBSV (UPLO,N,ND IAG,NRHS,A,LDA,B,LDB,\mathbb{NFO )}}\mathbf{N}\mathrm{ (NA,}
CHARACTER * 1 UPLO
\mathbb{NTEGERN,ND IAG,NRHS,LDA,LDB,NNFO}
DOUBLE PRECISION A (LDA,*),B(LDB,*)
SUBROUT\mathbb{NE DPBSV_64(UPLO,N,ND IAG,NRHS,A ,LDA,B,LDB, INFO)}
CHARACTER * 1UPLO
\mathbb{NTEGER*8N,ND IAG,NRHS,LDA,LD B, INFO}
DOUBLE PRECISION A (LDA,*),B (LDB,*)
```


## F95 INTERFACE

SU BROUTINE PBSV (UPLO, $\mathbb{N}], N D \mathbb{I A} G, \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N} F O])$

CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R:: N, N D \mathbb{A} G, N R H S, L D A, L D B, \mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (: : : : ::A, B

SU BROU T INE PBSV_64 (UPLO, $\mathbb{N}], N D \mathbb{I} G, \mathbb{N} R H S], A,[L D A], B,[L D B]$, [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R(8):: N, N D \mathbb{I} G, N R H S, L D A, L D B, \mathbb{N} F O$
REAL (8), D IM ENSION (:,:) ::A , B

## C INTERFACE

\#include <sunperfh>
void dpbsv (charuplo, intn, intndiag, intnrhs, double *a, int lda, double *b, int ldb, int *info);
void dpbsv_64 (charuplo, long n, long ndiag, long nins, double *a, long lda, double *b, long ldb, long *info);

## PURPOSE

dpbsv com putes the solution to a realsystem of linear equations
$A * X=B, w h e r e A$ is an $N$ boy $N$ sym m etric positive defintie band $m$ atrix and X and B are N -by-N R H S m atrices. The Cholesky decom position is used to factorA as
$A=U * * T * U$, if $U P L O=U$ ', or
$\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$, if $\mathrm{UPLO}=\mathrm{L}$ ',
$w$ here $U$ is an upper triangularband $m$ atrix, and $L$ is a low er triangular band $m$ atrix, $w$ ith the sam e num ber of superdiagonals or subdiagonals as A. The factored form of A is then used to solve the system ofequations $A * X=B$.

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ : U Uper triangle ofA is stored;
$=\mathrm{L}$ ': Low ertriangle ofA is stored.

N (input) The num ber of linearequations, i.e., the order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

ND IA G (input)
The num ber of superdiagonals of the $m$ atrix $A$ if $\mathrm{UPLO}=\mathrm{U}$ ', or the num ber of subdiagonals if $\mathrm{U} P L O$
$=\mathbb{L}$ '. NDIAG > $=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS $>=0$.

A (input/output)
O n entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstN D IA $G+1$ row s of the array. The $j$ th colum n of A is stored in the jth colum $n$ of the anray $A$ as follow $s$ : if
 $\max (1, j \mathrm{jND} \mathrm{IAG})<=i<=j$ if $\mathrm{UPLO}=\mathrm{L}^{\prime}, \mathrm{A}(1+i-j)$
$=A(i, j)$ for $j<i<=m$ in $(\mathbb{N}, j+N D$ IA G ). See below for furtherdetails.

On exit, if $\mathbb{N} F O=0$, the triangular factor $U$ orL
from the Cholesky factorization $A=U * * T * U$ or $A=$
$\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ of the band m atrix A , in the sam e storage form atas A.

LD A (input)
The leading dim ension of the aray A. LD A >= N D IA G +1.
B (input/output)
On entry, the N -by-NRHS righthand side m atrix B. On exit, if $\mathbb{N F O}=0$, the N boy -N RH S solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the array $\mathrm{B} . \operatorname{LD} \mathrm{B}>=$ $\max (1, N)$.

IN FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvałue $>0:$ if $\mathbb{N F O}=i$, the leading $m$ inoroforder iof $A$ is notpositive definite, so the factorization could notbe com pleted, and the solution has not been com puted.

## FURTHER DETAILS

The band storage scheme is illustrated by the follow ing exam ple, w hen $N=6, N D I A G=2$, and $U P L O=U ':$

On entry: On exit:

*     * a13 a24 a35 a46 * * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55 u66

Sim ilarly, ifU PLO = 'L 'the form atofA is as follow s:

On entry: On exit:

a31 a42 a53 a64 * * 131142153164 * *

A may elem entsm arked * are notused by the routine.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dpbsvx - use the Cholesky factorization $A=U * * T * U$ or $A=$ $\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ to com pute the solution to a realsystem of linear equations $A * X=B$,

## SYNOPSIS

```
SUBROUT\mathbb{NEDPBSVX FACT,UPLO,N,NDIAG,NRHS,A,LDA,AF,LDAF,}
    EQUED,S,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK 2,
    INFO)
```

CHARACTER * 1 FACT, UPLO, EQUED
$\mathbb{N}$ TEGERN,ND $\mathbb{I} G, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O$
$\mathbb{N}$ TEGERWORK2(*)
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA,*), AF (LDAF,*), S (*), B (LDB,*),
$\mathrm{X}(\mathrm{LD} \mathrm{X}, \star), \operatorname{FERR}\left({ }^{*}\right), \operatorname{BERR}(*), \mathrm{W} O R K(*)$
SU BROUTINEDPBSVX_64 FACT,UPLO,N,NDIAG,NRHS,A,LDA,AF,LDAF,
EQUED,S,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK2,
$\mathbb{N} F O$ )
CHARACTER * 1 FACT, UPLO, EQUED
$\mathbb{N} T E G E R * 8 N, N D \mathbb{I} G, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O$
$\mathbb{N}$ TEGER * 8 W ORK 2 ( ${ }^{\star}$ )
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA, *), AF (LDAF,*), S (*), B (LDB, $\left.{ }^{\star}\right)$,
$\mathrm{X}(\mathrm{LDX}, \star), \operatorname{FERR}\left({ }^{\star}\right), \operatorname{BERR}\left({ }^{\star}\right), \mathrm{W} O R K\left({ }^{*}\right)$

## F95 INTERFACE

SU BROUTINE PBSVX (FACT,UPLO, $\mathbb{N}], N D \mathbb{I} G, \mathbb{N R H S}], A,[L D A], A F,[L D A F]$, EQUED, $S, B,[L D B], X,[L D X], R C O N D, F E R R, B E R R,[W$ ORK ],
[W ORK2], [ $\mathbb{N F O}$ ])

CHARACTER ( $L E N=1$ ) : :FACT, UPLO, EQUED
$\mathbb{N} T E G E R:: N, N D \mathbb{A} G, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) ::W ORK2
REAL (8) :: RCOND
REAL (8), D $\mathbb{M}$ ENSION (:) :: S, FERR, BERR, W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) ::A,AF,B,X

SU BROUTINE PBSVX_64 (FACT, UPLO, $\mathbb{N}], N D I A G, ~ \mathbb{N} R H S], A,[L D A], A F$, [LDAF], EQUED, S, B, [LDB],X, [LDX],RCOND,FERR,BERR, [W ORK], [WORK2], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) ::FACT,UPLO, EQUED
$\mathbb{N}$ TEGER (8) ::N,NDIAG,NRHS,LDA,LDAF,LDB,LDX, $\mathbb{N} F O$
$\mathbb{N}$ TEGER (8), D $\mathbb{M}$ ENSION (:) ::W ORK2
REAL (8) :: RCOND
REAL (8), D $\mathbb{M} E N S I O N(:):: S, F E R R, B E R R, W$ ORK
REAL (8), D $\mathbb{I M}$ ENSION (:,:) ::A,AF,B,X

## C INTERFACE

\#include <sunperfh>
void dpbsvx (char fact, charuplo, int $n$, int ndiag, int nrhs, double *a, intlda, double *af, intldaf, char equed, double *s, double *b, int ldb, double ${ }^{*}{ }_{x}$, int $l d x$, double *roond, double *ferr, double *berr, int*info);
void dpbsvx_64 (char fact, charuplo, long n, long ndiag, long nrhs, double *a, long lda, double *af, long ldaf, charequed, double *s, double *b, long ldb, double *x, long ldx, double *rcond, double *ferr, double *berr, long *info);

## PURPOSE

dpbsvx uses the Cholesky factorization $A=U * * T * U$ or $A=$ L*L**T to com pute the solution to a realsystem of linear equations
$A * X=B, w$ here $A$ is an $N$ boy $-N$ sym $m$ etric positive definite band $m$ atrix and $X$ and $B$ are $N$ boy-N R H S m atrices.

Emorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:

1. IfFACT $=$ E', real scaling factors are computed to
equilibrate
the system :
$\operatorname{diag}(S) * A * \operatorname{diag}(S) * \operatorname{inv}(\operatorname{diag}(S)) * X=\operatorname{diag}(S) * B$
W hether or not the system w illbe equilibrated depends on the scaling of the m atrix A, but if equilibration is used, A is
overw ritten by diag $(S) \star A * d i a g(S)$ and $B$ by diag $(S) * B$.
2. IfFACT = N 'or E ', the Cholesky decom position is used to
factor them atrix A (afterequilibration ifFACT = E) as
$A=U * * T * U$, ifUPLO $=U$ ', or
$\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$, if $\mathrm{UPLO}=\mathrm{L}$ ',
$w$ here $U$ is an upper triangularband $m$ atrix, and $L$ is a
low er
triangularband $m$ atrix.
3. If the leading i-by-iprincipal m inor is not positive definite,
then the routine retums w ith $\mathbb{N} F O=$ i. O therw ise, the factored
form ofA is used to estim ate the condition num ber of the $m$ atrix
A. If the reciprocal of the condition num ber is less than $m$ achine precision, $\mathbb{N} F O=N+1$ is retumed as a w aming, but the routine
stillgoes on to solve for X and com pute errorbounds as described below .
4.The system of equations is solved for $X$ using the factored form of A.
4. Herative refinem ent is applied to im prove the com puted solution
$m$ atrix and calculate error bounds and backw ard error estim ates
for it.
5. Ifequilibration w as used, the $m$ atrix $X$ is prem ultiplied by diag (S) so that it solves the original system before equilibration.

## ARGUMENTS

## FACT (input)

Specifies w hether ornot the factored form of the $m$ atrix A is supplied on entry, and ifnot, w hether the m atrix A should be equilibrated before it is factored. = F : On entry, AF contains the factored form ofA. IfEQUED = $Y$ ', them atrix A has been equilibrated $w$ ith scaling factors given by $S$. A and AF w illnotbe m odified. $=\mathrm{N}$ ': Them atrix A w illbe copied to A F and factored.
= E : : The matrix A will be equilibrated if necessary, then copied to A F and factored.

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.
N (input) The num ber of linearequations, i.e., the order of the matrix A. $\mathrm{N}>=0$.
ND IA G (input)
The num ber of superdiagonals of the matrix A if UPLO = U',orthe num berof subdiagonals ifU PLO
= L'. ND IAG >= 0 .

NRHS (input)
The num ber of right-hand sides, i.e., the num ber of collm ns of the matrices B and X. NRHS >= 0 .

A (input/output)
On entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstND IA G +1 row s of the array, except ifFACT = F'and EQUED $=Y$ ', then A m ust contain the equilibrated $m$ atrix diag $(S){ }^{*} A$ *diag $(S)$. The jth colum n of A is stored in the $j$ th colum $n$ of the array $A$ as follow $s$ : if UPLO = U', A NDIAG+1+i-ji) =A (i, $\boldsymbol{j}$ ) for $\max (1, j$ ND IAG $)<=i<=j$ if UPLO $=\mathrm{L}$ ', A ( $1+i-j, j)$ $=A(i, j)$ for $j=i<=m$ in $(\mathbb{N}, \dot{j}+N D \mathbb{I A} G)$. See below for furtherdetails.

Onexit, ifFACT = E' and EQUED = Y', A is overw rilten by diag (S)*A *diag (S).

LDA (input)
The leading dim ension of the array A. LDA >= ND IA G +1.

AF (input/output)
If FACT = F ', then $A F$ is an inputargum entand on entry contains the triangular factorU or $L$ from the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=$
$L * L * * T$ of the band $m$ atrix $A$, in the sam e storage form atas A (see A). IfEQUED $=Y$ ', then $A F$ is the factored form of the equilibrated $m$ atrix A.

IfFACT $=N$ ', then AF is an output argum ent and on exit retums the triangular factorU orL from the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=$ L*L**T。

IfFACT = E', then AF is an output argum ent and on exitretums the triangular factor $U$ or $L$ from the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=$ $\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ of the equilibrated $m$ atrix $A$ (see the description of for the form of the equilibrated m atrix).
LD AF (input)
The leading dim ension of the array AF. LDAF >= $N D I A G+1$.

EQUED (input)
Specifies the form of equilibration thatw as done.
$=\mathrm{N}$ : N o equilibration (alw ays true iffACT = N 7 。
$=Y^{\prime}:$ Equilibration w as done, i.e., A has been replaced by diag $(\mathrm{S})$ * A * diag $(\mathrm{S})$. EQUED is an inputargum entifFACT = F'; otherw ise, it is an outputargum ent.

S (input/output)
The scale factors forA; notaccessed if EQUED = $N^{\prime} . S$ is an inputargum entifFACT = $F^{\prime}$; otherw ise, S is an outputargum ent. IfFACT $=\mathrm{F}^{\prime}$ and EQUED $=Y$ ', each elem entof m ustbe posilive.

B (input/output)
O n entry, the N Hoy-NRH S righthand side m atrix B.
On exit, if EQUED = $N^{\prime}$, $B$ is notm odified; if EQUED $=Y$ ', $B$ is overw ritten by diag $(S) * B$.

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.

X (output)
If $\mathbb{N} F O=0$ or $\mathbb{I N F O}=\mathrm{N}+1$, the N -by-NRHS solution
$m$ atrix $X$ to the original system of equations.
$N$ ote that if EQUED $=Y$ ', A and $B$ arem odified on exit, and the solution to the equilibrated system is inv (diag (S ) ) *X .

## LD X (input)

The leading dim ension of the anay X . LD X >= $\max (1, N)$.

## RCOND (output)

The estim ate of the reciprocal condition num ber of the $m$ atrix $A$ after equilibration (if done). If RCOND is less than them achine precision (in particular, ifRCOND $=0$ ), the $m$ atrix is singular to w orking precision. This condition is indicated by a retum code of $\mathbb{N} F O>0$.

FERR (output)
The estim ated forw ard emorbound for each solution vector $X(\mathcal{)}$ ) the $j$ th colum $n$ of the solution $m$ atrix X). If XTRUE is the true solution comesponding to $X(\mathcal{O})$, FERR ( $)$ is an estim ated upperbound for the $m$ agnitude of the largest ele$m$ ent in ( $\mathrm{X}(\mathcal{i})-\mathrm{X}$ TRU E ) divided by the m agnitude of the largestelem ent in $X(\mathcal{J})$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

## BERR (output)

The com ponentw ise relative backw ard error of each
solution vectorX (j) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( $\mathcal{j}$ ) an exactsolution).

W ORK (w orkspace)
dim ension ( $3 * \mathrm{~N}$ )
W ORK 2 (w orkspace)
dim ension (N)
$\mathbb{N} F O$ (output)
= 0 : successfulexit
< 0: if $\mathbb{N N}$ FO = -i, the i-th argum enthad an illegalvalue
$>0$ : if $\mathbb{N} F O=i$, and $i$ is
<= N : the leading $m$ inoroforderiof $A$ is not positive definite, so the factorization could not be com pleted, and the solution has not been com puted. RCOND $=0$ is retumed. $=\mathrm{N}+1$ : U is nonsingular, butRCOND is less than machine precision, $m$ eaning that the $m$ atrix is singular to $w$ orking precision. N evertheless, the solution and error bounds are com puted because there are a num ber of situations w here the com puted solution can bem ore accurate than the value ofRCOND w ould

## FURTHER DETAILS

The band storage scheme is illustrated by the follow ing exam ple, when $N=6, N D \mathbb{I A G}=2$, and UPLO $=U$ ':

Tw o-dim ensional storage of the sym $m$ etric $m$ atrix A : all al2 a13 a22 a23 a24
a33 a34 a35
a44 a45 a46
a55 a56
(aijong (ä̈)) a66
$B$ and storage of the upper triangle ofA :

*     * a13 a24 a35 a46
* a12 a23 a34 a45 a56
a11 a22 a33 a44 a55 a66

Sim ilarly, if U PLO = L'the form atofA is as follow s:
a11 a22 a33 a44 a55 a66
a21 a32 a43 a54 a65 *
a31 a42 a53 a64 * *
A ray elem entsm arked * are notused by the routine.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dpbte2 - oom pute the C holesky factorization of a real sym $m$ etric positive definite band $m$ atrix A

## SYNOPSIS

```
SU BROUT\mathbb{NE DPBTF2(UPLO,N,KD,AB,LDAB,\mathbb{NFO)}}\mathbf{N}\mathrm{ (N,}
CHARACTER * 1 UPLO
INTEGERN,KD,LDAB,INFO
DOUBLE PRECISION AB (LDAB,*)
SU BROUT\mathbb{NE DPBTF2_64(UPLO,N,KD,AB,LDAB,INFO)}
CHARACTER * 1 UPLO
INTEGER*8N,KD,LDAB,INFO
DOUBLE PRECISION AB (LDAB,*)
F95 INTERFACE
SU BROUT\mathbb{NE PBTF2 (UPLO, N ],KD,AB, [LDAB], [NFO])}
CHARACTER (LEN=1) ::UPLO
\mathbb{NTEGER ::N,KD,LDAB,INFO}
REAL (8),D IM ENSION (:,:) ::AB
SU BROUT\mathbb{NE PBTF2_64 (UPLO, N ],KD ,AB, [LDAB ], [NNO ])}
CHARACTER (LEN=1) ::UPLO
\mathbb{NTEGER (8)::N,KD,LDAB,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REAL (8),D IM ENSION (:,:) ::AB
```

void dpbtf2 (char uplo, intn, int kd, double *ab, int ldab, int*info);
void dpbtf2_64 (charuple, long n, long kd, double *ab, long ldab, long *info);

## PURPOSE

dpbtf2 com putes the C holesky factorization of a real sym $m$ etric positive definite band $m$ atrix A.

The factorization has the form

```
A = U'* U , ifUPLO = U',or
A = L * L', ifUPLO = L',
```

where $U$ is an uppertriangularm atrix, $U$ 'is the transpose ofU , and $L$ is low ertriangular.

This is the unblocked version of the algorithm, calling Level2 BLAS.

## ARGUMENTS

## UPLO (input)

Specifies w hether the upper or low er triangular
part of the sym $m$ etric $m$ atrix $A$ is stored:
= U ': U pper triangular
= L': Low er triangular

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

KD (input)
The num ber of super-diagonals of the m atrix A if $\mathrm{UPLO}=\mathrm{U}$ ', or the num ber of sub-diagonals if UPLO
$=\mathbb{L} . \mathrm{KD}>=0$ 。

A B (input/output)
O n entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstK $D+1$ row s of the array. The $j$ th colum n ofA is stored in the $j$ th colum n of the array A B as follow s: if $\mathrm{UPLO}=\mathrm{U}$ ', AB $(k d+1+i-j, j)=A(i, j)$ for $\max (1, j$ $\mathrm{kd})<=i<=j$ ifUPLO $=\mathrm{L}$ ', AB $(1+i-j, j=A(i, j)$ for $\dot{j}=i<=m$ in $(n, j+k d)$.

On exit, if $\mathbb{N} F O=0$, the triangular factor $U$ orL
from the Cholesky factorization $\mathrm{A}=\mathrm{U}$ * U or $\mathrm{A}=$ L * L ' of the band m atrix A , in the sam e storage form atas A.

LDAB (input)
The leading dim ension of the array AB. LD AB >= KD+1.
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N} F O=-k$, the $k$-th argum enthad an illegalvalue $>0:$ if $\mathbb{N} F O=k$, the leading $m$ inoroforderk is notpositive definite, and the factorization could notbe com pleted.

## FURTHER DETAILS

The band storage scheme is illustrated by the follow ing exam ple, when $\mathrm{N}=6, \mathrm{KD}=2$, and $\mathrm{U} P L O=\mathrm{U}$ ':

On entry: On exit:

*     * a13 a24 a35 a46 * * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55 u66

Sim ilarly, if UPLO = L'the form atofA is as follow s:
On entry: On exit:

166
a21 a32 a43 a54 a65 * $121 \quad 132143154165$
*
a31 a42 a53 a64 * * 131142153164 *

A ray elem entsm arked * are not used by the routine.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dpbtrf-com pute the C holesky factorization of a real sym $m$ etric positive definite band $m$ atrix A

## SYNOPSIS

```
SUBROUTINEDPBTRF(UPLO,N,KD,A,LDA, INFO)
CHARACTER * 1 UPLO
\mathbb{NTEGERN,KD,LDA,NFO}
DOUBLE PRECISION A (LDA,*)
SU BROUTINE DPBTRF_64 (UPLO,N ,KD,A ,LD A , IN FO )
CHARACTER * 1 UPLO
INTEGER*8N,KD,LDA, INFO
DOUBLE PRECISION A (LDA,*)
F95 INTERFACE
SUBROUT\mathbb{NE PBTRF (UPLO, N ],KD ,A ,[LDA ], [NNFO])}
CHARACTER (LEN=1) ::UPLO
INTEGER ::N,KD,LDA,}\mathbb{N}F
REAL (8),D IM ENSION (:,:) ::A
SU BROUT\mathbb{NE PBTRF_64 (UPLO, N ],KD ,A, [LDA ], [\mathbb{NFO ])}}\mathbf{(})=
CHARACTER (LEN=1)::UPLO
\mathbb{NTEGER (8) ::N,KD,LDA, \mathbb{NFO}}\mathbf{N}=\mp@code{N}
REAL (8),D IM ENSION (:,:) ::A
void dpbtrf(char uple, intn, intkd, double *a, int lda, int*info);
void dpbtrf_64 (charuplo, long n, long kd, double *a, long lda, long *info);

\section*{PURPOSE}
dpbtrf com putes the C holesky factorization of a real sym \(m\) etric positive definite band \(m\) atrix A.

The factorization has the form
\[
\begin{aligned}
& A=U * * T * U, \text { if } \mathrm{UPLO}=\mathrm{U}^{\prime} \text {, or } \\
& \mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}, \text { if } \mathrm{UPLO}=\mathrm{L}^{\prime},
\end{aligned}
\]
\(w\) here \(U\) is an uppertriangularm atrix and \(L\) is low er triangular.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) : U ppertriangle of A is stored;
\(=\mathrm{L}\) ': Low er triangle of \(A\) is stored.

N (input) The order of them atrix \(A . N>=0\).

KD (input)
The num berof superdiagonals of the \(m\) atrix \(A\) if
\(\mathrm{UPLO}=\mathrm{U}\) ', or the num ber of subdiagonals if \(\mathrm{U} P L O\)
\(=L^{\prime} . K D>=0\) 。

A (input/output)
O n entry, the upper or low er triangle of the sym \(m\) etric band \(m\) atrix \(A\), stored in the firstK \(D+1\) row s of the array. The \(j\) th colum n of A is stored in the \(j\) th colum \(n\) of the amay \(A\) as follow \(s\) : if \(\mathrm{UPLO}=U ', A(k d+1+i-j, j)=A(i, j)\) for \(m a x(1, j\) \(\mathrm{kd})<=i<=j\) if \(\mathrm{UPLO}={ }^{\prime}\) ', \(A(1+i-j, j)=A(i, j)\) for \(j<=i<=m\) in \((n, j+k d)\).

On exit, if \(\mathbb{N F O}=0\), the triangular factor \(U\) orL from the Cholesky factorization \(A=U * * T * U\) or \(A=\) L*L**T of the band \(m\) atrix \(A\), in the same storage form atas A.

The leading dim ension of the array A. LDA >= K D +1 .
\(\mathbb{N F O}\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the leading \(m\) inoroforder is notpositive definite, and the factorization could notbe com pleted.

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, when \(N=6, K D=2\), and \(U P L O=U:\)
On entry: On exit:
```

    * a13 a24 a35 a46 * * u13 u24 u35
    u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55
u66

```
Sim ilarly, if UPLO = L 'the form atofA is as follow s:
On entry: Onexit:
    a11 a22 a33 a44 a55 a66 \(111 \quad 122 \quad 133144 \quad 155\)
166
    a21 a32 a43 a54 a65 * \(121 \quad 132143154165\)
*
    a31 a42 a53 a64 * * 131142153164 *
*

A rray elem entsm arked * are notused by the routine.
C ontributed by
PeterM ayes and G inseppe Radicati, \(\mathbb{B M}\) EC SEC , Rom e, M arch 23,1989

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dpbtre - solve a system of linearequations A *X = B w ith a sym \(m\) etric positive definite band \(m\) atrix \(A\) using the C holesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\) com puted by SPBTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DPBTRS(UPLO,N,KD,NRHS,A,LDA,B,LDB,NNFO)}
CHARACTER * 1 UPLO
INTEGERN,KD,NRHS,LDA,LDB,}\mathbb{NFO
DOUBLE PRECISION A (LDA,*),B (LDB,*)
SUBROUT\mathbb{NEDPBTRS_64(UPLO,N,KD,NRHS,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N,KD,NRHS,LDA,LDB,NNFO}
DOUBLE PRECISION A (LDA,*),B (LDB,*)

```
F95 INTERFACE
    SU BROUTINE PBTRS (UPLO, \(\mathbb{N}], K D, \mathbb{N R H S}], A,[L D A], B,[L D B],[\mathbb{N F O}])\)
    CHARACTER (LEN=1) ::UPLO
    \(\mathbb{N}\) TEGER ::N,KD,NRHS,LDA,LDB, \(\mathbb{N}\) FO
    REAL (8), D \(\mathbb{M}\) ENSIO N (:,:) ::A,B
    SU BROUTINE PBTRS_64 (UPLO, \(\mathbb{N}], K D, \mathbb{N} R H S], A,[L D A], B,[L D B]\),
        [ \(\mathbb{N} F O\) ])
    CHARACTER (LEN=1) ::UPLO
    \(\mathbb{N}\) TEGER (8) ::N,KD,NRHS,LDA,LDB, \(\mathbb{N} F O\)
    REAL (8), D IM ENSION (:,:) ::A , B

\section*{C INTERFACE}
\#include <sunperfh>
void dpbtrs (charuplo, intn, intkd, intnrhs, double *a, int lda, double *b, int ldb, int *info);
void dpbtrs_64 (charuplo, long n, long kd, long nihs, double
*a, long lda, double *b, long ldb, long *info);

\section*{PURPOSE}
dpbtrs solves a system of linearequations A *X \(=\mathrm{B}\) w ith a sym \(m\) etric positive definite band \(m\) atrix \(A\) using the C holesky factorization \(A=U * * T * U\) orA \(=L * L * * T\) com puted by SPBTRF .

\section*{ARGUMENTS}

UPLO (input)
\(=U\) ': U pper triangular factor stored in A ;
\(=\mathrm{L}\) ': Low ertriangular factorstored in A.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

KD (input)
The num ber of superdiagonals of the \(m\) atrix \(A\) if
\(\mathrm{UPLO}=\mathrm{U}\) ', orthe num ber of subdiagonals ifU PLO
\(=\mathbb{L}^{\prime} . \mathrm{KD}>=0\) 。

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of the m atrix B. NRH S \(>=0\).

A (input) The triangular factor \(U\) or \(L\) from the Cholesky
factorization \(A=U * * T * U\) or \(A=L * L * * T\) of the band \(m\) atrix A, stored in the first KD +1 row \(s\) of the array. The \(j\) th colum \(n\) of \(U\) orL is stored in the jth colum \(n\) of the array A as follow s: if UPLO \(=U \prime, A(k d+1+i-j)=U(i, j)\) for \(m a x(1, j\)
\(\mathrm{kd})<=i<=\dot{j}\) ifUPLO \(=L \prime\) ' A \((1+i-j, j)=L(i, j)\)
for \(j<=i<=m\) in \((n, j+k d)\).

LD A (input)
The leading dim ension of the array A. LD A >= K D +1.

B (input/output)
O \(n\) entry, the right hand side m atrix B. On exit,
the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray B. LD B \(>=\) \(\max (1, N)\).

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dpocon -estim ate the reciprocal of the condition num ber (in the 1-norm ) of a realsym \(m\) etric positive definite \(m\) atrix using the C holesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\) com puted by SPO TRF

\section*{SYNOPSIS}

CHARACTER * 1 UPLO
\(\mathbb{N}\) TEGER \(N, L D A, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER W ORK 2 (*)
DOUBLE PRECISION ANORM,RCOND
D OUBLE PRECISION A (LDA,*),W ORK (*)
SU BROUTINEDPOCON_64 (UPLO,N,A,LDA,ANORM,RCOND,W ORK,W ORK2,
    \(\mathbb{N} F O\) )
CHARACTER * 1 UPLO
\(\mathbb{N}\) TEGER*8 N ,LDA, \(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER*8 W ORK 2 (*)
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION A (LDA,*),W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE POCON (UPLO, \(\mathbb{N}], A,[L D A], A N O R M, R C O N D,[W\) ORK ], [W ORK2], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER ::N,LDA, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) ::W ORK2
REAL (8) ::ANORM,RCOND

SUBROUTINE POCON_64 (UPLO, \(\mathbb{N}], A,[L D A], A N O R M, R C O N D,[W O R K],[W O R K 2]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER (8) :: N , LDA, \(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER (8), D \(\mathbb{M}\) ENSION (:) ::WORK2
REAL (8) ::ANORM,RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include < sunperfh>
void dpocon (charuplo, intn, double *a, int lda, double anorm , double *rcond, int *info);
void dpocon_64 (charuplo, long n, double *a, long lda, double anorm , double *roond, long *info);

\section*{PURPOSE}
dpocon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a realsym m etric positive definite \(m\) atrix using the C holesky factorization \(A=U * * T * U\) or \(A=L * L * * T\) com puted by SPO TRF .

A n estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND \(=1\) / (ANORM * norm (inv (A))).

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': Upper triangle of A is stored;
\(=\mathbb{L}\) ': Low ertriangle ofA is stored.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input) The triangular factor \(U\) or \(L\) from the Cholesky factorization \(A=U * * T * U\) orA \(=L * L * * T\), as com puted by SPO TRF.

LDA (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

ANORM (input)
The 1-norm (or infinity-norm ) of the symmetric \(m\) atrix A.

\section*{RCOND (output)}

The reciprocal of the condition number of the \(m\) atrix \(A\), com puted as RCOND \(=1\) ( \(A N O R M * A \mathbb{N} V N M\) ), where \(A \mathbb{N} V N M\) is an estim ate of the 1 -nom of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension ( \(3 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{I N F O}\) (output)
= 0 : successfiulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dpoequ - com pute row and colum \(n\) scalings intended to equilibrate a sym \(m\) etric positive definite \(m\) atrix \(A\) and reduce its condition num ber (w ith respect to the tw o-norm )

\section*{SYNOPSIS}

```

INTEGERN,LDA, INFO
DOUBLE PRECISION SCOND,AM AX
DOUBLE PRECISION A (LDA,*),SCALE (*)
SUBROUT\mathbb{NEDPOEQU_64 N,A,LDA,SCALE,SCOND,AM AX,\mathbb{NFO )}}\mathbf{N}={
INTEGER*8N,LDA, INFO
DOUBLE PRECISION SCOND,AMAX
D OU BLE PRECISION A (LDA,*),SCALE (*)

```
F95 INTERFACE
    SU BROUTINE POEQU ( \(\mathbb{N}\) ],A, [LDA ],SCALE, SCOND ,AMAX, [ \(\mathbb{N F O}\) ])
\(\mathbb{N} T E G E R:: N, L D A, \mathbb{N} F O\)
REAL (8) :: SCOND,AMAX
REAL (8),D \(\mathbb{I M}\) ENSION (:) ::SCALE
REAL (8),D IM ENSION (: : : : : A
SU BROUTINE POEQU_64 ( \(\mathbb{N}\) ],A, [LDA ],SCALE,SCOND,AMAX, \(\mathbb{N} F O]\) )
\(\mathbb{N}\) TEGER (8) ::N,LDA, \(\mathbb{N}\) FO
REAL (8) :: SCOND,AMAX
REAL (8),D \(\mathbb{I M}\) ENSION (:) ::SCALE
REAL (8),D IM ENSION (: : : : : A

\section*{C INTERFACE}
\#include <sunperfh>
void dpoequ (intn, double *a, int lda, double *scale, double
*scond, double *am ax, int *info);
void dpoequ_64 (long n, double *a, long lda, double *scale, double *scond, double *am ax, long *info);

\section*{PURPOSE}
dpoequ com putes row and colum \(n\) scalings intended to equilibrate a sym \(m\) etric positive definite \(m\) atrix A and reduce its condition num ber (w ith respect to the tw o-nom ). S contains the scale factors, \(S(i)=1 /\) squt \((A)(i, i))\), chosen so that the scaled \(m\) atrix \(B\) w ith elem ents \(B(i, j)=S(i) * A(i, j) * S(i)\) has ones on the diagonal. This choice of \(S\) puts the condition num berofB w ithin a factor \(N\) of the sm allestpossible condition num ber overallpossible diagonal scalings.

\section*{ARGUMENTS}

N (input) The order of the matrix A. \(\mathrm{N}>=0\).
A (input) The \(\mathrm{N}-\) by -N sym \(m\) etric positive definite \(m\) atrix whose scaling factors are to be com puted. O nly the diagonalelem ents ofA are referenced.

LD A (input)
The leading dim ension of the anay A. LDA >= \(\max (1, N)\).

SCA LE (output)
If \(\mathbb{N} F O=0\), SCA LE contains the scale factors for A.

SCOND (output)
If \(\mathbb{N} F O=0, S C A L E\) contains the ratio of the \(s m\) allest SCALE (i) to the largestSCA LE (i). IfSCOND \(>=0.1\) and AM AX is neither too large nor too sm all, it is notw orth scaling by SCA LE .

AM AX (output)
A bsolute value of largestm atrix elem ent. If A M A X is very close to overflow orvery close to underflow , the \(m\) atrix should be scaled.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=\) i, the ith diagonal elem ent is nonpositive.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dporfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is sym \(m\) etric positive definite,

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DPORFS (UPLO,N,NRHS,A,LDA,AF,LDAF,B,LDB,X,LDX,}
FERR,BERR,W ORK,W ORK2,INFO)
CHARACTER * 1 UPLO
\mathbb{NTEGER N,NRHS,LDA,LDAF,LDB,LDX,}\mathbb{N}FO
INTEGERW ORK2 (*)
D OUBLE PRECISION A (LDA,*),AF (LDAF,*), B (LDB,*), X (LDX,*),
FERR (*),BERR (*),W ORK (*)
SUBROUT\mathbb{NE DPORFS_64 UPLO,N,NRHS,A,LDA,AF,LDAF,B,LDB,X,LDX,}
FERR,BERR,W ORK,W ORK2, INFO)
CHARACTER * 1 UPLO
INTEGER*8 N,NRHS,LDA,LDAF,LDB,LDX,NNFO
\mathbb{NTEGER*8 W ORK2 (*)}
DOUBLE PRECISION A (LDA,*),AF (LDAF,*), B (LDB,*), X (LDX,*),
FERR (*),BERR (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE PORFS (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F], B,[L D B]\), X, [LDX],FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) ::W ORK2
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK

SU BROUTINE PORFS_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F], B,[L D B]\), \(\mathrm{X},[\mathrm{LD} \mathrm{X}], \mathrm{FERR}, \mathrm{BERR},[\mathrm{W} O R K],[W O R K 2],[\mathbb{N} F O])\)

CHARACTER ( \(L E N=1\) ) : : UPLO
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):\) W ORK2
REAL (8), D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A, AF, B, X

\section*{C INTERFACE}
\#include <sunperfh>
void dporfs (charuplo, intn, intnms, double *a, int lda, double *af, intldaf, double *b, int ldb, double
\({ }^{*} x\), int \(l d x\), double *ferr, double *berr, int *info);
void dporfs_64 (charuplo, long n, long nrhs, double *a, long lda, double *af, long ldaf, double *b, long ldb, double *x, long ldx, double * ferr, double *berr, long *info);

\section*{PURPOSE}
dporfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is sym \(m\) etric positive definite, and provides emorbounds and backw ard emoresti\(m\) ates for the solution.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': Upper triangle ofA is stored;
\(=\mathbb{L}\) ': Low ertriangle ofA is stored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of right hand sides, i.e., the num ber
of colum ns of the m atrices B and X. NRHS \(>=0\).

A (input) The sym \(m\) etric \(m\) atrix A. If \(\mathrm{UPLO}=\mathrm{U}\) ', the leading N -by -N uppertriangularpart of A contains the uppertriangularpant of the \(m\) atrix \(A\), and the strictly low ertriangularpart of \(A\) is notreferenced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading N -by-N low er
triangularpart ofA contains the low er triangular partof the \(m\) atrix A, and the strictly upper triangular part ofA is not referenced.

LD A (input)
The leading dim ension of the anay A. LDA >= \(\max (1, N)\).

AF (input)
The triangular factor \(U\) or \(L\) from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(\mathrm{A}=\mathrm{L}{ }^{*} \mathrm{~L}{ }^{* *} \mathrm{~T}\), as com puted by SPO TRF.
LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, \mathbb{N})\).
\(B\) (input) The righthand side \(m\) atrix \(B\).
LD B (input)
The leading dim ension of the aray \(B\). LD B \(>=\) \(\max (1, \mathbb{N})\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by SPO TRS. On exit, the im proved solution \(m\) atrix \(X\).

\section*{LD X (input)}

The leading dim ension of the array X . LD X >= \(\max (1, \mathbb{N})\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X(\mathcal{)}\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix \(X\) ). If XTRUE is the true solution comesponding to \(X(\mathcal{O}), \operatorname{FERR}(\mathcal{O})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(\mathrm{X}(\mathcal{1})-\mathrm{XTRUE}\) ) divided by the magnitude of the largestelem ent in \(X(j)\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector \(X(\mathcal{)}\) (i.e., the sm allest relative change in any elem entofA orB thatm akes X (i) an exactsolution).

W ORK (w orkspace)
dim ension ( \(3 * \mathrm{~N}\) )

W ORK2 (w orkspace)
dim ension \(\mathbb{N}\) )

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the \(i\)-th argum ent had an illegal value

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dposv - com pute the solution to a real system of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDPOSV (UPLO,N,NRHS,A,LDA,B,LDB, INFO)}
CHARACTER * 1 UPLO
INTEGERN,NRHS,LDA,LDB,\mathbb{NFO}
D OU BLE PRECISION A (LDA,*),B (LDB,*)
SUBROUT\mathbb{NEDPOSV_64(UPLO,N,NRHS,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N,NRHS,LDA,LDB,INFO}
DOUBLE PRECISION A (LDA,*),B(LDB,*)

```
F95 INTERFACE
    SU BROUTINE POSV (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N F O}])\)
    CHARACTER (LEN=1) ::UPLO
    \(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, \mathbb{N} F O\)
    REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B
    SU BROUTINE POSV_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO
    \(\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D B, \mathbb{N} F O\)
    REAL (8), D \(\mathbb{M}\) ENSION (: ::) ::A, B
C INTERFACE
    \#include <sunperfh>
void dposv (charuplo, intn, intnれs, double *a, int lda, double *b, int ldb, int *info);
void dposv_64 (charuplo, long n, long nrhs, double *a, long lda, double *b, long ldb, long *info);

\section*{PURPOSE}
dposv com putes the solution to a realsystem of linear equations
\(A * X=B\), where \(A\) is an \(N\) boy \(-N\) sym m etric positive definte \(m\) atrix and \(X\) and \(B\) are \(N\) boy \(N\) R H S \(m\) atrices.

The Cholesky decom position is used to factorA as
\(A=U * * T * U\), if \(U P L O=U\) ', or
\(A=L * L \star * T\), if \(\mathrm{UPLO}=\mathrm{L}\) ',
\(w\) here \(U\) is an uppertriangularm atrix and \(L\) is a low er triangular \(m\) atrix. The factored form of \(A\) is then used to solve the system ofequations \(A * X=B\).

\section*{ARGUMENTS}

UPLO (input)
\(=U\) ': U pper triangle of \(A\) is stored;
\(=\mathbb{L}\) ': Low ertriangle ofA is stored.

N (input) The num ber of linearequations, i.e., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of the m atrix B. NRH S \(>=0\).

A (input/output)
O n entry, the sym m etric m atrix A. If UPLO \(=U\) ', the leading N -oy N uppertriangularpartofA contains the upper triangularpart of the \(m\) atrix \(A\), and the strictly low er triangularpart of A is not referenced. If UPLO \(=\mathrm{L}\) ', the leading N -by N low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of A is notreferenced.

On exit, if \(\mathbb{N F O}=0\), the factor \(U\) orL from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\).

LD A (input)
The leading dim ension of the anay A. LD A >= \(\max (1, N)\).

B (input/output)
On entry, the N -by-NRHS righthand side matrix B.
On exi, if \(\mathbb{N} F O=0\), the \(N\) by \(N\) RH S solution
matrix \(X\).

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO = -i, the i-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N F O}=\mathrm{i}\), the leading m inoroforderiof
A is notpositive definite, so the factorization could not.be com pleted, and the solution has not been com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dposvx - use the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\) to com pute the solution to a realsystem of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDPOSVX (FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,EQUED,}
S,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK2,INFO)
CHARACTER * 1FACT,UPLO,EQUED
INTEGER N,NRHS,LDA,LDAF,LDB,LDX,INFO
INTEGERW ORK2 (*)
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA,*), AF (LDAF,*), S (*), B (LDB,*),
X (LDX ,*),FERR (*),BERR (*),W ORK (*)
SUBROUT\mathbb{NEDPOSVX_64(FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,EQUED,}
S,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK2, INFO)
CHARACTER * 1 FACT,UPLO,EQUED
INTEGER*8N,NRHS,LDA,LDAF,LDB,LDX,INFO
\mathbb{NTEGER*8W ORK2 (*)}
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA,*), AF (LDAF,*), S (*), B (LDB,*),
X (LDX,*),FERR (*),BERR (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE POSVX \(\mathbb{F} A C T, U P L O, \mathbb{N}], \mathbb{N R H S}], A,[L D A], A F,[L D A F]\), EQUED, \(S, B,[L D B], X,[L D X], R C O N D, F E R R, B E R R,[W O R K]\), [W ORK 2], [ \(\mathbb{N F F O}\) ])

CHARACTER (LEN=1)::FACT,UPLO,EQUED
\(\mathbb{N} T E G E R:\) N, NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) ::W ORK2
REAL (8) :: RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::S,FERR,BERR,W ORK
REAL (8), D \(\mathbb{M} \operatorname{ENSION}(:,:):: A, A F, B, X\)

SU BROUTINE POSVX_64 (FACT, UPLO, \(\mathbb{N}], \mathbb{N R H S}], A,[L D A], A F,[L D A F]\), \(E Q U E D, S, B,[L D B], X,[L D X], R C O N D, F E R R, B E R R,[W O R K]\), [W ORK2], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::FACT,UPLO, EQUED
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{M}\) ENSION (:) ::W ORK2
REAL (8) :: RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) :: S, FERR,BERR,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A \(, A F, B, X\)

\section*{C INTERFACE}
\#include <sunperfh>
void dposvx (char fact, char uplo, intn, int nrhs, double
*a, int lda, double *af, int ldaf, char equed, double *s, double *b, int ldb, double *x, int ldx, double *rcond, double *ferr, double *berr, int *info);
void dposvx_64 (char fact, charuple, long n, long nrhs, double *a, long lda, double *af, long ldaf, char equed, double *s, double *b, long ldb, double *x, long ldx, double *roond, double *ferr, double *berr, long *info);

\section*{PURPOSE}
dposvx uses the Cholesky factorization \(A=U * * T * U\) or \(A=\) L*L**T to compute the solution to a realsystem of linear equations
\(A * X=B, w\) here \(A\) is an \(N\) boy \(-N\) sym \(m\) etric posilive definte \(m\) atrix and \(X\) and \(B\) are \(N\) boy-N R H S \(m\) atrioes.

Emorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT = E', real scaling factors are computed to equilibrate
the system :
\(\operatorname{diag}(S) * A * \operatorname{diag}(S) * \operatorname{inv}(\operatorname{diag}(S)) * X=\operatorname{diag}(S) * B\)
W hether or not the system w illbe equilibrated depends on
the
scaling of the m atrix A , but ifequilibration is used, A is
overw rilten by diag \((\mathrm{S}) \star A\) *diag \((\mathrm{S})\) and B by diag \((\mathrm{S}) \star \mathrm{B}\).
2. IfFACT = N 'or E', the Cholesky decom position is used to
factorthem atrix A (afterequilibration ifFACT = E )
as
\[
\begin{aligned}
& A=U * * T * U, \text { if } U P L O=U ' \text {, or } \\
& A=L * L * * T, \text { if } U P L O=' L^{\prime},
\end{aligned}
\]
\(w\) here \(U\) is an upper triangularm atrix and \(L\) is a low er triangular
\(m\) atrix.
3. If the leading iboy-iprincipal \(m\) inor is not positive definite,
then the routine retums w ith \(\mathbb{N F O}=\) i. O therw ise, the factored
form ofA is used to estim ate the condition num ber of the \(m\) atrix
A. If the reciprocal of the condition num ber is less than \(m\) achine precision, \(\mathbb{N} F O=\mathrm{N}+1\) is retumed as a w aming, but the routine still goes on to solve for \(X\) and com pute emorbounds as described below .
4. The system ofequations is solved forX using the factored form
```

ofA.

```
5. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates for it.
6. If equilibration \(w\) as used, the \(m\) atrix \(X\) is prem ultiplied by diag (S) so that it solves the original system before equilibration.

\section*{ARGUMENTS}

FACT (input)
Specifies w hether or not the factored form of the \(m\) atrix \(A\) is supplied on entry, and ifnot, w hether
the m atrix A should be equilibrated before it is factored. = F : O n entry, AF contains the factored form of \(A\). IfEQUED \(=Y\) ', the matrix A has been equilibrated \(w\) ith scaling factors given by \(S\). A and AF w illnot.be m odified. \(=\mathrm{N}\) ': The m atrix A w illbe copied to A F and factored.
\(=\mathrm{E}\) ': The matrix A w ill be equilibrated if necessary, then copied to A F and factored.

UPLO (input)
\(=U\) ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.
N (input) The num ber of linear equations, i.e., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrices B and X. NRHS >=0.
A (input/output)
On entry, the sym m etric \(m\) atrix A, except ifFA C T = F' and EQUED = Y', then A mustcontain the equilibrated \(m\) atrix diag \((S) * A\) *diag \((S)\). If UPLO \(=\) U', the leading \(N\)-by -N uppertriangular partofA contains the upper triangularpart of the \(m\) atrix
A, and the strictly low ertriangularpartofA is not referenced. If U PLO \(=\mathrm{L}\) ', the leading N -by -N low er triangularpartofA contains the low ertriangularpart of the matrix A, and the strictly upper triangular partofA is not referenced. A is notm odified ifFACT = F or \(\mathrm{N}^{\prime}\), or if \(\mathrm{FACT}=\) E'and EQUED = N 'on exit.

On exit, ifFACT = E' and EQUED = \(\mathrm{Y}^{\prime}\), A is overw ritten by diag \((S) \star A\) *diag \((S)\).

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

\section*{AF (input/output)}

If \(F A C T=F\) ', then \(A F\) is an inputargum ent and on entry contains the triangular factor \(U\) or \(L\) from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\), in the sam e storage form at as A. IfEQUED ne. \(N\) ', then \(A F\) is the factored form of the equilibrated \(m\) atrix diag \((S) \star A * \operatorname{diag}(S)\).

IfFACT = N ', then \(A \mathrm{~F}\) is an output argum ent and on exit retums the triangular factor \(U\) or \(L\) from
the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\) of the originalm atrix A.

IfFACT \(=E\) ', then AF is an output argum ent and on exitretums the triangular factor \(U\) orL from the Cholesky factorization \(A=U * * T * U\) or \(A=\) \(L * L * * T\) of the equilibrated \(m\) atrix \(A\) (see the description ofA for the form of the equilibrated matrix) .

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, N)\).

EQUED (input)
Specifies the form of equilibration thatw as done.
\(=\mathrm{N}\) ': N o equilibration (alw ays true ifFACT = N 7 。
\(=Y\) ': Equilibration w as done, i.e., A has been
replaced by diag \((\mathrm{S})\) * \(A\) * diag \((\mathrm{S})\). EQUED is an inputargum entiffACT = \(\mathrm{F}^{\prime}\); otherw ise, it is an output argum ent.

S (input/output)
The scale factors forA; not accessed if EQUED =
\(\mathrm{N}^{\prime} . \mathrm{S}\) is an inputargum entifFACT=F'; other-
W ise, S is an outputargum ent. IfFACT \(=\mathrm{F}^{\prime}\) and
\(E Q U E D=Y\) ', each elem entof m ust.be positive.

B (input/output)
On entry, the N boy-N RH S righthand side m atrix B.
On exit, if EQUED = \(N^{\prime}\) ', B is notm odified; if
EQUED \(=Y ', B\) is overw ritten by diag \((S) * B\).

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=N+1\), the N -by -N R H S solution
\(m\) atrix \(X\) to the original system of equations.
\(N\) ote that if EQ UED \(=Y\) ', \(A\) and \(B\) are m odified on
exit, and the solution to the equilibrated system
is inv \((\) diag \((S)) \star X\).

LD X (input)
The leading dim ension of the anay X . LD X >= \(\max (1, N)\).

The estim ate of the reciprocal condition num ber of the matrix A after equilibration (if done). If RCOND is less than the \(m\) achine precision (in particular, ifRCOND \(=0\) ), the \(m\) atrix is singular to w orking precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO \(>0\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X()\) ) the \(j\) th column of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{H})\), FERR ( \()\) ) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{D})-X\) TRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(j)\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector \(X(\mathcal{)}\) ) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension ( \(3 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthad an ille-
galvalue
\(>0:\) if \(\mathbb{N F O}=i\), and \(i\) is
\(<=N\) : the leading \(m\) inor oforderiof \(A\) is not positive definite, so the factorization could not be com pleted, and the solution has not been com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1\) : U is nonsingular, butRCOND is less than \(m\) achine precision, m eaning that the \(m\) atrix is singularto w orking precision. Nevertheless, the solution and error bounds are com puted because there are a num berof situationsw here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dpotf2 - com pute the C holesky factorization of a real sym \(m\) etric positive definite \(m\) atrix \(A\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDPOTF2(UPLO,N,A,LDA, INFO)}

```
CHARACTER * 1 UPLO
\(\mathbb{N} T E G E R N, L D A, \mathbb{N} F O\)
DOUBLE PRECISION A (LDA,*)
SU BROUTINE DPOTF2_64 (UPLO ,N,A,LDA, \(\mathbb{N} F O\) )
CHARACTER * 1 UPLO
\(\mathbb{N}\) TEGER*8N,LDA, \(\mathbb{N} F O\)
DOUBLE PRECISION A (LDA,*)
```

F95 INTERFACE

```
    SU BROUTINE POTF2 (UPLO, \(\mathbb{N}], A,[L D A],[\mathbb{N F O}])\)
    CHARACTER (LEN=1) ::UPLO
    \(\mathbb{N} T E G E R:: N, L D A, \mathbb{N F O}\)
    REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A
    SU BROUTINE POTF2_64 (UPLO, \(\mathbb{N}], A,[L D A],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO
    \(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{LD} A, \mathbb{N} F O\)
    REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A
C INTERFACE
    \#include <sunperfh>
void dpotf2 (char uple, int \(n\), double *a, int lda, int *info);
void dpotf2_64 (charuplo, long n, double *a, long lda, long *info);

\section*{PURPOSE}
dpotf2 com putes the C holesky factorization of a real sym \(m\) etric posilive definite m atrix A .

The factorization has the form
\(A=U^{\prime} * U\), if \(U P L O=U '\), or
\(A=L * L \prime\), if UPLO = L',
\(w\) here \(U\) is an uppertriangularm atrix and \(L\) is low er triangular.

This is the unblocked version of the algorithm, calling Level2 BLAS.

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the upper or low er triangular
part of the symm etric m atrix A is stored. \(=U\) ':
U pper triangular
= 'L ': Low er triangular

N (input) The order of them atrix A. \(\mathrm{N}>=0\).

A (input/output)
O n entry, the sym m etric m atrix A. If UPLO \(=U^{\prime}\) ', the leading \(n\) by \(n\) upper triangularpart of A contains the upper triangularpart of the \(m\) atrix \(A\), and the strictly low ertriangularpart of A is not referenced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading n by n low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of A is notreferenced.

On exit, if \(\mathbb{N F O}=0\), the factor \(U\) orL from the
Cholesky factorization \(A=U\) * U orA \(=\mathrm{L} * \mathrm{~L}{ }^{\prime}\).

LD A (input)
The leading dim ension of the array \(A . L D A>=\) \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-\mathrm{k}\), the k -th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N} F O=k\), the leading \(m\) inor oforder \(k\) is notpositive definite, and the factorization could notbe com pleted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dpotrf-com pute the C holesky factorization of a real sym \(m\) etric positive definite \(m\) atrix \(A\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDPOTRF(UPLO,N,A,LDA, INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGERN,LDA,}\mathbb{N}FO
DOUBLE PRECISION A (LDA,*)
SU BROUT\mathbb{NEDPOTRF_64(UPLO,N,A,LDA, INFO )}
CHARACTER * 1 UPLO
INTEGER*8N,LDA,INFO
DOUBLE PRECISION A (LDA,*)
F95 INTERFACE
SUBROUT\mathbb{NE POTRF (UPLO, NN,A, [LDA ], [NFO])}
CHARACTER (LEN=1) ::UPLO
INTEGER ::N,LDA,\mathbb{NFO}
REAL (8),D IM ENSION (:,:) ::A
SUBROUT\mathbb{NEPOTRF_64 (UPLO, N ],A,[LDA],[NFO])}
CHARACTER (LEN=1) ::UPLO
\mathbb{NTEGER (8) ::N,LDA,\mathbb{NFO}}0=1/2
REAL (8),D IM ENSION (:,:) ::A
C INTERFACE
\#include <sunperfh>

```
void dpotrf(charuple, int \(n\), double *a, int lda, int *info);
void dpotrf_64 (charuplo, long n, double *a, long lda, long *info);

\section*{PURPOSE}
dpotrf com putes the C holesky factorization of a real sym \(m\) etric posilive definite m atrix A .

The factorization has the form
\(A=U * * T * U\), if \(U P L O=U\) ', or
\(A=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\), if \(\mathrm{UPLO}=\mathrm{L}\) ',
\(w\) here \(U\) is an uppertriangularm atrix and \(L\) is low er triangular.

This is the block version of the algorithm, calling Level 3 BLAS .

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': Upper triangle of A is stored;
\(=\mathbb{L}\) ': Low er triangle of \(A\) is stored.

N (input) The order of them atrix A. N \(>=0\).

A (input/output)
On entry, the sym m etric m atrix A. If UPLO \(=U\) ', the leading N -by N uppertriangularpartof u contains the uppertriangularpart of the \(m\) atrix \(A\), and the strictly low er triangularpart of A is not referenced. If UPLO = 'L', the leading N -by N low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of A is not referenced.

On exit, if \(\mathbb{N} F O=0\), the factor \(U\) orL from the Cholesky factorization \(A=U * * T * U\) or \(A=L * L * * T\).

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).
= 0: successfulexit
< 0 : if \(\mathbb{N}\) FO \(=-\)-i, the i-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the leading \(m\) inoroforder \(i\) is not positive definite, and the factorization could notbe com pleted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dpotri-com pute the inverse of a real sym \(m\) etric positive definite \(m\) atrix \(A\) using the Cholesky factorization \(A=\) \(\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\) com puted by SPO TRF

\section*{SYNOPSIS}
```

SUBROUTINE DPOTRI(UPLO,N,A,LDA, INFO)
CHARACTER * 1UPLO
\mathbb{NTEGER N,LDA,}\mathbb{N}FO
DOUBLE PRECISION A (LDA,*)
SUBROUT\mathbb{NE DPOTRI_64(UPLO,N,A,LDA, INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N,LDA,}\mathbb{N}FO
DOUBLE PRECISION A (LDA,*)
F95 INTERFACE
SU BROUT\mathbb{NE POTRI(UPLO, N ],A , [LDA ], [NNO ])}
CHARACTER (LEN=1) ::UPLO
\mathbb{NTEGER ::N,LDA,}\mathbb{N}FO
REAL (8),D IM ENSION (:,:) ::A
SU BROUT\mathbb{NE POTRI_64 (UPLO, N ],A, [LDA ], [NNO ])}
CHARACTER (LEN=1) ::UPLO
\mathbb{NTEGER (8)::N,LDA,}\mathbb{NFO}
REAL (8),D IM ENSION (:,:) ::A
void dpotri(charuplo, int $n$, double *a, int lda, int *info);
void dpotri_ 64 (charuplo, long n, double *a, long lda, long *info);

## PURPOSE

dpotricom putes the inverse of a real symm etric positive definite $m$ atrix $A$ using the Cholesky factorization $A=$ $\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ orA $=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ com puted by SPO TRF .

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': Uppertriangle of $A$ is stored;
$=\mathbb{L}$ ': Low er triangle of $A$ is stored.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
O $n$ entry, the triangular factor $U$ or $L$ from the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$, as com puted by SPO TRF. On exit, the upper or low er triangle of the (sym m etric) inverse of A, overw riting the input factor $U$ orL.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, N)$.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N F O}=-i$, the i-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N} F O=i$, the $(i, i)$ elem entof the factor
U orL is zero, and the inverse could not.be com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
dpotrs-solve a system of linearequationsA *X = B w th a
symm etric positive definitem atrix A using the C holesky fac-
torization A = U **T * U orA = L*L**T com puted by SPO TRF
```


## SYNOPSIS

```
SU BROUT\mathbb{NE DPOTRS (UPLO,N,NRHS,A,LDA,B,LDB, NNFO)}
CHARACTER * 1 UPLO
INTEGERN,NRHS,LDA,LDB,INFO
DOUBLE PRECISION A (LDA,*),B (LDB,*)
SU BROUT\mathbb{NE DPOTRS_64 (UPLO ,N,NRHS,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO
INTEGER*8N,NRHS,LDA,LDB,\mathbb{NFO}
DOUBLE PRECISION A (LDA,*),B (LDB,*)
```

F95 INTERFACE
SU BROUTINE POTRS (UPLO, $\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N F O}])$
CHARACTER (LEN=1)::UPLO
$\mathbb{N}$ TEGER ::N,NRHS,LDA,LDB, $\mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSIO N (:,:) ::A,B
SU BROUTINE POTRS_64 (UPLO, $\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N F O}])$
CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER (8) ::N,NRHS,LDA,LDB, $\mathbb{N}$ FO
REAL (8), D $\mathbb{M}$ ENSION (:,:) ::A , B
void dpotrs (charuplo, intn, intnrhs, double *a, int lda, double *b, int ldb, int *info);
void dpotrs_64 (charuplo, long n, long nrhs, double *a, long lda, double *b, long lalb, long *info);

## PURPOSE

dpotrs solves a system of linearequations $A * X=B$ w th a sym $m$ etric positive definite $m$ atrix A using the Cholesky factorization $A=U * * T * U$ or $A=L * L * * T$ com puted by SPOTRF.

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': Uppertriangle of $A$ is stored;
$=\mathbb{L}$ ': Low er triangle ofA is stored.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRH S >=0.

A (input) The triangular factor $U$ or $L$ from the Cholesky
factorization $A=U * * T * U$ orA $=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$, as com puted by SPO TRF .

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, N)$.

B (input/output)
On entry, the right hand side m atrix B. On exit, the solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N ~ F O ~}=-$ i, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dppcon -estim ate the reciprocal of the condition num ber (in the 1 -norm ) of a realsym $m$ etric positive definite packed $m$ atrix using the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=$ L *L**T com puted by SPPTRF

## SYNOPSIS

```
SUBROUT\mathbb{NEDPPCON(UPLO,N,A,ANORM ,RCOND,W ORK,W ORK2,\mathbb{NFO)}}\mathbf{N}\mathrm{ (N,A}
```

```
CHARACTER * 1 UPLO
```

$\mathbb{N}$ TEGER $N, \mathbb{I N F O}$
$\mathbb{I N}$ TEGERW ORK 2 (*)
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION A (*),W ORK (*)
SU BROUTINEDPPCON_64 (UPLO,N,A,ANORM,RCOND,WORK,WORK2, $\mathbb{N} F O$ )
CHARACTER * 1 UPLO
$\mathbb{N}$ TEGER*8 $\mathrm{N}, \mathbb{N}$ FO
$\mathbb{N}$ TEGER*8 W ORK 2 (*)
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION A (*), W ORK (*)

## F95 INTERFACE

SUBROUTINE PPCON (UPLO,N,A,ANORM,RCOND, [WORK], [W ORK2], [ $\mathbb{N} F O]$ )
CHARACTER (LEN=1)::UPLO
$\mathbb{N}$ TEGER :: N, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) ::W ORK 2
REAL (8) ::ANORM,RCOND
REAL (8),D IM ENSION (:) ::A,W ORK

CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R(8):: N, \mathbb{N F O}$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: W$ ORK2
REAL (8) ::ANORM, RCOND
REAL (8), D $\mathbb{M}$ ENSION (:) ::A , W ORK

## C INTERFACE

\#include <sunperfh>
void dppcon (charuple, intn, double *a, double anorm , double *roond, int*info);
void dppcon_64 (charuplo, long n, double *a, double anorm, double *rcond, long *info);

## PURPOSE

dppoon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a realsym m etric positive definite packed $m$ atrix using the Cholesky factorization $A=U * * T * U$ or $A=$ L*L**T com puted by SPPTRF .

A $n$ estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND = 1 / (ANORM * norm (inv (A))).

## ARGUMENTS

UPLO (input)
= U : U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrix A. $\mathrm{N}>=0$.
A (input) The triangular factorU or $L$ from the Cholesky
factorization $A=U * * T * U$ orA $=\mathrm{L} * \mathrm{~L} * * T$, packed colum nw ise in a linearanay. The jth colum $n$ of $U$ or $L$ is stored in the array A as follow s: if UPLO = U', A $(i+(j-1) * j 2)=U(i, j)$ for $1<=i<=j$ if UPLO = L', A (i+ ( $j-1)^{\star}(2 n-j / 2)=L(i, 7)$ for j=i<=n.

ANORM (input)
The 1-norm (or infinity-norm) of the symmetric matrix A.

## RCOND (output)

The reciprocal of the condition num ber of the
$m$ atrix $A$, com puted as RCOND = 1/(ANORM *A $\mathbb{N} V N M)$,
where $A \mathbb{N} V N M$ is an estim ate of the 1 -nom of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension ( $3 * \mathrm{~N}$ )
W ORK 2 (w orkspace)
dim ension (N)
$\mathbb{I N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum enthad an ille-
galvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dppequ - com pute row and colum $n$ scalings intended to equilibrate a symm etric positive definite $m$ atrix $A$ in packed storage and reduce its condition num ber (w ith respect to the tw o-norm )

## SYNOPSIS

```
SUBROUT\mathbb{NE DPPEQU (UPLO,N,A,SCALE,SCOND,AM AX,INFO)}
```

```
CHARACTER * 1 UPLO
```

$\mathbb{N}$ TEGER $N, \mathbb{I N F O}$
DOUBLE PRECISION SCOND,AMAX
DOUBLE PRECISION A (*), SCALE (*)
SU BROUTINE DPPEQU_64 (UPLO,N,A,SCALE,SCOND,AMAX, $\mathbb{N} F O$ )
CHARACTER * 1 UPLO
$\mathbb{N}$ TEGER*8N, $\mathbb{N} F O$
DOUBLE PRECISION SCOND,AMAX
DOUBLE PRECISION A (*), SCALE (*)

## F95 INTERFACE

SU BROUTINE PPEQU (UPLO, $\mathbb{N}], A, S C A L E, S C O N D, A M A X,[\mathbb{N} F O])$
CHARACTER (LEN=1)::UPLO
$\mathbb{N} T E G E R:: N, \mathbb{N F O}$
REAL (8) :: SCOND ,AMAX
REAL (8), D $\mathbb{M}$ ENSION (:) ::A, SCALE

SU BROUTINE PPEQU_64 (UPLO, $\mathbb{N}], A, S C A L E, S C O N D, A M A X,[\mathbb{N} F O$ ])
CHARACTER (LEN=1)::UPLO
$\mathbb{N} T E G E R(8):: N, \mathbb{N} F O$
REAL (8) :: SCOND ,AMAX
REAL (8),D $\mathbb{I M}$ ENSION (:) ::A ,SCALE

## C INTERFACE

\#include <sunperfh>
void dppequ (charuplo, intn, double *a, double *scale, double *scond, double *am ax, int *info);
void dppequ_64 (charuple, long n, double *a, double *scale, double *scond, double *am ax, long *info);

## PURPOSE

dppequ com putes row and colum $n$ scalings intended to equilibrate a symm etric positive definite $m$ atrix $A$ in packed storage and reduce its condition num ber (w ith respect to the two-norm). S contains the scale factors, $S(i)=1 /$ sqit $(A(i, i))$, chosen so that the scaled $m$ atrix $B$ w ith elem ents $B(i, j)=S(i) * A(i, j) * S(i)$ has ones on the diagonal. This choige ofS puts the condition num ber of B w ithin a factor $N$ of the sm allestpossible condition num berover all possible diagonal scalings.

## ARGUMENTS

```
UPLO (input)
```

= U ': U pper triangle ofA is stored;
= L': Low er triangle ofA is stored.
N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input) The upper or low er triangle of the sym m etric $m$ atrix A, packed colum nw ise in a linear array. The $j$ th column of A is stored in the array A as follow s: if UPLO = $U^{\prime}, A(i+(j-1) * j 2)=A(i, 7)$ for $1<=i<=\dot{j}$ ifUPLO $=\mathrm{L}$ ', A ( $\left.i+(j-1)^{*}(2 n-j) / 2\right)$ $=A(i, j)$ for $j=i<=n$.

SCA LE (output)
If $\mathbb{N} F O=0, S C A L E$ contains the scale factors for A.

SCOND (output)
If $\mathbb{N} F O=0, S C A L E$ contains the ratio of the $s m$ allest SCA LE (i) to the largestSCA LE (i). If SCOND
>= 0.1 and AMAX is neither too large nor too sm all, it is notw orth scaling by SCA LE .

AMAX (output)
A bsolute value of largestm atrix elem ent. IfA M A X
is very close to overflow orvery close to underflow , the m atrix should be scaled.
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the i-th argum enthad an illegalvalue
> 0 : if $\mathbb{N F O}=$ i, the $i$-th diagonal elem ent is nonpositive.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dppris -im prove the com puted solution to a system of linear equations when the coefficientm atrix is sym $m$ etric positive definite and packed, and provides emrorbounds and backw ard enrorestim ates for the solution

## SYNOPSIS

```
SU BROUT\mathbb{NE DPPRFS (UPLO,N,NRHS,A,AF,B,LDB,X,LDX,FERR,BERR,}
    W ORK,WORK2,INFO)
CHARACTER * 1 UPLO
INTEGERN,NRHS,LDB,LDX,INFO
\mathbb{NTEGERWORK2(*)}
DOUBLE PRECISION A (*),AF (*),B (LDB ,*), X (LDX ,*),FERR (*),
BERR (*),WORK (*)
SUBROUT\mathbb{NE DPPRFS_64 (UPLO,N,NRHS,A,AF,B,LDB,X,LDX,FERR,}
    BERR,W ORK,WORK2,\mathbb{NFO)}
CHARACTER * 1 UPLO
\mathbb{N}TEGER*8N,NRHS,LDB,LDX,INFO
INTEGER*8W ORK2 (*)
DOUBLE PRECISION A (*),AF (*),B (LDB,*), X (LDX ,*), FERR (*),
BERR (*),W ORK (*)
```


## F95 INTERFACE

```
SU BROUTINE PPRFS (UPLO,N, NRHS],A,AF,B, [LDB],X, [LDX],FERR, BERR, [W ORK], [W ORK2], [ \(\mathbb{N} F O\) ])
CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION (:) ::W ORK2
```

SUBROUTINE PPRFS_64 (UPLO,N, NRHS],A,AF,B, [LDB],X,[LDX],FERR, BERR, [WORK], [WORK2], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER (8) ::N,NRHS,LDB,LDX, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I M}$ ENSION (:) ::W ORK2
REAL (8), D $\mathbb{M}$ ENSION (:) ::A,AF,FERR,BERR,W ORK REAL (8),D IM ENSION (: : : : : B, X

## C INTERFACE

\#include < sunperfh>
void dpprfs (charuplo, intn, intnirs, double *a, double
*af, double *b, int ldb, double *x, int ldx, dou-
ble *ferr, double *berr, int*info);
void dpprfs_64 (char uplo, long n, long nrhs, double *a, double *af, double *b, long ldb, double *x, long ldx, double *ferr, double *berr, long *info);

## PURPOSE

dpprif im proves the com puted solution to a system of linear equations when the coefficientm atrix is sym $m$ etric positive definite and packed, and provides errorbounds and backw ard errorestim ates for the solution.

## ARGUMENTS

## UPLO (input)

$=\mathrm{U}$ : U pper triangle of A is stored;
= L': Low ertriangle ofA is stored.
N (input) The order of the matrix $\mathrm{A} . \mathrm{N}>=0$.
NRHS (input)
The num ber of righthand sides, i.e., the num ber
of collm ns of the m atrices B and X. NRH S >=0.
A (input) The upper or low er triangle of the sym metric $m$ atrix A, packed colum nw ise in a linear array. The $j$ th column of A is stored in the array A as

$$
\text { follow s: if UPLO = } U^{\prime}, A(i+(j 1) \star j 2)=A(i, 7)
$$

for $1<=i<=j$ ifUPLO $=\mathrm{L}$ ', A $(i+(j-1) *(2 n-j) / 2)$
$=A(i, 7)$ for $j=i<=n$.

## AF (input)

The triangular factor $U$ or $L$ from the Cholesky factorization $A=U * * T * U$ or $A=L * L * *$, as com puted by SPPTRF /CPPTRF, packed colum nw ise in a linear aray in the sam e form at as A (see A).
$B$ (input) The righthand side $m$ atrix $B$.
LD B (input)
The leading dim ension of the aray $B$. LD B >= $\max (1, N)$.
$X$ (input/output)
On entry, the solution $m$ atrix $X$, as com puted by SPPTRS. On exit, the im proved solution $m$ atrix $X$.

LD X (input)
The leading dim ension of the aray X . LD X >= $\max (1, \mathbb{N})$.

FERR (output)
The estim ated forw ard errorbound for each solution vector $X()$ (the $j$ th colum $n$ of the solution $m$ atrix X). If XTRUE is the true solution corresponding to $X(\mathcal{O}), \operatorname{FERR}(\underset{)}{(1)}$ is an estim ated upperbound for the $m$ agnitude of the largest ele$m$ ent in ( $\mathrm{X}(\mathcal{j})-\mathrm{XTRUE}$ ) divided by the $m$ agnitude of the largestelem ent in $X(\mathcal{J})$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true enror.

## BERR (output)

The com ponentw ise relative backw ard emor of each
solution vector $X(\mathcal{j})$ (i.e., the sm allest relative
change in any elem entofA orB thatm akes X ( $\mathcal{j}$ ) an exactsolution).

W ORK (w orkspace)
dím ension ( $3 * \mathrm{~N}$ )
W ORK 2 (w orkspace)
dim ension (N)
$\mathbb{N} F O$ (output)
= 0 : successfinlexit
<0: if $\mathbb{I N}$ FO $=-i$, the $i$-th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dppsv - com pute the solution to a real system of linear equations $A * X=B$,

## SYNOPSIS

```
SUBROUT\mathbb{NE DPPSV (UPLO,N,NRHS,A,B,LDB,INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER N,NRHS,LDB,INFO}
DOUBLE PRECISION A (*),B(LDB,*)
SU BROUTINE DPPSV_64 (UPLO ,N ,NRHS,A ,B,LD B, INFO)
CHARACTER * 1 UPLO
INTEGER*8N,NRHS,LDB,INFO
DOUBLE PRECISION A (*),B (LDB,*)
F95 INTERFACE
SUBROUT\mathbb{NE PPSV (UPLO,N, NRHS],A,B, [LDB],[NFO])}
CHARACTER (LEN=1) ::UPLO
\mathbb{NTEGER ::N,NRHS,LDB,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REAL (8),D IM ENSION (:) ::A
REAL (8),D IM ENSION (:,:) ::B
```



```
CHARACTER (LEN=1) ::UPLO
\mathbb{NTEGER (8)::N,NRHS,LDB,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REAL (8),D IM ENSION (:) ::A
REAL (8),D IM ENSIO N (:,:) ::B
```


## C INTERFACE

\#include <sunperfh>
void dppsv (charuplo, intn, intnrhs, double *a, double *b, int ldb, int *info);
void dppsv_64 (charuple, long n, long nihs, double *a, double *b, long ldb, long *info);

## PURPOSE

dppsv com putes the solution to a real system of linear equations
A * $X=B$, where $A$ is an $N$-by-N sym m etric positive defintie $m$ atrix stored in packed form at and $X$ and $B$ are $N$ by $-N$ RH $S$ $m$ atrices.

The Cholesky decom position is used to factorA as
$A=U * * T * U$, if $U P L O=U '$, or
$\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$, if $\mathrm{UPLO}=\mathrm{L}$ ',
$w$ here $U$ is an upper triangularm atrix and $L$ is a low er triangular $m$ atrix. The factored form of $A$ is then used to solve the system of equations $A * X=B$.

## ARGUMENTS

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low er triangle ofA is stored.
N (input) The num ber of linear equations, ie., the order of the matrix A. $N>=0$.

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of them atrix B. NRHS $>=0$.

A (input/output)
O $n$ entry, the upper or low ertriangle of the sym $m$ etric $m$ atrix $A$, packed colum nw ise in a linear array. The $j$ th colum $n$ of $A$ is stored in the array A as follows: if UPLO $=U^{\prime}, \mathrm{A}(i+(j$
 $(j-1)^{*}(2 n-j / 2)=A(i, 7)$ for $\dot{j}=i<=n$. See below for further details.

On exit, if $\mathbb{N F O}=0$, the factor $U$ orL from the Cholesky factorization $A=U * * T * U$ or $A=L * L * * T$, in the sam e storage form at as A.

B (input/output)
On entry, the N -by-NRHS righthand side m atrix B. On ex弌, if $\mathbb{N F O}=0$, the N boy $-\mathrm{NRH} S$ solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the anay $\mathrm{B} . \operatorname{LDB}>=$ $\max (1, N)$.
$\mathbb{N}$ FO (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N F O}=-i$, the $i-$ th argum enthad an illegalvalue
$>0:$ if $\mathbb{N}$ FO $=i$, the leading $m$ inoroforder iof $A$ is notposilive definite, so the factorization could notbe com pleted, and the solution has not been com puted.

## FURTHER DETAILS

The packed storage schem e is illustrated by the follow ing exam ple when $N=4, \mathrm{UPLO}=\mathrm{U}$ ':

Tw o-dim ensional storage of the sym $m$ etric $m$ atrix A :

```
al1 a12 al3 a14
    a22 a23 a24
        a33 a34 (aij= con\g (aï))
        a44
```

Packed storage of the upper triangle ofA:
$A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]$

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dppsvx - use the Cholesky factorization $A=U * * T * U$ or $A=$ $\mathrm{L}{ }^{*} \mathrm{~L}^{* *} \mathrm{~T}$ to com pute the solution to a realsystem of linear equations $A * X=B$,

## SYNOPSIS

```
SUBROUT\mathbb{NE DPPSVX (FACT,UPLO,N,NRHS,A,AF,EQUED,S,B,LDB,}
    X,LDX,RCOND,FERR,BERR,W ORK,W ORK2,INFO)
CHARACTER * 1FACT,UPLO,EQUED
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
INTEGER W ORK2(*)
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (*), AF (*), S (*), B (LDB,*), X (LDX,*),
FERR (*),BERR (*),W ORK (*)
SU BROUT\mathbb{NEDPPSVX_64 (FACT,UPLO,N,NRHS,A,AF,EQUED,S,B,}
    LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK 2, INFO)
CHARACTER * 1FACT,UPLO,EQUED
\mathbb{NTEGER*8N,NRHS,LDB,LDX,NNFO}
INTEGER*8 W ORK2 (*)
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (*), AF (*), S (*), B (LDB,*), X (LDX **),
FERR (*),BERR (*),W ORK (*)
```


## F95 INTERFACE

SUBROUTINE PPSVX $\mathbb{F} A C T, U P L O, \mathbb{N}], \mathbb{N} R H S], A, A F, E Q U E D, S, B$, [LDB],X, [LDX],RCOND ,FERR, BERR, [W ORK], [W ORK 2], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1) ::FACT,UPLO,EQUED
$\mathbb{N}$ TEGER : : N,NRHS,LDB,LDX, $\mathbb{N}$ FO
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):$ ORK2
REAL (8) :: RCOND
REAL (8), D $\mathbb{M}$ ENSION (:) ::A,AF,S,FERR,BERR,W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) :: B, X

SUBROUTINE PPSVX_64 (FACT, UPLO, $\mathbb{N}], \mathbb{N} R H S], A, A F, E Q U E D, S, B$, $[$ LDB $], \mathrm{X},[\llbracket D X], R C O N D, F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N} F O])$

CHARACTER (LEN=1) ::FACT,UPLO, EQUED
$\mathbb{N} T E G E R(8):: N, N R H S, L D B, L D X, \mathbb{N} F O$
$\mathbb{N}$ TEGER (8), D $\mathbb{M}$ ENSION (:) ::W ORK2
REAL (8) ::RCOND
REAL (8), D $\mathbb{M}$ ENSION (:) ::A,AF,S,FERR,BERR,W ORK
REAL (8), D $\mathbb{M}$ ENSION (: $:$ :) : : B, X

## C INTERFACE

\#include <sunperfh>
void dppsvx (char fact, char uplo, intn, int nrhs, double *a, double *af, charequed, double *s, double *b, int ldlb, double *x, int ldx, double *rcond, double *ferr, double *berr, int *info);
void dppsvx_64 (char fact, charuplo, long n, long nrhs, double *a, double *af, charequed, double *s, double *b, long ldb, double *x, long ldx, double *rcond, double *ferr, double *berr, long *info);

## PURPOSE

dppsvx uses the Cholesky factorization $A=U * * T * U$ or $A=$ $\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ to com pute the solution to a realsystem of linear equations
$A * X=B$, where $A$ is an $N$ boy $-N$ sym m etric positive defintem atrix stored in packed form atand $X$ and $B$ are $N$ foy-N RH S m atrices.

E rrorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:

1. IfFACT = E', real scaling factors are com puted to equilibrate
the system :
$\operatorname{diag}(S) * A * \operatorname{diag}(S) * \operatorname{inv}(\operatorname{diag}(S)) * X=\operatorname{diag}(S) * B$
W hether or not the system w illbe equilibrated depends on the
scaling of the m atrix A , but ifequilibration is used, A
overw rilten by diag $(\mathrm{S}) \star A$ *diag $(\mathrm{S})$ and B by diag $(\mathrm{S}) \star$ B .
2. IfFACT = N 'or E', the Cholesky decom position is used to
factor them atrix A (afterequilibration ifFACT =E)
as

$$
\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}, \text { if } \mathrm{UPLO}=\mathrm{U} \text { ', or }
$$

$\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$, ifUPLO = L ',
w here U is an upper triangularm atrix and L is a low er triangular
$m$ atrix.
3. If the leading i-by-iprincipal m inor is not positive definite,
then the routine retums $w$ ith $\mathbb{N F O}=$ i. O therw ise, the factored
form ofA is used to estim ate the condition num ber of the $m$ atrix
A. If the reciprocal of the condition num ber is less than $m$ achine
precision, $\mathbb{N} F O=N+1$ is retumed as a w aming, but the routine
still goes on to solve forX and com pute errorbounds as described below .
4. The system ofequations is solved forX using the factored form
of A.
5. Iterative refinem ent is applied to im prove the com puted solution
$m$ atrix and calculate error bounds and backw ard error estim ates
for it.
6. If equilibration $w$ as used, the $m$ atrix $X$ is prem ultiplied by
diag (S) so that it solves the original system before equilibration.

## ARGUMENTS

FACT (input)
Specifies w hether or not the factored form of the $m$ atrix $A$ is supplied on entry, and if not, whether them atrix A should be equilibrated before it is factored. = F': On entry, AF contains the fac-
tored form ofA. IfEQUED $=Y$ ', them atrix $A$ has been equilibrated $w$ ith scaling factors given by $S$. A and AF w illnotbe m odified. $=\mathrm{N}$ ': Them atrix A w illbe copied to A F and factored.
$=\mathrm{E}$ : The matrix A w ill be equilibrated if necessary, then copied to AF and factored.

UPLO (input)
= U : U ppertriangle ofA is stored;
= L': Low er triangle ofA is stored.

N (input) The num ber of linear equations, i.e., the order of them atrix A. $N>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrices B and X. NRHS >=0.
A (input/output)
O $n$ entry, the upper or low er triangle of the sym $m$ etric $m$ atrix $A$, packed colum nw ise in a linear array, exceptifFACT = F'and EQUED = $Y^{\prime}$ ', then A $m$ ust contain the equilibrated $m$ atrix diag $(S) * A * \operatorname{diag}(S)$. The $j$ th column ofA is stored in the array A as follow s: if UPLO = U', A (i+ $(j-1) * j 2)=A(i, 7)$ for $1<=i<=j$ if $U P L O=L \prime$, $A(i+(j-1) *(2 n-j) / 2)=A(i, j)$ for $j=i<=n$. See below for further details. A is not $m$ odified if FACT = F' or $\mathrm{N}^{\prime}$, orifFACT = E'andEQUED = N 'on exit.

On exit, ifFACT = E' and EQUED = $\mathrm{Y}^{\prime}$, A is overw ritten by diag (S)*A *diag (S).

AF (input/output)
$(\mathbb{N} *(N+1) / 2)$ IfFACT $=F$ ', then $A F$ is an input argum ent and on entry contains the triangular factor $U$ orL from the Cholesky factorization $\mathrm{A}=$ $U{ }^{*} U$ or $A=L * L$ ', in the sam e storage form atas $A$. IfEQUED ne. $N$ ', then $A F$ is the factored form of the equilibrated $m$ atrix $A$.

IfFACT = N ', then AF is an output argum ent and on exit retums the triangular factorU orL from the Cholesky factorization $\mathrm{A}=\mathrm{U}$ * U orA $=\mathrm{L} \star \mathrm{L}$ 'of the originalm atrix A.

IfFACT=E', then AF is an output argum ent and on exit retums the triangular factor $U$ orL from the Cholesky factorization $\mathrm{A}=\mathrm{U}$ * U orA $=\mathrm{L}$ * L 'of the equilibrated $m$ atrix A (see the description of

A for the form of the equilibrated $m$ atrix).

EQUED (input)
Specifies the form of equilibration thatw as done.
$=N^{\prime}:$ No equilibration (alw ays true ifFA CT = N).
$=Y$ ': Equilibration w as done, i.e., A has been replaced by diag (S) * A * diag (S). EQUED is an input argum ent ifFACT = F '; otherw ise, it is an outputargum ent.

S (output)
The scale factors forA ; not accessed if EQUED = $\mathrm{N}^{\prime} . \mathrm{S}$ is an inputargum ent ifFACT = $\mathrm{F}^{\prime}$; otherw ise, S is an outputargum ent. IfFACT $=\mathrm{F}^{\prime}$ and EQUED = Y', each elem entofS m ustbe positive.
B (input/output)
On entry, the $N-b y-N R H S$ righthand side m atrix $B$.
On exit, if EQUED = N ', B is notm odified; if
EQUED $=Y$ ', $B$ is overw ritten by diag $(S) * B$.

LD B (input)
The leading dim ension of the aray B. LD B >= $\max (1, N)$.

X (output)
If $\mathbb{N} F O=0$ or $\mathbb{N} F O=N+1$, the $N$-by-NRHS solution
$m$ atrix $X$ to the original system of equations.
$N$ ote that if EQUED = $Y$ ', A and B arem odified on exit, and the solution to the equilibrated system is inv (diag $(S)) * X$.

LD X (input)
The leading dim ension of the array X . LD X >= $\max (1, N)$.

## RCOND (output)

The estim ate of the reciprocal condition num ber of the matrix A afterequilibration (if done). If
RCOND is less than them achine precision (in particular, ifRCOND $=0$ ), the $m$ atrix is singular to w orking precision. This condition is indicated by a retum code of $\mathbb{N} \mathrm{FO}>0$.

FERR (output)
The estim ated forw ard errorbound for each solution vector $X(\neg)$ the $j$ th colum $n$ of the solution matrix X). If XTRUE is the true solution comesponding to $X(\mathcal{O}), \operatorname{FERR}(\mathcal{O})$ is an estim ated upperbound for the $m$ agnitude of the largest ele-
$m$ ent in $(X(\mathcal{O})$ X TRUE $)$ divided by the $m$ agnitude of the largestelem ent in $X(\mathcal{J})$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

## BERR (output)

The com ponentw ise relative backw ard error of each solution vector $X(\mathcal{)}$ ) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( $\mathcal{j}$ ) an exactsolution).
W ORK (w orkspace)
dím ension ( $3 * \mathrm{~N}$ )
W ORK 2 (w orkspace)
dim ension (N)
$\mathbb{I N F O}$ (output)
= 0 : successfulexit
<0: if $\mathbb{N N}$ FO = -i, the i-th argum enthad an illegalvalue
$>0$ : if $\mathbb{N} F O=i$, and $i$ is
$<=\mathrm{N}$ : the leading m inor oforderiof A is not positive definite, so the factorization could not be com pleted, and the solution has not been com puted. RCOND $=0$ is retumed. $=\mathrm{N}+1: \mathrm{U}$ is nonsingular, butRCOND is less than machine precision, $m$ eaning that the $m$ atrix is singularto working precision. Nevertheless, the solution and error bounds are com puted because there are a num berof siluations w here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

## FURTHER DETAILS

The packed storage schem e is illustrated by the follow ing exam plewhen $\mathrm{N}=4, \mathrm{UPLO}=\mathrm{U}$ ':

Tw o-dim ensional storage of the sym $m$ etric $m$ atrix A :

```
al1 a12 al3 a14
    a22 a23 a24
        a33 a34 (aij= conjg (aji))
```

            a44
    Packed storage of the upper triangle ofA :
$A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]$

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dpptrf - com pute the Cholesky factorization of a real sym $m$ etric positive definite m atrix A stored in packed form at

## SYNOPSIS

```
SUBROUT\mathbb{NE DPPTRF (UPLO,N,A , INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER N, INFO}
DOUBLE PRECISION A (*)
SU BROUTINE DPPTRF_64(UPLO,N,A,\mathbb{NFO)}
CHARACTER * 1 UPLO
INTEGER*8 N, INFO
DOUBLE PRECISION A (*)
F95 INTERFACE
```



```
CHARACTER (LEN=1)::UPLO
\mathbb{NTEGER ::N,\mathbb{NFO}}\mathbf{N}=0
REAL (8),D IM ENSION (:) ::A
SUBROUTINE PPTRF_64 (UPLO,N,A,[\mathbb{NFO ])}
CHARACTER (LEN=1)::UPLO
\mathbb{NTEGER (8)::N,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REAL (8),D IM ENSION (:) ::A
```

void dpptrf(char uple, intn, double *a, int *info);
void dpptrf_64 (charuplo, long n, double *a, long *info);

## PURPOSE

dpptrf com putes the C holesky factorization of a real sym $m$ etric positive definite $m$ atrix A stored in packed form at.

The factorization has the form
$A=U * * T * U$, if $U P L O=U '$ 'or
$A=L \star L^{* *} \mathrm{~T}$, ifUPLO $=\mathrm{L} \prime$,
$w$ here $U$ is an upper triangularm atrix and $L$ is low er triangular.

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': U pper triangle ofA is stored;
= L': Low er triangle of A is stored.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
O n entry, the upper or low er triangle of the sym $m$ etric $m$ atrix $A$, packed colum nw ise in a linear amray. The jth colum n of $A$ is stored in the anay $A$ as follows: if UPLO $=U '$ 'A (i+ (j 1) $j^{2} 2$ ) $=A(i, j)$ for $1<=i<=j$ ifUPLO $=~ L '$ ' A ( $i+$ $(j-1) *(2 n-j / 2)=A(i, j)$ for $j=i<=n$. See below for further details.

On exit, if $\mathbb{N} F O=0$, the triangular factor $U$ orL
from the Cholesky factorization $A=U * * T * U$ or $A=$ $\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$, in the sam e storage form atas A .
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the $i$ th argum enthad an illegalvałue
$>0:$ if $\mathbb{N} F O=i$, the leading $m$ inoroforder is notpositive definite, and the factorization could notbe com pleted.

## FURTHER DETAILS

The packed storage schem e is illustrated by the follow ing exam ple w hen $N=4, \mathrm{UPLO}=\mathrm{U}$ ':

Tw o-dim ensionalstorage of the sym $m$ etric $m$ atrix A :

```
a11 a12 a13 a14
    a22 a23 a24
        a33 a34 (aij= a\ddot{i})
        a44
```

Packed storage of the upper triangle ofA :

$$
A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]
$$

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dpptri-com pute the inverse of a real sym $m$ etric positive definite $m$ atrix $A$ using the Cholesky factorization $A=$ $U * * T * U$ orA $=L * L * * T$ com puted by SPPTRF

## SYNOPSIS

```
SU BROUT\mathbb{NE DPPTRI(UPLO,N,A ,NNFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER N, INFO}
DOUBLE PRECISION A (*)
SU BROUT\mathbb{NE DPPTRI_64(UPLO,N,A,INFO)}
CHARACTER * 1 UPLO
INTEGER*8 N, INFO
DOUBLE PRECISION A (*)
```


## F95 INTERFACE

SU BROUTINE PPTRI(UPLO , N, A , [ $\mathbb{N F O}$ ])
CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::A
SU BROUTINE PPTRI_64 (UPLO,N,A, [NFO ])
CHARACTER (LEN=1)::UPLO
$\mathbb{N}$ TEGER (8) :: $\mathrm{N}, \mathbb{I N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::A
void dpptri(charuplo, intn, double *a, int *info);
void dpptri_64 (charuplo, long n, double *a, long *info);

## PURPOSE

dpptricom putes the inverse of a real symm etric positive definite $m$ atrix $A$ using the Cholesky factorization $A=$ $\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ orA $=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ com puted by SPPTRF.

## ARGUMENTS

UPLO (input)
$=U^{\prime}:$ U pper triangular factor is stored in $A$;
$=L^{\prime}:$ Low er triangular factor is stored in A.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
O n entry, the triangular factor $U$ or $L$ from the Cholesky factorization $A=U * * T * U$ or $A=L * L * * T$, packed colum nw ise as a linear array. The jth colum $n$ ofU orL is stored in the amay A as fol low s: if UPLO $=U^{\prime}, A(i+(j 1) \star j 2)=U(i, 1)$ for $1<=i<=j$ if UPLO $=L$ ', A $(i+(j 1) *(2 n-j) / 2)$
$=L(i, j)$ for $\dot{j}=i<=n$.

O n exit, the upper or low er triangle of the (sym -
$m$ etric) inverse of $A$, overw riting the input factor U orL.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N F O}=-i$, the i-th argum enthad an illegalvałue
$>0:$ if $\mathbb{N} F O=i$, the $(i, i)$ elem entof the factor
U orL is zero, and the inverse could notbe com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dpptrs-solve a system of linearequations $A * X=B$ w th a sym $m$ etric positive definite $m$ atrix $A$ in packed storage using the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ orA $=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ computed by SPPTRF

## SYNOPSIS

```
SUBROUT\mathbb{NE DPPTRS(UPLO,N,NRHS,A,B,LDB, INFO)}
```

CHARACTER * 1 UPLO
$\mathbb{N}$ TEGERN,NRHS,LDB, $\mathbb{N} F O$
DOUBLE PRECISION A (*), B (LDB, *)
SU BROUTINE DPPTRS_64 (UPLO ,N,NRHS,A,B,LDB, $\mathbb{N} F O$ )
CHARACTER * 1 UPLO
$\mathbb{N}$ TEGER*8N,NRHS,LDB, $\mathbb{N} F O$
DOUBLE PRECISION A (*), B (LDB, ${ }^{*}$ )

## F95 INTERFACE

SU BROUTINE PPTRS (UPLO,N, NRHS],A,B,[LDB],[NFO])

CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER :: N,NRHS,LDB, $\mathbb{N}$ FO
REAL (8),D $\mathbb{M}$ ENSION (:) ::A
REAL (8), D IM ENSIO N (:,:) ::B
SU BROUTINE PPTRS_64 (UPLO ,N, $\mathbb{N} R \mathrm{R}$ S], A, B, [LDB], [ $\mathbb{N} F \mathrm{FO}$ ])

CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER (8) :: N,NRHS,LDB, $\mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::A

REAL (8), D IM ENSION (:,:) ::B

## C INTERFACE

\#include <sunperfh>
void dpptes (charuplo, intn, intnrhs, double *a, double *b, int ldb, int *info);
void dpptrs_64 (charuplo, long n, long nrhs, double *a, double *b, long ldb, long *info);

## PURPOSE

dpptrs solves a system of linearequations A *X $=\mathrm{B}$ w ith a sym $m$ etric positive definite $m$ atrix A in packed storage using the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ com puted by SPPTRF.

## ARGUMENTS

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The orderof the m atrix A. $\mathrm{N}>=0$.

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of them atrix B. NRHS $>=0$.

A (input) The triangular factorU or L from the Cholesky
factorization $A=U * * T * U$ orA $=L * L * * T$, packed
colum nw ise in a linearanay. The jth collm $n$ of
U or L is stored in the array A as follow s: if UPLO = U', A $(i+(j-1) * j 2)=U(i, j)$ for $1<=i<=j$ if UPLO $=L \prime$, A $\left(i+(j-1)^{*}(2 n-j) / 2\right)=L(i, j)$ for j=i<=n。

B (input/output)
On entry, the righthand side m atrix B. On exit, the solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the array B . LD B >= $\max (1, N)$.
= 0: successfulexit
< 0: if $\mathbb{N}$ FO $=-$ i, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dptoon - com pute the reciprocal of the condition num ber (in the 1 -norm ) of a realsym $m$ etric positive definite tridiagonalm atrix using the factorization $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ or $\mathrm{A}=$ $\mathrm{U} * * \mathrm{~T} * \mathrm{D}$ * U com puted by SPTTRF

## SYNOPSIS

```
SU BROUTINE DPTCON \(\mathbb{N}, D \mathbb{I} G, O F F D, A N O R M, R C O N D, W\) ORK, \(\mathbb{N} F O\) )
```

$\mathbb{N}$ TEGER $N, \mathbb{N} F O$
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION DIAG (*),OFFD (*),WORK (*)
SUBROUTINEDPTCON_64 $\mathbb{N}, D \mathbb{I} G, O F F D, A N O R M, R C O N D, W O R K, \mathbb{N} F O$ )
$\mathbb{N}$ TEGER*8N, $\mathbb{N} F O$
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISIONDIAG (*), OFFD ( $\left.{ }^{( }\right)$, W ORK ( ${ }^{*}$ )

## F95 INTERFACE

SUBROUTINE PTCON ( $\mathbb{N}], D \mathbb{I} G, O F F D, A N O R M, R C O N D,[W O R K],[\mathbb{N} F O])$
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
REAL (8) ::ANORM,RCOND
REAL (8),D $\mathbb{M}$ ENSION (:) ::D $\mathbb{A} G, O F F D, W$ ORK
SU BROUTINE PTCON_64 ( $\mathbb{N}], D \mathbb{I} G, O F F D, A N O R M, R C O N D,[W O R K],[\mathbb{N} F O])$
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{I N} F \mathrm{O}$
REAL (8) ::ANORM,RCOND
REAL (8),D $\mathbb{I}$ ENSION (:) ::D IA G,OFFD,W ORK

## C INTERFACE

\#include < sunperfh>
void dptcon (intn, double *diag, double *offd, double anorm , double *rcond, int *info);
void dptcon_64 (long n, double *diag, double *offd, double anorm , double *rcond, long *info);

## PURPOSE

dptoon com putes the reciprocal of the condition num ber (in the 1 -norm ) of a realsym $m$ etric positive definite tridiagonalm atrix using the factorization $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ or $\mathrm{A}=$ $\mathrm{U} * * \mathrm{~T} * \mathrm{D}$ * U com puted by SPTTRF.
Norm (inv (A)) is com puted by a direct method, and the reciprocal of the condition num ber is com puted as

$$
\text { RCOND }=1 /(\operatorname{ANORM} \text { * nom (inv (A))). }
$$

## ARGUMENTS

N (input) The order of the matrix $A . N>=0$.

D IA G (input)
Then diagonalelem ents of the diagonal $m$ atrix D IA G from the factorization of $A$, as com puted by SPTTRF.

OFFD (input)
The ( $n-1$ ) off-diagonalelem ents of the unit bidiagonal factorU orL from the factorization ofA, as com puted by SPTTRF .

## ANORM (input)

The 1-norm of the originalm atrix A.

## RCOND (output)

The reciprocal of the condition number of the $m$ atrix $A$, com puted as RCOND $=1 /(A N O R M * A \mathbb{N} V N M)$, where $A \mathbb{N} V N M$ is the 1 -nom of inv ( $A$ ) com puted in this routine.

W ORK (w orkspace)
dim ension $(\mathbb{N})$
$=0$ : successfulexit
< 0: if $\mathbb{N}$ FO = -i, the i-th argum enthad an illegalvalue

## FURTHER DETAILS

Them ethod used is described in N icholas J . H igham, "E fficient A lgorithm s for C om puting the C ondition N um berofa TridiagonalM atrix", SIA M J. Sci.Stat. C om put., V ol. 7, No. 1, January 1986.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dpteqr - com pute alleigenvalues and, optionally, eigenvectors of a sym $m$ etric positive definite tridiagonalm atrix by first factoring the $m$ atrix using SPTTRF, and then calling SBD SQR to com pute the singular values of the bidiagonal factor

## SYNOPSIS

```
SUBROUT\mathbb{NE DPTEQR(COMPZ,N,D,E,Z,LDZ,WORK,INFO)}
CHARACTER * 1 COM PZ
NNTEGERN,LDZ,INFO
DOUBLE PRECISION D (*),E (*),Z (LD Z,*),W ORK (*)
SUBROUT\mathbb{NEDPTEQR_64(COMPZ,N,D,E,Z,LD Z,W ORK,INFO)}
CHARACTER * 1 COM PZ
INTEGER*8N,LDZ,INFO
DOUBLE PRECISIOND (*),E (*),Z (LD Z ,*),W ORK (*)
```


## F95 INTERFACE

```
SU BROUTINE PTEQR (COMPZ, \(\mathbb{N}], D, E, Z,[L D Z],[W O R K],[\mathbb{N} F O])\)
CHARACTER (LEN=1) ::COM PZ
\(\mathbb{N} T E G E R:: N, L D Z, \mathbb{N F O}\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK
REAL (8),D \(\mathbb{I}\) ENSION (:,:) :: Z
SU BROUTINE PTEQR_64 (COMPZ, \(\mathbb{N}], D, E, Z,[L D Z],[W O R K],[\mathbb{N} F O])\)
```

CHARACTER (LEN=1) ::COM PZ
$\mathbb{N}$ TEGER (8) :: N, LD Z, $\mathbb{N} F O$

REAL (8), D IM ENSION (:) ::D ,E,W ORK
REAL (8),D IM ENSION (:,:) ::Z

## C INTERFACE

\#include <sunperfh>
void dpteqr(char com pz, intn, double *d, double *e, double * $z$, intldz, int *info);
void dpteqr_64 (charcom pz, long n, double *d, double *e, double *z, long ldz, long *info);

## PURPOSE

dpteqr computes all eigenvalues and, optionally, eigenvectors of a symm etric positive definite tridiagonal $m$ atrix by first factoring the $m$ atrix using SPTTRF, and then calling SBD SQR to com pute the singularvalues of the bidiagonal factor.

This routine com putes the eigenvalues of the positive definite tridiagonal m atrix to high relative accuracy. This $m$ eans that if the eigenvalues range overm any orders ofm agnitude in size, then the sm alleigenvalues and corresponding eigenvectorsw illlbe com puted m ore accurately than, for exam ple, w th the standard $Q R$ m ethod.

The eigenvectors of a fullorband sym $m$ etric positive definthe $m$ atrix can also be found ifSSY TRD, SSPTRD, orSSBTRD has been used to reduce this $m$ atrix to tridiagonal form . (The reduction to tridiagonal form, how ever, $m$ ay preclude the possibility of obtaining high relative accuracy in the sm all eigenvalues of the originalm atrix, if these eigenvalues range overm any orders ofm agnitude.)

## ARGUMENTS

COMPZ (input)
$=\mathrm{N}^{\prime}:$ C om pute eigenvalues only.
$=\mathrm{V}$ ': C om pute eigenvectors of original sym $m$ etric
m atrix also. A ray Z contains the orthogonal $m$ atrix used to reduce the originalm atrix to tridiagonal form . = I': C om pute eigenvectors of tridiagonalm atrix also.

N (input) The order of the m atrix. $\mathrm{N}>=0$.

D (input/output)
O n entry, the n diagonalelem ents of the tridiagonalm atrix. On norm alexit, $D$ contains the eigenvalues, in descending order.

E (input/output)
O $n$ entry, the ( $n-1$ ) subdiagonal elem ents of the tridiagonal matrix. On exit, E hasbeen destroyed.

Z (input) On entry, if $\mathrm{COMPZ}=\mathrm{V}$ ', the orthogonal m atrix used in the reduction to tridiagonal form. On exit, if $C O M P Z=V$ ', the orthonorm aleigenvectors of the original sym $m$ etric $m$ atrix; if $C O M P Z=I$ ', the orthonorm aleigenvectors of the tridiagonal $m$ atrix. If $\mathbb{N} F O>0$ on exit, $Z$ contains the eigenvectors associated with only the stored eigenvalues. If $C O M P Z=N$ ', then $Z$ is not referenced.

LD Z (input)
The leading din ension of the array $\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1$, and ifCOM PZ $=V$ 'or $I$ ', $L D Z>=m a x(1, N)$.

W ORK (w orkspace)
dim ension (4*N )
$\mathbb{N} F O$ (output)
= 0: successfulexit.
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum enthad an illegalvahue.
> 0: if $\mathbb{N} F O=i$, and $i$ is: <= $N$ the Cholesky factorization of the $m$ atrix could notbe perform ed because the $i$-th principalm inorw as not positive definite. > N the SVD algorithm failed to converge; if $\mathbb{N} F O=\mathrm{N}+$ i, ioff-diagonal elem ents of the bidiagonal factor did not converge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dptrfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is sym $m$ etric positive definite and tridiagonal, and provides error bounds and backw ard errorestim ates for the solution

## SYNOPSIS

```
SUBROUT\mathbb{NE DPTRFS N,NRHS,D IAG,OFFD,D IA GF,OFFDF,B,LD B,X,LDX,}
    FERR,BERR,W ORK, \mathbb{NFO)}
\mathbb{NTEGER N,NRHS,LDB,LDX, NNFO}
DOUBLE PRECISION DIAG (*),OFFD (*), DIAGF (*), OFFDF (*),
B([LDB,*),X (LDX , *),FERR (*),BERR (*),WORK (*)
SUBROUT\mathbb{NE DPTRFS_64 N,NRHS,DIAG,OFFD,D IA GF,OFFDF,B,LD B ,X,}
    LD X ,FERR,BERR,WORK, NNFO)
\(\mathbb{N}\) TEGER*8N,NRHS,LDB,LDX, \(\mathbb{N}\) FO
DOUBLE PRECISION DIAG (*), OFFD (*), DIAGF (*), OFFDF (*) , \(\mathrm{B}(\mathrm{LDB}, \star), \mathrm{X}(\mathrm{LDX}, \star), \operatorname{FERR}(\star), \operatorname{BERR}(*), \mathrm{W} O R K(*)\)
```


## F95 INTERFACE

SU BROUT $\mathbb{N} E \operatorname{PTRFS}(\mathbb{N}],[N R H S], D \mathbb{A} G, O F F D, D \mathbb{A} G F, O F F D F, B,[L D B], X$, [LDX],FERR,BERR, [WORK],[ $\mathbb{N} F O]$ )
$\mathbb{N}$ TEGER :: N, NRHS,LDB,LDX, $\mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (:) :: D IA G, OFFD, D IAGF, OFFDF, FERR, BERR, W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) :: B, X
SU BROUTINEPTRFS_64 (N), NRHS],D IA G,OFFD,D IA GF,OFFDF,B,[LDB], $\mathrm{X},[\mathrm{LD} \mathrm{X}], \mathrm{FERR}, \mathrm{BERR},[\mathrm{W} O R K],[\mathbb{N F O}])$
$\mathbb{N}$ TEGER (8) :: N , NRHS,LDB,LDX, $\mathbb{N}$ FO
REAL (8), D $\mathbb{M} E N S I O N(:):: D \mathbb{I} G, O F F D, D \mathbb{A G F}, \mathrm{OFFDF}, \mathrm{FERR}$, BERR, W ORK
REAL (8), D $\mathbb{M} E N S I O N(:,:): B, X$

## C INTERFACE

\#include <sunperfh>
void dptrfs (intn, intnrhs, double *diag, double *offd, double *diagf, double *offfdf, double *b, int ldl, double *x, intldx, double *ferr, double *berr, int*info);
void dptrfs_64 long n, long nrhs, double *diag, double *offd, double *diagf, double *offfdf, double *b, long ldb, double *x, long ldx, double * ferr, double *berr, long *info);

## PURPOSE

dptrfs im proves the com puted solution to a system of linear equations $w$ hen the coefficientm atrix is sym $m$ etric positive definite and tridiagonal, and provides error bounds and backw ard errorestim ates for the solution.

## ARGUMENTS

N (input) The order of them atrix A. $\mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS $>=0$.

D IA G (input)
The n diagonalelem ents of the tridiagonal matrix
A.

OFFD (input)
The ( $n-1$ ) subdiagonalelem ents of the tridiagonal $m$ atrix $A$.

D IA GF (input)
The $n$ diagonalelem ents of the diagonal $m$ atrix
D IA G from the factorization com puted by SPTTRF .

OFFDF (input)
The ( $n-1$ ) subdiagonalelem ents of the unitbidiag-
onal factor $L$ from the factorization com puted by SPTTRF.
$B$ (input) The righthand side $m$ atrix $B$.
LD B (input)
The leading dim ension of the array $B$. LD B >= $\max (1, N)$.
$X$ (input/output)
On entry, the solution $m$ atrix $X$, as com puted by SPTTRS. On exit, the im proved solution $m$ atrix $X$.

## LD X (input)

The leading dim ension of the array X . LD X >= $\max (1, N)$.

## FERR (output)

The forw ard enrorbound foreach solution vector $X$ ( $\mathcal{I}$ ) (the $j$ th collm $n$ of the solution $m$ atrix $X$ ). IfXTRUE is the true solution corresponding to $X(\mathcal{j})$, FERR ( $\mathcal{I}$ ) is an estim ated upperbound for the $m$ agnitude of the largestelem ent in ( $X(\mathcal{J}$-X TRUE) divided by the $m$ agnitude of the largestelem ent in $\mathrm{X}(\mathrm{j})$.

BERR (output)
The com ponentw ise relative backw ard emorof each
solution vector $X(\mathcal{j})$ (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( $\mathcal{J}$ ) an exactsolution).

W ORK (w orkspace)
dim ension $(2 * N)$
$\mathbb{N} F O$ (output)
= 0: successfulexit
<0: if $\mathbb{I N}$ FO $=-$ i, the $i$ th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dptsv -com pute the solution to a real system of linear equations $A * X=B$, where $A$ is an $N$ by $N$ symm etric positive definite tridiagonalm atrix, and X and B are N -by-NRHS m atrices.

## SYNOPSIS

```
SUBROUT\mathbb{NEDPTSV N,NRHS,D IAG,SUB,B,LDB,INFO)}
\mathbb{NTEGERN,NRHS,LDB,INFO}
DOUBLE PRECISION D IA G (*),SUB (*),B (LDB,*)
SU BROUT\mathbb{NE DPTSV_64 N,NRHS,D IAG,SUB,B,LDB,INFO)}
\mathbb{NTEGER*8N,NRHS,LDB,INFO}
DOUBLE PRECISION DIAG (*),SUB (*),B (LDB,*)
```

F95 INTERFACE
SUBROUTINE PTSV ( $\mathbb{N}], \mathbb{N} R H S], D \mathbb{A} G, S U B, B,[L D B],[\mathbb{N} F O])$
$\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::D $\mathbb{A} G, S U B$
REAL (8),D $\mathbb{I}$ ENSION (:,:) ::B
SUBROUTINEPTSV_64 ( $\mathbb{N}], \mathbb{N} R H S], D \mathbb{A} G, S U B, B,[L D B],[\mathbb{N} F O])$
$\mathbb{N}$ TEGER (8) ::N,NRHS,LDB, $\mathbb{N}$ FO
REAL (8), D $\mathbb{M}$ ENSION (:) ::D $\mathbb{A} G$, SUB
REAL (8),D IM ENSION (:,:) ::B
C INTERFACE
\#include <sunperfh>
void dptsv (intn, intnrhs, double *diag, double *sub, double *b, int ldb, int *info);
void dptsv_64 (long n, long nihs, double *diag, double *sub, double *b, long ldb, long *info);

## PURPOSE

dptsv com putes the solution to a realsystem of linear equations $A * X=B$, where $A$ is an $N$ boy $-N$ sym m etric posilive definite tridiagonalm atrix, and $X$ and $B$ are $N$ by NRHS m atrices.
$A$ is factored as $A=L * D * L * * T$, and the factored form of $A$ is then used to solve the system of equations.

## ARGUMENTS

N (input) The order of them atrix $A . N>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS $>=0$.

D IA G (input/output)
O n entry, the $n$ diagonalelem ents of the tridiago-
nalm atrix A. On exit, the $n$ diagonalelem ents of the diagonalm atrix D IA G from the factorization $A$ $=\mathrm{L} * \mathrm{D} \mathbb{I} \mathrm{G} * \mathrm{~L} * * \mathrm{~T}$.

SU B (input/output)
O $n$ entry, the $(n-1)$ subdiagonal elem ents of the tridiagonalm atrix A. On exit, the $(n-1)$ subdiagonalelem ents of the unitbidiagonal factorL from the L *D IA G *L **T factorization ofA. (SU B can also be regarded as the superdiagonal of the unitbidiagonal factor $U$ from the $U * * T * D I A G * U$ factorization ofA.)

B (input/output)
O n entry, the N boy-N RH S righthand side m atrix B. On exit, if $\mathbb{N F O}=0$, the N boy-NRH S solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the array B. LD B >=
$\max (1, N)$.
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the i-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N F O}=i$, the leading $m$ inoroforder $i$ is not positive definite, and the solution has not
been com puted. The factorization has not been com pleted unless $i=N$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dptsvx -use the factorization $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ to com pute the solution to a realsystem of linearequations $A * X=B$, where $A$ is an $N$ by $-N$ symm etric positive definite tridiagonal $m$ atrix and $X$ and $B$ are $N$-by $-N$ R H S $m$ atrices

## SYNOPSIS

```
SU BROUTINE DPTSVX (FACT,N,NRHS,DIAG,SUB,D IAGF,SUBF,B,LDB,X,
    LDX,RCOND,FERR,BERR,W ORK, INFO )
CHARACTER*1FACT
\mathbb{N TEGER N,NRHS,LDB,LDX, IN FO}
DOUBLE PRECISION RCOND
DOUBLE PRECISION DIAG (*), SUB (*), D IAGF (*), SUBF (*),
B (LD B ,*),X (LDX **), FERR (*), BERR (*),W ORK (*)
SUBROUT\mathbb{NE DPTSVX_64(FACT,N,NRHS,DIAG,SUB,DIAGF,SUBF,B,LDB,}
    X,LDX,RCOND,FERR,BERR,W ORK, IN FO)
CHARACTER*1FACT
INTEGER*8N,NRHS,LDB,LDX, IN FO
DOUBLE PRECISION RCOND
DOUBLE PRECISION DIAG (*), SUB (*), DIAGF (*), SUBF (*),
B (LD B ,*),X (LDX ,*),FERR (*),BERR (*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINEPTSVX $\mathbb{E A C T}, \mathbb{N}], \mathbb{N} R H S], D \mathbb{A} G, S U B, D \mathbb{A} G F, S U B F, B,[L D B]$, $\mathrm{X},[\operatorname{LD} X], R C O N D, F E R R, B E R R,[\mathbb{O} \mathrm{ORK}],[\mathbb{N} F O])$

CHARACTER (LEN=1)::FACT
$\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N} F O$
REAL (8) :: RCOND

REAL (8),D $\mathbb{M} \operatorname{ENSION}(:):: D \mathbb{I A} G, S U B, D \mathbb{A} G F, S U B F, F E R R, B E R R$,
W ORK
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::B,X
 [LDB],X, [LDX],RCOND,FERR,BERR, [WORK], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1) ::FACT
$\mathbb{N} T E G E R(8):: N, N R H S, L D B, L D X, \mathbb{N} F O$
REAL (8) ::RCOND
REAL (8),D $\mathbb{M}$ ENSION (:) ::D IA G, SUB,D $\mathbb{A} G F, S U B F, F E R R, B E R R$,
W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) ::B,X

## C INTERFACE

\#include < sunperfh>
void dptsvx (char fact, intn, int nirhs, double *diag, double *sub, double *diagf, double *subf, double *b, int ldb, double *x, int ldx, double *rcond, double * ferrs, double *berr, int *info);
void dptsvx_64 (char fact, long n, long nrhs, double *diag, double *sub, double *diagf, double *subf, double *b, long lalb, double *x, long ldx, double *rcond, double *ferr, double *berr, long *info);

## PURPOSE

dptsvx uses the factorization $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ to com pute the solution to a real system of linear equations $A * X=B$, where $A$ is an $N-b y-N$ symmetric positive definite tridiagonal $m$ atrix and $X$ and $B$ are $N$ by-N RH S $m$ atrices.

E rrorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:

1. IfFACT $=N$ ', the $m$ atrix $A$ is factored as $A=L * D * L * * T$, where L
is a unit low erbidiagonalm atrix and $D$ is diagonal. The factorization can also be regarded as having the form
$\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{D} * \mathrm{U}$.
2. If the leading i-by-iprincipal m inor is not positive definite,
then the routine retums w ith $\mathbb{N} F O=$ i. O therw ise, the factored
form ofA is used to estim ate the condition num ber of the
m atrix
A. If the reciprocal of the condition num ber is less than $m$ achine
precision, $\mathbb{N}$ FO $=\mathrm{N}+1$ is retumed as a w aming, but the routine
still goes on to solve for X and com pute emorbounds as described below .
3.The system ofequations is solved forX using the factored form of A.
3. Iterative refinem ent is applied to im prove the com puted solution
$m$ atrix and calculate error bounds and backw ard error estim ates
for it.

## ARGUMENTS

FACT (input)
Specifies w hether ornot the factored form of $A$ has been supplied on entry. = F': On entry, D IA GF and SUBF contain the factored form of A.
D IA G , SUB, D IA G F , and SUBF w illnotbe m odified.
$=N$ ': Them atrix A w illbe copied to D IA GF and SUBF and factored.

N (input) The order of them atrix $A . N>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber
of colum ns of the $m$ atriges $B$ and X. NRHS $>=0$.

D IA G (input)
The $n$ diagonalelem ents of the tridiagonal matrix A.

SU B (input)
The ( $n-1$ ) subdiagonalelem ents of the tridiagonal $m$ atrix A.

D IA GF (input/output)
IfFACT = $\mathrm{F}^{\prime}$, then $D \mathrm{IAGF}$ is an inputargum entand on entry contains the $n$ diagonalelem ents of the diagonalm atrix D IA G from the $L$ *D IA G *L **T factorization ofA. IfFACT $=N$ ', then D IA GF is an outputargum entand on exitcontains the $n$ diagonal
elem ents of the diagonal $m$ atrix $D \mathbb{I A}$ from the $\mathrm{L} * \mathrm{D}$ IA $G * \mathrm{~L} * *$ T factorization ofA.

## SU BF (input/output)

IfFACT $=\mathrm{F}^{\prime}$, then SUBF is an inputargum ent and on entry contains the ( $n-1$ ) subdiagonalelem ents of the unit bidiagonal factor $L$ from the $\mathrm{L} * \mathrm{D} \mathbb{I A}{ }^{*} \mathrm{~L} * * \mathrm{~T}$ factorization of A . If $\mathrm{FACT}=\mathrm{N}^{\prime}$, then SUBF is an outputargum entand on exit contains the ( $n-1$ ) subdiagonalelem ents of the unit bidiagonal factorL from the $L * D \operatorname{IAG} * L * * T$ factorization ofA.
$B$ (input) The N by -N RH S righthand side $m$ atrix $B$.
LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.

X (output)
If $\mathbb{N} F O=0$ of $\mathbb{N} F O=N+1$, the N -by-NRHS solution $m$ atrix $X$.

LD X (input)
The leading dim ension of the amay X . LD X >= $\max (1, N)$.

## RCOND (output)

The reciprocalcondition num berof the $m$ atrix $A$. If RCOND is less than the $m$ achine precision (in particular, ifRCOND $=0$ ), the $m$ atrix is singular to working precision. This condition is indicated by a retum code of $\mathbb{N}$ FO $>0$.

FERR (output)
The forw ard emorbound foreach solution vector $X$ (j) (the $j$ th colum $n$ of the solution $m$ atrix $X$ ). IfXTRUE is the true solution comesponding to $X(\mathcal{j})$, FERR $(\mathcal{j})$ is an estim ated upperbound for the $m$ agnitude of the largestelem ent in (X ( $)$-X TRU E) divided by the $m$ agnitude of the largestelem ent in $\mathrm{X}(\underset{)}{ }$.

BERR (output)
The com ponentw ise relative backw ard emorof each solution vector $X(\mathcal{j})$ (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( $\mathcal{j}$ ) an exactsolution).

W ORK (w orkspace)
dim ension $(2 * N)$
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N F O}=-$ i, the i-th argum enthad an illegalvalue
$>0$ : if $\mathbb{N} F O=i$, and $i$ is
$<=N$ : the leading $m$ inor oforderiof $A$ is not positive definite, so the factorization could not
be com pleted, and the solution has not been com puted. RCOND $=0$ is retumed. $=\mathrm{N}+1: \mathrm{U}$ is nonsingular, butRCOND is less than machine precision, $m$ eaning that the $m$ atrix is singular to w orking precision. Nevertheless, the solution and error bounds are com puted because there are a num ber of situations w here the com puted solution can bem ore accurate than the value ofRC OND w ould suggest.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dpttrf-com pute the L *D *L 'factorization of a real sym m etric positive definite tridiagonalm atrix A

## SYNOPSIS

SU BROUTINEDPTTRF $\mathbb{N}, D \mathbb{I} G, O F F D, \mathbb{N} F O$ )
$\mathbb{N}$ TEGER $N, \mathbb{N} F O$
DOUBLE PRECISION DIAG (*),OFFD (*)

SU BROUTINE DPTTRF_64 $\mathbb{N}, D \mathbb{I A}, \quad$ OFFD , $\mathbb{N} F O$ )
$\mathbb{N} T E G E R * 8 N, \mathbb{N} F O$
DOUBLE PRECISION D IAG (*), OFFD (*)

## F95 INTERFACE

SU BROUTINE PTTRF ( $\mathbb{N}], D \mathbb{I} G, O F F D,[\mathbb{N} F O])$
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::D IA G , OFFD
SU BROUTINE PTTRF_64 ( $\mathbb{N}], D \mathbb{I A}$, OFFD, $[\mathbb{N} F O]$ )
$\mathbb{N}$ TEGER (8) :: $\mathrm{N}, \mathbb{I N} F \mathrm{O}$
REAL (8),D $\mathbb{M}$ ENSION (:) ::D IA G , OFFD

## C INTERFACE

\#include <sunperfh>
void dpturf(intn, double *diag, double *offd, int*info);
void dpttrf_ 64 (long n, double *diag, double *offd, long

## PURPOSE

dpttrf com putes the $L * D$ *L 'factorization of a realsym $m$ etric positive definite tridiagonalm atrix A. The factorization $m$ ay also be regarded as having the form $A=U * D * U$.

## ARGUMENTS

N (input) The order of the m atrix A. $\mathrm{N}>=0$.
D IA G (input/output)
O n entry, the $n$ diagonalelem ents of the tridiagonalm atrix A. On exit, the $n$ diagonalelem ents of the diagonalm atrix D IA G from the $L * D \mathbb{I A G}{ }^{\prime}{ }^{\prime}$ factorization of A.

OFFD (input/output)
O $n$ entry, the $(n-1)$ subdiagonal elem ents of the tridiagonalm atrix $A$. On exit, the $(n-1)$ subdiagonalelem ents of the unit.bidiagonal factorL from the $L * D I A G * L$ ' factorization ofA. OFFD can also be regarded as the superdiagonal of the unitbidiagonal factor $U$ from the $U$ *D IA G *U factorization ofA.
$\mathbb{I N} F O$ (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N} F O=-\mathrm{k}$, the k -th argum enthad an illegalvalue
$>0$ : if $\mathbb{N} F O=k$, the leading $m$ inoroforderk is notpositive definite; if $k<N$, the factorization could notbe com pleted, while if $\mathrm{k}=\mathrm{N}$, the factorization w as com pleted, butD $\mathbb{I A} G \mathbb{N})=0$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dptters-solve a tridiagonalsystem of the form $A * X=B$ using the $L * D * L$ 'factorization of $A$ com puted by SPTTRF

## SYNOPSIS

```
SUBROUT\mathbb{NEDPTTRS N,NRHS,D IAG,OFFD,B,LDB, NNFO)}
\mathbb{NTEGER N,NRHS,LDB,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
DOUBLE PRECISIONDIAG(*),OFFD (*),B (LDB,*)
SUBROUT\mathbb{NE DPTTRS_64 N,NRHS,D IAG,OFFD,B,LDB,INFO )}
\mathbb{N}TEGER*8N,NRHS,LDB,NNFO
DOUBLE PRECISION DIAG (*),OFFD (*),B (LDB,*)
```


## F95 INTERFACE

```
SU BROUTINE PTTRS ( \(\mathbb{N}], \mathbb{N} R H S], D \mathbb{A} G, O F F D, B,[L D B],[\mathbb{N} F O])\)
\(\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::D IA G , OFFD
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::B
SU BROUTINEPTTRS_64 ( \(\mathbb{N}], \mathbb{N} R H S], D \mathbb{I} G, O F F D, B,[L D B],[\mathbb{N} F O])\)
\(\mathbb{N} T E G E R(8):: N, N R H S, L D B, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::D \(\mathbb{I A} G, O F F D\)
REAL (8), D \(\mathbb{M}\) ENSIO N (:,:) ::B
```


## C INTERFACE

```
\#include < sunperfh>
void dpttrs (intn, intnihs, double *diag, double *offd,
```

double *b, int ldb, int*info);
void dpttrs_64 (long n, long nrhs, double *diag, double *offd, double *b, long ldb, long *info);

## PURPOSE

dpters solves a tridiagonal system of the form
A * $\mathrm{X}=\mathrm{B}$ using the $\mathrm{L} * \mathrm{D}$ *L 'factorization of A com puted by SPTTRF. $D$ is a diagonalm atrix specified in the vectorD, $L$ is a unitbidiagonalm atrix $w$ hose subdiagonal is specified in the vector $E$, and $X$ and $B$ are $N$ by NRH S $m$ atrices.

## ARGUMENTS

N (input) The order of the tridiagonalm atrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the m atrix B. NRHS $>=0$.

D IA G (input)
Then diagonalelem ents of the diagonal matrix
D IA G from the $L * D \mathbb{I A}$ * $L$ 'factorization of $A$.

OFFD (input/output)
The ( $n-1$ ) subdiagonalelem ents of the unitbidiagonal factor L from the $L$ *D IA G *L 'factorization of A. OFFD can also be regarded as the superdiagonal of the unitbidiagonal factor $U$ from the factorization $A=U$ *D $\mathbb{A} G * U$.

B (input/output)
On entry, the righthand side vectors B for the system of linearequations. On exit, the solution vectors, $X$.

LD B (input)
The leading dim ension of the array $B$. LD B >= $\max (1, N)$.
$\mathbb{N} F O$ (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-\mathrm{k}$, the k -th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dptts2 -solve a tridiagonal system of the form $A * X=B$ using the $L * D * L$ 'factorization of $A$ com puted by SPTTRF

## SYNOPSIS

SUBROUTINEDPTTS2 $\mathbb{N}, N R H S, D, E, B, L D B)$
$\mathbb{I N} T E G E R N, N R H S, L D B$
DOUBLE PRECISIOND (*) , E (*) , B (LDB,*)
SU BROUTINE DPTTS2_64 $\mathbb{N}, N R H S, D, E, B, L D B)$
$\mathbb{N}$ TEGER*8N,NRHS,LDB
DOUBLE PRECISION D (*), E (*), B (LDB,*)

## F95 INTERFACE

SU BROUTINE DPTTS2 $\mathbb{N}, N R H S, D, E, B, L D B)$
$\mathbb{N} T E G E R:: N, N R H S, L D B$
REAL (8), D IM ENSION (:) :: D , E
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::B
SU BROUTINEDPTTS2_64 $\mathbb{N}, N R H S, D, E, B, L D B)$
$\mathbb{N} T E G E R(8):: N, N R H S, L D B$
REAL (8), D IM ENSION (:) ::D , E
REAL (8),D IM ENSIO N (:,:) ::B

## C INTERFACE

\#include <sunperfh>
void dptts2 (intn, intnrhs, double *d, double *e, double
void dptts2_64 (long n, long nihs, double *d, double *e, double *b, long ldb);

## PURPOSE

dptts2 solves a tridiagonal system of the form
A * $\mathrm{X}=\mathrm{B}$ using the L *D *L 'factorization of A com puted by SPTTRF. D is a diagonalm atrix specified in the vectorD, $L$ is a unitbidiagonalm atrix $w$ hose subdiagonal is specified in the vectorE, and X and B are N by N RH S m atrioes.

## ARGUMENTS

N (input) The order of the tridiagonalm atrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the m atrix B. NRHS $>=0$.
$D$ (input) The $n$ diagonalelem ents of the diagonal $m$ atrix $D$ from the $L * D$ ${ }^{*}$ 'factorization ofA.

E (input) The ( $\mathrm{n}-1$ ) subdiagonalelem ents of the unitbidiagonal factor $L$ from the $L * D * L$ 'factorization of $A$. E can also be regarded as the superdiagonal of the unit bidiagonal factor $U$ from the factorization $A$ $=U * D * U$.

B (input/output)
O $n$ entry, the righthand side vectors $B$ for the system of linearequations. On exit, the solution vectors, X .

LD B (input)
The leading dim ension of the anay $B$. LD B $>=$ $\max (1, N)$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
dqdota - compute a double precision constant plus an
```

extended precision constantplus the extended precision dot
product of tw o double precision vectors x and y .

## SYNOPSIS

DOUBLE PRECISION FUNCTION DQDOTA $\mathbb{N}, D B, Q C, D X, \mathbb{N C X}, D Y, \mathbb{N} C Y)$
$\mathbb{N}$ TEGER $N, \mathbb{I N C X}, \mathbb{N} C Y$
REAL * 16 QC
DOUBLE PRECISION DB DOUBLE PRECISION DX (*), DY (*)

DOUBLE PRECISION FUNCTION DQDOTA_64 $\mathbb{N}, D B, Q C, D X, \mathbb{N} C X, D Y, \mathbb{N} C Y$ )
$\mathbb{N} T E G E R * 8 N, \mathbb{N} C X, \mathbb{N} C Y$
REAL * 16 Q C
DOUBLE PRECISION DB
DOUBLE PRECISION DX (*), DY (*)

## F95 INTERFACE

REAL (8) FUNCTION DQDOTA $\mathbb{N}, \mathrm{DB}, Q \mathrm{C}, \mathrm{DX}, \mathbb{N} C X, D Y, \mathbb{N C Y})$
$\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y$
REAL (16) :: Q C
REAL (8) ::DB
REAL (8),D $\mathbb{M}$ ENSION (:) ::DX,DY
REAL (8) FUNCTION DQDOTA_64 $\mathbb{N}, \mathrm{DB}, Q \mathrm{C}, \mathrm{DX}, \mathbb{N C X}, \mathrm{DY}, \mathbb{N} C Y)$
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{I N C X}, \mathbb{N} C Y$
REAL (16) :: QC

REAL (8) :: DB
REAL (8),D $\mathbb{M}$ ENSION (:) ::DX,DY

## C INTERFACE

\#include <sunperfh>
double dqdota (intn, double db, long double *qc, double *dx, int incx, double *dy, intincy);
double dqdota_64 (long n, double db, long double *qc, double *dx, long incx, double *dy, long incy);

## PURPOSE

dqdota com pute a double precision constantplus an extended precision constant plus the extended precision dotproduct oftw o double precision vectors $x$ and $y$.

## ARGUMENTS

N (input)
O n entry, $N$ specifies the num ber of elem ents in the vector. If $\mathrm{N}<=0$ then the function retums the value D B +Q C. U nchanged on exit.

D B (input)
O n entry, the constant that is added to the dot product before the result is retumed. U nchanged on exit.

QC (input/output)
O n entry, the extended precision constant to be added to the dotproduct. On exit, the extended precision result.

DX (input)
O n entry, the increm ented array D X m ust contain the vectorx. U nchanged on exit.
$\mathbb{N} C X$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents ofD $\mathrm{X} . \mathbb{N} C X$ m ustnotbe zero. U nchanged on exit.

DY (input)
On entry, the increm ented anay DY m ust contain the vectory. U nchanged on exit.
$\mathbb{N} C Y$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents ofD Y. $\mathbb{N}$ CY m ustnotbe zero. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dqdoti-com pute a constantplus the extended precision dot product oftw o double precision vectors x and y .

## SYNOPSIS

DOUBLE PRECISION FUNCTION DQDOTIN,DB,QC,DX, $\mathbb{N} C X, D Y, \mathbb{N C Y})$
$\mathbb{N} T E G E R N, \mathbb{N C X}, \mathbb{N} C Y$
REAL * 16 QC
DOUBLE PRECISION DB
DOUBLE PRECISION DX (*), DY (*)
DOUBLE PRECISION FUNCTION DQDOTI_64N,DB,QC,DX, $\mathbb{N} C X, D Y, \mathbb{N} C Y)$
$\mathbb{N} T E G E R * 8 N, \mathbb{N} C X, \mathbb{N} C Y$
REAL * 16 QC
DOUBLE PRECISION DB
DOUBLE PRECISION DX (*),DY (*)

## F95 INTERFACE

REAL (8) FUNCTION DQDOTIN,DB,QC,DX, $\mathbb{N} C X, D Y, \mathbb{N} C Y)$
$\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y$
REAL (16) :: Q C
REAL (8) ::DB
REAL (8),D $\mathbb{M}$ ENSION (:) ::DX,DY
REAL (8) FUNCTION DQDOTI_64N,DB,QC,DX, $\mathbb{N C X}, \mathrm{DY}, \mathbb{N} C Y)$
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{I N C X}, \mathbb{N} C Y$
REAL (16) :: Q C
REAL (8) :: DB

## C INTERFACE

\#include <sunperfh>
double dqdoti(intn, double db, long double *qc, double *dx, intincx, double *dy, intincy);
double dqdoti_ 64 (long n, double db, long double *qC, double *dx, long incx, double *dy, long incy);

## PURPOSE

dqdoticom putes a constantplus the double precision dot product of $x$ and $y$ where $x$ and $y$ are double precision $n-$ vectors.

## ARGUMENTS

N (input)
O n entry, $N$ specifies the num ber of elem ents in the vector. If $\mathrm{N}<=0$ then the function retums the valueD B. U nchanged on exit.

D B (input)
O $n$ entry, the constant that is added to the dot product before the result is retumed. U nchanged on exit.

QC (output)
O n exit, the extended precision result.

D X (input)
O n entry, the increm ented array D X m ust contain the vectorx. U nchanged on exit.
$\mathbb{N} C X$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents ofD X. $\mathbb{N}$ CX m ustnotbe zero. Unchanged on exit.

D Y (input)
O n entry, the increm ented array D Y m ust contain the vectory. U nchanged on exit.
$\mathbb{N} C Y$ (input)
O n entry, $\mathbb{N N C Y}$ specifies the increm ent for the
elem ents ofD Y. $\mathbb{N} C Y$ m ustnotbe zero. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

drot-A pply a G iven's rotation constructed by SRO TG .

## SYNOPSIS



```
\mathbb{NTEGERN,}\mathbb{NCX,\mathbb{NCY}}\mathbf{}\mathrm{ CN}
DOUBLE PRECISION C,S
DOUBLE PRECISION X (*),Y(*)
SUBROUT\mathbb{NEDROT_64 N,X,NNCX,Y,\mathbb{NCY,C,S)}}\mathbf{N},\mp@code{C}
INTEGER*8N,\mathbb{NCX,INCY}
DOUBLE PRECISION C,S
DOUBLE PRECISION X (*),Y (*)
F95 INTERFACE
```



```
\mathbb{NTEGER ::N,\mathbb{NCX,}\mathbb{NCY}}\mathbf{N}={
REAL (8) ::C,S
REAL (8),D IM ENSION (:) ::X,Y
```



```
\mathbb{NTEGER (8)::N,\mathbb{NCX,INCY}}\mathbf{N}={
REAL (8) ::C,S
REAL (8),D IM ENSION (:) ::X,Y
```


## C INTERFACE

```
\#include <sunperfh>
```

void drot(intn, double * $x$, intincx, double * $y$, int incy, double c, double s);
void drot_64 (long n, double *x, long incx, double *y, long incy, double c, double s);

## PURPOSE

drot A pply a G iven s rotation constructed by SRO TG .

## ARGUMENTS

N (input)
O $n$ entry, $N$ specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.

X (input)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C X))$. Before entry, the increm ented array $X$ must contain the vectorx. U nchanged on exit.
$\mathbb{N} C X$ (input/output)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of $\mathrm{X} . \mathbb{N} C X$ m ustnotbe zero. U nchanged on exit.

Y (input/output)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))$. On entry, the increm ented array $Y$ m ust contain the vectory. On exit, $Y$ is overw ritten by the updated vectory.
$\mathbb{N} C Y$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of Y. $\mathbb{N} C Y$ m ustnotbe zero. U nchanged on exit.

C (input) On entry, the C rotation value constructed by SRO TG. U nchanged on exit.
$S$ (input) On entry, the $S$ rotation value constructed by SRO TG. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
drotg -C onstructa G iven 's plane rotation
```


## SYNOPSIS

$$
\text { SUBROUTINEDROTG }(A, B, C, S)
$$

DOUBLE PRECISIONA,B,C,S SUBROUTINEDROTG_64(A,B,C,S) DOUBLE PRECISION A, B, C, S

## F95 INTERFACE

SU BROUTINEROTG ( $\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{S}$ )
REAL (8) :: A, B, C, S
SU BROUTINEROTG_64(A,B,C,S)

REAL (8) :: A, B, C , S
C INTERFACE
\#include <sunperfh>
void drotg (double *a, double *b, double *c, double *s);
void drotg_64 (double *a, double *b, double *c, double *s);

## PURPOSE

## ARGUMENTS

A (input/output)
On entry, A contains the entry in the firstvector that comesponds to the elem ent to be annihilated in the second vector. On exit, contains the nonzero elem ent of the rotated vector.

B (input/output)
On entry, $B$ contains the entry to be annihilated in the second vector. On exit, contains eithers or 1/C depending on which of the inputvalues of A and $B$ is larger.

C (output)
On exit, $C$ and $S$ are the elem ents of the rotation $m$ atrix thatw illlibe applied to anninilate $B$.

S (output)
See the description of C .

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS


## NAME

droti- A pply an indexed G ívens rotation.

## SYNOPSIS

```
SUBROUTINE DROTINZ,X,NNDX,Y,C,S)
INTEGER NZ
INTEGER INDX (*)
DOUBLE PRECISION C,S
DOUBLE PRECISION X (*),Y (*)
SUBROUT\mathbb{NE DROTI_64 NZ,X,INDX,Y,C,S)}
INTEGER*8NZ
INTEGER*8 \mathbb{NDX (*)}
DOUBLE PRECISION C,S
DOUBLE PRECISION X (*),Y (*)
F95 IN TERFACE
SUBROUTINEROTI(NZ],X,\mathbb{NDX,Y,C,S)}
\mathbb{NTEGER ::NZ}
\mathbb{NTEGER,D IM ENSION (:)::\mathbb{NDX}}\mathbf{N}={
REAL (8)::C,S
REAL (8),D IM ENSION (:) ::X,Y
```



```
INTEGER (8)::N Z
\mathbb{NTEGER (8),D IM ENSION (:) ::\mathbb{NDX}}\mathbf{N}={
REAL (8) ::C,S
REAL (8),D IM ENSION (:) ::X,Y
```

D RO T I-A pplies a G ivens rotation to a sparse vector $x$ stored in com pressed form and anothervectory in full storage form

```
do \(i=1, n\)
    tem \(p=-s^{\star} x(i)+c^{\star} y\) (indx (i))
    \(x(i)=c^{*} x(i)+s^{*} y(i n d x(i))\)
    \(y(\) indx (i) \()=\) tem \(p\)
    enddo
```


## ARGUMENTS

N Z (input) - $\mathbb{N}$ TEGER
N um ber of elem ents in the com pressed form .
U nchanged on exit.
X (input)
V ector containing the values of the com pressed form .
$\mathbb{N} D \mathrm{X}$ (input) - $\mathbb{N}$ TEGER
V ector containing the indiges of the com pressed
form. It is assum ed that the elem ents in $\mathbb{N D}$ D are
distinctand greater than zero. U nchanged on exit.

Y (input/output)
V ector on inputw hich contains the vector $Y$ in full
storage form. O n exit, only the elem ents
corresponding to the indices in $\mathbb{N}$ D X have been
m odified.

C (input)
Scalardefining the G ivens rotation
$S$ (input)
Scalardefining the G ivens rotation

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

drotm -A pply a Gentlem an Sm odified G iven S rotation constructed by SRO TM G .

## SYNOPSIS

```
SUBROUT\mathbb{NE DROTM N,X,NNCX,Y,INCY,PARAM)}
\mathbb{NTEGERN,}\mathbb{NCX,\mathbb{NCY}}\mathbf{T}\mathrm{ \}
DOUBLE PRECISION X (*),Y(*),PARAM (*)
```



```
INTEGER*8N,\mathbb{NCX,INCY}
DOUBLE PRECISION X (*),Y (*),PARAM (*)
F95 INTERFACE
    SUBROUT\mathbb{NE ROTM (N ],X, [\mathbb{NCX ],Y, [NCY ],PARAM)}}\mathbf{N}\mathrm{ (NS}
    \mathbb{NTEGER::N,\mathbb{NCX,}\mathbb{NCY}}\mathbf{N}\mathrm{ \}
    REAL (8),D IM ENSION (:) ::X,Y,PARAM
```



```
    \mathbb{NTEGER (8)::N, \mathbb{NCX,}\mathbb{NCY}}\mathbf{M}\mathrm{ (%)}
    REAL (8),D IM ENSION (:) ::X,Y,PARAM
```


## C INTERFACE

```
\#include <sunperfh>
void drotm (intn, double *x, intincx, double *y, int incy, double *param );
```

void drotm _64 (long n, double *x, long incx, double *y, long incy, double *param );

## PURPOSE

drotm A pply a G iven 5 rotation constructed by SRO TM G .

## ARGUMENTS

N (input)
On entry, N specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.
X (input/output)
$(1+(n-1) * a b s(\mathbb{N} C X))$. On entry, the increm ented array $X$ m ustcontain the vectorx. On exit, $X$ is overw ritten by the updated vector $x$.
$\mathbb{N} C X$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of $\mathrm{X} . \mathbb{N} C X$ m ustnot.be zero. U nchanged on exit.

Y (input/output)
( $1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C Y)$ ). On entry, the increm ented array $Y$ mustcontain the vectory. On exit, $Y$ is overw rilten by the updated vectory.
$\mathbb{N} C Y$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of Y. $\mathbb{N} C Y$ m ustnot.be zero. Unchanged on exit.

PARAM (input)
On entry, the rotation values constructed by SRO TM G. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

drotm g -C onstruct a Gentlem an's m odified G iven 's plane rotation

## SYNOPSIS

```
SUBROUTINE DROTMG (D 1,D 2,B1,B2,PARAM)
```

DOUBLE PRECISION D 1,D 2,B1,B2
DOUBLE PRECISION PARAM (*)
SUBROUTINEDROTM G_64(D 1,D 2,B1,B2,PARAM)
DOUBLE PRECISION D 1,D 2, B1,B2
DOUBLE PRECISION PARAM (*)

## F95 INTERFACE

SU BROUTINE ROTM G (D1,D 2,B1,B2,PARAM)
REAL (8) ::D 1, D 2, B1,B2
REAL (8),D $\mathbb{I}$ ENSION (:) ::PARAM
SUBROUTINEROTM G_64D 1,D 2,B1,B2,PARAM)

REAL (8) :: D 1,D 2, B1,B2
REAL (8),D $\mathbb{M}$ ENSION (:) ::PARAM

## C INTERFACE

\#include <sunperfh>
void drotm g (double d1, double d2, double b1, double b2, double *param );
void drotm g_64 (double d1, double d2, double b1, double b2, double *param );

## PURPOSE

drotm g C onstruct $G$ entlem an sm odified a G iven splane rotation thatw illannihilate an elem ent of a vector.

## ARGUMENTS

D 1 (input/output)
On entry, the first diagonal entry in the $H$ $m$ atrix. O $n$ exit, changed to reflect the effectof the transform ation.
D 2 (input/output)
On entry, the second diagonal entry in the $H$ $m$ atrix. On exit, changed to reflect the effect of the transform ation.

B1 (input/output)
On entry, the firstelem ent of the vector to which the H matrix is applied. On exit, changed to reflect the effect of the transform ation.

## B2 (input)

O $n$ entry, the second elem ent of the vector to which the $H \mathrm{~m}$ atrix is applied. U nchanged on exit.

PARAM (output)
On exit, PARAM (1) describes the form of the rotation matrix H, and PARAM (2.5) contain the $H$ $m$ atrix.

IfPARAM (1) $=-2$ then $H=I$ and no elem ents of PARAM are modified.

IfPARAM $(1)=-1$ then PARAM $(2)=h 11$, PARAM $(3)=$ h21, PARAM (4) $=\mathrm{h} 12$, and PARAM (5) $=\mathrm{h} 22$.

IfPARAM $(1)=0$ then $\mathrm{h} 11=\mathrm{h} 22=1$, PARAM $(3)=$ h21, and PARAM (4) $=$ h12.

IfPARAM $(1)=1$ then $h 12=1, h 21=-1$, PARAM $(2)=$ h11, and PARAM (5) = h22.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dsbev - com pute all the eigenvalues and, optionally, eigenvectors of a realsym $m$ etric band $m$ atrix A

## SYNOPSIS

```
SUBROUT\mathbb{NEDSBEV (OBBZ,UPLO,N,KD,A,LDA,W ,Z,LD Z,W ORK, INFO)}
CHARACTER * 1 JOBZ,UPLO
INTEGERN,KD,LDA,LD Z, INFO
DOUBLE PRECISION A (LDA,*),W (*),Z (LD Z,*),W ORK (*)
SU BROUTINE DSBEV_64(JOBZ,UPLO,N,KD,A,LDA,W ,Z,LD Z,W ORK,
    \mathbb{NFO)}
CHARACTER * 1 OOBZ,UPLO
INTEGER*8N,KD,LDA,LD Z, INFO
DOUBLE PRECISION A (LDA,*),W (*),Z (LD Z,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE SBEV (JOBZ,UPLO, $\mathbb{N}], K D, A,[L D A], W, Z,[L D Z],[W O R K]$, [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1)::JOBZ,UPLO
$\mathbb{N}$ TEGER :: N, KD, LDA, LD $Z, \mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL (8), D IM ENSIO N (:,:) ::A, Z
SU BROUTINE SBEV_64 (JOBZ, UPLO, $\mathbb{N}], K D, A,[L D A], W, Z,[L D Z]$, [ $\mathrm{W} O \mathrm{RK}$ ], [ $\mathbb{N} F \mathrm{O}]$ )

CHARACTER (LEN=1):: JOBZ, UPLO
$\mathbb{N} T E G E R(8):: N, K D, L D A, L D Z, \mathbb{N F O}$

REAL (8),D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) ::A , Z

## C INTERFACE

\#include < sunperfh>
void dsbev (char jobz, char uplo, intn, int kd, double *a, int lda, double ${ }^{*}$, double ${ }^{*}$, int ldz, int *info);
void dsbev_64 (char jobzz, charuplo, long n, long kd, double
*a, long lda, double *w, double *z, long ldz, long
*info);

## PURPOSE

dsbev com putes all the eigenvalues and, optionally, eigenvectors of a real sym m etric band $m$ atrix A .

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}^{\prime}$ : C om pute eigenvalues only;
= V ': C om pute eigenvalues and eigenvectors.
UPLO (input)
= U ': U pper triangle ofA is stored;
= LL': Low ertriangle ofA is stored.

N (input) The order of the m atrix A. $\mathrm{N}>=0$.
KD (input)
The num ber of superdiagonals of the $m$ atrix $A$ if UPLO = U',orthe num berof subdiagonals ifU PLO = L'. KD >= 0 .

A (input/output)
O n entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the first $K D+1$ row s of the array. The $j$ th colum $n$ of $A$ is stored in the jth colum $n$ of the anray A as follow s: if $\mathrm{UPLO}=\mathrm{U}, \mathrm{A}(\mathrm{kd}+1+i-j, j)=A(i, j)$ for max $(1, j$ $\mathrm{kd})<=i<=j$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{A}(1+i-j, j)=A(i, j)$ for $j<=i<=m$ in $(n, j+k d)$.

On exit, A is overw rilten by values generated during the reduction to tridiagonal form . If $\mathrm{PLO}=$

U', the firstsuperdiagonal and the diagonal of the tridiagonal m atrix T are retumed in row sK D and $K D+1$ ofA, and if $U P L O=\Sigma$ ', the diagonaland first subdiagonal of $T$ are retumed in the first tw o row sofA.

LD A (input)
The leading dim ension of the array A. LD A >=KD + 1.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.
$Z$ (input) If $\mathcal{O B Z}=\mathrm{V}$ ', then if $\mathbb{N} F O=0, Z$ contains the orthonorm aleigenvectors of the $m$ atrix $A, w$ ith the $i$-th colum $n$ of $Z$ holding the eigenvector associated with W (i). If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced.

LD $Z$ (input)
The leading din ension of the amay $\mathrm{Z} . \operatorname{LD} \mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD $Z>=\mathrm{max}(1, N)$.

W ORK (w orkspace)
dim ension MAX ( $1,3 * \mathrm{~N}-2)$ )
$\mathbb{N} F O$ (output)
= 0 : successfulexit
< 0: if $\mathbb{I N}$ FO = -i, the i-th argum ent had an illegalvalue
> 0 : if $\mathbb{N} F O=$ i, the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dsbevd - com pute all the eigenvalues and, optionally, eigenvectors of a realsym $m$ etric band $m$ atrix $A$

## SYNOPSIS

```
SU BROUT\mathbb{NEDSBEVD (OBZ,UPLO,N,KD,AB,LDAB,W,Z,LD Z,W ORK,}
    LW ORK,IN ORK,L\mathbb{IN ORK,INFO)}
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGERN,KD,LDAB,LD Z,LW ORK,LIN ORK,\mathbb{NFO}}\mathbf{N}\mathrm{ , LN}
INTEGER IN ORK (*)
DOUBLE PRECISION AB (LDAB,*),W (*),Z (LDZ ,*),W ORK (*)
SU BROUTINE D SBEVD_64 (JOBZ,UPLO,N,KD,AB,LDAB,W ,Z,LD Z,W ORK,
    LW ORK,IN ORK,LIN ORK,\mathbb{NFO)}
```

CHARACTER * 1 JOBZ, UPLO
$\mathbb{N} T E G E R * 8 N, K D, L D A B, L D Z, L W O R K, L \mathbb{I N} O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I N}$ ORK ( ${ }^{*}$ )
DOUBLE PRECISION AB (LDAB,*),W (*), Z (LDZ,*),WORK (*)

## F95 INTERFACE

SU BROUTINE SBEVD (JOBZ, UPLO, $\mathbb{N}], K D, A B,[L D A B], W, Z,[L D Z],[W O R K]$, [LW ORK ], [ $\mathbb{W}$ ORK ], [LIN ORK ], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1)::JOBZ, UPLO
$\mathbb{N} T E G E R:: N, K D, L D A B, L D Z, L W O R K, L \mathbb{I N} O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N}$ ORK
REAL (8),D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL (8), D IM ENSION (:,:) ::AB,Z
SU BROUTINE SBEVD_64 (JOBZ,UPLO, $\mathbb{N}$ ],KD, AB, [LDAB],W,Z, [LD Z],
[W ORK ], [LW ORK ], [IN ORK ], [LINORK], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1)::JOBZ,UPLO
$\mathbb{N}$ TEGER (8) ::N,KD,LDAB,LDZ,LW ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{I N}$ ORK
REAL (8), D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL (8),D IM ENSION (:,:) ::AB,Z

## C INTERFACE

\#include < sunperfh>
void dsbevd (char jobz, char uplo, intn, intkd, double *ab, int ldab, double *w, double *z, intldz, int *info);
void dsbevd_64 (char jobz, charuplo, long n, long kd, double *ab, long ldab, double *w , double *z, long ldz, long *info);

## PURPOSE

dsbevd com putes all the eigenvalues and, optionally, eigenvectors of a realsym $m$ etric band $m$ atrix $A$. If eigenvectors are desired, ituses a divide and conquer algorithm .

The divide and conqueralgorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on $m$ achines w ith a guard digit in add/subtract, or on those binary $m$ achines $w$ thout guard digits w hich subtract like the $C$ ray X M P , C ray Y M P , C ray C-90, orC ray-2. It could conceivably fail on hexadecim al or decim al $m$ achines $w$ ithout guard digits, butw e know of none.

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}^{\prime}$ : C om pute eigenvalues only;
$=\mathrm{V}$ ': C om pute eigenvalues and eigenvectors.
UPLO (input)
$=\mathrm{U}$ :: U pper triangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the matrix $\mathrm{A} . \mathrm{N}>=0$.
$K D$ (input)
The num ber of superdiagonals of the $m$ atrix A if
UPLO $=\mathrm{U}$ ', or the num berof subdiagonals if UPLO
$=L^{\prime} . K D>=0$.

AB (input/output)
O $n$ entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstK $D+1$ row s of the amay. The $j$ th colum $n$ of $A$ is stored in the jth colum $n$ of the array AB as follow s: if $\mathrm{UPLO}=\mathrm{U}$ ', $\mathrm{AB}(\mathrm{kd}+1+\mathrm{i}-j)=\mathrm{j}(i, j)$ for $\mathrm{max}(1, j$ $\mathrm{kd})<=i<=j$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{AB}(1+i-j, j)=A(i, 7)$ for $j<=i<=m$ in $(n, j+k d)$.

On exit, AB is overw rilten by values generated during the reduction to tridiagonal form. If PLO = U', the first superdiagonal and the diagonal of the tridiagonal m atrix T are retumed in row SKD and $K D+1$ of $B$, and if $U P L O=L$ ', the diagonal and first subdiagonal of $T$ are retumed in the first tw o row sofA B.

LDAB (input)
The leading dim ension of the array AB. LD AB $>=K D$ +1 .

W (output) If $\mathbb{N}$ FO $=0$, the eigenvalues in ascending order.

Z (output)
If $\mathrm{JOBZ}=\mathrm{V}$ ', then if $\mathbb{N} F O=0, \mathrm{Z}$ contains the orthonorm aleigenvectors of the $m$ atrix $A, w$ ith the i-th colum $n$ of $Z$ holding the eigenvector associated w ith $W$ (i). If $J O B Z=N$ ', then $Z$ is not referenced.

LD Z (input)
The leading dim ension of the amay $Z . L D Z>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD Z $>=\max (1, N)$.

W ORK (w orkspace)
dim ension (LW ORK) On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim allW ORK.

LW ORK (input)
The dim ension of the array $\mathrm{W} O R \mathrm{~K}$. If $\mathrm{N}<=1$, LW ORK must.be at least1. If 0 BZ $=N$ 'and $N>$
2, LW ORK mustbe at least $2 * \mathrm{~N}$. If $\mathrm{JOBZ}=\mathrm{V}$ 'and
$\mathrm{N}>2$,LW ORK mustbe at least ( $1+5 * \mathrm{~N}+2 \mathrm{~N}_{\mathrm{N}}$ **2
).
If LW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim al size of
the W ORK amray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERB LA .

IN ORK (w orkspace/output)
On exit, if $\mathbb{N} F O=0, \mathbb{I V}$ ORK (1) retums the optim al LIN ORK.
LIV ORK (input)
The dim ension of the array LIN ORK. If $\mathrm{JOBZ}=\mathrm{N}^{\prime}$ or $\mathrm{N}<=1$, LIN ORK mustbe at least1. If $\operatorname{JOBZ}=$ $V$ 'and $N>2$, LIN ORK m ust.be at least $3+5 * N$.

If LIV ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the IV ORK array, retums this value as the first entry of the $\mathbb{I V}$ ORK array, and no errorm essage related to $L \mathbb{I N} O R K$ is issued by XERBLA.
$\mathbb{I N F O}$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the i-th argum ent had an illegalvalue
> 0 : if $\mathbb{N} F O=$ i, the algonithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dsbevx - com pute selected eigenvalues and, optionally, eigenvectors of a realsym $m$ etric band $m$ atrix $A$

## SYNOPSIS

```
SUBROUT\mathbb{NEDSBEVX (OBZ,RANGE,UPLO,N,KD,A,LDA,Q,LDQ,VL,}
```



```
CHARACTER * 1 JOBZ,RANGE,UPLO
\mathbb{NTEGER N,KD,LDA,LDQ,\mathbb{L},\mathbb{U},NFOUND,LD Z, INFO}
\mathbb{NTEGER IN ORK2(*),\mathbb{FA | (*)}}\mathbf{(*)}
DOUBLE PRECISION VL,VU,ABTOL
DOUBLE PRECISION A (LDA,*),Q (LDQ ,*),W (*),Z (LD Z,*),W ORK (*)
SU BROUT\mathbb{NEDSBEVX_64(JOBZ,RANGE,UPLO,N,KD,A,LDA,Q,LDQ,VL,}
```


CHARACTER * 1 JOBZ,RANGE,UPLO
$\mathbb{N} T E G E R * 8 N, K D, L D A, L D Q, \mathbb{L}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I V}$ ORK 2 (*), $\mathbb{F A} \mathbb{H}$ (*)
DOUBLE PRECISION VL,VU,ABTOL
DOUBLE PRECISION A (LDA, *), Q (LDQ , *), W (*), Z (LDZ, *), W ORK (*)

## F95 INTERFACE

SU BROUTINE SBEVX ( $\mathrm{O} B \mathrm{~B}, \mathrm{RANGE,UPLO}, \mathbb{N}], K D, A,[L D A], Q,[L D Q]$, VL, VU, $\mathbb{I}, \mathbb{U}, A B T O L, N F O U N D, W, Z,[L D Z],[W O R K],[\mathbb{W}$ ORK2], $\mathbb{F A} \mathbb{L},[\mathbb{N F O}])$

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
$\mathbb{N} T E G E R:: N, K D, L D A, L D Q, \mathbb{L}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I}$ ENSION (:) :: $\mathbb{I W}$ ORK2, $\mathbb{F} A \mathbb{I}$
REAL (8) ::VL,VU,ABTOL

REAL (8), D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) ::A, Q,Z
SU BROUTINE SBEVX_64 (OBZ,RANGE,UPLO, $\mathbb{N}], K D, A,[L D A], Q,[L D Q]$, VL,VU, $\mathbb{H}, \mathbb{U}, A B T O L, N F O U N D, W, Z,[L D Z],[\mathbb{W} O R K],[\mathbb{I W} O R K 2]$, $\mathbb{F A} \mathbb{I},[\mathbb{N F O}])$

CHARACTER (LEN=1):: JOBZ,RANGE,UPLO
$\mathbb{N}$ TEGER (8) :: N, KD,LDA,LDQ, $\mathbb{L}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{I W}$ ORK2, $\mathbb{F A} \mathbb{I}$
REAL (8) ::VL,VU, ABTOL
REAL (8),D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL (8), D IM ENSIO N (:,:) ::A , Q , Z

## C INTERFACE

\#include < sunperfh>
void dsbevx (char j̀bz, char range, char uplo, intn, int kd, double *a, int lda, double *q, int ldq, double vl, double vu, int il, int iu, double abtol, int ${ }^{*}$ nfound, double ${ }^{*}$, double ${ }^{*}$ z, int ldz, int *ifail, int*info);
void dsbevx_64 (char jobz, char range, char uplo, long n, long kd, double *a, long lda, double *q, long ldq, double vl, double vu, long il, long iu, double abtol, long *nfound, double *w, double *z, long ldz, long *ifail, long *info);

## PURPOSE

dsbevx com putes selected eigenvalues and, optionally, eigenvectors of a realsym $m$ etric band $m$ atrix A. Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

## ARGUMENTS

```
JO BZ (input)
    = N ': C om pute eigenvalues only;
    = V ': C om pute eigenvalues and eigenvectors.
RANGE (input)
= A ': alleigenvalues w ill.be found;
= V ::alleigenvalues in the half-open interval
(NL,VU] will be found; = ' 1 ': the \(\mathbb{I}\)-th through
\(\mathbb{I U}\)-th eigenvaluesw illle found.
```

UPLO (input)
= U ': U ppertriangle ofA is stored;
= LL': Low ertriangle ofA is stored.
N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.
KD (input)
The num ber of superdiagonals of the $m$ atrix $A$ if UPLO = U',orthe num berof subdiagonals ifU PLO $=\mathbb{L} \cdot \mathrm{KD}>=0$.

A (input/output)
On entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstKD +1 row s of the amay. The $j$ th colum $n$ of $A$ is stored in the $j$ th colum n of the amay $A$ as follow $s$ : if $\mathrm{UPLO}=\mathrm{U}^{\prime}, \mathrm{A}(\mathrm{kd}+1+\mathrm{i}-j, j)=A(i, j)$ for $\mathrm{max}(1, j$ $\mathrm{kd})<=i<=j$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{A}(1+i-j, j)=A(i, j)$ for $j<=i<=m$ in $(n, j+k d)$.

On exit, A is overw rilten by values generated during the reduction to tridiagonal form. If $\mathrm{PLO}=$ U ', the first superdiagonal and the diagonal of the tridiagonal m atrix T are retumed in row SKD and KD +1 of , and if $\mathrm{PLO}=\mathrm{L}$ ', the diagonal and first subdiagonal of T are retumed in the first two row sofA.

LD A (input)
The leading dim ension of the array A. LD A >=KD + 1.

Q (output)
If $\mathrm{JOBZ}=\mathrm{V}$ ', the $\mathrm{N}-$ by N orthogonal m atrix used in the reduction to tridiagonal form. If $\mathrm{JO} \mathrm{BZ}=$ N ', the array Q is not referenced.

LD Q (input)
The leading dim ension of the array $Q$. If $J 0 B Z=$ V ', then LD $Q>=\max (1, N)$.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU. N ot referenced ifRANGE = A 'or I'.

VU (input)
Se the description of V L .
II (input)

If RA N GE=I', the indiges (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{H}<=\mathbb{Z}<=N$, if $N>0$; $\mathbb{H}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N otreferenced ifRANGE $=$ A 'or V'.

IU (input)
See the description of II.

ABTOL (input)
The absolute emortolerance forthe eigenvalues.
A $n$ approxim ate eigenvalue is accepted as converged w hen itis determ ined to lie in an interval [a,b]
of w idth less than orequal to
$A B T O L+E P S * \max (a|, b|)$,
where EPS is them achine precision. If ABTOL is less than or equal to zero, then EPS*|I|will.be used in its place, w here $\mathrm{F} \mid$ is the 1 -norm of the tridiagonal m atrix obtained by reducing A to tridiagonalform .

E igenvalues w illbe com puted m ostaccurately when ABTOL is set to tw ice the underflow threshold 2*SLAMCH (S ), notzero. If this routine retums w ith $\mathbb{N}$ FO $>0$, indicating that som e eigenvectors did not converge, try setting ABTOL to 2*SLAM CH (S ).

See "C om puting Sm allSingularV alues of B idiagonal M atrices $w$ ith G uaranteed H igh Relative A ccuracy," by D em m eland K ahan, LAPA CK W orking N ote \#3.

NFOUND (output)
The total num ber of eigenvalues found. $0<=$
NFOUND <= N. IfRANGE = A', NFOUND = N, and if RANGE $=$ 'I', NFOUND $=\mathbb{I}-\mathbb{L}+1$ 。

W (output)
The first NFOUND elem ents contain the selected eigenvalues in ascending order.
$Z$ (input) If $\mathrm{OB} Z=V^{\prime}$, then if $\mathbb{N} F O=0$, the first $N F O U N D$ colum ns of $Z$ contain the orthonorm aleigenvectors of the matrix A corresponding to the selected eigenvalues, w ith the $i-$ th colum n of $Z$ holding the eigenvector associated w ith $W$ (i). If an eigenvector fails to converge, then that colum n of Z contains the latest approxim ation to the eigenvector, and the index of the eigenvector is retumed in

ㅍAII. If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced.
$N$ ote: the user must ensure that at least
$m$ ax ( $1, N F O U N D$ ) colum ns are supplied in the array $Z$;
if RANGE $=V$ ', the exactvalue ofNFOUND is not know $n$ in advance and an upperbound $m$ ustbe used.
LD Z (input)
The leading dim ension of the array Z. LD Z $>=1$, and if $\mathrm{OBB}=\mathrm{V}^{\prime}$, LD Z > = $\mathrm{max}(1, N)$.

W ORK (w orkspace)
dim ension $(7 \star \mathrm{~N})$

IW ORK 2 (w orkspace)

FAII (output)
If $\mathrm{OBZ}=\mathrm{V}^{\prime}$ ', then if $\mathbb{N F O}=0$, the first NFOUND
elements of $\mathbb{F A} I L$ are zero. If $\mathbb{N F O}>0$, then
IFA II contains the indices of the eigenvectors
that failed to converge. If $\mathrm{OBZ}=\mathrm{N}$ ', then
$\mathbb{F} A \mathbb{I}$ is notreferenced.
$\mathbb{N}$ FO (output)
= 0: successfulexit.
$<0$ : if $\mathbb{N} F O=-i$, the i-th argum enthad an illegalvalue.
$>0:$ if $\mathbb{N} F O=i$, then ieigenvectors failed to converge. Their indiges are stored in array正AII.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dsogst-reduce a realsym $m$ etric-definite banded generalized eigenproblem $\mathrm{A} * \mathrm{x}=\operatorname{lam}$ bda* $\mathrm{B} * \mathrm{x}$ to standard form $\mathrm{C} * \mathrm{y}=$ lam bda夫y,

## SYNOPSIS

```
SUBROUT\mathbb{NEDSBGST NECT,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,X,LDX,}
    W ORK,INFO)
```

CHARACTER * 1 VECT,UPLO
$\mathbb{N}$ TEGER N,KA, KB,LDAB,LDBB,LDX, $\mathbb{N} F O$
DOUBLE PRECISION AB (LDAB,*),BB (LDBB,*), X (LDX,*), WORK (*)
SU BROUTINEDSBGST_64 NECT, UPLO,N,KA,KB,AB,LDAB,BB,LDBB,X,
LD $\mathrm{X}, \mathrm{W}$ ORK, $\mathbb{N} F \mathrm{O}$ )
CHARACTER * 1 VECT,UPLO
$\mathbb{N} T E G E R * 8 N, K A, K B, L D A B, L D B B, L D X, \mathbb{N} F O$
DOUBLE PRECISION AB (LDAB,*), BB (LDBB,*), X (LD X, $\left.{ }^{\star}\right), \mathrm{W} O R K\left({ }^{*}\right)$

## F95 INTERFACE

SU BROUTINE SBGST $N E C T, U P L O, \mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B], X$, [LDX], [W ORK], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) ::VECT,UPLO
$\mathbb{N} T E G E R:: N, K A, K B, L D A B, L D B B, L D X, \mathbb{N} F O$
REAL (8),D IM ENSION (:) ::W ORK
REAL (8), D IM ENSION (:,:) ::AB,BB,X

SU BROUTINE SBGST_64 NECT,UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B]$, $\mathrm{X},[\mathrm{LDX}], \mathbb{W} \mathrm{ORK}],[\mathbb{N} F \mathrm{O}])$

CHARACTER (LEN=1) ::VECT,UPLO
$\mathbb{N}$ TEGER (8) ::N,KA, KB,LDAB,LDBB,LDX, $\mathbb{N} F O$
REAL (8),D IM ENSION (:) ::W ORK
REAL (8), D IM ENSION (:,:)::AB,BB,X

## C INTERFACE

\#include <sunperfh>
void dsbogst(charvect, charuplo, intn, int ka, int kb, double *ab, int ldab, double *bb, int labb, double * $x$, int ldx $x$, int *info);
void dsbgst_64 (charvect, charuplo, long n, long ka, long kb, double *ab, long ldab, double *bb, long ldbb, double *x, long ldx, long *info);

## PURPOSE

dsogst reduces a real sym $m$ etric-definite banded generalized eigenproblem $A * x=\operatorname{lam}$ bda*B*x to standard form $C * y=$ lam bda*y, such thatC has the sam e bandw idth asA .

B m usthave been previously factorized as $\mathrm{S}^{* *} \mathrm{~T} * \mathrm{~S}$ by SPBSTF, using a split Cholesky factorization. A is overw ritten by C $=\mathrm{X} * * \mathrm{~T} * \mathrm{~A} * \mathrm{X}$, where $\mathrm{X}=\mathrm{S} * *(-1) * \mathrm{Q}$ and Q is an orthogonal $m$ atrix chosen to preserve the bandw idth ofA.

## ARGUMENTS

## VECT (input)

$=N$ ': do not form the transform ation $m$ atrix $X$;
= V ': form X .

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrioes A and $\mathrm{B} . \mathrm{N}>=0$.

KA (input)
The num ber of superdiagonals of the $m$ atrix $A$ if UPLO = U',orthe num berof subdiagonals ifUPLO $=L^{\prime} . K A>=0$.

K B (input)
The num ber of superdiagonals of the $m$ atrix $B$ if
UPLO = U',orthe num berof subdiagonals ifU PLO
$=L^{\prime} . K A>=K B>=0$.

AB (input/output)
O $n$ entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstka+1 row s of the amay. The $j$ th colum $n$ of $A$ is stored in the $j$ th colum $n$ of the amay $A B$ as follows: if $\mathrm{UPLO}=\mathrm{U}, \mathrm{AB}(k a+1+i-j)=A(i, j)$ for $m a x(1, j$ $\mathrm{ka})<=\mathrm{i}<=j$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{AB}(1+i-j, j)=A(i, j)$ for $j<=i<=m$ in $(n, j+k a)$.

On exit, the transform ed matrix $X * * T * A * X$, stored in the sam e form at as $A$.

LD AB (input)
The leading dim ension of the array $A B$. LD AB >= KA+1.

BB (input)
The banded factors from the split Cholesky factorization ofB, as retumed by SPBSTF, stored in the first $K B+1$ row sof the array.

LD BB (input)
The leading dim ension of the array BB. LD BB >= K B+1.

X (output)
IfVECT = $V$ ', the $n-b y-n m$ atrix $X$. If VECT $=$ N ', the array X is not referenced.

LD X (input)
The leading dim ension of the aray X . LD X >= $\max (1, N)$ if VECT $=V$ ';LD X >= 1 otherw ise.

W ORK (w orkspace)
dim ension $(2 * N)$
$\mathbb{N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{N}$ FO = -i, the i-th argum ent had an illegalvahue.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dsbogv - com pute allthe eigenvalues, and optionally, the eigenvectors of a real generalized sym $m$ etric-definite banded eigenproblem, of the form $A * x=(l a m ~ b d a) * B * x$

## SYNOPSIS

```
SUBROUTINEDSBGV (OOBZ,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,W,Z,
    LD Z,W ORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGER N,KA,KB,LDAB,LDBB,LD Z, IN FO}
DOUBLE PRECISION AB (LDAB,*), BB (LDBB,*),W (*), Z (LDZ,*),
W ORK (*)
SUBROUT\mathbb{NEDSBGV_64(JOBZ,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,W,Z,}
    LD Z,WORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
INTEGER*8N,KA,KB,LDAB,LDBB,LD Z, INFO
DOUBLE PRECISION AB (LDAB,*), BB (LDBB,*),W (*), Z (LD Z,*),
W ORK (*)
```


## F95 INTERFACE

SU BROUTINE SBGV (OBBZ, UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B], W$, Z, [LD Z], [W ORK], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) :: JOBZ, UPLO
$\mathbb{N}$ TEGER :: N,KA,KB,LDAB,LDBB,LD Z, $\mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) ::AB,BB,Z
SU BROUTINE SBGV_64 (JOBZ, UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B]$,

W , Z, [LD Z], [W ORK ], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1)::JOBZ,UPLO
$\mathbb{N}$ TEGER (8) ::N,KA,KB,LDAB,LDBB,LD, $\mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::AB,BB,Z

## C INTERFACE

\#include <sunperfh>
void dsbogv (char jobz, charuplo, intn, int ka, int kb, dou-
ble *ab, int ldab, double *bb, int ldbb, double
*w , double *z, intldz, int *info);
void dsbgv_64 (char jobz, charuplo, long n, long ka, long
kb, double *ab, long ldab, double *bb, long ldbb, double *w , double *z, long ldz, long *info);

## PURPOSE

dsogv com putes all the eigenvalues, and optionally, the eigenvectors of a realgeneralized sym $m$ etric-definite banded eigenproblem, of the form $A$ *x $=(l a m ~ b d a) * B * x$. H ere A and B are assum ed to be sym $m$ etric and banded, and $B$ is also positive definite.

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}$ ': C om pute eigenvalues only;
$=\mathrm{V}$ ': C om pute eigenvalues and eigenvectors.
UPLO (input)
$=\mathrm{U}$ ': U ppertriangles of $A$ and $B$ are stored;
$=\mathrm{L}$ ': Low ertriangles of $A$ and $B$ are stored.
$N$ (input) The order of the $m$ atrices $A$ and $B . N>=0$.

KA (input)
The num ber of superdiagonals of the $m$ atrix $A$ if UPLO $=\mathrm{U}$ ', or the num berof subdiagonals if P PLO
$=\mathbb{L} \cdot \mathrm{KA}>=0$.

KB (input)
The num ber of superdiagonals of the $m$ atrix $B$ if UPLO $=\mathrm{U}$ ', or the num berof subdiagonals if UPLO $=\mathbb{L} \cdot \mathrm{KB}>=0$.

AB (input/output)
O $n$ entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstka+1 row s of the array. The $j$ th colum n of A is stored in the $j$ th colum $n$ of the array AB as follow s: if UPLO = U',AB $(k a+1+i-j)=A(i, j)$ for max ( $1, j$ $\mathrm{ka})<=\mathrm{i}<=j$; if UPLO $=\mathrm{L}, \mathrm{AB}(1+i-j, 7)=A(i, j)$ for $j<=i<=m$ in $(n, j+k a)$.

On exit, the contents of $A B$ are destroyed.
LDAB (input)
The leading dim ension of the array AB. LD AB >= KA+1.

BB (input/output)
O $n$ entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $B$, stored in the first kb+1 row sof the array. The $j$ th colum n of $B$ is stored in the $j$ th colum $n$ of the array $B B$ as follow s: if $\mathrm{UPLO}=\mathrm{U}, \mathrm{BB}(\mathrm{kb}+1+\mathrm{i}-j)=\mathrm{j}(1,7)$ for $\mathrm{max}(1, j$ $\mathrm{kb})<=\mathrm{i}<=\dot{j}$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{BB}(1+i-j, j)=B(i, 7)$ for $j<=i<=m$ in $(n, j+k b)$.

On exit, the factors from the splitCholesky factorization $B=S * * T * S$, as retumed by SPBSTF .

## LD BB (input)

The leading dim ension of the array BB. LD BB >= K B+1.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.
$Z$ (input) If $\mathrm{OOBZ}=\mathrm{V}^{\prime}$, then if $\mathbb{N} F O=0, \mathrm{Z}$ contains the $m$ atrix $Z$ ofeigenvectors, $w$ ith the $i$-th colum $n$ of Z holding the eigenvector associated w ith W (i). The eigenvectors are norm alized so that $\mathrm{Z} * * \mathrm{~T} * \mathrm{~B} * \mathrm{Z}=$ I. If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced.

LD Z (input)
The leading dim ension of the array Z . LD $\mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD Z >= N .

W ORK (w orkspace)
dim ension ( $3 * \mathrm{~N}$ )
$\mathbb{N} F O$ (output)
= 0: successfulexit
<0: if $\mathbb{N N}$ FO = -i, the i-th argum ent had an illegalvalue
$>0$ : if $\mathbb{N} F O=i$, and $i$ is:
<= N : the algorithm failed to converge: i offdiagonal elem ents of an interm ediate tridiagonal form did notconverge to zero; $>\mathrm{N}$ : if $\mathbb{N} F \mathrm{FO}=\mathrm{N}$ +i , for $1<=\mathrm{i}<=\mathrm{N}$, then SPBSTF
retumed $\mathbb{N} F O=i: B$ is not positive definite. The factorization ofB could notbe com pleted and no eigenvalues or eigenvectors w ere com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dsogvd -com pute all the eigenvalues, and optionally, the eigenvectors of a realgeneralized sym $m$ etric-definite banded eigenproblem, of the form $A * x=(l a m ~ b d a) * B * x$

## SYNOPSIS

```
SU BROUT\mathbb{NEDSBGVD (JOBZ,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,W,Z,}
    LD Z,W ORK,LW ORK,IN ORK,LIN ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGER N,KA,KB,LDAB,LDBB,LD Z,LW ORK,LIN ORK, INFO}
\mathbb{NTEGER IN ORK (*)}
DOUBLE PRECISION AB (LDAB,*), BB (LDBB,*),W (*), Z (LD Z,*),
W ORK (*)
SU BROUT\mathbb{NE D SBGVD_64 (OOBZ,UPLO ,N,KA,KB,AB,LDAB,BB,LDBB,W ,Z,}
    LD Z,W ORK,LW ORK,INORK,LIN ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
```



```
INTEGER*8 IN ORK (*)
DOUBLE PRECISION AB (LDAB,*), BB (LDBB,*),W (*), Z (LD Z,*),
W ORK (*)
```


## F95 INTERFACE

SU BROUTINE SBGVD ( $\mathrm{O} B \mathrm{~B}, \mathrm{UPLO}, \mathbb{N}], \mathrm{KA}, \mathrm{KB}, \mathrm{AB},[L D A B], B B,[L D B B], W$,


CHARACTER (LEN=1)::JOBZ,UPLO
$\mathbb{N} T E G E R:: N, K A, K B, L D A B, L D B B, L D Z, L W O R K, L \mathbb{I N} O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{I N}$ ORK

REAL (8), D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::AB,BB,Z
SUBROUTINE SBGVD_64 (OOBZ,UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B]$,
W, Z, [LD Z], [W ORK ], [LW ORK], [IW ORK ], [LINORK], [ $\mathbb{N} F O]$ )
CHARACTER (LEN=1)::JOBZ,UPLO
$\mathbb{N} T E G E R(8):: N, K A, K B, L D A B, L D B B, L D Z, L W O R K, L \mathbb{I W} O R K$, $\mathbb{N}$ FO
$\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{I W}$ ORK
REAL (8),D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) ::AB,BB, Z

## C INTERFACE

\#include <sunperfh>
void dsbgvd (char jobz, charuple, intn, int ka, int kb, double *ab, int ldab, double *bb, int ldbb, double *W , double *z, int ldz, int *info);
void dsogvd_64 (char jobz, charuplo, long n, long ka, long kb, double *ab, long ldab, double *bb, long ldbb, double *w, double *z, long ldz, long *info);

## PURPOSE

dsogvd com putes all the eigenvalues, and optionally, the eigenvectors of a realgeneralized sym $m$ etric-definite banded eigenproblem, of the form $A * x=(\operatorname{lam} b d a) * B * x$. H ere A and B are assum ed to be sym $m$ etric and banded, and $B$ is also positive definite. If eigenvectors are desired, it uses a divide and conqueralgorithm .

The divide and conquer algorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on $m$ achines w th a guard digit in add/subtract, or on those binary $m$ achines $w$ thout guard digits w hich subtract like the $C$ ray X M P , C ray Y M P , C ray C -90, orC ray-2. It could conceivably fail on hexadecim al or decim al machines w thout guard digits, butw e know of none.

## ARGUMENTS

JOBZ (input)
= N ': C om pute eigenvalues only;
$=\mathrm{V}$ ': C om pute eigenvalues and eigenvectors.
$=\mathrm{U}:$ : U pper triangles of $A$ and $B$ are stored;
= L': Low ertriangles ofA and B are stored.
N (input) The order of the m atrices A and $\mathrm{B} . \mathrm{N}>=0$.

KA (input)
The num ber of superdiagonals of the matrix A if UPLO $=\mathrm{U}$ ', or the num berof subdiagonals if UPLO
$=\mathrm{L}$ '. KA >= 0 .

KB (input)
The num ber of superdiagonals of the $m$ atrix $B$ if UPLO = U',orthe num berof subdiagonals ifU PLO $=\mathbb{L}^{\prime} . \mathrm{KB}>=0$.
AB (input/output)
O $n$ entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstka+1 row s of the array. The $j$ th colum $n$ of $A$ is stored in the $j$ th colum n of the array AB as follows: if $\mathrm{UPLO}=\mathrm{U}$ ', AB $(k a+1+i-j)=A(i, 1)$ for $m a x(1, j$ $\mathrm{ka})<=i<=j ;$ ifUPLO $=\mathrm{L}, \mathrm{AB}(1+i-j, j)=A(i, j)$ for $j=i<=m$ in $(n, j+k a)$.

On exit, the contents of $A B$ are destroyed.

LDAB (input)
The leading dim ension of the array AB. LDAB >= KA+1.

BB (input/output)
O $n$ entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $B$, stored in the first kb+1 row sof the array. The $j$ th colum n of $B$ is stored in the jth colum $n$ of the array $B B$ as follow s: if $\mathrm{UPLO}=\mathrm{U}, \mathrm{BB}(k a+1+i-j)=\mathrm{B}(i, j)$ for $\max (1, j$ $\mathrm{kb})<=\mathrm{i}<=\dot{j}$ if P PLO $=\mathrm{L}, \mathrm{BB}(1+i-j, j)=B(i, 7)$ for $j=i<=m$ in $(n, j+k b)$.

On exit, the factorS from the splitCholesky factorization $B=S * * T * S$, as retumed by SPBSTF .

LD BB (input)
The leading dim ension of the array BB. LD BB >= K B+1.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.
$Z$ (input) If $\mathcal{O B Z}=\mathrm{V}^{\prime}$, then if $\mathbb{N} F O=0, \mathrm{Z}$ contains the $m$ atrix $Z$ ofeigenvectors, $w$ ith the $i$-th colum n of
$Z$ holding the eigenvector associated $w$ ith $W$ (i). The eigenvectors are norm alized so $\mathrm{Z} * * \mathrm{~T} * \mathrm{~B} * \mathrm{Z}=\mathrm{I}$. If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced.

LD $Z$ (input)
The leading dm ension of the array Z . LD $\mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD Z $>=\mathrm{max}(1, \mathrm{~N})$.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array W ORK. If $\mathrm{N}<=1$, LW ORK >= 1. If JOBZ $=N$ 'and $N>1$,LW ORK $>=$ 3*N. If $\operatorname{OOBZ}=\mathrm{V}$ 'and $\mathrm{N}>1$,LW ORK >=1 + 5*N + $2 * N * * 2$ 。

If LW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IV ORK (w orkspace/output)
On exit, if LIN ORK > 0, $\mathbb{I N}$ ORK (1) retums the optim all IV ORK.

LIV ORK (input)
The dim ension of the array $\mathbb{I N} O R K$. If $0 \mathrm{OBZ}=\mathrm{N}^{\prime}$ orN <=1,L $\mathbb{I N} O R K>=1$. If $J O B Z=V$ 'and $N>1$, LIN ORK >= $3+5 * N$.

If LIV ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the $\mathbb{I W}$ ORK array, retums this value as the first entry of the $\mathbb{I W}$ ORK array, and no errorm essage related to $L \mathbb{I N} O R K$ is issued by X ERBLA.
$\mathbb{I N F O}$ (output)
= 0 : successfinlexit
<0: if $\mathbb{N N}$ FO = -i, the i-th argum ent had an illegalvalue
> 0 : if $\mathbb{N F O}=\mathrm{i}$, and is:
$<=\mathrm{N}$ : the algorithm failed to converge: i offdiagonal elem ents of an interm ediate tridiagonal form did not converge to zero; > N : if $\mathbb{N} F O=\mathrm{N}$ $+i$, for $1<=i<=N$, then SPBSTF
retumed $\mathbb{N} F O=i: B$ is not positive definite. The factorization ofB could notbe com pleted and
no eigenvalues oreigenvectors w ere com puted.

## FURTHER DETAILS

B ased on contributions by
$M$ ark Fahey,D epartm entofM athem atics, U niv. ofK entucky, U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dsbogvx - com pute selected eigenvahues, and optionally, eigenvectors of a real generalized sym $m$ etric-definite banded eigenproblem, of the form $A$ *x (lam bda) *B *x

## SYNOPSIS

```
SUBROUT\mathbb{NEDSBGVX(OBZ,RANGE,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,}
    Q,LDQ,VL,VU,\mathbb{L},\mathbb{U},ABSTOL,M,W,Z,LD Z,W ORK,IN ORK, FFA IL,
    \mathbb{NFO)}
```

CHARACTER * 1 JOBZ,RANGE,UPLO
$\mathbb{N} T E G E R N, K A, K B, L D A B, L D B B, L D Q, \mathbb{L}, \mathbb{U}, M, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R \mathbb{I}$ ORK (*), $\mathbb{F A} \mathbb{L}(*)$
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION AB (LDAB,*), BB (LDBB,*), Q (LDQ,*), W (*),
Z (LD Z, $\left.{ }^{*}\right), \mathrm{W}$ ORK (*)
SU BROUTINEDSBGVX_64(JBZ,RANGE,UPLO,N,KA,KB,AB,LDAB,BB,
LD BB, $Q, L D Q, V L, V U, \mathbb{L}, \mathbb{U}, A B S T O L, M, W, Z, L D Z, W O R K, \mathbb{I N} O R K$,
$\mathbb{F A} \mathbb{L}, \mathbb{N} F O$ )
CHARACTER * 1 JOBZ,RANGE,UPLO
$\mathbb{N}$ TEGER*8N,KA,KB,LDAB,LDBB,LDQ, $\mathbb{L}, \mathbb{U}, M, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I N}$ ORK $(*), \mathbb{F A} \mathbb{I}(*)$
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION AB (LDAB,*), BB (LDBB,*), Q (LDQ,*), W (*),
Z (LD Z, $\left.{ }^{\star}\right)$, WORK (*)

## F95 INTERFACE

SU BROUTINE SBGVX (OBZ,RANGE,UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B$, $\left[\begin{array}{ll}\text { D BB }], Q,[L D Q], V L, V U, \mathbb{L}, \mathbb{U}, A B S T O L, M, W, Z,[L D Z],[W O R K],\end{array}\right.$
[ $\mathbb{N}$ ORK $], \mathbb{F A} \mathbb{I},[\mathbb{N} F O])$

CHARACTER (LEN=1) :: JOBZ,RANGE, UPLO
$\mathbb{N} T E G E R:: N, K A, K B, L D A B, L D B B, L D Q, \Pi, \mathbb{U}, M, L D Z, \mathbb{N} F O$
$\mathbb{N}$ TEGER,D $\mathbb{M} E N S I O N(:):: \mathbb{I} O R K, \mathbb{F A} \mathbb{I}$
REAL (8) :: VL,VU,ABSTOL
REAL (8), D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL (8), D $\mathbb{M}$ ENSION (:,:) ::AB,BB, Q , Z

SUBROUTINE SBGVX_64 (OBZ,RANGE, UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B$, $[[\mathrm{D} B \mathrm{~B}], \mathrm{Q},[\mathrm{LD} Q], \mathrm{VL}, \mathrm{VU}, \mathbb{Z}, \mathbb{U}, \mathrm{ABSTOL}, \mathrm{M}, \mathrm{W}, \mathrm{Z},[\mathrm{LD} \mathrm{Z}],[\mathrm{W} O R K]$, [ $\mathbb{I N}$ ORK], $\mathbb{F} A \mathbb{I},[\mathbb{N} F O]$ )

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{KA}, \mathrm{KB}, \mathrm{LD} A B, L D B B, L D Q, \mathbb{L}, \mathbb{U}, M, L D Z$,
$\mathbb{N} \mathrm{FO}$
$\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{I W} O R K, \mathbb{F A} \mathbb{I}$
REAL (8) ::VL,VU, ABSTOL
REAL (8), D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL (8), D $\mathbb{M}$ ENSION (: : : : : AB, BB , $2, Z$

## C INTERFACE

\#include <sunperfh>
void dsogvx (char jobz, char range, charuplo, intn, intka, int kb , double *ab, int ldab, double *bb, int ldbb, double *q, int ldq, double vl, double vu, int il, int in, double abstol, int *m, double ${ }^{*}$, double *z, int ldz, int *ifail, int *info);
void dsbgvx_64 (char jobz, charrange, char uplo, long n, long ka, long kb, double *ab, long ldab, double *bb, long ldbb, double *q, long ldq, double vl, double vu, long il, long iu, double abstol, long *m , double *w, double *z, long ldz, long *ifail, long *info);

## PURPOSE

d.sogvx com putes selected eigenvalues, and optionally , eigenvectors of a real generalized sym m etric-definite banded eigenproblem, of the form $A * x=(l a m ~ b d a) * B * x . H$ ere $A$ and $B$ are assum ed to be sym $m$ etric and banded, and $B$ is also positive definite. Eigenvahues and eigenvectors can be selected by specifying either alleigenvalues, a range of values or a range of indices for the desired eigenvalues.

## ARGUMENTS

JO BZ (input)
$=\mathrm{N}$ : C om pute eigenvalues only;
$=\mathrm{V}$ : C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvalues w illbe found.
$=V$ ': alleigenvalues in the half-open interval
(VL, VU] w ill be found. = I': the IL th through $\mathbb{I U}$-th eigenvalues $w$ illbe found.

UPLO (input)
$=U U^{\prime}:$ Uppertriangles of $A$ and $B$ are stored;
$=\mathbb{L}$ ': Low ertriangles of $A$ and $B$ are stored.

N (input) The order of them atriges $A$ and $B . N>=0$.
KA (input)
The num ber of superdiagonals of the $m$ atrix $A$ if
$\mathrm{UPLO}=\mathrm{U}$ ', orthe num berof subdiagonals ifU PLO
$=\mathbb{L}^{\prime} . \mathrm{KA}>=0$ 。

K B (input)
The num ber of superdiagonals of the $m$ atrix $B$ if U PLO $=\mathrm{U}$ ', orthe num ber of subdiagonals if U PLO
$=L^{\prime} . \mathrm{KB}>=0$ 。

A B (input/output)
O n entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstka+1 row s of the array. The $j$ th colum n ofA is stored in the $j$ th colum $n$ of the array A B as follow s: if $\mathrm{UPLO}=\mathrm{U}$ ', AB $(k a+1+i-j)=A(i, j)$ for $m a x(1, j$ $\mathrm{ka})<=i<=j$ ifUPLO $={ }^{\prime}$ ', AB $(1+i-j, j)=A(i, j)$ for $\dot{j}=i<=m$ in $(n, j+k a)$.

On exit, the contents of A B are destroyed.

LD A B (input)
The leading dim ension of the array AB. LDAB >= $K A+1$.

BB (input/output)
O $n$ entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $B$, stored in the firstkb+1 row s of the anray. The $j$ th colum n of $B$ is stored in the $j$ th colum n of the array BB as follow $s$ : if $\mathrm{UPLO}=\mathrm{U}, \mathrm{BB}(\mathrm{ka}+1+i-j)=\mathrm{B}(1,1)$ for $\mathrm{max}(1, j$ $\mathrm{kb})<=\dot{i}<=\dot{j}$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{BB}(1+i-j, j)=B(i, 7)$ for $j<=i<=m$ in $(n, j+k b)$.

On exit, the factors from the splitCholesky factorization $B=S * * T * S$, as retumed by SPBSTF .

LD BB (input)
The leading dim ension of the array BB. LD B B >= K B +1 .

Q (output)
If $\mathrm{OBBZ}=\mathrm{V}^{\prime}$, the $\mathrm{n}-\mathrm{by}-\mathrm{n}$ matrix used in the reduction of $A{ }_{x}=\left(\operatorname{lam}\right.$ bda) ${ }^{*} \mathrm{~B} *_{x}$ to standard form, i.e. $\mathrm{C} *_{\mathrm{x}}=(\operatorname{lam} . \mathrm{bda}){ }^{*} \mathrm{x}$, and consequently C to tridiagonal form. If $\operatorname{OBB}=\mathrm{N}$ ', the amray $Q$ is not referenced.

LD Q (input)
The leading dim ension of the array $Q$. If $J B Z=$ $N^{\prime}, L D Q>=1$. If $O B Z=V^{\prime}, L D Q>=\max (1, N)$.

VL (input)
IfRANGE=V', the low er and upper bounds of the interval to be searched foreigenvalues. VL < VU. N otreferenced ifRANGE=A'or I'.

V U (input)
See the description ofV L .

II (input)
If RA N GE= I', the indiges (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{L}<=\mathbb{U}<=N$, if $N>0 ; \Pi=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N otreferenced ifRANGE $=$ A 'or V'.

IU (input)
See the description of II .

ABSTOL (input)
The absolute emortolerance for the eigenvalues. A $n$ approxim ate eigenvalue is accepted as converged $w$ hen it is determ ined to lie in an interval [a,b]
of w idth less than orequal to
ABSTOL + EPS * max $(|,||$,$) ,$
where EPS is them achine precision. IfABSTOL is less than or equal to zero, then EPS* $\mid$ | w illbe used in its place, w here $\mathrm{F} \mid$ is the 1 -norm of the tridiagonal $m$ atrix obtained by reducing $A$ to tridiagonalform.

E igenvalues w illbe com puted m ostaccurately when ABSTOL is set to tw ice the underflow threshold $2 *$ SLA M CH (S ), not zero. If this routine retums w ith $\mathbb{N} F O>0$, indicating that som e eigenvectors did not converge, try setting ABSTO L to $2 \star$ SLAM CH (S ).

M (output)
The totalnum ber ofeigenvalues found. $0<=\mathrm{M}$ <= N . IfRANGE $=A \prime, \mathrm{M}=\mathrm{N}$, and ifRANGE $=\mathrm{I}^{\prime}, \mathrm{M}=$ $\mathbb{I}-\mathbb{L}+1$.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.

Z (input) If $\mathrm{JOBZ}=\mathrm{V}$ ', then if $\mathbb{N} F O=0, \mathrm{Z}$ contains the $m$ atrix $Z$ ofeigenvectors, $w$ th the $i$-th colum $n$ of $Z$ holding the eigenvector associated $w$ ith $W$ (i). The eigenvectors are norm alized so $\mathrm{Z} * * \mathrm{~T} * \mathrm{~B} * \mathrm{Z}=\mathrm{I}$. If $J O B Z=N$ ', then $Z$ is not referenced.

LD Z (input)
The leading dim ension of the array Z . LD $\mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{max}(1, \mathrm{~N})$.

W ORK (w orkspace)
dim ension ( $7 * \mathrm{~N}$ )

IN ORK (w orkspace/output)
dim ension $(5 * N)$

FAII (input)
If $\mathrm{JOBZ}=\mathrm{V}$ ', then if $\mathbb{N F O}=0$, the firstM ele-
$m$ ents of $\mathbb{F} A \mathbb{I}$ are zero. If $\mathbb{N} F O>0$, then $\mathbb{F} A \mathbb{I}$
contains the indioes of the eigenvalues that
failed to converge. If $\mathrm{JOBZ}=\mathrm{N}$ ', then $\mathbb{F} A \mathbb{I}$ is
not referenced.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0$ : if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an illegalvalue
$<=\mathrm{N}:$ if $\mathbb{N} \mathrm{FO}=\mathrm{i}$, then ieigenvectors failed to converge. Their indioes are stored in $\mathbb{F A} \mathbb{I} .>\mathrm{N}$ :SPBSTF retumed an errorcode; ie., if $\mathbb{N N} F=\mathrm{N}$
$+i$, for $1<=i<=N$, then the leading $m$ inor of orderiofB is notpositive definite. The factorization of B could not be com pleted and no eigenvalues or eigenvectors w ere com puted.

## FURTHER DETAILS

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv. of K entucky, USA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dsom v -perform the m atrix-vectoroperation $\mathrm{y}:=\operatorname{alpha} \mathrm{A} \mathrm{A} *$

+ beta*y


## SYNOPSIS

```
SUBROUT\mathbb{NE DSBM V (UPLO,N,K,ALPHA,A,LDA,X,INCX,BETA,Y,}
    INCY)
CHARACTER * 1UPLO
\mathbb{NTEGERN,K,LDA,INCX,}\mathbb{N}CY
DOUBLE PRECISION ALPHA,BETA
D OUBLE PRECISION A (LDA,*),X (*),Y (*)
SU BROU T\mathbb{NE D SBM V_64 (UPLO ,N,K,A LPHA,A,LDA,X, IN CX,BETA,Y,}
    INCY)
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N,K,LDA, NNCX,}\mathbb{N}CY
DOUBLE PRECISION ALPHA,BETA
D OU BLE PRECISION A (LDA,*),X (*),Y (*)
```


## F95 INTERFACE

SU BROUTINE SBMV (UPLO, $\mathbb{N}], K, A L P H A, A,[L D A], X,[\mathbb{N C X}], B E T A$, Y, [ $\mathbb{N} C Y])$

CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER ::N,K,LDA, $\mathbb{N C X}, \mathbb{N} C Y$
REAL (8) :: A LPHA, BETA
REAL (8), D IM ENSION (:) :: X,Y
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::A
SU BROUTINE SBM V_64 (UPLO, $\mathbb{N}], K, A L P H A, A,[L D A], X,[\mathbb{N} C X]$,

CHARACTER (LEN=1) :: UPLO
$\mathbb{N} T E G E R(8):: N, K, L D A, \mathbb{N C X}, \mathbb{N C Y}$
REAL (8) ::ALPHA,BETA
REAL (8), D $\mathbb{M}$ ENSION (:) :: X , Y
REAL (8), D $\mathbb{I M}$ ENSION (:,:) ::A

## C INTERFACE

\#include <sunperfh>
void dsbm $v$ (charuplo, intn, intk, double alpha, double *a, int lda, double *x, int incx, double beta, double *y, intincy);
void dsbm v_64 (charuplo, long $n$, long $k$, double alpha, double *a, long lda, double *x, long incx, double beta, double *y, long incy);

## PURPOSE

dsom v perform sthe $m$ atrix-vector operation $y:=$ alpha*A *x + beta* $y$, w here alpha and beta are scalars, $x$ and $y$ are $n$ ele$m$ ent vectors and $A$ is an $n$ by $n$ sym $m$ etric band $m$ atrix, $w$ ith k super-diagonals.

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the band $m$ atrix $A$ is being supplied as follow s:
$\mathrm{UPLO}=\mathrm{U}$ 'or L ' The upper triangularpart of $A$ is being supplied.

UPLO = L'or I' The low er triangularpart of A is being supplied.

U nchanged on exit.

N (input)
O n entry, $N$ specifies the order of the $m$ atrix A. $\mathrm{N}>=0$. U nchanged on exit.

K (input)
On entry, $K$ specifies the number of superdiagonals of them atrix A.K $>=0$. U nchanged on
exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
Before entry w ith UPLO = U 'or G ', the leading ( $k+1$ ) by $n$ part of the array A $m$ ust contain the upper triangular band part of the symmetric $m$ atrix, supplied colum $n$ by colum $n$, $w$ th the leading diagonal of the $m$ atrix in row ( $k+1$ ) of the array, the first super-diagonalstarting atposition 2 in row $k$, and so on. The top left $k$ by $k$ triangle of the anray $A$ is not referenced. The follow ing program segm entw illtransfer the upper triangular part of a sym $m$ etric band $m$ atrix from conventional fullm atrix storage to band storage:

$$
\begin{aligned}
& \text { DO } 20, J=1, N \\
& \text { M }=\mathrm{K}+1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{M} A X(1, \mathrm{~J}-\mathrm{K}), \mathrm{J} \\
& \text { A }(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \text { CONTINUE } \\
& 20 \text { CONTINUE }
\end{aligned}
$$

Before entry w ith UPLO = L 'or 1', the leading ( $\mathrm{k}+1$ ) by n partof the array A m ustcontain the low er triangular band part of the symmetric $m$ atrix, supplied colum $n$ by colum $n$, $w$ th the leading diagonalof them atrix in row 1 of the amay, the first sub-diagonalstarting atposition 1 in row 2 , and so on. The bottom right k by $k$ triangle of the array $A$ is not referenced. The follow ing program segm entw illtransfer the low ertriangular part of a sym $m$ etric band $m$ atrix from conventional fullm atrix storage to band storage:

$$
\begin{aligned}
& \text { DO } 20, J=1, N \\
& \mathrm{M}=1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{J}, \mathrm{M} \mathbb{N}(\mathrm{~N}, \mathrm{~J}+\mathrm{K}) \\
& \mathrm{A}(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \text { CONTINUE } \\
& 20 \mathrm{CONTINUE}
\end{aligned}
$$

U nchanged on exit.

LDA (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD A $>=$ (
$\mathrm{k}+1$ ). Unchanged on exit.
X (input)
$(1+(n-1) * a b s(\mathbb{N} C X))$. Before entry, the increm ented array $X$ must contain the vectorx.
U nchanged on exit.
$\mathbb{N C X}$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents ofX. $\mathbb{N} C X<>0$. U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. U nchanged on exit.

Y (input/output)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))$. Before entry, the increm ented array $Y$ m ustcontain the vectory. On exit, $Y$ is overw ritten by the updated vectory.
$\mathbb{N C Y}$ (input)
O n entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $Y$. $\mathbb{N C Y}$ <> 0 . Unchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

dsbtrd -reduce a realsym $m$ etric band $m$ atrix $A$ to sym $m$ etric tridiagonal form T by an orthogonal sim ilarity transform ation

## SYNOPSIS

```
SUBROUT\mathbb{NEDSBTRD NECT,UPLO,N,KD,AB,LDAB,D,E,Q,LDQ,W ORK,}
    INFO)
CHARACTER * 1 VECT,UPLO
NNTEGER N,KD,LDAB,LDQ,NNFO
DOUBLE PRECISION AB (LDAB,*),D (*),E (*),Q (LDQ,*),W ORK (*)
SUBROUT\mathbb{NEDSBTRD_64NECT,UPLO,N,KD,AB,LDAB,D,E,Q,LDQ,W ORK,}
    \mathbb{NFO)}
CHARACTER * 1 VECT,UPLO
\mathbb{NTEGER*8N,KD,LDAB,LDQ,NNFO}
DOUBLE PRECISION AB (LDAB,*),D (*),E (*),Q (LDQ & *),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE SBTRD (NECT,UPLO, $\mathbb{N}], K D, A B,[L D A B], D, E, Q,[L D Q]$, [W ORK], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1)::VECT,UPLO
$\mathbb{N} T E G E R:: N, K D, L D A B, L D Q, \mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::D ,E,W ORK
REAL (8), D IM ENSION (:,:) ::AB,Q
SU BROUTINE SBTRD_64NECT, UPLO, $\mathbb{N}], K D, A B,[L D A B], D, E, Q,[L D Q]$, [ $\mathbb{W}$ ORK], [ $\mathbb{N F O}$ ])

CHARACTER ( $\llcorner E N=1$ ) : : VECT, UPLO
$\mathbb{N} \operatorname{TEGER}(8):: N, K D, L D A B, L D Q, \mathbb{N} F O$
REAL (8), D $\mathbb{M}$ ENSION (:) ::D ,E,W ORK
REAL (8), D $\mathbb{I}$ ENSION (:,:) ::AB, Q

## C INTERFACE

\#include <sunperfh>
void dsbtrd (charvect, charuplo, intn, int kd, double *ab, int ldab, double *d, double *e, double *q, int ldq, int *info);
void dsotrd_64 (charvect, charuplo, long n, long kd, double *ab, long ldab, double *d, double *e, double *q, long ldq, long *info);

## PURPOSE

dsbtrd reduces a real sym $m$ etric band $m$ atrix $A$ to sym $m$ etric tridiagonal form T by an orthogonalsim ilarity transform ation: $\mathrm{Q} * * \mathrm{~T} * A * Q=T$.

## ARGUMENTS

VECT (input)
$=\mathrm{N}$ : do notform Q ;
$=\mathrm{V}$ : form Q ;
$=U$ : update a $m$ atrix $X$, by form ing $X * Q$.

UPLO (input)
$=\mathrm{U}$ ': Uppertriangle of A is stored;
$=\mathbb{L}:$ Low ertriangle ofA is stored.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

KD (input)
The num ber of superdiagonals of the $m$ atrix $A$ if $\mathrm{UPLO}=\mathrm{U}$ ', or the num ber of subdiagonals ifUPLO $=\mathbb{L}^{\prime} . \mathrm{KD}>=0$ 。

A B (input/output)
O n entry, the upper or low ertriangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstK $D+1$ row s of the array. The $j$ th colum n of A is stored in the $j$ th colum n of the array A B as follow s: if $\mathrm{UPLO}=\mathrm{U}$ ', AB $(k d+1+i-j, j)=A(i, j)$ for $m a x(1, j$
$\mathrm{kd})<=i<=\dot{j}$ ifUPLO $=\mathrm{L}, \mathrm{AB}(1+i-j)=A(i, 7)$ for $j=i<=m$ in $(n, j+k d)$. O $n$ exit, the diagonalele$m$ ents of AB are overw ritten by the diagonalele$m$ ents of the tridiagonalm atrix $T$; if $K D>0$, the elem ents on the first superdiagonal (if UPLO = U ) or the first subdiagonal (ifU PLO = L ) are overw ritten by the off-diagonalelem ents of $T$; the restofA $B$ is overw ritten by values generated during the reduction.

LD AB (input)
The leading dim ension of the array AB. LD AB >= K D +1 .

D (output)
The diagonalelem ents of the tridiagonalm atrix T .
E (output)
The off-diagonal elem ents of the tridiagonal $m$ atrix $T: E(i)=T(i, i+1)$ if $U P L O=U ; E(i)=$ $T(i+1, i)$ if $\mathrm{PLO}=\mathrm{L}^{\prime}$.

Q (input/output)
On entry, ifVECT $=U$ ', then Q must contain an N by $-\mathrm{N} m$ atrix X ; if $\mathrm{VECT}=\mathrm{N}$ 'or $V$ ', then $Q$ need notbe set.

On exit: if $\mathrm{VECT}=\mathrm{V}$ ', Q contains the N -by -N orthogonalm atrix $Q$; if VECT $=U$ ', $Q$ contains the product $\mathrm{X} * \mathrm{Q}$; ifVECT $=\mathrm{N}$ ', the array Q is not referenced.

LDQ (input)
The leading dim ension of the array $\mathrm{Q} . \mathrm{LD} \mathrm{Q}>=1$, and $L D Q>=N$ ifVECT $=V$ 'or $U '$.

W ORK (w orkspace)
dim ension (N)
$\mathbb{I N F O}$ (output)
= 0 : successfulexit
< $0:$ if $\mathbb{N} F O=-i$, the $i$-th argum enthad an illegalvalue

## FURTHER DETAILS

M odified by Linda K aufm an, BellLabs.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
dscal-C om pute y := alpha * y
```


## SYNOPSIS

```
SUBROUTINEDSCALN,ALPHA,Y,INCY)
\mathbb{NTEGERN,\mathbb{NCY}}\mathbf{N}=\mp@code{N}
DOUBLE PRECISION ALPHA
DOUBLE PRECISION Y (*)
SUBROUTINE DSCAL_64 N,ALPHA,Y, INCY)
\mathbb{NTEGER*8N,\mathbb{NCY}}\mathbf{N}=1
DOUBLE PRECISION ALPHA
DOUBLE PRECISION Y (*)
F95 INTERFACE
```



```
\mathbb{NTEGER ::N,\mathbb{NCY}}\mathbf{N}=\mp@code{N}
REAL (8) ::A LPHA
REAL (8),D IM ENSION (:) ::Y
SU BROUTINE SCAL_64 (N ],ALPHA,Y,[\mathbb{N CY ])}
\mathbb{NTEGER (8) ::N,\mathbb{NCY}}\mathbf{T}=\mp@code{N}
REAL (8) ::A LPHA
REAL (8),D\mathbb{M ENSION (:) ::Y}
```


## C INTERFACE

\#include <sunperfh>
void dscal(intn, double alpha, double *y, int incy);
void dscal 64 (long n, double alpha, double *y, long incy);

## PURPOSE

dscalC om pute $y:=$ alpha * $y$ w here alpha is a scalar and $y$ is an n-vector.

## ARGUMENTS

N (input)
O $n$ entry, $N$ specifies the num ber of elem ents in the vector. $N$ must be at least one for the subroutine to have any visible effect. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

Y (input/output)
(1+(n-1)*abs( $\mathbb{N} C Y)$ ). On entry, the increm ented array $Y$ m ustcontain the vectory. On exit, $Y$ is overw rilten by the updated vectory.
$\mathbb{N C C Y}$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $Y . \mathbb{I N C Y}$ m ustnotbe zero. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS


## NAME

dsctr-Scatters elem ents from $x$ into $y$.

## SYNOPSIS

```
SUBROUTINEDSCTR NZ,X,NNDX,Y)
DOUBLE PRECISION X (*),Y (*)
INTEGER NZ
INTEGER \mathbb{NDX (*)}
SUBROUT\mathbb{NEDSCTR_64 NZ,X,\mathbb{NDX,Y)}}\mathbf{N}=(
DOUBLE PRECISION X (*),Y (*)
INTEGER*8NZ
INTEGER*8 \mathbb{NDX (*)}
F95 INTERFACE
SUBROUTINE SCTR (NZ],X,NDD,Y)
REAL (8),D IM ENSION (:) ::X,Y
\mathbb{NTEGER ::NZ}
\mathbb{NTEGER,D IM ENSION (:) :: \mathbb{NDX}}\mathbf{N}=\mp@code{N}
SUBROUT\mathbb{NE SCTR_64(NZ],X,NNDX,Y)}
REAL (8),D IM ENSION (:) ::X,Y
INTEGER (8)::NZ
\mathbb{NTEGER (8),D IM ENSION (:) :: \mathbb{NDX}}\mathbf{N}={
```


## PURPOSE

D SCTR -Scatters the com ponents of a sparse vector $x$ stored in com pressed form into specified com ponents of a vectory
in fullstorage form .
do $i=1, n$
$y($ indx (i) $)=x(i)$
enddo

## ARGUMENTS

N Z (input) - $\mathbb{N}$ TEGER
N um ber of elem ents in the com pressed form .
U nchanged on exit.

X (input)
V ector containing the values to be scattered from com pressed form into fill storage form. U nchanged on exit.
$\mathbb{N} D X$ (input) $-\mathbb{N} T E G E R$
$V$ ector containing the indiges of the com pressed form. It is assum ed that the elem ents in $\mathbb{N} D \mathrm{X}$ are distinctand greater than zero. U nchanged on exit.

Y (output)
V ectorw hose elem ents specified by indx have been set to the corresponding entries ofx. Only the elem ents corresponding to the indices in indx have been m odified.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

dsdot-com pute the double precision dotproductoftw o single precision vectors x and y .

## SYNOPSIS

```
    DOUBLE PRECISION FUNCTION DSDOTN,X,\mathbb{NCX,Y,INCY)}
    INTEGERN,\mathbb{NCX,\mathbb{NCY}}\mathbf{}\mathrm{ \}
    REALX (*),Y (*)
    DOUBLE PRECISION FUNCTION DSDOT_64 N,X,INCX,Y,INCY)
    INTEGER*8N,\mathbb{NCX,INCY}
    REALX (*),Y (*)
F95 INTERFACE
    REAL (8) FUNCTIONDSDOTN,X,INCX,Y,INCY)
    \mathbb{NTEGER::N,\mathbb{NCX,INCY}}\mathbf{N}={
    REAL,D IM ENSION (:) ::X,Y
```




```
    REAL,D IM ENSION (:) ::X,Y
C INTERFACE
    #include <sunperfh>
    double dsdot(intn, float *x, int incx, float *y, int incy);
    double dsdot_64 (long n, float *x, long incx, float *y, long
```


## PURPOSE

dsdot com pute the double precision dotproductof $x$ and $y$ $w$ here $x$ and $y$ are single precision $n$-vectors.

## ARGUMENTS

N (input)
O n entry, N specifies the num ber of elem ents in the vector. IfN is not positive then the function retums the value 0.0 . U nchanged on exit.
$X$ (input)
( $1+(\mathrm{n}-1) * \mathrm{abs}(\mathbb{N} C X)$ ). On entry, the
increm ented array $X$ must contain the vectorx. U nchanged on exit.
$\mathbb{N C X}$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of X. $\mathbb{N} C X$ m ustnotbe zero. U nchanged on exit.
$Y$ (input)
( $1+(\mathrm{n}-1) * \mathrm{abs}(\mathbb{N} C Y)$ ). On entry, the increm ented array $Y$ m ust contain the vectory. U nchanged on exit.
$\mathbb{N C Y}$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of Y. $\mathbb{N} C Y$ m ustnotbe zero. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE


## NAME

```
dsecnd - retum the user tim e for a process in seconds
```


## SYNOPSIS

DOUBLE PRECISION FUNCTION DSECND ()

DOUBLE PRECISION FUNCTION DSECND_640

## F95 INTERFACE

REAL (8) FUNCTION DSECND ()
REAL (8) FUNCTION DSECND_640)
C INTERFACE
\#include < sunperfh>
double dsecnd ();
double dsecnd_64 ();

## PURPOSE

dsecnd retums the user tim e for a process in seconds. This version gets the tim efrom the system function ETIME.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

dsinqb -synthesize a Fourier sequence from its representation in term sofa sine series $w$ th odd $w$ ave num bers. The SIN Q operations are unnorm alized inverses of them selves, so a call to $S \mathbb{N} Q F$ follow ed by a call to $S \mathbb{N} Q B$ w illm ultiply the inputsequence by 4 * $N$.

## SYNOPSIS

```
SUBROUT\mathbb{NEDSINQB N,X,W SAVE)}
```

$\mathbb{N}$ TEGER N
DOUBLE PRECISION X (*),W SAVE (*)
SU BROUTINEDSINQB_64 $\mathbb{N}, \mathrm{X}, \mathrm{W}$ SAVE)
$\mathbb{N}$ TEGER*8 N
DOUBLE PRECISION X (*), W SAVE (*)

F95 INTERFACE
SU BROUTINESINQB ( $\mathbb{N}$ ],X,W SAVE)
$\mathbb{N} T E G E R:: N$
REAL (8), D $\mathbb{M}$ ENSION (:) ::X,W SAVE

SU BROUTINE SNQB_64 ( $\mathbb{N}$ ],X,W SAVE)
$\mathbb{N} T E G E R(8):: N$
REAL (8),D $\mathbb{M}$ ENSION (:) ::X,W SAVE

## C INTERFACE

\#include <sunperfh>
void dsinqb (intn, double *x, double *w save);

```
void dsinqb_64 (long n, double *x, double *w save);
```


## ARGUMENTS

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product
of sm allprim es. $\mathrm{N}>=0$.
X (input/output)
On entry, an array of length N containing the sequence to be transform ed. On exit, the quarterw ave sine synthesis of the input.
W SAVE (input)
O n entry, an array with dim ension of at least (3 *
$\mathrm{N}+15$ ) for scalar subroutines, initialized by
$S \mathbb{N} Q$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

dsinqf-com pute the Fouriercoefficients in a sine series representation w ith only odd w ave num bers. The $S \mathbb{N} Q$ operations are unnorm alized inverses of them selves, so a call to $S \mathbb{N} Q F$ follow ed by a call to $S \mathbb{N} Q B$ w illm ultiply the input sequence by 4 * $N$.

## SYNOPSIS

$$
\text { SU BROUTINEDSINQF } \mathbb{N}, \mathrm{X}, \mathrm{~W} \text { SAVE) }
$$

$\mathbb{N}$ TEGER N
DOUBLE PRECISIONX (*) , W SAVE (*)
SU BROUTINEDSINQF_64 $\mathbb{N}, \mathrm{X}, \mathrm{W}$ SAVE)
$\mathbb{N}$ TEGER*8N
DOUBLE PRECISION X (*), W SAVE (*)
F95 INTERFACE
SUBROUTINESTNQF( $\mathbb{N}], X, W$ SAVE)
$\mathbb{N} T E G E R:: N$
REAL (8),D $\mathbb{M}$ ENSION (:) ::X,W SAVE
SUBROUTINESINQF_64( $\mathbb{N}$ ], $X, W$ SAVE)
$\mathbb{N}$ TEGER (8) :: N
REAL (8), D IM ENSION (:) ::X,W SAVE

## C INTERFACE

\#include <sunperfh>
void dsingf(intn, double *x, double *w save);

## ARGUMENTS

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product of sm allprim es. $\mathrm{N}>=0$.

X (input/output)
On entry, an array of length N containing the sequence to be transform ed. On exit, the quarter-w ave sine transform of the input.
W SAVE (input)
On entry, an array with dim ension of at least (3

* $N+15$ ) for scalar subroutines, initialized by
$S \mathbb{N} Q$ I.


## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

dsinqi-initialize the array xW SAVE, which is used in both $S \mathbb{N} Q F$ and $S \mathbb{N} Q B$.

## SYNOPSIS

> SU BROUTINEDSINQIN,W SAVE)
$\mathbb{N}$ TEGER N
DOUBLE PRECISION W SAVE (*)
SU BROUTINEDSINQI_64 $\mathbb{N}, W$ SAVE)
$\mathbb{N}$ TEGER*8 N
DOUBLE PRECISION W SAVE (*)

## F95 INTERFACE

SU BROUTINESINQIN,W SAVE)
$\mathbb{N} T E G E R:: N$
REAL (8),D IM ENSION (:) ::W SAVE

SUBROUTINESINQI_64 (N,W SAVE)
$\mathbb{N} T E G E R(8):: N$
REAL (8), D $\mathbb{M}$ ENSION (:) ::W SAVE

## C INTERFACE

\#include <sunperfh>
void dsinqi(intn, double *w save);
void dsinqi_ 64 (long n, double *w save);

## ARGUMENTS

N (input) Length of the sequence to be transform ed. The m ethod is m ost efficientw hen N is a product of sm allprim es.

W SAVE (input)
On entry, an array ofdim ension ( 3 * $\mathrm{N}+15$ ) or greater. SIN Q I needs to be called only once to intialize W SA VE before calling $S \mathbb{N} Q F$ and/orS $\mathbb{N} Q B$ if N and W SAVE rem ain unchanged betw een these calls. Thus, subsequent transform s or inverse transforms of same size can be obtained faster than the first since they do not require initialization of the w orkspace.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

dsint-com pute the discrete Fourier sine transform of an odd sequence. The $S \mathbb{N} T$ transform s are unnorm alized inverses of them selves, so a callofSIN T follow ed by another call of SNN T w illm ultiply the input sequence by 2 * $(\mathbb{N}+1)$.

## SYNOPSIS

$$
\text { SU BROUTINEDSINT } \mathbb{N}, \mathrm{X}, \mathrm{~W} \text { SAVE) }
$$

$\mathbb{N}$ TEGER N
DOUBLE PRECISIONX (*), W SAVE (*)
SU BROUTINEDSINT_64 $\mathbb{N}, \mathrm{X}, \mathrm{W}$ SAVE)
$\mathbb{N}$ TEGER*8N
DOUBLE PRECISIONX (*),W SAVE (*)

## F95 INTERFACE

SU BROUTINE SNT ( $\mathbb{N}$ ],X,W SAVE)
$\mathbb{N} T E G E R:: N$
REAL (8),D $\mathbb{M}$ ENSION (:) ::X,W SAVE
SU BROUTINE SINT_64 (N ],X,W SAVE)
$\mathbb{N}$ TEGER (8) :: N
REAL (8),D $\mathbb{M}$ ENSION (:) ::X,W SAVE

## C INTERFACE

\#include <sunperfh>
void dsint(intn, double *x, double *w save);

## ARGUMENTS

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen $\mathrm{N}+1$ is a productof sm all prim es. $\mathrm{N}>=0$.

X (input/output)
Onentry, an aray of length N containing the sequence to be transform ed. On exit, the sine transform of the input.
W SAVE (input/output)
On entry, an array w ith dimension of at least int(2.5*N + 15) initialized by SIN TI.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

dsinti-initialize the anray W SA VE, which is used in subroutine $S \mathbb{N}$ T .

## SYNOPSIS

```
SUBROUT\mathbb{NE DSINTIN,W SAVE)}
INTEGER N
DOUBLE PRECISION W SAVE (*)
SUBROUT\mathbb{NEDSINTI_64 N,W SAVE)}
INTEGER*8 N
DOUBLE PRECISION W SAVE (*)
F95 INTERFACE
    SUBROUT\mathbb{NESINTIN,W SAVE)}
    \mathbb{NTEGER ::N}
    REAL (8),D IM ENSION (:) ::W SAVE
    SUBROUTINESINTI_64N,W SAVE)
    INTEGER (8) ::N
    REAL (8),D IM ENSION (:) ::W SAVE
C INTERFACE
    #include <sunperfh>
    void dsinti(intn, double *W save);
    void dsinti_64 (long n, double *w save);
```


## ARGUMENTS

N (imput) Length of the sequence to be transform ed. $\mathrm{N}>=0$.

W SAVE (input/output)
On entry, an array ofdim ension ( $2 \mathrm{~N}+\mathrm{N} / 2+15$ ) or greater. SIN TI is called once to initializeW SA VE before calling $S \mathbb{N} T$ and need notbe called again betw een calls to SINT if N and W SAVE rem ain unchanged. Thus, subsequent transform s of sam e size can be obtained faster than the first since they do not require initialization of the w orkspace.

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

dskym m -Skyline form atm atrix-m atrix multiply

## SYNOPSIS

```
SUBROUT\mathbb{NEDSKYMM(TRANSA,M,N,K,ALPHA,DESCRA,}
* VAL,PNTR,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER TRANSA,M,N,K,DESCRA (5),
* LDB,LDC,LWORK
\mathbb{NTEGER PNTR(*),}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
```

SU BROUTINEDSKYMM_64(TRANSA, M,N,K,ALPHA,DESCRA,

* VAL,PNTR,
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
$\mathbb{N}$ TEGER*8 TRANSA, $\mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{DESCRA}$ (5),
* LDB,LDC,LW ORK
$\mathbb{N} T E G E R * 8 \operatorname{PNTR}$ (*),
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL (NNZ),B (LDB,*), C (LDC, $\left.{ }^{\star}\right)$, W ORK (LW ORK)
where NNZ = PNTR ( $(\mathbb{1}+1)$ PN TR ( 1 ) (upper triangular)
NN Z = PNTR $M+1$ ) PNTR (1) (low er triangular)
PN TR 0 ) size $=(\mathbb{K}+1)$ (upper triangular)
PNTR () size $=(M+1)$ (low ertriangular)


## F95 INTERFACE

SUBROUTINESKYMM (TRANSA, M, $\mathbb{N}], K, A L P H A, D E S C R A, V A L$,

* PNTR, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
$\mathbb{N} T E G E R$ TRANSA, M, K
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: DESCRA, PNTR

DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION ,D $\mathbb{M}$ ENSION (:) :: VAL
DOUBLE PRECISION ,D IM ENSION (:, :) :: B , C

SUBROUTINE SKYM M_64 (TRANSA, M, $\mathbb{N}], K, A L P H A, D E S C R A, V A L$,

* PNTR, B, [LDB],BETA, C , [LDC], [W ORK], [LW ORK])
$\mathbb{N}$ TEGER*8 TRANSA, M, K
$\mathbb{N} T E G E R * 8, D \mathbb{M}$ ENSION (:) :: DESCRA, PNTR
DOUBLEPRECISION ALPHA,BETA
DOUBLE PRECISION,D $\mathbb{I M} E N S I O N(:):: V A L$
DOUBLE PRECISION ,D IM ENSION (:, :) :: B , C


## DESCRIPTION

C <-alpha op (A ) B + beta C
where A LPHA and BETA are scalar, $C$ and $B$ are dense $m$ atrices, $A$ is a m atrix represented in skyline form at and op (A) is one of
$o p(A)=A$ or $o p(A)=A^{\prime}$ or op $(A)=\operatorname{conjg}\left(A^{\prime}\right)$.
( 'indicates m atrix transpose)

## ARGUMENTS

TRANSA Indicates how to operate $w$ th the sparse $m$ atrix
0 : operate w ith m atrix
1 : operate $w$ th transpose $m$ atrix
2 : operate $w$ ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M $\quad N$ um ber of row $s$ in $m$ atrix A

N $\quad N$ um ber of colum ns in m atrix $C$

K $\quad \mathrm{N}$ um berof colum ns in matrix A

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
D ESCRA (1) m atrix structure
0 : general $\mathbb{N O T}$ SUPPORTED )
1 : symm etric ( $A=A$ )
2: Herm itian ( $A=C O N J G(A)$ )
3 : Triangular
4 : Skew (A nti)-Symm etric ( $A=-A$ )
5 :D iagonal
6 : Skew Herm titian ( $A=-\operatorname{CON}$ J ( $A$ ) )

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 :non-unit
1 : unit
DESCRA (4) A ray base $\mathbb{N} O T \mathbb{M}$ PLEM ENTED)
0 :C C++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? $\mathbb{N} O T \mathbb{I}$ PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () array contain the nonzeros ofA in skyline profile form. Row -oriented ifD ESCRA (2) = 1 (low er triangular), colum n oriented ifD ESCRA (2) = 2 (upper triangular).
PNTR 0) integer anay of length $M+1$ (low ertriangular) or $\mathrm{K}+1$ (upper triangular) such thatPN TR (I) PN TR (1)+1 points to the location in VAL of the firstelem entof the skyline profile in row (colum n) I.

B 0 rectangular array w ith first dim ension LD B.
LD B leading dim ension of $B$
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension of $C$

W ORK ( scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is notreferenced in the currentversion.

## SEE ALSO

N IST FO RTRAN Sparse B las U ser's G uide available at: http://m ath nistgov/n csd/Staff/k Rem ington/Aspoblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

## NOTES/BUGS

The SK Y data structure is not supported for a generalm atrix structure (DESCRA (1)=0).

A lso not supported:

1. low er triangularm atrix $A$ of size $m$ by $n$ where $m>n$
2. uppertriangularm atrix $A$ of size $m$ by $n$ where $m<n$

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

dskysm -Skyline form at triangular solve

## SYNOPSIS

```
SUBROUTINE D SKYSM(TRANSA,M ,N ,UNITD,DV,A LPHA,DESCRA,
* VAL,PNTR,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,M,N,UNITD,DESCRA (5),}
* LDB,LDC,LWORK
\mathbb{NTEGER PNTR (*),}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION DV M),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
```

SUBROUTINEDSKYSM_64(TRANSA, M,N,UNITD,DV,ALPHA,DESCRA,

* VAL,PNTR,
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
$\mathbb{N} T E G E R * 8$ TRANSA, M,N,UNITD,DESCRA (5),
* LDB,LDC,LWORK
$\mathbb{N} T E G E R * 8 \operatorname{PNTR}$ (*),
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION DV (M),VAL $\mathbb{N N Z}), B(L D B, \star), C(L D C, \star), W$ ORK (LW ORK)
where $\mathrm{NN} Z=\operatorname{PN} \operatorname{TR}(\mathrm{M}+1)$-PNTR (1) (uppertriangular)
$\mathrm{NN} Z=\operatorname{PNTR}(\mathbb{K}+1)$-PNTR (1) (low ertriangular)
PNTR 0 size $=(M+1)$ (uppertriangular)
PNTR 0 ) size $=(\mathbb{K}+1)$ (low ertriangular)


## F95 INTERFACE

SUBROUTINE SKYSM (TRANSA,M, $\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L$, * PNTR, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
$\mathbb{N} T E G E R$ TRANSA,M,UNITD
$\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:)::$ DESCRA, PNTR

DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION ,D $\mathbb{I M} E N S I O N(:):: V A L, D V$
D OUBLE PRECISION ,D IM ENSION (:, :) :: B , C

SU BROUTINE SKY SM _64 (TRANSA, M, $\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A$,

* VAL,PNTR, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
$\mathbb{N} T E G E R * 8$ TRANSA, M, UNITD
$\mathbb{N} T E G E R * 8, D \mathbb{M}$ ENSION (:) :: DESCRA, PNTR
DOUBLEPRECISION ALPHA,BETA
DOUBLE PRECISION ,D $\mathbb{M}$ ENSION (:) :: VAL, DV
DOUBLE PRECISION ,D IM ENSION (:, :) :: B , C


## DESCRIPTION

$$
C<-A L P H A \quad \text { op (A) B + BETA } C \quad C<-A L P H A D \text { op (A) B + BETA C }
$$ $C<-A L P H A \operatorname{Op}(A) D B+B E T A C$ where A LPHA andBETA are scalar, $C$ and $B$ are $m$ by $n$ dense $m$ atrices, $D$ is a diagonalscaling $m$ atrix, $A$ is a unit, ornon-unit, upper or low er triangularm atrix represented in skyline form at and $o p(A)$ is one of

```
op (A ) = inv (A ) or op (A ) = inv (A ) or op (A ) = inv (oonjg (A')).
``` (inv denotesm atrix inverse, 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRA N SA Indicates how to operate \(w\) ith the sparse \(m\) atrix 0 : operate \(w\) ith \(m\) atrix
1 : operate w th transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um ber of row \(s\) in matrix A
\(N \quad N\) um berof colum \(n s\) in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identily \(m\) atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)
4 :A utom atic row orcolum n scaling (see section N O TES for further details)

DV () A rray of length M containing the diagonalentries of the scaling diagonalm atrix D .

ALPHA Scalarparam eter
```

D ESCRA () D escriptor argum ent. Fi̇ve elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric (A=A)
2:H erm itian (A = CON JG (A ))
3:Triangular
4 : Skew (A nti)-Symm etric (A=-A )
5 :D iagonal
6:Skew Herm itian (A= CON JG (A ) )
N ote: For the routine, D ESCRA (1)=3 is only supported.
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 :non-unit
1 :unit
DESCRA (4) A nay base $\mathbb{N} O T \mathbb{M}$ PLEM ENTED)
$0: C / C++$ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? (NOT $\mathbb{M}$ PLEM ENTED)
0 : unknown
1 : no repeated indices

```

VAL () array contain the nonzeros ofA in skyline profile form .
Row -oriented ifD ESCRA (2) = 1 (low er triangular), colum n oriented ifD ESCRA (2) \(=2\) (upper triangular) .

PN TR () integer array of length \(M+1\) (low ertriangular) or
\(\mathrm{K}+1\) (upper triangular) such thatPN TR (I)-PN TR (1)+1 points to the location in VAL of the firstelem ent of the skyline profile in row (colum n) I.

B 0 rectangular anay w ith first dim ension LD B .

LD B leading dim ension ofB

BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch amay of length LW ORK.
On exit, ifLW ORK \(=-1, W\) ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK array. LW ORK should be at leastM .

Forgood perform ance, LW O RK should generally be larger.

For optim um perform ance on \(m\) ultiple processors, LW ORK \(>=M\) *N _CPU S where N_CPUS is the m axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser'S G uide available at:
http://m ath nist.gov/m cso/Staff/K Rem ington/Espblas/
"D ocum ent forthe B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlio .org/utk/papers/sparse _ps

\section*{NOTES /BUGS}
1.A lso notsupported:
a. low er triangularm atrix A ofsizem by n wherem \(>n\)
b. upper triangularm atrix \(A\) of size \(m\) by \(n\) where \(m<n\)
2. N o test for singularity ornear-singularity is included in this routine. Such tests m ust.be perform ed before calling this routine.
3. If U N ITD \(=4\), the routine scales the row s of \(A\) if \(D E S C R A(2)=1\) and the colum ns ofA if \(D E S C R A(2)=2\) such that their 2 -norm s are one. The scaling \(m\) ay im prove the accuracy of the com puted solution. C orresponding entries of V A L are changed only in this particular case. O n retum D V m atrix stored as a vector contains the diagonalm atrix by w hich the row \(s\) (colum ns) have been scaled. U N ITD = 2 if \(D E S C R A(2)=1\) and UN ITD \(=3\) if \(D E S C R A(2)=2\) should be used for the next calls to the routine \(w\) ith overw rilten \(V A L\) and \(D V\).

WORK \((1)=0\) on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here i is the row (colum n) num berw hich 2 -norm is exactly zero.
4. If \(D E S C R A(3)=1\) and \(U \mathrm{~N}\) ITD \(<4\), the unit diagonalelem ents
\(m\) ightorm ightnotbe referenced in the SK Y representation of a sparse m atrix. They are not used anyw ay in these cases. ButifUN ITD = 4, the unit diagonalelem ents M U ST be referenced in the SK Y representation.
5.The routine can be applied for solving triangular system \(s\) w hen the upper or low er triangle of the general sparse \(m\) atrix \(A\) is used. H ow ever DESCRA (1) m ustbe equal to 3 in this case.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dspcon -estim ate the reciprocal of the condition num ber (in
the 1-norm ) of a realsymm etric packedm atrix A using the
factorization A =U*D *U**T or A = L*D *L**T com puted by
D SPTRF

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSPCON (UPLO,N,AP,\mathbb{P}\mathbb{IVOT,ANORM,RCOND,W ORK,IN ORK2,}}\mathbf{N},\textrm{N},\textrm{N}
\mathbb{NFO )}

```
CHARACTER * 1 UPLO
\(\mathbb{N} T E G E R N, \mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{P} \mathbb{V} O T\left({ }^{*}\right)\), \(\mathbb{I N}\) ORK 2 (*)
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION AP (*) , W ORK (*)
SU BROUTINEDSPCON_64 (UPLO,N,AP, \(\mathbb{P} \mathbb{I V O T}, A N O R M, R C O N D, W\) ORK, \(\mathbb{I N}\) ORK2,
    \(\mathbb{N} F O\) )
```

CHARACTER * 1 UPLO

```
\(\mathbb{N}\) TEGER*8 \(\mathrm{N}, \mathbb{N} F \mathrm{O}\)
\(\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{I V O T}\left({ }^{*}\right), \mathbb{I} \operatorname{ORK} 2\) ( \({ }^{*}\) )
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION AP (*), W ORK (*)

\section*{F95 INTERFACE}
 [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V O T}, \mathbb{I N}\) ORK2

REAL (8) ::ANORM,RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::AP,W ORK
SU BROUTINE SPCON_64 (UPLO,N,AP, \(\mathbb{P} \mathbb{I} O T, A N O R M, R C O N D,[W O R K]\), [ \(\mathbb{I W}\) ORK2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathbb{I N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}, \mathbb{I N}\) ORK 2
REAL (8) ::ANORM,RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::AP,W ORK

\section*{C INTERFACE}
\#include < sunperfh>
void dspcon (charuplo, intn, double *ap, int *ipivot, double anorm , double *roond, int *info);
void dspoon_64 (charuplo, long n, double *ap, long *ipivot, double anorm, double * rcond, long *info);

\section*{PURPOSE}
dspcon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a realsymm etric packed \(m\) atrix A using the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by D SPTRF.

A n estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND = 1 / ANORM * norm (inv (A))).

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) ': U pper triangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\);
= L': Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\).
N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
AP (input)
D ouble precision amay, dim ension \((\mathbb{N} * \mathbb{N}+1) / 2\) ) The
block diagonal \(m\) atrix \(D\) and the \(m\) ultipliers used to obtain the factorU orL as com puted by D SPTRF, stored as a packed triangularm atrix.

Integer anray, dim ension \(\mathbb{N}\) ) D etails of the interchanges and the block structure ofD as determ ined by D SPTRF.

\section*{ANORM (input)}

The 1-norm of the originalm atrix A.

\section*{RCOND (output)}

The reciprocal of the condition num ber of the \(m\) atrix \(A\), com puted as RCOND \(=1 /(A N O R M * A \mathbb{N} V N M)\), where \(A \mathbb{N} V N M\) is an estim ate of the 1 -nom of inv (A) com puted in this routine.
W ORK (w orkspace)
D ouble precision array, dím ension ( \(2 * \mathrm{~N}\) )
IV ORK 2 (w orkspace)
Integer array, dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
< 0: if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dspev - com pute all the eigenvalues and, optionally, eigenvectors of a realsym \(m\) etric \(m\) atrix \(A\) in packed storage

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DSPEV (JOBZ,UPLO,N,AP,W,Z,LD Z,W ORK,INFO )}
CHARACTER * 1 JOBZ,UPLO
INTEGERN,LDZ,\mathbb{NFO}
DOU BLE PRECISION AP (*),W (*),Z (LD Z ,*),W ORK (*)
SUBROUTINE DSPEV_64(JOBZ,UPLO,N,AP,W ,Z,LD Z,W ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
INTEGER*8N,LDZ,INFO
D OUBLE PRECISION AP (*),W (*),Z (LD Z ,*),W ORK (*)

```
F95 INTERFACE
    SU BROUTINE SPEV (JOBZ, UPLO ,N,AP,W,Z, [LD Z], [W ORK ], [ \(\mathbb{N} F O]\) )
    CHARACTER (LEN=1)::JOBZ,UPLO
    \(\mathbb{N} T E G E R:: N, L D Z, \mathbb{N} F O\)
    REAL (8),D IM ENSION (:) ::AP,W,W ORK
    REAL (8),D \(\mathbb{M}\) ENSIO N (:,:) :: Z
    SU BROUTINE SPEV_64 (JOBZ, UPLO,N,AP,W,Z, [LD Z], [W ORK], [NFO])
    CHARACTER (LEN=1): : JOBZ,UPLO
    \(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} \mathrm{Z}, \mathbb{N}\) FO
    REAL (8), D IM ENSION (:) ::AP,W ,W ORK
    REAL (8),D \(\mathbb{M}\) ENSION (: : : : : Z

\section*{C INTERFACE}
\#include <sunperfh>
void dspev (char jobz, charuplo, intn, double *ap, double
\({ }^{*}\) W , double * \(z\), int \(1 d z\), int *info);
void dspev_64 (char j̣bz, charuple, long n, double *ap, double *w , double *z, long ldz, long *info);

\section*{PURPOSE}
dspev com putes all the eigenvalues and, optionally, eigenvectors of a realsym \(m\) etric \(m\) atrix \(A\) in packed storage.

\section*{ARGUMENTS}

JOBZ (input)
\(=\mathrm{N}^{\prime}:\) C om pute eigenvalues only;
\(=\mathrm{V}\) ': C om pute eigenvalues and eigenvectors.

UPLO (input)
\(=\mathrm{U}\) ': U pper triangle ofA is stored;
\(=\mathbb{L}\) ': Low ertriangle of A is stored.

N (input) The order of them atrix A. \(\mathrm{N}>=0\).

A P (input/output)
D ouble precision anray, dim ension \((\mathbb{N} *(\mathbb{N}+1) / 2)\) On entry, the upper or low er triangle of the sym \(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linear array. The jth column of \(A\) is stored in the anray AP as follow \(s\) : if UPLO \(=U ', A P(i+(j\) \(\left.1)^{\star} j 2\right)=A\left(i, \gamma\right.\) for \(1<=i<=j\) ifUPLO \(=L^{\prime}, A P(i\)
\(\left.+(j-1)^{\star}(2 * n-j) / 2\right)=A(i, j)\) for \(j<i<=n\).

On exit, AP is overw ritten by values generated during the reduction to tridiagonal form . IfU PLO \(=U\) ', the diagonal and firstsuperdiagonal of the tridiagonal \(m\) atrix \(T\) overw rite the corresponding elem ents of \(A\), and if UPLO = L', the diagonal and first subdiagonal of \(T\) overw rite the comesponding elem ents ofA.

W (output)
D ouble precision array, dim ension \((\mathbb{N})\) If \(\mathbb{N} F O=0\), the eigenvalues in ascending order.

Z (output)
D ouble precision array, dim ension (LD Z , N ) If JO B Z
\(=\mathrm{V}\) ', then if \(\mathbb{N} F O=0, \mathrm{Z}\) contains the orthonor-
m aleigenvectors of the \(m\) atrix \(A\), \(w\) ith the \(i\)-th
column of Z holding the eigenvector associated
with \(W\) (i). If \(\mathrm{OBZ}=\mathrm{N}\) ', then Z is not referenced.

LD Z (input)
The leading din ension of the amray Z . LD \(\mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{max}(1, N)\).

W ORK (w orkspace)
D ouble precision array, dim ension ( \(3{ }^{\star} \mathrm{N}\) )
\(\mathbb{N} F O\) (output)
= 0: successfulexit.
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvahue.
>0: if \(\mathbb{N} F O=\) i, the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dspevd - com pute all the eigenvalues and, optionally, eigenvectors of a realsym \(m\) etric \(m\) atrix A in packed storage

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE D SPEVD (JOBZ,UPLO,N,AP,W ,Z,LD Z,W ORK,LW ORK,IN ORK,}
LIN ORK,\mathbb{NFO )}
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGER N,LD Z,LW ORK,LIN ORK,INFO}
INTEGER IN ORK (*)
DOUBLE PRECISION AP (*),W (*),Z (LD Z ,*),W ORK (*)
SU BROUTINE D SPEVD_64 (JOBZ,UPLO,N,AP,W ,Z,LD Z,W ORK,LW ORK,
IN ORK,L\mathbb{N ORK,\mathbb{NFO)}}\mathbf{~}=()

```
CHARACTER * 1 JOBZ, UPLO
\(\mathbb{I N}\) TEGER*8N,LDZ,LW ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER*8 \(\mathbb{I N}\) ORK (*)


\section*{F95 INTERFACE}

SU BROUTINE SPEVD (JOBZ, UPLO, \(\mathbb{N}], A P, W, Z,[L D Z],[W O R K],[L W ~ O R K]\), [ \(\mathbb{I N}\) ORK ], [LIN ORK ], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1): : JOBZ, UPLO
\(\mathbb{N} T E G E R:: N, L D Z, L W\) ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL (8), D IM ENSION (:) ::AP,W,W ORK
REAL (8), D \(\mathbb{I}\) ENSION (:,:) :: Z
SU BROUTINE SPEVD _64 (OBZ,UPLO, \(\mathbb{N}], A P, W, Z,[L D ~ Z],[\mathbb{W}\) ORK ], [LW ORK ],
[ \(\mathbb{I N}\) ORK], [LIN ORK ], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1):: JOBZ, UPLO
\(\mathbb{N}\) TEGER (8) :: \(N\), LD Z, LW ORK, LIW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{I W} O R K\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::AP,W ,W ORK
REAL (8), D \(\mathbb{I}\) ENSION (:,:) ::Z

\section*{C INTERFACE}
\#include <sunperfh>
void dspevd (char j̀bz, char uplo, intn, double *ap, double *W , double * \(z\), int ldz, int *info);
void dspevd_64 (char jंbz, charuplo, long n, double *ap, double *w , double *z, long ldz, long *info);

\section*{PURPOSE}
dspevd com putes all the eigenvalues and, optionally, eigenvectors of a realsymm etric \(m\) atrix \(A\) in packed storage. If eigenvectors are desired, ituses a divide and conquer algorithm .

The divide and conquer algorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on m achines w ith a guard digit in add/subtract, or on those binary \(m\) achines \(w\) thout guard digits \(w\) hich subtract like the \(C\) ray
 fail on hexadecim al or decim al machines \(w\) ithout guard digits, butw e know of none.

\section*{ARGUMENTS}

JO B Z (input)
\(=\mathrm{N}\) ': C om pute eigenvalues only;
\(=V^{\prime}:\) C om pute eigenvalues and eigenvectors.

UPLO (input)
\(=\mathrm{U}\) ': U pper triangle ofA is stored;
\(=\mathrm{L}\) ': Low er triangle of A is stored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

AP (input/output)
D ouble precision aray, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2)\) On
entry, the upper or low er triangle of the sym -
\(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linear
array. The \(j\) th column of \(A\) is stored in the array AP as follow \(s\) : if UPLO \(=U ', A P(i+(j\)
 \(+(j-1)^{\star}(2 \star n-j / 2)=A(i, 7)\) for \(j<=i<=n\).

On exit, AP is overw rilten by values generated during the reduction to tridiagonal form. If PLO \(=U\) ', the diagonal and first superdiagonal of the tridiagonal \(m\) atrix \(T\) overw rite the corresponding elem ents ofA, and if UPLO = L', the diagonal and first subdiagonal of \(T\) overw rite the comesponding elem ents ofA.
W (output)
D ouble precision array, dim ension \(\mathbb{N}\) ) If \(\mathbb{N} F O=0\), the eigenvalues in ascending order.
\(Z\) (input) \(D\) ouble precision array, dim ension ( \(\mathrm{LD} \mathrm{Z}, \mathrm{N}\) ) If O B Z \(=\mathrm{V}\) ', then if \(\mathbb{N} F O=0, \mathrm{Z}\) contains the orthonorm aleigenvectors of the \(m\) atrix \(A\), \(w\) ith the \(i\)-th column of Z holding the eigenvector associated w th W (i). If \(\mathrm{OBZ}=\mathrm{N}\) ', then Z is not referenced.

LD \(Z\) (input)
The leading dim ension of the array \(Z\). LD \(Z>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD \(Z>=\mathrm{max}(1, \mathrm{~N})\).

W ORK (w orkspace)
Realarray, dim ension (LW ORK)On exit, if \(\mathbb{N} F O=\) \(0, W\) ORK (1) retums the optim allW ORK .

LW ORK (input)
The dim ension of the array \(\mathrm{W} O R \mathrm{O}\). If \(\mathrm{N}<=1\), LW ORK must be at least1. If \(\mathrm{JOBZ}=\mathrm{N}\) 'and \(\mathrm{N}>\) \(1, L W\) ORK m ustbe at least \(2{ }^{*} \mathrm{~N}\). If \(\mathcal{J O} \mathrm{BZ}=\mathrm{V}^{\prime}\) and \(\mathrm{N}>1\), LW ORK mustbe at least \(1+6 * \mathrm{~N}+\mathrm{N} * * 2\).

IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IV ORK (w orkspace/output)
Integer array, dim ension (L IN ORK) On exit, if \(\mathbb{I N}\) FO \(=0, \mathbb{I V}\) ORK (1) retums the optim alL IV ORK.

LIV ORK (input)
The dim ension of the array \(\mathbb{I N} O R K\). If \(J O B Z=N^{\prime}\) or \(N<=1, L \mathbb{I N} O R K\) mustbe at least1. If JOBZ \(=\)

V'and N > 1,LIN ORK m ustbe at least \(3+5 * N\).

IfLIW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the \(\mathbb{I W}\) ORK array, retums this value as the first entry of the \(\mathbb{I W}\) ORK aray, and no errorm essage related to LIN ORK is issued by XERBLA .
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue.
\(>0:\) if \(\mathbb{N F O}=i\), the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did notconverge to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dspevx - com pute selected eigenvalues and, optionally, eigenvectors of a realsym \(m\) etric \(m\) atrix A in packed storage

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE DSPEVX (JOBZ,RANGE,UPLO,N,AP,VL,VU, IL,\mathbb{U,ABTOL,}}\mathbf{~},\textrm{N},\textrm{N}
NFOUND,W,Z,LD Z,WORK,INORK2,\mathbb{FA}\mathbb{L},\mathbb{N}FO)
CHARACTER * 1 JOBZ,RANGE,UPLO
\mathbb{NTEGERN,}\mathbb{N},\mathbb{U},NFOUND,LDZ,\mathbb{NFO}
\mathbb{NTEGER IN ORK2(*),\mathbb{FA L (*)}}\mathbf{(*)}
DOUBLE PRECISION VL,VU,ABTOL
DOUBLE PRECISION AP (*),W (*),Z (LDZ ,*),W ORK (*)
SU BROUT\mathbb{NE DSPEVX_64(JOBZ,RANGE,UPLO,N,AP,VL,VU,IL,\mathbb{U},ABTOL,}
NFOUND,W,Z,LDZ,WORK,INORK2,\mathbb{FA}\mathbb{L},\mathbb{N}FO)

```
CHARACTER * 1 JOBZ,RANGE,UPLO
\(\mathbb{N} T E G E R * 8 N, \mathbb{I}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{I V}\) ORK 2 (*), \(\mathbb{F} A \mathbb{L}\) (*)
DOUBLE PRECISION VL,VU,ABTOL
D OUBLE PRECISIONAP ( \(\left.{ }^{*}\right), \mathrm{W}\left({ }^{( }\right), \mathrm{Z}(\mathrm{LD} \mathrm{Z}, \star), \mathrm{W} O R K(*)\)

\section*{F95 INTERFACE}

SUBROUTINE SPEVX (JOBZ,RANGE, UPLO,N,AP,VL,VU, IL, IU, ABTOL, NFOUND, W, Z, [LD Z], [W ORK], [IW ORK 2], \(\mathbb{F A} \mathbb{I},[\mathbb{N} F O\) ])

CHARACTER (LEN=1):: JOBZ,RANGE,UPLO
\(\mathbb{N} T E G E R:: N, \mathbb{I}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{I W}\) ORK2, \(\mathbb{F A} \mathbb{I}\)
REAL (8) ::VL,VU, ABTOL
REAL (8), D IM ENSION (:) ::AP,W,W ORK

SU BROUTINE SPEVX_64 (OBZ,RANGE,UPLO,N,AP,VL,VU, \(\mathbb{Z}, \mathbb{Z}, A B T O L\), NFOUND,W,Z,[LD Z], [WORK],[ \(\mathbb{W} O R K 2], \mathbb{F} A \mathbb{I},[\mathbb{N} F O]\) )

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
\(\mathbb{N}\) TEGER (8) :: N , \(\mathbb{I}, \mathbb{I U}, \mathrm{NFOUND}, \mathrm{LD} Z, \mathbb{N}\) FO
\(\mathbb{N}\) TEGER (8), D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{I N}\) ORK2, \(\mathbb{F} A \operatorname{II}\)
REAL (8) ::VL, VU, ABTOL
REAL (8), D \(\mathbb{M}\) ENSION (:) ::AP,W ,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::Z

\section*{C INTERFACE}
\#include <sunperfh>
void dspevx (char jobz, char range, charuplo, intn, double
*ap, double vl, double vu, int il, intiu, double
abtol, int *nfound, double \({ }^{2}\), double \({ }^{2}\), intldz, int *ifail, int*info);
void dspevx_64 (char jobz, char range, char uplo, long n, double *ap, double vl, double vu, long il, long iu, double abtol, long *nfound, double *w, double * z, long ldz, long *ifail, long *info);

\section*{PURPOSE}
dspevx com putes selected eigenvalues and, optionally , eigenvectors of a real symm etric \(m\) atrix A in packed storage. E igenvalues/vectors can be selected by specifying either a range of vahues or range of indices for the desired eigenvalues.

\section*{ARGUMENTS}

JOBZ (input)
\(=\mathrm{N}\) : C om pute eigenvalues only;
\(=\mathrm{V}\) : C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvalues w illbe found;
\(=\mathrm{V}\) ::alleigenvalues in the half-open interval
(VL, VU] w ill be found; = \(\mathrm{I}^{\prime}\) : the \(I \mathrm{I}\)-th through
\(\mathbb{I U}\)-th eigenvalues w ill be found.

UPLO (input)
\(=\mathrm{U}\) ': Upper triangle ofA is stored;
\(=\mathbb{L}\) ': Low er triangle of A is stored.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
AP (input/output)
D ouble precision array, dim ension \(\mathbb{N} * \mathbb{N}+1\) )/2) On entry, the upper or low er triangle of the sym \(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linear array. The \(j\) th colum \(n\) of \(A\) is stored in the aray AP as follow s: if UPLO = U', AP (i + \((j\)
 \(+(j-1)^{\star}(2 * n-j / 2)=A(i, 7)\) for \(j<=i<=n\).

On exit, AP is overw ritten by values generated during the reduction to tridiagonal form. If PLO
= U', the diagonal and first superdiagonal of the tridiagonal \(m\) atrix \(T\) overw rite the corresponding elem ents ofA, and if UPLO = 5 ', the diagonal and first subdiagonal of \(T\) overw rite the comesponding elem ents ofA.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE=A 'or I '.

VU (input)
See the description of V L .

II (input)
If RANGE= 'I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{Z}<=\mathbb{U}<=N\), if \(N>0\); \(\mathbb{H}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N ot referenced ifRANGE= A 'or V'.

IU (input)
See the description of II.

ABTOL (input)
The absolute error tolerance for the eigenvalues. A \(n\) approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b] of w idth less than orequal to

ABTOL + EPS * max ( \(k|\),\(| |) ,\)
\(w\) here EPS is the m achine precision. If ABTOL is less than or equalto zero, then EPS* \(\mid\) | w illube used in its place, w here \(|T|\) is the 1 -norm of the tridiagonalm atrix obtained by reducing AP to tri-
diagonal form .
E igenvahues w illbe com puted m ostaccurately when ABTOL is set to tw ice the underflow threshold \(2 \star\) SLAM CH (S ), notzero. If this routine retums w ith \(\mathbb{N}\) FO \(>0\), indicating that som e eigenvectors did not converge, try setting ABTOL to \(2 *\) SLAM CH (S ).

See "C om puting Sm allSingularV alues ofB idiagonal M atrices w ith G uaranteed H igh Relative A ccuracy," by D em m eland \(K\) ahan, LA PA CK W orking \(N\) ote \#3. NFOUND (output)

The total num ber of eigenvalues found. \(0<=\) NFOUND <= N. IfRANGE = A', NFOUND = N, and if RANGE \(=I^{\prime}\), NFOUND \(=\mathbb{U}-\mathbb{L}+1\).

W (output)
D ouble precision array, dim ension \(\mathbb{N}\) ) If \(\mathbb{N} F O=0\), the selected eigenvalues in ascending order.

Z (output)
D ouble precision array, dim ension (LD Z, max (1, M ))
If \(J O B Z=V\) ', then if \(\mathbb{N} F O=0\), the firstNFOUND colum ns of \(Z\) contain the orthonorm al eigenvectors of the \(m\) atrix A comesponding to the selected eigenvalues, \(w\) ith the i-th colum \(n\) of \(Z\) holding the eigenvector associated w ith \(W\) (i). If an eigenvector fails to converge, then thatcolum \(n\) of \(Z\) contains the latest approxim ation to the eigenvector, and the index of the eigenvector is retumed in \(\mathbb{F A} \mathbb{I}\). If \(\mathrm{JOBZ}=\mathrm{N}\) ', then Z is not referenced. \(N\) ote: the user must ensure that at least m ax (1,NFOUND) colum ns are supplied in the anay \(Z\); ifRANGE = V', the exact value of FOUND is not know \(n\) in advance and an upperbound \(m\) ustbe used.

LD \(Z\) (input)
The leading \(d i m\) ension of the array \(Z\). LD \(Z>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD Z >= \(\mathrm{max}(1, N)\).

W ORK (w orkspace)
D ouble precision array, dim ension ( \(8 * N\) )
IV ORK 2 (w orkspace)
Integer amay, dim ension ( 5 *N )
FA \(\mathbb{I}\) (output)
Integer array, dim ension \((\mathbb{N})\) If JO B Z \(=\mathrm{V}\) ', then if
\(\mathbb{N} F O=0\), the firstNFOUND elem ents of \(\mathbb{F} A \mathbb{I}\) are
zero. If \(\mathbb{N F O}>0\), then \(\mathbb{F A} I I\) contains the
indices of the eigenvectors that failed to converge. If \(J 0 B Z=N\) ', then \(\mathbb{F A} \mathbb{I}\) is not referenced.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N F O}=-i\), the \(i\)-th argum ent had an illegal value
\(>0\) : if \(\mathbb{N F O}=\mathrm{i}\), then ieigenvectors failed to converge. Their indioes are stored in array FAI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dspgst-reduce a realsym \(m\) etric-definite generalized eigenproblem to standard form, using packed storage

\section*{SYNOPSIS}

SUBROUTINEDSPGST (TTYPE, UPLO,N,AP,BP, \(\mathbb{N} F O\) )
CHARACTER * 1 UPLO
\(\mathbb{N} T E G E R \operatorname{ITYPE}, N, \mathbb{N} F O\)
D OUBLE PRECISION AP (*), BP (*)
SU BROUTINE DSPGST_64 (TTYPE,UPLO,N,AP,BP, \(\mathbb{N} F O\) )

CHARACTER * 1 UPLO
\(\mathbb{N} T E G E R * 8 \mathbb{I T Y P E}, N, \mathbb{N} F O\)
DOUBLE PRECISION AP (*), BP (*)

\section*{F95 INTERFACE}

SUBROUTINE SPGST (TTYPE, UPLO, N,AP, BP, [ \(\mathbb{N} F O]\) )
CHARACTER (LEN=1) ::UPLO
\(\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, \mathbb{N} F O\)
REAL (8),D \(\mathbb{I}\) ENSION (:) ::AP,BP
SU BROUTINE SPGST_64 (TTYPE, UPLO ,N , AP, BP, [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER (8) :: \(\mathbb{T} Y \mathrm{PE}, \mathrm{N}, \mathbb{I N F O}\)
REAL (8), D \(\mathbb{I M}\) ENSION (:) ::AP,BP

\section*{C INTERFACE}
\#include <sunperfh>
void dspgst(int itype, charuplo, intn, double *ap, double *bp, int *info);
void dspgst_64 (long itype, charuplo, long n, double *ap, double *bp, long *info);

\section*{PURPOSE}
dspgst reduces a real sym \(m\) etric-definite generalized eigenproblem to standard form, using packed storage.

If ITY PE \(=1\), the problem is \(A * x=\) lam bda*B \({ }^{*} \mathrm{X}\), and \(A\) is overw ritten by inv \((U * * T) * A * \operatorname{inv}(U)\) or inv ( \((\mathrm{A})\) *A *inv ( \((\mathrm{L} * * \mathrm{~T})\)
If ITYPE \(=2\) or 3 , the problem is \(A * B *_{x}=\) lam bda* x or \(B *_{A} *_{X}=\operatorname{lam}\) bda* \(x\), and \(A\) is overw ritten by \(U * A * U * * T\) or L**T*A*L。

B m usthave been previously factorized as U ** T * U or \(\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\) by SPPTRF.

\section*{ARGUMENTS}

ITYPE (input)
\(=1:\) compute \(\quad \operatorname{inv}(\mathrm{U} * * T) \star A * \operatorname{inv}(\mathrm{U})\) or
\(\operatorname{inv}(\amalg) \star A\) *inv \((\amalg * * T)\);
\(=2\) or 3 : com pute \(\mathrm{U} * \mathrm{~A} * \mathrm{U} * * \mathrm{~T}\) orL \({ }^{* *} \mathrm{~T} * \mathrm{~A} * \mathrm{~L}\).

UPLO (input)
\(=U^{\prime}\) : Uppertriangle of \(A\) is stored and \(B\) is factored as \(\mathrm{U} * * \mathrm{~T} * \mathrm{U} ;=\mathrm{L}\) : L Low er triangle ofA is stored and B is factored as \(\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\).

N (input) The order of the \(m\) atriges \(A\) and \(B . N>=0\).

A P (input/output)
D ouble precision array, dim ension \((\mathbb{N} *(\mathbb{N}+1) / 2)\) On entry, the upper or low er triangle of the sym \(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linear amay. The \(j\) th column of \(A\) is stored in the array AP as follow s: if UPLO \(=\mathrm{U}\) ', AP (i \(+(j\) 1) \(\star j 2\) ) \(=A(i, 7)\) for \(1<=i<=j\) ifUPLO \(=L^{\prime}, A P(i\) \(+(j-1)^{\star}(2 n-j / 2)=A(i, j)\) for \(j=i<=n\).

On exit, if \(\mathbb{N F O}=0\), the transform ed \(m\) atrix, stored in the sam e form at as A.

BP (input)
D ouble precision array, dim ension \((\mathbb{N} *(N+1) / 2)\) The triangular factor from the Cholesky factorization ofB, stored in the sam e form atasA, as retumed by SPPTRF.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dspgv - com pute all the eigenvalues and, optionally, the
eigenvectors of a real generalized sym $m$ etric-definite eigen-
problem, of the form $A * x=(\operatorname{lam} . b d a) * B * x, A * B x=(\operatorname{lam} . b d a) * x$, or
B *A *x= (lam bda) *x

```

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE D SPGV (TTYPE,JOBZ,UPLO,N,AP,BP,W ,Z,LD Z,W ORK,INFO )}
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGER ITYPE,N,LD Z,INFO}
DOUBLE PRECISION AP (*),BP (*),W (*),Z (LD Z,*),W ORK (*)
SU BROUTINE D SPGV_64 (ITYPE,JOBZ,UPLO,N,AP,BP,W ,Z,LD Z,W ORK,
INFO)
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGER*8 ITYPE,N,LDZ,INFO}
DOUBLE PRECISION AP (*),BP (*),W (*),Z (LD Z,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE SPGV (TTYPE, JOBZ, UPLO,N,AP,BP,W,Z,[LD Z], [W ORK ], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1): : JOBZ,UPLO
\(\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D Z, \mathbb{N F O}\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::AP,BP,W ,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) :: Z

SU BROUTINE SPGV_64 (TTYPE, JOBZ, UPLO ,N,AP,BP,W,Z,[LDZ], [W ORK], [ \(\mathbb{N}\) FO ])

CHARACTER ( \(\llcorner E N=1\) ) : : OBZ,\(~ U P L O\)
\(\mathbb{N}\) TEGER (8) :: \(\mathbb{I T Y} \mathrm{PE}, \mathrm{N}, \mathrm{LD} \mathrm{Z}, \mathbb{N} F O\)
REAL (8), D \(\mathbb{M} E N S I O N(:):: A P, B P, W, W O R K\)
REAL (8), D \(\mathbb{M}\) ENSION (:,:) :: Z

\section*{C INTERFACE}
\#include <sunperfh>
void dspgv (int itype, char jobz, charuple, int n, double
*ap, double *bp, double *w , double *z, intldz, int*info);
void dspgv_64 (long itype, char j̀bz, charuplo, long n, dou-
ble *ap, double *bp, double *w , double *z, long ldz, long *info);

\section*{PURPOSE}
dspgv com putes all the eigenvalues and, optionally, the eigenvectors of a realgeneralized sym m etric-definite eigenproblem, of the form \(A * x=(\operatorname{lam} . b d a) * B * x, A * B x=\left(\operatorname{lam}\right.\) bda) \({ }^{*} x\), or \(B *_{A} *_{x}=\left(l a m\right.\) bda) \({ }^{*} x\). H ere \(A\) and \(B\) are assum ed to be sym \(m\) etric, stored in packed form at, and B is also positive definite.

\section*{ARGUMENTS}

ITYPE (input)
Specifies the problem type to be solved:
\(=1: A{ }^{*} \mathrm{X}=\left(\operatorname{lam}\right.\) bda)\({ }^{*} \mathrm{~B}{ }^{*} \mathrm{X}\)
\(=2: A * B *_{X}=\left(\operatorname{lam}\right.\) bda) \({ }^{*} \mathrm{X}\)
\(=3: B{ }^{*} A{ }^{*} \mathrm{x}=\left(\operatorname{lam}\right.\) bda)\({ }^{\star} \mathrm{x}\)

JOBZ (input)
\(=\mathrm{N}^{\prime}:\) C om pute eigenvalues only;
\(=\mathrm{V}\) : C om pute eigenvalues and eigenvectors.

UPLO (input)
\(=\mathrm{U}\) : U pper triangles of \(A\) and \(B\) are stored;
\(=\mathbb{L}\) ': Low ertriangles of \(A\) and \(B\) are stored.

N (input) The order of them atriges A and \(\mathrm{B} . \mathrm{N}>=0\).

A P (input/output)
D ouble precision anray, dim ension \((\mathbb{N} *(\mathbb{N}+1) / 2)\) On
entry, the upper or low er triangle of the sym -
\(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linear
array. The jth column of \(A\) is stored in the
array AP as follow s: if UPLO \(=U\) ', AP \((i+(j\)
 \(\left.+(j-1)^{\star}(2 * n-j) / 2\right)=A(i, 7)\) for \(j=i<=n\).

On exit, the contents ofA are destroyed.
BP (input/output)
D ouble precision anray, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2\) ) On entry, the upper or low er triangle of the sym \(m\) etric \(m\) atrix \(B\), packed colum nw ise in a linear array. The \(j\) th column of \(B\) is stored in the anay BP as follow s: if UPLO \(=\mathrm{U}\) ', BP \((i+(j\) 1) \(* \mathfrak{j} 2\) ) \(=\mathrm{B}(i, j)\) for \(1<=\mathrm{i}<=\dot{j}\) if \(\mathrm{UPLO}=\mathrm{L}, \mathrm{BP}(i\) \(+(j-1)^{\star}(2 * n-j / 2)=B(i, j)\) for \(j=i<=n\).

On exit, the triangular factor \(U\) or \(L\) from the Cholesky factorization \(\mathrm{B}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(\mathrm{B}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\), in the sam e storage form at as \(B\).

W (output)
D ouble precision array, dim ension \(\mathbb{N}\) ) If \(\mathbb{N} F O=0\), the eigenvalues in ascending order.

Z (output)
D ouble precision array, dim ension (LD Z , N ) If JO B Z
\(=\mathrm{V}^{\prime}\), then if \(\mathbb{N} \mathrm{FO}=0, \mathrm{Z}\) contains the m atrix Z
of eigenvectors. The eigenvectors are norm alized
as follows: if ITYPE \(=1\) or \(2, Z * * T * B * Z=I\); if \(\operatorname{ITYPE}=3, Z * * T * \operatorname{inv}(B) * Z=I\). If \(J O B Z=N\) ', then Z is not referenced.

LD Z (input)
The leading dim ension of the array \(\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{max}(1, N)\).

W ORK (w orkspace)
D ouble precision array, dim ension ( \(3 * N\) )
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
> 0: SPPTRF orSSPEV retumed an errorcode:
\(<=N:\) if \(\mathbb{N} F O=i, S S P E V\) failed to converge; i off-diagonalelem ents of an interm ediate tridiagonal form did notconverge to zero. \(>\mathrm{N}\) : if \(\mathbb{N} F O\) \(=n+i\), for \(1<=i<=n\), then the leading \(m\) inor of orderiofB is not positive definite. The factorization of B could notbe com pleted and no eigenvalues or eigenvectors w ere com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
```

dspgvd -com pute all the eigenvalues, and optionally, the
eigenvectors of a realgeneralized symm etric-definite eigen-
problem, of the form A *x= (lam bda)*B *x, A *B x= (lam bda)*x, or
B *A *X= (lam boda) *}\textrm{X

```

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE DSPGVD(TTYPE,NOBZ,UPLO,N,AP,BP,W,Z,LD Z,W ORK,}
LW ORK,IN ORK,L\mathbb{IN ORK,INFO)}
CHARACTER * 1 JOBZ,UPLO

```

```

INTEGER IN ORK (*)
DOUBLE PRECISION AP (*),BP (*),W (*),Z (LD Z ,*),W ORK (*)
SU BROUT\mathbb{NEDSPGVD_64(TTYPE,NOBZ,UPLO,N,AP,BP,W ,Z,LD Z,W ORK,}

```

```

CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGER*8 ITYPE,N,LDZ,LW ORK,LIN ORK,INFO}
INTEGER*8 IN ORK (*)
DOUBLE PRECISION AP (*),BP (*),W (*),Z (LD Z,*),W ORK (*)

```

\section*{F95 INTERFACE}

SUBROUTINE SPGVD (TTYPE, JOBZ, UPLO,N,AP,BP,W,Z,[LDZ], [W ORK], [LW ORK], [ \(\mathbb{W}\) ORK], [LIN ORK ], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1): : JOBZ, UPLO
\(\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D Z, L W\) ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL (8),D \(\mathbb{M}\) ENSION (:) ::AP,BP,W,WORK

SU BROUTINE SPGVD_64 (TTYPE, JOBZ,UPLO,N,AP,BP,W,Z, [LD Z], [W ORK ], [LW ORK], [IN ORK], [LINORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1)::JOBZ,UPLO
\(\mathbb{N} T E G E R(8)::\) ITY PE,N,LDZ,LW ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
REAL (8),D \(\mathbb{M}\) ENSION (:) ::AP,BP,W ,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (: : : : : Z

\section*{C INTERFACE}
\#include <sunperfh>
void dspgvd (int itype, char jobz, charuplo, int n, double
*ap, double *bp, double *w, double *z, int ldz, int*info);
void dspgvd_64 (long itype, char jobz, char uplo, long n, double *ap, double *bp, double *w , double *z, long ldz, long *info);

\section*{PURPOSE}
dspgvd com putes all the eigenvalues, and optionally, the eigenvectors of a realgeneralized sym \(m\) etric-definite eigenproblem, of the form \(A * x=\left(\operatorname{lam}\right.\) bda) \({ }^{*} \mathrm{~B} * \mathrm{x}, \mathrm{A} * \mathrm{~B}\) x \(=\left(\operatorname{lam}\right.\) bda) \({ }^{*} \mathrm{x}\), or \(B * A * x=\left(\operatorname{lam}\right.\) bda) \({ }^{*} x\). Here \(A\) and \(B\) are assum ed to be sym \(m\) etric, stored in packed form at, and \(B\) is also positive definite. Ifeigenvectors are desired, ituses a divide and conquer algorithm .

The divide and conqueralgorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on \(m\) achines w ith a guard digit in add/subtract, or on those binary \(m\) achines \(w\) ithout guard digits w hich subtract like the \(C\) ray X M P , C ray Y M P , C ray C-90, or C ray-2. It could conceivably fail on hexadecim al or decim al \(m\) achines \(w\) ithout guard digits, butw e know of none.

\section*{ARGUMENTS}

ITYPE (input)
Specifies the problem type to be solved:
\(=1: A{ }^{*} \mathrm{X}=\left(\operatorname{lam}\right.\) bda)\({ }^{*} \mathrm{~B}^{*} \mathrm{X}\)
\(=2: \mathrm{A} * \mathrm{~B} * \mathrm{x}=\left(\operatorname{lam}\right.\) bda) \({ }^{*} \mathrm{x}\)
\(=3: B * A * x=(\operatorname{lam} . b d a){ }^{*} \mathrm{x}\)

JOBZ (input)
= N ': C om pute eigenvahues only;
\(=\mathrm{V}:\) : C om pute eigenvahues and eigenvectors.

UPLO (input)
\(=\mathrm{U}\) ': U pper triangles of \(A\) and \(B\) are stored;
\(=\mathrm{L}\) ': Low ertriangles of \(A\) and \(B\) are stored.
N (input) The order of the m atrices A and \(\mathrm{B} . \mathrm{N}>=0\).
AP (input/output)
D ouble precision array, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2\) ) On entry, the upper or low er triangle of the sym \(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linear array. The \(j\) th colum \(n\) of \(A\) is stored in the array AP as follow s: if UPLO = U', AP (i \(+(j\) \(1) \star \dot{j} 2)=A(i, 7)\) for \(1<=i<=j\) ifUPLO \(=L^{\prime}, A P(i\) \(\left.+(j-1)^{\star}\left(2{ }^{*} n-j\right) / 2\right)=A(i, 7)\) for \(j=i<=n\).

On exit, the contents ofA \(P\) are destroyed.
BP (input/output)
D ouble precision anay, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2\) ) On entry, the upper or low er triangle of the sym \(m\) etric \(m\) atrix \(B\), packed colum nw ise in a linear anray. The jth column of \(B\) is stored in the array BP as follow s: if UPLO \(=\mathrm{U}\) ', \(\mathrm{BP}(i+(j\) \(\left.1)^{\star} \mathfrak{j} 2\right)=\mathrm{B}(i, 7)\) for \(1<=\mathrm{i}<=\dot{j}\) if \(\mathrm{UPLO}=\mathrm{L}, \mathrm{BP}(i\) \(\left.+(j-1)^{*}(2 \star n-j) / 2\right)=B(i, 7)\) for \(j=i<=n\).

On exit, the triangular factor \(U\) or \(L\) from the Cholesky factorization \(\mathrm{B}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(\mathrm{B}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\), in the sam e storage form at as \(B\).

W (output)
D ouble precision array, dim ension \(\mathbb{N}\) ) If \(\mathbb{N} F O=0\), the eigenvalues in ascending order.

Z (output)
D ouble precision array, dim ension (LD Z,N) If JO B Z
\(=\mathrm{V}^{\prime}\), then if \(\mathbb{N} \mathrm{FO}=0, \mathrm{Z}\) contains the m atrix Z
of eigenvectors. The eigenvectors are norm alized
as follow s : if ITY \(\mathrm{PE}=1\) or \(2, \mathrm{Z} * * \mathrm{~T} * \mathrm{~B} * \mathrm{Z}=\mathrm{I}\); if \(\operatorname{ITYPE}=3, Z * * T * \operatorname{inv}(B) * Z=I\). If \(J O B Z=N\) ', then \(Z\) is not referenced.

LD Z (input)
The leading dim ension of the array \(Z . L D Z>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{max}(1, N)\).

W ORK (w orkspace/output)
D ouble precision array, dim ension (LW ORK) O n exit, if \(\mathbb{N F O}=0, \mathrm{~W} O \mathrm{RK}(1)\) retums the optim alLW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. If \(\mathrm{N}<=1\), LW ORK \(>=1\). If \(\mathrm{OBZ}=\mathrm{N}\) 'and \(\mathrm{N}>1\), LW ORK \(>=\) \(2 * N\). If \(\mathrm{OBZ}=\mathrm{V}\) 'and \(\mathrm{N}>1\),LW ORK \(>=1+6 * N+\) \(2 * N * * 2\) 。

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK aray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IN ORK (w orkspace/output)
Integer array, dim ension ( \((\mathbb{I}\) W ORK) On exit, if \(\mathbb{N}\) FO \(=0, \mathbb{I W}\) ORK (1) retums the optim alL \(\mathbb{I N}\) ORK.

LIV ORK (input)
The dim ension of the anay \(\mathbb{I W}\) ORK. If \(O B Z=N^{\prime}\) or \(\mathrm{N}<=1, \mathrm{~L} \mathbb{W} O R K>=1\). If \(\mathrm{OBZ}=\mathrm{V}\) 'and \(N>1\), LIW ORK \(>=3+5 \star \mathrm{~N}\) 。

IfLIW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the \(\mathbb{I W}\) ORK array, retums this value as the first entry of the \(\mathbb{I W}\) ORK array, and no errorm essage related to \(L \mathbb{I W}\) ORK is issued by XERBLA.

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue
>0: SPPTRF orSSPEVD retumed an emorcode: \(<=\mathrm{N}:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{SSPEVD}\) failed to converge; i offf-diagonal elem ents of an interm ediate tridiagonal form did notconverge to zero; \(>\mathrm{N}\) : if \(\mathbb{N F O}\) \(=N+i\), for \(1<=i<=N\), then the leading \(m\) inor oforderiofB is not positive definite. The factorization of \(B\) could notbe com pleted and no eigenvalues or eigenvectors w ere com puted.

\section*{FURTHER DETAILS}

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv. of K entucky, USA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
```

dspgvx - com pute selected eigenvahues, and optionally,
eigenvectors of a realgeneralized symm etric-definite eigen-
problem, of the form A *x= (lam bda)*B *x, A *B x= (lam bda)*x, or
B *A *X= (lam boda) *}\textrm{X

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE D SPGVX (TTYPE,NOBZ,RANGE,UPLO,N,AP,BP,VL,VU,IL,}
\mathbb{U},ABSTOL,M,W ,Z,LDZ,W ORK,IN ORK,\mathbb{FA}\mathbb{I},\mathbb{NFO)}
CHARACTER * 1 JobZ,RANGE,UPLO
\mathbb{NTEGER ITYPE,N,\mathbb{N},\mathbb{U},M,LDZ,INFO}
\mathbb{NTEGER IN ORK (*), \mathbb{FAIL (*)}}\mathbf{(})
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION AP (*),BP (*),W (*),Z (LD Z,*),W ORK (*)
SU BROUTINE D SPGVX_64 (ITY PE,NOBZ,RANGE,UPLO,N,AP,BP,VL,VU,IL,
\mathbb{U},ABSTOL,M,W,Z,LD Z,W ORK,INORK,\mathbb{FA}\mathbb{L},\mathbb{N}FO)
CHARACTER * 1 JOBZ,RANGE,UPLO

```

```

INTEGER*8 IN ORK (*), \mathbb{FA IU (*)}
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION AP (*),BP (*),W (*),Z (LD Z,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E \operatorname{SPGVX(TTYPE,~JOBZ,RANGE,UPLO,N,AP,BP,VL,VU,~\amalg ,~}\) \(\mathbb{U}, A B S T O L, M, W, Z,[L D Z],[W O R K],[\mathbb{N} O R K], \mathbb{F} A \mathbb{I},[\mathbb{N} F O])\)

CHARACTER (LEN=1):: DBZ,RANGE,UPLO
\(\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, \mathbb{L}, \mathbb{U}, M, L D Z, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{I N} O R K, \mathbb{F} A \mathbb{L}\)
REAL (8) :: VL, VU, ABSTOL
REAL (8), D \(\mathbb{M}\) ENSION (:) ::AP, BP ,W ,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::Z

SU BROUTINE SPGVX_64 (TTYPE, JOBZ,RANGE, UPLO, N, AP, BP, VL, VU, \(\mathbb{I}, \mathbb{U}, A B S T O L, M, W, Z,[L D Z],[W O R K],[\mathbb{W} O R K], \mathbb{F} A \mathbb{I},[\mathbb{N} F O])\)

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
\(\mathbb{N}\) TEGER (8) :: ITYPE, \(N, \mathbb{I}, \mathbb{U}, \mathrm{M}, \mathrm{LD} \mathrm{Z}, \mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{I W} O R K, \mathbb{F A} \mathbb{I}\)
REAL (8) :: VL, VU, ABSTOL
REAL (8), D \(\mathbb{M} E N S I O N(:):: A P, B P, W, W O R K\)
REAL (8), D IM ENSION (:,:) :: Z

\section*{C INTERFACE}
\#include <sunperfh>
void dspgvx (int itype, char jंbz, char range, charuplo, int n , double *ap, double *bp, double vl, double vu, intil, intin, double abstol, int \({ }^{m}\), double \({ }^{*}\), double *z, int ldz, int *ifail, int *info);
void dspgvx_64 (long itype, char jobz, char range, charuplo, long \(n\), double *ap, double *bp, double vl, double vu, long il, long iu, double abstol, long *m, double *w, double *z, long ldz, long *ifail, long *info);

\section*{PURPOSE}
dspgvx com putes selected eigenvalues, and optionally , eigenvectors of a realgeneralized sym \(m\) etric-definite eigenproblem, of the form \(A * x=(l a m . b d a) * B * x, A * B x=(l a m . b d a) * x\), or \(B{ }^{\star} A{ }^{*} x=(\operatorname{lam} . b d a){ }^{\star} x\). H ere \(A\) and \(B\) are assum ed to be sym \(m\) etric, stored in packed storage, and \(B\) is also positive definite.
E igenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvahues.

\section*{ARGUMENTS}

ITYPE (input)
Specifies the problem type to be solved:
\(=1: A *_{\mathrm{x}}=(\operatorname{lam} \mathrm{bda}){ }^{*} \mathrm{~B}{ }^{*} \mathrm{x}\)
\(=2: A * B{ }^{*} \mathrm{x}=(\operatorname{lam} \mathrm{bda}){ }^{\star} \mathrm{x}\)
\(=3: \mathrm{B}^{\star} \mathrm{A}{ }^{*} \mathrm{X}=(\operatorname{lam} \mathrm{bda})^{\star} \mathrm{X}\)

JOBZ (input)
\(=N^{\prime}:\) C om pute eigenvalues only;
= V ': C om pute eigenvalues and eigenvectors.
RANGE (input)
= A ': alleigenvalues w illbe found.
\(=\mathrm{V}\) ::alleigenvalues in the half-open interval ( \(\mathrm{L}, \mathrm{VU}] \mathrm{w}\) ill be found. = 'I': the \(\mathbb{I}\)-th through \(\mathbb{I U}\)-th eigenvaluesw illbe found.

UPLO (input)
= U ': U ppertriangle of A and B are stored;
= L ': Lowertriangle of A and B are stored.
\(N\) (input) The order of the \(m\) atrix pencil \((A, B) . N>=0\).
AP (input/output)
D ouble precision array, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2\) ) On entry, the upper or low er triangle of the sym \(m\) etric \(m\) atrix A, packed colum nw ise in a linear array. The \(j\) th colum \(n\) of \(A\) is stored in the array AP as follow s: if UPLO = U', AP (i + (j \(\left.1)^{\star} \mathfrak{j} 2\right)=A(i, j)\) for \(1<=i<=j\) ifUPLO \(=L^{\prime}, A P(i\) \(\left.+(j-1)^{\star}(2 \star n-j) / 2\right)=A(i, 7)\) for \(j<=i<=n\).

On exit, the contents ofAP are destroyed.
BP (input/output)
D ouble precision array, dim ension \(\mathbb{N}^{*}(\mathbb{N}+1) / 2\) ) On entry, the upper or low er triangle of the sym \(m\) etric \(m\) atrix \(B\), packed colum nw ise in a linear array. The \(j\) th colum \(n\) of \(B\) is stored in the array BP as follow s: if \(\mathrm{UPLO}=\mathrm{U}^{\prime}, \mathrm{BP}(i+(j\)
 \(+(j-1)^{\star}\left(2{ }^{\star} n-j / 2\right)=B(i, j)\) for \(j=i<=n\).

On exit, the triangular factor \(U\) or \(L\) from the Cholesky factorization \(\mathrm{B}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) orB \(=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\), in the sam e storage form atas \(B\).

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE=A'or I'.

VU (input)
See the description of V L .

II (input)
IfRA N G E= I', the indices (in ascending order) of the smallest and largest eigenvalues to be
retumed. \(1<=\mathbb{I}<=\mathbb{U}<=\mathrm{N}\), if \(\mathrm{N}>0\); \(\mathbb{I}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N ot referenced ifRANGE \(=\) A 'or V'.

IU (input)
See the description of II.

ABSTOL (input)
The absolute errortolerance for the eigenvalues.
A \(n\) approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b] of w idth less than orequal to

ABSTOL + EPS * max ( \(\mid\) |, \(\mid\) ) ,
where EPS is them achine precision. IfA BSTOL is less than or equal to zero, then EPS* \(\mid\) | w illbe used in its place, where \(T\) | is the 1 -nom of the tridiagonal \(m\) atrix obtained by reducing \(A\) to tridiagonalform.

E igenvalues w illbe com puted m ost accurately when ABSTOL is set to tw ice the underflow threshold \(2 *\) SLAM CH ( S ), not zero. If this routine retums w ith \(\mathbb{N}\) FO \(>0\), indicating that som e eigenvectors did not converge, try setting ABSTO L to \(2 *\) SLAM CH (S ).

M (output)
The total num ber of eigenvalues found. \(0<=\mathrm{M}\) <= N . IfRANGE \(=\mathrm{A}^{\prime}, \mathrm{M}=\mathrm{N}\), and ifRANGE \(=\mathrm{I}^{\prime}, \mathrm{M}=\) \(\mathbb{U}-\mathbb{L}+1\).

W (output)
D ouble precision array, dim ension \(\mathbb{N}\) ) On norm al exit, the first \(M\) elem ents contain the selected eigenvalues in ascending order.

Z (output)
D ouble precision array, dim ension (LD Z, max (1, M )) If \(\mathrm{JOBZ}=\mathrm{N}\) ', then Z is not referenced. If JOBZ \(=\mathrm{V}\) ', then if \(\mathbb{N} F O=0\), the first \(M\) colum ns of \(Z\) contain the orthonom aleigenvectors of the \(m\) atrix A comesponding to the selected eigenvalues, w ith the i-th colum \(n\) of \(Z\) holding the eigenvectorassociated w ith W (i). The eigenvectors are norm alized as follows: if ITYPE \(=1\) or \(2, Z * * T * B * Z=I\); if TTYPE \(=3, Z * * T * \operatorname{inv}(B) * Z=I\).

If an eigenvector fails to converge, then that colum \(n\) of \(Z\) contains the latestapproxim ation to
the eigenvector, and the index of the eigenvector is retumed in \(\mathbb{F A} \mathbb{I}\). N ote: the userm ustensure that at leastm ax \((1, M)\) colum ns are supplied in the aray \(Z\); ifRANGE = V', the exactvalue of \(M\) is notknow \(n\) in advance and an upper bound \(m\) ust be used.
LD \(Z\) (input)
The leading dim ension of the array \(\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD Z >= \(\mathrm{max}(1, \mathrm{~N})\).

W ORK (w orkspace)
D ouble precision array, dim ension ( \(8{ }^{\star} \mathrm{N}\) )
IV ORK (w orkspace)
Integer anay, dim ension ( 5 *N )
FAII (output)
Integer array, dim ension \(\mathbb{N}\) ) If \(\mathrm{JOBZ}=\mathrm{V}^{\prime}\), then if \(\mathbb{N} F O=0\), the firstM elem ents of \(\mathbb{F} A \mathbb{I}\) are zero. If \(\mathbb{N F O}>0\), then \(\mathbb{F A} \mathbb{I}\) contains the indices of the eigenvectors that failed to converge. If \(\mathrm{JOBZ}=\mathrm{N}\) ', then \(\mathbb{F} A \mathbb{I}\) is not referenced.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
> 0: SPPTRF orSSPEVX retumed an errorcode: \(<=N\) : if \(\mathbb{N} F O=i, S S P E V X\) failed to converge; i eigenvectors failed to converge. Their indices are stored in array \(\mathbb{F} A \mathbb{I} .>N\) : if \(\mathbb{N} F O=N+\) \(i\), for \(1<=i<=N\), then the leading \(m\) inorof orderiofB is notpositive definite. The factorization of B could not be com pleted and no eigenvalues or eigenvectors w ere com puted.

\section*{FURTHER DETAILS}

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv . of K entucky, USA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dspm v-perform them atrix-vectoroperation y := alpha*A *x

```
+ beta*y

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSPMV (UPLO,N,ALPHA,A,X, INCX,BETA,Y,INCY)}
CHARACTER * 1 UPLO
\mathbb{NTEGERN,\mathbb{NCX,INCY}}\mathbf{N}=1
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION A (*),X (*),Y (*)
SU BROUT\mathbb{NEDSPM V_64 (UPLO,N,ALPHA,A,X,INCX,BETA,Y, INCY)}
CHARACTER * 1 UPLO
INTEGER*8N,INCX,\mathbb{NCY}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION A (*),X (*),Y (*)

```

\section*{F95 INTERFACE}

SU BROUTINE SPMV (UPLO,N,ALPHA,A,X, [NCX],BETA,Y,[INCY])

CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y\)
REAL (8) ::ALPHA,BETA
REAL (8), D IM ENSION (:) ::A, X,Y
SU BROUTINE SPM V_64 (UPLO, N,ALPHA, A, X, [ \(\mathbb{N C X}], B E T A, Y,[\mathbb{N C Y}])\)
CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R(8):: N, \mathbb{N} C X, \mathbb{N} C Y\)
REAL (8) :: A LPHA,BETA

REAL (8), D IM ENSION (:) ::A, X,Y

\section*{C INTERFACE}
\#include <sunperfh>
void dspm v (charuplo, intn, double alpha, double *a, double \({ }^{*} x\), int incx, double beta, double \({ }^{*} y\), int incy);
void dspm v_64 (charuplo, long n, double alpha, double *a, double *x, long incx, double beta, double *y, long incy);

\section*{PURPOSE}
dspm v perform s the m atrix-vectoroperation \(y:=a l p h a * A * x+\) beta* \(y\), w here alpha and beta are scalars, \(x\) and \(y\) are \(n\) ele\(m\) entvectors and \(A\) is an \(n\) by \(n\) sym \(m\) etric \(m\) atrix, supplied in packed form .

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the \(m\) atrix A is supplied in the packed array A as follow s:

UPLO = U'or U ' The uppertriangularpartofA is supplied in A.

UPLO = L'or I' The low ertriangularpart of A is supplied in A.

U nchanged on exit.

N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix \(A\). \(\mathrm{N}>=0\). U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
\(\left(\left(n^{*}(n+1)\right) / 2\right)\). Before entry \(w\) ith UPLO \(=\) U' or L ', the array A mustcontain the upper triangularpartof the sym m etric \(m\) atrix packed sequentially, column by column, so thatA (1)
containsa(1,1), A (2) and A (3) contain a( 1,2 ) and a( 2,2 ) respectively, and so on. Before entry w ith UPLO = L'or 1', the array A \(m\) ustcontain the low er triangularpartof the sym \(m\) etric \(m\) atrix packed sequentially, column by colum \(n\), so thatA (1) contains a ( 1,1 ), A (2) and A (3) contain a \((2,1)\) and \(a(3,1)\) respectively, and so on. U nchanged on exit.

X (input)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C X))\). Before entry, the increm ented array \(X\) must contain the \(n\) elem ent vectorx. U nchanged on exit.
\(\mathbb{N} C X\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X<>0\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then \(Y\) need notbe seton input. U nchanged on exit.

Y (input/output)
\((1+(n-1) \star a b s(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) m ust contain the \(n\) elem ent vectory. On exit, \(Y\) is overw rilten by the updated vectory.
\(\mathbb{N} C Y\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y . \mathbb{N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dspr- perform the symmetric rank 1 operation A := alpha*x*x'+A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSPR (UPLO,N,ALPHA,X, NNCX,A)}
CHARACTER * 1 UPLO
\mathbb{NTEGER N, INCX}
DOUBLE PRECISION ALPHA
DOUBLE PRECISION X (*),A (*)
SUBROUT\mathbb{NEDSPR_64(UPLO,N,ALPHA,X,INCX,A)}
CHARACTER * 1 UPLO
INTEGER*8N,\mathbb{NCX}
DOUBLE PRECISION ALPHA
DOUBLE PRECISION X (*),A (*)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE SPR (UPLO,N,ALPHA,X,[NCX ],A)}

```
CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R:: N, \mathbb{N C X}\)
REAL (8) ::A LPHA
REAL (8), D IM ENSION (:) :: X,A
SU BROUTINE SPR_64 (UPLO,N,ALPHA, X, [ \(\mathbb{N} C X], A)\)
CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R(8):: N, \mathbb{N} C X\)
REAL (8) ::ALPHA

REAL (8),D IM ENSION (:) :: X,A

\section*{C INTERFACE}
\#include < sunperfh>
void dspr(charuplo, intn, double alpha, double *x, int incx, double *a);
void dspr_64 (charuplo, long n, double alpha, double *x, long incx, double *a);

\section*{PURPOSE}
dsprperform sthe sym m etric rank 1 operation \(A:=a l p h a * x^{*} x^{\prime}\)
\(+A\), where alpha is a realscalar, x is an n elem ent vector and \(A\) is an \(n\) by \(n\) sym \(m\) etric \(m\) atrix, supplied in packed form.

\section*{ARGUMENTS}

UPLO (input)
Onentry,UPLO specifies whether the upper or low er triangular part of the \(m\) atrix \(A\) is supplied in the packed array A as follow s:

UPLO = U 'or L ' The uppertriangularpartof \(A\) is supplied in A.

UPLO = L'or I' The low ertriangularpartof A is supplied in A.

U nchanged on exit.

N (input)
O n entry, N specifies the order of the m atrix A . \(\mathrm{N}>=0\). U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
\((1+(n-1) \star a b s(\mathbb{N} C X))\). Before entry, the increm ented array X must contain the n elem ent vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)

On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X<>0\). U nchanged on exit.

A (input/output)
\(\left(\left(n^{*}(n+1)\right) / 2\right)\). Before entry \(w\) ith UPLO \(=\) U ' or G ', the anay A m ustcontain the upper triangularpartof the symm etric \(m\) atrix packed sequentially, column by colum n, so thatA (1) containsa(1,1), A (2) and A (3) contain a( 1,2 ) and a (2,2) respectively, and so on. On exit, the array A is overw rilten by the upper triangular part of the updated \(m\) atrix. Before entry with UPLO = L'or I', the array A m ust contain the low er triangularpart of the sym \(m\) etric \(m\) atrix packed sequentially, colum \(n\) by colum \(n\), so that A ( 1 ) contains a ( 1,1 ), A (2) and A (3) contain \(a(2,1)\) and \(a(3,1)\) respectively, and so on. On exit, the array A is overw ritten by the low er triangular part of the updated \(m\) atrix.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dspr2-perform the symmetric rank 2 operation \(A:=\) alpha*x*y'+ alpha*y*x'+ A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DSPR2 (UPLO,N,ALPHA,X,NNCX,Y,INCY,AP)}
CHARACTER * 1 UPLO
INTEGERN,\mathbb{NCX,INCY}
DOUBLE PRECISION ALPHA
DOUBLE PRECISION X (*),Y (*),AP (*)
SUBROUTINEDSPR2_64(UPLO,N,ALPHA,X, INCX,Y, INCY,AP)
CHARACTER * 1 UPLO
INTEGER*8N,INCX,INCY
DOUBLE PRECISION ALPHA
DOUBLE PRECISION X (*),Y (*),AP (*)

```

\section*{F95 INTERFACE}

SU BROUTINE SPR2 (UPLO, \(\mathbb{N}], A L P H A, X,[\mathbb{N} C X], Y,[\mathbb{N} C Y], A P)\)
CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y\)
REAL (8) ::ALPHA
REAL (8), D IM ENSION (:) :: X,Y,AP
SU BROUTINE SPR2_64 (UPLO, \(\mathbb{N}], A L P H A, X,[\mathbb{N} C X], Y,[\mathbb{N} C Y], A P)\)
CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R(8):: N, \mathbb{I N C X}, \mathbb{N} C Y\)
REAL (8) :: ALPHA

REAL (8),D \(\mathbb{M} \operatorname{ENSION}(:):: X, Y, A P\)

\section*{C INTERFACE}
\#include <sunperfh>
void dspr2 (charuplo, intn, double alpha, double *x, int incx, double *y, int incy, double *ap);
void dspr2_64 (charuple, long n, double alpha, double *x, long incx, double *y, long incy, double *ap);

\section*{PURPOSE}
dspr2 performs the symm etric rank 2 operation \(A:=\) alpha*x*y' + alpha*y*x'+ A, w here alpha is a scalar, \(x\) and \(y\) are \(n\) elem ent vectors and \(A\) is an \(n\) by \(n\) sym \(m\) etric \(m\) atrix, supplied in packed form .

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the \(m\) atrix \(A\) is supplied in the packed amay AP as follow s:

UPLO = U 'or \(\mathrm{L}^{\prime}\) ' The uppertriangularpartofA is supplied in AP .

UPLO = L'or I' The low ertriangularpartof A is supplied in A P .

U nchanged on exit.

N (input)
O n entry, N specifies the order of the m atrix A . \(\mathrm{N}>=0\). U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
D ouble precision array, dimension (1+(n-
1)*abs ( \(\mathbb{N} C X)\) ) Before entry, the increm ented array

X m ust contain the n elem entvectorx. Unchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X<>0\). U nchanged on exit.

Y (input)
D ouble precision array, dimension (1 + (n -
1) *abs ( \(\mathbb{N} C Y\) )) B efore entry, the increm ented amay Y m ustcontain the \(n\) elem entvectory. U nchanged on exit.
\(\mathbb{N C C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(. \mathbb{N} C Y<>0\). U nchanged on exit.
AP (input/output)
D ouble precision array, dim ension ((n* \((n+1)) / 2)\) Before entry w ith \(\mathrm{UPLO}=\mathrm{U}\) 'or U ', the amay AP \(m\) ust contain the upper triangularpart of the sym \(m\) etric \(m\) atrix packed sequentially, colum \(n\) by colum \(n\), so thatAP (1) contains a (1, 1) , AP ( 2 ) and AP ( 3 ) contain \(a(1,2)\) and \(a(2,2)\) respectively, and so on. O n exit, the array AP is overw ritten by the uppertriangularpart of the updated \(m\) atrix. Before entry w th UPLO \(=\) L' or I', the amay A P m ustcontain the low ertriangularpart of the sym m etric matrix packed sequentially, colum \(n\) by colum \(n\), so thatA P (1 ) contains \(a(1,1), A P(2)\) and AP (3) contain \(a(2,1)\) and \(\mathrm{a}(3,1)\) respectively, and so on. On exit, the array A P is overw rilten by the low er triangularpart of the updated \(m\) atrix.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}

> dsprfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is sym \(m\) etric indefinite and packed, and provides emorbounds and backw ard emror estim ates for the solution

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE DSPRFS (UPLO,N,NRHS,AP,AF,\mathbb{PIVOT,B,LDB,X,LDX,FERR,}}\mathbf{N},\textrm{N},\textrm{N}
BERR,W ORK,W ORK2,INFO)
CHARACTER * 1 UPLO
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*),W ORK2(*)}
D OU BLE PRECISION AP (*),AF (*),B (LD B ,*),X (LDX ,*), FERR (*),
BERR (*),W ORK (*)
SU BROUTINE D SPRFS_64 (UPLO,N,NRHS,AP,AF, \mathbb{PIVOT,B,LDB,X,LDX,}
FERR,BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 UPLO
\mathbb{N}TEGER*8N,NRHS,LDB,LDX,\mathbb{NFO}
INTEGER*8 \mathbb{PIVOT (*),W ORK2 (*)}
DOU BLE PRECISION AP (*),AF (*),B (LDB,*),X (LDX ,*), FERR (*),
BERR (*),W ORK (*)

```

\section*{F95 INTERFACE}

SUBROUTINE SPRFS (UPLO,N, NRHS],AP,AF, \(\mathbb{P} \mathbb{I} O T, B,[L D B], X,[L D X]\), FERR, BERR, [W ORK], [W ORK 2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}, \mathrm{W} O R K 2\)

REAL (8), D \(\mathbb{M}\) ENSION (:) ::A,AF,FERR,BERR,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) :: B, X
SUBROUT \(\mathbb{N} E\) SPRFS_64 (UPLO, N, \(\mathbb{N} R H S], A P, A F, \mathbb{P} \mathbb{I} O T, B,[L D B], X,[L D X]\), FERR, BERR, [W ORK], [W ORK 2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER ( 8 ), D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T , W}\) ORK2
REAL (8),D \(\mathbb{M}\) ENSION (:) ::AP,AF,FERR,BERR,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::B, X

\section*{C INTERFACE}
\#include < sunperfh>
void dsprfs (char uplo, intn, intnrhs, double *ap, double *af, int *ípivot, double *b, int ldb, double *x, int Idx, double *fers, double *berr, int *info);
void dspnfs_64 (charuplo, long n, long nihs, double *ap, double *af, long *ịíivot, double *b, long ldb, double *x, long ldx, double *ferr, double *berr, long *info);

\section*{PURPOSE}
dsprfs in proves the com puted solution to a system of linear equations \(w\) hen the coefficientm atrix is sym \(m\) etric indefinite and packed, and provides errorbounds and backw ard error estim ates forthe solution.

\section*{ARGUMENTS}

UPLO (input)
= U ': U pper triangle ofA is stored;
= LL': Low ertriangle ofA is stored.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber
of colum ns of the m atrices B and X. NRHS >=0.

AP (input)
D ouble precsion array, dim ension \((\mathbb{N} *(N+1) / 2)\) The upper or low er triangle of the sym \(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linear array. The jth colum \(n\) of A is stored in the array A as follow s:
if UPLO \(=U U^{\prime}, A P(i+(j-1) \star j 2)=A(i, j)\) for
\(1<=i<=j\) ifUPLO \(=L \prime\) ' AP \(\left(i+(j-1)^{*}(2 * n-j) / 2\right)=\)
A \((i, j)\) for \(j=i<=n\).
AF (input)
D ouble precsion amay, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2\) ) The factored form of them atrix A. AF contains the block diagonalm atrix D and the multipliers used to obtain the factorU orL from the factorization \(A=U * D * U * * T\) or \(A=L * D * L * * T\) as computed by SSPTRF, stored as a packed triangularm atrix.
\(\mathbb{P I V O T}\) (input)
Integer array, dim ension \(\mathbb{N}\) ) D etails of the interchanges and the block structure ofD as determ ined by SSPTRF.

B (input) D ouble precision amray, dim ension (LDB,NRHS) The right hand side \(m\) atrix \(B\).

LD B (input)
The leading din ension of the array \(B\). LD B >= \(\max (1, N)\).

X (input/output)
D ouble precision array, dim ension (LDX,NRHS) On entry, the solution \(m\) atrix \(X\), as com puted by SSPTRS. On exit, the im proved solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the anay X . LDX >= \(\max (1, \mathbb{N})\).

FERR (output)
D ouble precision array, dimension (NHS) The estim ated forw ard enror bound foreach solution vectorX ( \(\mathcal{F}\) ) the \(j\) th column of the solution \(m\) atrix \(X)\). If XTRUE is the true solution corresponding to \(\mathrm{X}(\mathcal{j})\), FERR ( \(\mathcal{1}\) ) is an estim ated upper bound for the \(m\) agnitude of the largestele\(m\) ent in ( \(\mathrm{X}(\mathcal{\nu})-\mathrm{X}\) TRUE) divided by the m agnitude of the largest elem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

BERR (output)
D ouble precision aray, dim ension (NRHS) The componentw ise relative backw ard error of each solution vector X ( ) (i.e., the smallest relative change in any elem entof \(A\) orB thatm akes \(X(\mathcal{j})\) an
exactsolution).

W ORK (w orkspace)
D ouble precision array, dim ension ( \(3 \star \mathrm{~N}\) )

W ORK2 (w orkspace)
Integer aray, dim ension \((\mathbb{N})\)
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dspsv - com pute the solution to a real system of linear equations \(A * X=B\),

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
\mathbb{NTEGER N,NRHS,LDB,INFO}
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION A (*),B (LDB,*)
SUBROUT\mathbb{NEDSPSV_64(UPLO,N,NRHS,A,}\mathbb{P}\mathbb{IVOT,B,LDB,INFO)}
CHARACTER * 1 UPLO
INTEGER*8N,NRHS,LDB,INFO
INTEGER*8 \mathbb{PIVOT (*)}
DOUBLE PRECISION A (*),B (LDB,*)

```

\section*{F95 INTERFACE}

```

CHARACTER (LEN=1)::UPLO
$\mathbb{N}$ TEGER :: N, NRHS,LDB, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T$
REAL (8),D $\mathbb{M}$ ENSION (:) ::A
REAL (8),D IM ENSION (:,:) ::B
SUBROUTINESPSV_64 (UPLO,N, NRHS],A, $\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\operatorname{LDB}],[\mathbb{N} F O])$
CHARACTER (LEN=1) ::UPLO

```
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LD B, \(\mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \mathrm{ENSION}(:):: \mathbb{P} \mathbb{I} O T\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::A
REAL (8), D \(\mathbb{I M} E N S I O N(:,:):: B\)

\section*{C INTERFACE}
\#include <sunperfh>
void dspsv (charuplo, int \(n\), int nihs, double *a, int *ịivot, double *b, int ldlo, int *info);
void dspsv_64 (charuplo, long n, long nrhs, double *a, long *ịivot, double *b, long ldb, long *info);

\section*{PURPOSE}
dspsv com putes the solution to a real system of linearequations
\(A * X=B\), where \(A\) is an \(N\) boy \(-N\) symm etric \(m\) atrix stored in packed form at and \(X\) and \(B\) are \(N\) boy \(-N\) RH S \(m\) atrices.

The diagonalpivoting \(m\) ethod is used to factorA as
\(A=U * D * U * * T\), if \(U P L O=U\) ', or
\(A=L * D * L * * T\), if \(U P L O=L '\),
w here U (orL) is a productof perm utation and unit upper (low er) triangularm atrioes, \(D\) is sym m etric and block diagonalw th 1 -by -1 and 2 -by -2 diagonal blocks. The factored form ofA is then used to solve the system of equations A * \(X=B\).

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': Upper triangle ofA is stored;
\(=\mathrm{L}\) ': Low er triangle of A is stored.

N (input) The num ber of linearequations, ie., the order of them atrix A. N >=0.

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of them atrix B. NRH S \(>=0\).

A (input/output)
D ouble precision aray, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2)\) On
entry, the upper or low er triangle of the sym -
\(m\) etric \(m\) atrix A, packed colum nw ise in a linear
aray. The \(j^{\text {th }}\) column of \(A\) is stored in the array A as follow s: if UPLO \(=U\) ', A (i \(+(j\)
 \((j-1) *(2 n-j / 2)=A(i, j)\) for \(j=i<=n\). See below for furtherdetails.

On exit, the block diagonalm atrix \(D\) and the \(m u l\) tipliers used to obtain the factor \(U\) orL from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) as com puted by SSPTRF, stored as a packed triangular \(m\) atrix in the sam e storage form atasA.

IPIVOT (output)
Integer anray, dim ension \(\mathbb{N}\) ) D etails of the interchanges and the block structure of \(D\), as determ ined by SSPTRF. If IP IV OT (k) > 0, then row s and colum nsk and \(\mathbb{P} \mathbb{I} O T(k)\) were interchanged, and \(D(k, k)\) is a 1-by-1 diagonalblock. If \(U P L O=U^{\prime}\) and \(\mathbb{P} \mathbb{I V O T}(k)=\mathbb{P} \mathbb{I V O T}(k-1)<0\), then row sand colum nsk-1 and \(-\mathbb{P} \mathbb{I V}\) OT(k) were interchanged and D (k-1 \(k, k-1 k)\) is a \(2-b y-2\) diagonalblock. If \(\mathrm{UPLO}=\mathrm{L}\) 'and \(\mathbb{P} \mathbb{I V O T}(k)=\mathbb{P} \mathbb{I V O T}(k+1)<0\), then row s and colum ns \(k+1\) and \(-\mathbb{P}\) IV OT ( \(k\) ) were interchanged and \(D(k: k+1, k: k+1)\) is a \(2-b y-2\) diagonal block.

B (input/output)
D ouble precision array, dim ension (LD B, NRHS) On entry, the N -by-NRHS righthand sidem atrix B. On exit, if \(\mathbb{N} F O=0\), the N -by-NRHS solution matrix X.

LD B (input)
The leading dim ension of the anay \(B\). LD B \(>=\) \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{D}(\mathrm{i}, \mathrm{i})\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, so the solution could notbe com puted.

\section*{FURTHER DETAILS}

The packed storage scheme is illustrated by the follow ing exam ple when \(\mathrm{N}=4, \mathrm{UPLO}=\mathrm{U}\) ':

Tw o-dim ensional storage of the sym m etric m atrix A:
```

a11 a12 a13 a14
a22 a23 a24
a33 a34 (aij= aj̈)
a44

```

Packed storage of the upper triangle ofA :
\[
A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]
\]

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
```

dspsvx - use the diagonal pivoting factorization A =
U *D *U **T or A = L *D *L **T to com pute the solution to a real
system of linearequations A * X = B,where A is an N by N
symm etric m atrix stored in packed form at and X and B are N -
by-N RH S m atrices

```

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NEDSPSVX (FACT,UPLO,N,NRHS,AP,AF, PPIVOT,B,LDB,X,LDX,}
RCOND,FERR,BERR,W ORK,W ORK2, INFO)
CHARACTER * 1 FACT,UPLO
\mathbb{NTEGERN,NRHS,LDB,LDX,}\mathbb{NFO}
INTEGER PIVOT (*),W ORK2(*)
DOUBLE PRECISION RCOND
DOUBLE PRECISION AP (*),AF (*),B (LDB,*),X (LDX,*),FERR (*),
BERR (*),W ORK (*)
SUBROUT\mathbb{NEDSPSVX_64(FACT,UPLO,N,NRHS,AP,AF, IPIVOT,B,LDB,X,}
LDX,RCOND,FERR,BERR,W ORK,W ORK 2, \mathbb{NFO)}
CHARACTER * 1 FACT,UPLO
\mathbb{N}TEGER*8N,NRHS,LDB,LDX,}\mathbb{N}F
INTEGER*8 \mathbb{PIVOT (*),W ORK2 (*)}
DOUBLE PRECISION RCOND
DOUBLE PRECISION AP (*),AF (*),B (LDB,*),X (LDX,*),FERR (*),
BERR (*),W ORK (*)

```

\section*{F95 INTERFACE}

SUBROUTINE SPSVX (FACT,UPLO,N, NRHS],AP,AF, \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\operatorname{LDB}], \mathrm{X}\), [LDX],RCOND,FERR,BERR, [W ORK], [W ORK2], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1) ::FACT, UPLO
\(\mathbb{N}\) TEGER : \(: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} \mathrm{B}, \mathrm{LD} \mathrm{X}, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T, W O R K 2\)
REAL (8) ::RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::AP, AF,FERR, BERR, W ORK
REAL (8), D \(\mathbb{I}\) ENSION (:,:) ::B,X
 [LDX ],RCOND ,FERR,BERR, [WORK], [WORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1): :FACT,UPLO
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENS} \mathbb{O}\) (:) :: \(\mathbb{P} \mathbb{I} V O T, W\) ORK 2
REAL (8) ::RCOND
REAL (8), D \(\mathbb{M} E N S I O N(:):: A P, A F, F E R R, B E R R, W O R K\)
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::B,X

\section*{C INTERFACE}
\#include <sunperfh>
void dspsvx (char fact, charuplo, intn, int nrhs, double
*a, double *af, int*ipivot, double *b, int ldb, double *x, int ldx, double *roond, double *ferr, double *berr, int*info);
void dspsvx_64 (char fact, charuplo, long n, long nrhs, double *a, double *af, long *ipivot, double *b, long ldb, double *x, long ldx, double *roond, double *ferr, double *berr, long *info);

\section*{PURPOSE}

D SPSVX uses the diagonalpivoting factorization \(A=U * D * U * * T\) or \(A=L \star D * L * * T\) to com pute the solution to a realsystem of linear equations \(A * X=B\), where \(A\) is an \(N\) boy -N symm etric \(m\) atrix stored in packed form atand \(X\) and \(B\) are \(N\) boy-NRHS \(m\) atrices.

E morbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT \(=N\) ', the diagonalpivoting \(m\) ethod is used to factorA as
\(A=U * D * U * * T\), if \(U P L O=U\) ', or
\(A=L * D * L * * T\), ifUPLO \(=L^{\prime}\) ',
where \(U\) (orL) is a product ofperm utation and unitupper (low er)
triangularm atrioes and \(D\) is sym \(m\) etric and block diagonal w ith
1-by-1 and 2 -by-2 diagonalblocks.
2. If som eD \((i, i)=0\), so thatD is exactly singular, then the routine
retums w ith \(\mathbb{N}\) FO \(=\) i. \(O\) therw ise, the factored form of \(A\) is used
to estim ate the condition num ber of the \(m\) atrix \(A\). If the reciprocal of the condition num ber is less than \(m\) achine precision,
\(\mathbb{N} F O=N+1\) is retumed as a waming, but the routine stillgoes on
to solve for \(X\) and com pute error bounds as described below.
3. The system ofequations is solved for \(X\) using the factored form of A.
4. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for it.

\section*{ARGUMENTS}

FACT (input)
Specifies whether ornot the factored form of A has been supplied on entry. = F : O O entry, A F and \(\mathbb{P} \mathbb{I V O T}\) contain the factored form of A. AP, AF and \(\mathbb{P} \mathbb{I} O T \mathrm{w}\) ill not be modified. \(=\mathrm{N}\) : The \(m\) atrix A w illlbe copied to A F and factored.

UPLO (input)
\(=\mathrm{U}:\) : U pper triangle ofA is stored;
= L': Low ertriangle ofA is stored.
N (input) The num ber of linear equations, ie., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, ie., the num ber
of collum ns of the \(m\) atrices \(B\) and \(X\). NRHS \(>=0\).
AP (input)
D ouble precision array, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2\) ) The upper or low er triangle of the sym \(m\) etric \(m\) atrix \(A\),
packed colum nw ise in a linear array. The \(j\) th
colum \(n\) ofA is stored in the array AP as follow \(s\) :
ifUPLO = U',AP \((i+(j-1) * j 2)=A(i, j)\) for
\(1<=\mathrm{i}<=\dot{j}\) ifUPLO \(=\mathrm{L}\) ', AP \(\left(i+(j-1)^{*}\left(22^{*}-j\right) / 2\right)=\)
A \((i, 1)\) for \(j=i<=n\). See below for further details.

AF (input/output)
D ouble precision array, dim ension \(\mathbb{N}^{*}(\mathbb{N}+1) / 2\) ) If FACT = \(F^{\prime}\), then \(A F\) is an inputargum ent and on entry contains the block diagonalm atrix \(D\) and the m ultipliers used to obtain the factorU orL from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) as com puted by SSPTRF, stored as a packed triangular \(m\) atrix in the sam e storage form atas \(A\).

IfFACT \(=N\) ', then \(A F\) is an output argum ent and on exit contains the block diagonalm atrix \(D\) and the m ultipliers used to obtain the factorU or L from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) as com puted by SSPTRF, stored as a packed triangularm atrix in the sam e storage form at as A.

IPIVOT (inputoroutput)
Integer array, dim ension (N) IfFACT = F', then \(\mathbb{P} \mathbb{V O T}\) is an inputargum ent and on entry contains details of the interchanges and the block structure ofD, as determ ined by SSPTRF. If \(\mathbb{P} \mathbb{V} O T(k)\) \(>0\), then row sand colum nsk and \(\mathbb{P} \mathbb{I} O T(k)\) were interchanged and \(\mathrm{D}(\mathrm{k}, \mathrm{k})\) is a 1 -by-1 diagonal block. If \(\mathrm{PLO}=\mathrm{U}\) 'and \(\mathbb{P} \mathbb{I V O T}(k)=\mathbb{P} \mathbb{V} O T(k-1)\) \(<0\), then row s and colum ns k-1 and - \(\mathbb{P}\) IV O T (k) w ere interchanged and \(D(k-1 k, k-1 k)\) is a 2 -by-2 diagonal block. If UPLO \(=\mathbb{L}\) ' and \(\mathbb{P} \mathbb{I V O T}(k)=\) PIVOT \((k+1)<0\), then row \(s\) and colum ns \(k+1\) and \(-\mathbb{P} \mathbb{I V O T}(k)\) were interchanged and \(D(k: k+1, k: k+1)\) is a 2 -by-2 diagonalblock.

IfFACT = \(\mathrm{N}^{\prime}\), then \(\mathbb{P} \mathbb{I V O T}\) is an output argum ent and on exit contains details of the interchanges and the block structure of D, as determ ined by SSPTRF.

B (input) D ouble precision array, dim ension (LDB,NRHS) The N -by-N RH S righthand side m atrix B .

LD B (input)
The leading dim ension of the array \(B\). LD B >= \(\max (1, \mathbb{N})\).

\section*{X (output)}

D ouble precision anray, dim ension (LD X, NRHS) If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=N+1\), the \(N\)-by \(-N\) RH S solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the anay \(\mathrm{X} . \mathrm{LD} \mathrm{X}>=\) \(\max (1, N)\).

RCOND (output)
The estim ate of the reciprocal condition num ber of the \(m\) atrix A. IfRCOND is less than them achine precision (in particular, if RCOND \(=0\) ), the \(m\) atrix is singular to working precision. This condition is indicated by a retum code of \(\mathbb{N F O}\) > 0.

FERR (output)
D ouble precision array, dimension (NRHS) The estim ated forw ard error bound foreach solution vectorX ( \(\mathcal{j}\) ) the \(j\) th column of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{D})\), FERR ( \(\mathcal{H}\) ) is an estim ated upper bound for the \(m\) agnitude of the largestele\(m\) ent in ( \(X(\mathcal{\nu})\)-X TRU E) divided by the \(m\) agnitude of the largest elem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

D ouble precision array, dim ension (NRHS) The com ponentw ise relative backw ard error of each solution vector \(X(\mathcal{O}\) (ie., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{(})\) an exactsolution).

W ORK (w orkspace)
D ouble precision array, dim ension ( \(3{ }^{*} \mathrm{~N}\) )
W ORK 2 (w orkspace)
Integer anay, dim ension (N)
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N} F O=i\), and \(i\) is
\(<=\mathrm{N}: \mathrm{D}(i, i)\) is exactly zero. The factorization has been completed but the factorD is exactly singular, so the solution and error bounds could
not be com puted. \(\mathrm{RCOND}=0\) is retumed. \(=\mathrm{N}+1: \mathrm{D}\) is nonsingular, butRCOND is less than \(m\) achine precision, \(m\) eaning that the \(m\) atrix is singular to w orking precision. N evertheless, the solution and error bounds are com puted because there are a num ber of situationsw here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{FURTHER DETAILS}

The packed storage schem e is illustrated by the follow ing
exam plewhen \(\mathrm{N}=4, \mathrm{UPLO}=\mathrm{U}\) ':
Tw o-dim ensional storage of the sym \(m\) etric \(m\) atrix A :
```

a11 a12 a13 a14
a22 a23 a24
a33 a34 (aij= aj̈)
a44

```

Packed storage of the upper triangle ofA :
\(A P=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]\)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dsptrd -reduce a real sym \(m\) etric \(m\) atrix A stored in packed form to sym \(m\) etric tridiagonal form \(T\) by an orthogonalsim ilarity transform ation

\section*{SYNOPSIS}
```

SUBROUTINEDSPTRD(UPLO,N,AP,D,E,TAU,\mathbb{NFO)}
CHARACTER * 1 UPLO
INTEGERN,\mathbb{NFO}
DOUBLE PRECISION AP (*),D (*),E (*),TAU (*)
SUBROUT\mathbb{NEDSPTRD_64(UPLO,N,AP,D ,E,TAU, INFO)}
CHARACTER * 1 UPLO
INTEGER*8N,\mathbb{NFO}
DOUBLE PRECISION AP (*),D (*),E (*),TAU (*)
F95 INTERFACE
SUBROUT\mathbb{NE SPTRD (UPLO,N,AP,D,E,TAU, [NNFO])}
CHARACTER (LEN=1) ::UPLO
\mathbb{NTEGER::N,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REAL (8),D IM ENSION (:) ::AP,D,E,TAU

```

```

    CHARACTER (LEN=1) ::UPLO
    \mathbb{NTEGER (8)::N,\mathbb{NFO}}\mathbf{N}\mathrm{ ()}
    REAL (8),D IM ENSION (:) ::AP,D,E,TAU
    ```

\section*{C INTERFACE}
\#include <sunperfh>
void dsptrd (charuple, intn, double *ap, double *d, double *e, double *tau, int *info);
void dsptrd_64 (charuple, long n, double *ap, double *d, double *e, double *tau, long *info);

\section*{PURPOSE}
dsptrd reduces a real sym \(m\) etric \(m\) atrix A stored in packed form to symm etric tridiagonal form T by an orthogonalsim ilarity transform ation: \(Q * * T * A * Q=T\).

\section*{ARGUMENTS}

UPLO (input)
\(=U\) ': U pper triangle ofA is stored;
\(=\mathbb{L}\) ': Low ertriangle of \(A\) is stored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

AP (input)
D ouble precision array, dim ension \(\left.\mathbb{N}^{*}(\mathbb{N}+1) / 2\right)\) On entry, the upper or low er triangle of the sym \(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linear array. The jth column of \(A\) is stored in the array AP as follow s: if UPLO \(=U^{\prime}, A P(i+(j\) \(1) \star j 2)=A(i, 7)\) for \(1<=i<=j\) ifUPLO \(=L^{\prime}, A P(i\) \(+(j-1) *(2 * n-j / 2)=A(i, j)\) for \(j=i<=n\). On exit, if \(\mathrm{UPLO}=\mathrm{U}\) ', the diagonal and first superdiagonalofA are overw rilten by the corresponding elem ents of the tridiagonalm atrix T , and the ele\(m\) ents above the first superdiagonal, w ith the anray TAU, represent the orthogonalm atrix Q as a productofelem entary reflectors; ifUPLO = L', the diagonaland first subdiagonal of A are overw ritten by the corresponding elem ents of the tridiagonal \(m\) atrix \(T\), and the elem ents below the first subdiagonal, w ith the aray \(T A U\), represent the orthogonalm atrix \(Q\) as a productofelem entary reflectors. See FurtherD etails.

D (output)
D ouble precision aray, dim ension \(\mathbb{N}\) ) The diagonal
elem ents of the tridiagonal matrix T:D (i)=

A (i,i).

E (output)
D ouble precision array, dim ension \((\mathbb{N}-1\) ) The offdiagonal elem ents of the tridiagonalm atrix T : \(\mathrm{E}(\mathrm{i})=\mathrm{A}(\mathrm{i}, \mathrm{i}+1)\) if \(\mathrm{UPLO}=\mathrm{U}^{\prime}, \mathrm{E}(\mathrm{i})=\mathrm{A}(\mathrm{i}+1, \mathrm{i})\) if UPLO = L'.

TAU (output)
D ouble precision anay, dim ension \((\mathbb{N}-1)\) The scalar factors of the elem entary reflectors (see Further D etails).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{FURTHER DETAILS}

If \(\mathrm{PLO}=\mathrm{U}\) ', the matrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(n-1) \ldots H(2) H(1) .
\]

Each \(H\) (i) has the form
H (i) \(=I-\tan * V^{*} V^{\prime}\)
where tau is a real scalar, and \(v\) is a realvectorw ith \(v(i+1 \mathrm{n})=0\) and \(\mathrm{v}(\mathrm{i})=1\); \(\mathrm{v}(1: i-1)\) is stored on exitin \(A P\), overw riting A ( \(1: 1-1, i+1)\), and tau is stored in TAU (i).

If \(\mathrm{PLO}=\mathrm{L}\) ', the m atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(1) H(2) \ldots H(n-1) .
\]

Each H (i) has the form
H (i) \(=I-\tan * V^{*} V^{\prime}\)
where tau is a real scalar, and \(v\) is a realvectorw ith \(v(1: i)=0\) and \(v(i+1)=1\); \(v(i+2 n)\) is stored on exitin AP, overw riting A (i+2 \(\mathrm{n}, \mathrm{i}\) ), and tau is stored in TAU (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dsptrf-com pute the factorization of a real symm etric \(m\) atrix A stored in packed form atusing the B unch-K aufm an diagonalpivoting \(m\) ethod

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
\mathbb{NTEGERN, \mathbb{NFO}}0\mathrm{ (})=
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION AP (*)

```

```

CHARACTER * 1 UPLO
\mathbb{NTEGER*8 N, INFO}
NNTEGER*8 \mathbb{PIVOT (*)}
DOUBLE PRECISION AP (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE SPTRF (UPLO ,N,AP, $\mathbb{P} \mathbb{I V O T},[\mathbb{N} F O]$ )
CHARACTER (LEN=1)::UPLO
$\mathbb{N} T E G E R:: N, \mathbb{N F O}$
$\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V O T}$
REAL (8),D $\mathbb{M}$ ENSION (:) ::AP
SU BROUTINE SPTRF_64 (UPLO, N,AP, $\mathbb{P} \mathbb{I V O T},[\mathbb{N} F O])$

```

CHARACTER (LEN=1) ::UPLO
```

$\mathbb{N}$ TEGER ( 8 ) :: $\mathrm{N}, \mathbb{N} F \mathrm{O}$

```
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V O T}\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::AP

\section*{C INTERFACE}
\#include < sunperfh>
void dsptrf(charuplo, intn, double *ap, int *ipivot, int *info);
void dsptrf_ 64 (char uplo, long n, double *ap, long *ipivot, long *info);

\section*{PURPOSE}
dsptrf com putes the factorization of a realsym \(m\) etric \(m\) atrix A stored in packed form at using the Bunch \(-K\) aufm an diagonal pivoting \(m\) ethod:
\[
\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T} \text { or } \mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}
\]
where \(U\) (orL) is a productof perm utation and unit upper (low er) triangular \(m\) atrices, and \(D\) is sym \(m\) etric and block diagonalw th 1 -by -1 and 2 -by- 2 diagonalblocks.

\section*{ARGUMENTS}

UPLO (input)
= U ': U pper triangle ofA is stored;
= L': Low er triangle ofA is stored.
N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
AP (input/output)
D ouble precision anay, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2\) ) On entry, the upper or low er triangle of the sym \(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linear array. The \(j\) th column of \(A\) is stored in the array AP as follow s: if UPLO = U', AP (i + \((j\) \(\left.1)^{\star} \mathfrak{j} 2\right)=A(i, 7)\) for \(1<=i<=j\) ifUPLO \(=L^{\prime}, A P(i\) \(+(j-1)^{\star}(2 n-7 / 2)=A(i, j)\) for \(j=i<=n\).

On exit, the block diagonalm atrix \(D\) and the \(m u l\) tipliers used to obtain the factorU orL, stored as a packed triangularm atrix overw riting A (see below for further details).

\section*{IPIVOT (output)}

Integer amay, dim ension \(\mathbb{N}\) ) D etails of the interchanges and the block structure of D. If \(\mathbb{P} \mathbb{I V O T}(\mathrm{k})>0\), then rows and colum ns k and IP IV O T \((k)\) w ere interchanged and \(D(k, k)\) is a 1 -by-1 diagonalblock. IfUPLO \(=\mathrm{U}^{\prime}\) and \(\mathbb{P} \mathbb{I V O T}(k)=\) \(\mathbb{P} \mathbb{I V O T}(k-1)<0\), then row s and colum nsk-1 and - \(\mathbb{P}\) IV O T \((k)\) w ere interchanged and \(D(k-1 * k, k-1 k)\) is a 2 -by-2 diagonal block. If UPLO = L'and \(\mathbb{P} \mathbb{I V} \circ T(k)=\mathbb{P} \mathbb{I} \circ T(k+1)<0\), then row s and colum \(n s\) \(\mathrm{k}+1\) and \(-\mathbb{P} \mathbb{I V O T}(\mathrm{k})\) were interchanged and D \((k: k+1, k: k+1)\) is a 2 -by -2 diagonalblock.

\section*{\(\mathbb{N} F O\) (output)}
= 0: successfiulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the \(i\)-th argum ent had an illegal value
\(>0:\) if \(\mathbb{N F O}=i, D(i, i)\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix D is exactly singular, and division by zero w illoccur if it is used to solve a system of equations.

\section*{FURTHER DETAILS}

5-96 - B ased on m odifications by J. Lew is, B oeing C om puter Services

C om pany

If \(\mathrm{U} P \mathrm{LO}=\mathrm{U}\) ', then \(A=U * D * U\) ', where
\(U=P(n) \star U(n)^{\star} \ldots{ }^{*} P(k) U(k)^{\star} \ldots\),
i.e., \(U\) is a productof term \(s P(k) * U(k)\), where \(k\) decreases from \(n\) to 1 in steps of 1 or 2 , and \(D\) is a block diagonal \(m\) atrix \(w\) ith 1 -by-1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I V} O T(k)\), and \(U(k)\) is a unituppertriangularm atrix, such that if the diagonal
block \(D(k)\) is of orders \((s=1\) or 2 ), then
```

    ( I v 0 ) k-s
    U (k)=(0 I 0 ) s
( 0 0 I ) n-k
k-s s n-k

```

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(1, k-\) \(1, k)\). If \(s=2\), the uppertriangle of \((k)\) overw rites \(A(k-\) \(1, k-1), A(k-1, k)\), and \(A(k, k)\), and \(v\) overw rites \(A(1 k-2, k-\) 1 k).
```

IfU PLO = L', then A = L *D *L',w here
L = P (l)*L (l)* ... *P (k)*L (k)* ...,

```
i.e., \(L\) is a productofterm \(S P(k) * L(k)\), where \(k\) increases from 1 to \(n\) in steps of 1 or2, and \(D\) is ablock diagonal \(m\) atrix \(w\) th 1 -by -1 and 2 -by- 2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I V O T}(k)\), and \(L(k)\) is a unit low ertriangularm atrix, such that if the diagonal block \(D(k)\) is of orders ( \(s=1\) or 2 ), then
```

    ( I 0 0 ) k-1
    L (k)=( 0 I 0 ) s
    ( 0 v I ) n-k-s+1
        k-1 s n-k-s+1
    ```

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites A \((k+1 n, k)\). If \(s=2\), the low er triangle ofD ( \(k\) ) overw rites A \((k, k), A(k+1, k)\), and \(A(k+1, k+1)\), and \(v\) overw rites \(A(k+2 n, k k+1)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsptri-com pute the inverse of a realsym \(m\) etric indefinite \(m\) atrix \(A\) in packed storage using the factorization \(A=\) \(\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by SSPTRF

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
\mathbb{NTEGER N,}\mathbb{N}FO
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION AP (*),W ORK (*)

```

```

CHARACTER * 1 UPLO
\mathbb{NTEGER*8 N,\mathbb{NFO}}\mathbf{N}\mathrm{ ( }
INTEGER*8 \mathbb{PIVOT (*)}
DOUBLE PRECISION AP (*),W ORK (*)

```
F95 INTERFACE
    SU BROUT \(\mathbb{N} E \operatorname{SPTRI(UPLO,N,AP,~\mathbb {P}\mathbb {I}OT,[WORK],[\mathbb {N}FO])~}\)
    CHARACTER (LEN=1) ::UPLO
    \(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
    \(\mathbb{N}\) TEGER,D \(\mathbb{I}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V} O T\)
    REAL (8),D \(\mathbb{M}\) ENSION (:) ::AP,W ORK
    SU BROUTINE SPTRI_64 (UPLO,N,AP, \(\mathbb{P} \mathbb{I V O T},[\mathbb{W}\) ORK \(],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO
    \(\mathbb{N} T E G E R(8):: N, \mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::AP,WORK

\section*{C INTERFACE}
\#include <sunperfh>
void dsptri(char uplo, intn, double *a, int *ipivot, int *info);
void dsptri_64 (charuplo, long n, double *a, long *ipivot, long *info);

\section*{PURPOSE}
dsptricom putes the inverse of a real sym \(m\) etric indefinite \(m\) atrix \(A\) in packed storage using the factorization \(A=\) \(U * D * U * * T\) orA \(=L * D * L * * T\) com puted by SSPTRF.

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) :: U pper triangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}\) **T;
\(={ }^{L} \mathrm{~L}:\) : Low er triangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\).

N (input) The order of them atrix A. \(\mathrm{N}>=0\).

A P (input/output)
D ouble precision array, dim ension \((\mathbb{N} *(\mathbb{N}+1) / 2\) ) On entry, the block diagonalm atrix D and the m ultipliers used to obtain the factor \(U\) or \(L\) as com puted by SSPTRF, stored as a packed triangular \(m\) atrix.

On exit, if \(\mathbb{N F O}=0\), the (sym m etric) inverse of the originalm atrix, stored as a packed triangular \(m\) atrix. The \(j\) th colum \(n\) of inv \((A)\) is stored in the array AP as follow s: ifUPLO = U',AP (i+ (j 1) \(\star \dot{j} 2)=\operatorname{inv}(A)(i, j)\) for \(1<=i<=j ;\) ifUPLO \(=L^{\prime}\) ', \(A P(i+(j-1) *(2 n-j) / 2)=\operatorname{inv}(A)(i, j)\) for \(\dot{<}=i<=n\).

IPIVOT (input)
Integer array, dim ension \(\mathbb{N}\) ) D etails of the interchanges and the block structure ofD as determ ined by D SPTRF.

W ORK (w ork.space)
D ouble precision array, dim ension \((\mathbb{N})\)
\(\mathbb{N}\) FO (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N N F O}=i, D(i, i)=0\); the \(m\) atrix is singular and its inverse could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dsptrs-solve a system of linearequationsA *X = B w ith a
real symm etric m atrix A stored in packed form atusing the
factorization A = U *D *U**T or A = L*D *L**T com puted by
D SPTRF

```

\section*{SYNOPSIS}

CHARACTER * 1 UPLO
\(\mathbb{N}\) TEGERN,NRHS,LDB, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R \mathbb{P} \mathbb{V} O T\left({ }^{( }\right)\)
D OUBLE PREC ISION AP (*), B (LD B , *)
SU BROUTINEDSPTRS_64 (UPLO,N,NRHS,AP, \(\mathbb{P} \mathbb{I V O T}, \mathrm{B}, \mathrm{LD} B, \mathbb{N} F O)\)
CHARACTER * 1 UPLO
\(\mathbb{N}\) TEGER*8N,NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{I V} O T(\star)\)
DOUBLE PRECISION AP (*), B (LDB, \(\left.{ }^{\star}\right)\)

\section*{F95 INTERFACE}

SU BROUTINE SPTRS (UPLO,N, NRHS],AP, \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\operatorname{LDB}],[\mathbb{N F O}])\)
CHARACTER (LEN=1)::UPLO
\(\mathbb{N}\) TEGER ::N,NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V} O T\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::AP
REAL (8),D IM ENSIO N (:,:) ::B
SU BROUTINE SPTRS_64 (UPLO ,N, NRHS],AP, \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER (8) :: N,NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P I V O T}\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::AP
REAL (8),D \(\mathbb{M}\) ENSION (:,:) :: B

\section*{C INTERFACE}
\#include <sunperfh>
void dsptrs (charuplo, intn, int nrhs, double *ap, int
*ịíiot, double *b, int ldb, int *info);
void dsptrs_64 (charuplo, long n, long nrhs, double *ap, long *ipìivot, double *b, long ldb, long *info);

\section*{PURPOSE}
dsptrs solves a system of linear equations \(A * X=B\) with a real sym \(m\) etric \(m\) atrix A stored in packed form atusing the factorization \(A=U * D * U * * T\) or \(A=L * D * L * * T\) com puted by D SPTRF .

\section*{ARGUMENTS}

\section*{UPLO (input)}

Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) : : U ppertriangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\);
\(=\mathrm{L}\) ': Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\).

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS >=0.

AP (input)
D ouble precision array, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2\) ) The
block diagonal \(m\) atrix \(D\) and the \(m\) ultipliers used to obtain the factorU orL as com puted by D SPTRF, stored as a packed triangularm atrix.

PIVOT (input)
Integer array, dim ension \(\mathbb{N}\) ) D etails of the interchanges and the block structure ofD as determ ined by D SPTRF.

B (input/output)
D ouble precision anay, dim ension (LD B ,NRHS) On
entry, the righthand side m atrix B. On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dstebz -com pute the eigenvalues of a sym m etric tridiagonal \(m\) atrix \(T\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSTEBZ RANGE,ORDER,N,VL,VU,\mathbb{L},\mathbb{U},ABSTOL,D,E,M,}
NSPLIT,W,\mathbb{BLOCK,ISPLIT,W ORK,INORK,\mathbb{NFO)}}\mathbf{N}\mathrm{ (T)}
CHARACTER * 1 RANGE,ORDER
\mathbb{N}TEGERN,\mathbb{L},\mathbb{U},M,NSPLTT,\mathbb{NFO}
\mathbb{NTEGER BBLOCK (*),ISPLIT (*), IN ORK (*)}
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION D (*),E (*),W (*),W ORK (*)
SUBROUT\mathbb{NEDSTEBZ_64 RANGE,ORDER,N,VL,VU,|,IU,ABSTOL,D,E,}
M,NSPLIT,W,\mathbb{BLOCK},ISPLIT,WORK,INORK,\mathbb{NFO)}

```
CHARACTER * 1 RANGE, ORDER
\(\mathbb{N} T E G E R * 8 N, \mathbb{I}, \mathbb{U}, M, N S P L \mathbb{T}, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{B L O C K}(*)\), \(\operatorname{ISPLIT}(*), \mathbb{I N}\) ORK (*)
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISIOND (*), E (*), W (*), WORK (*)

\section*{F95 INTERFACE}

SU BROUTINE STEBZ (RANGE,ORDER,N,VL,VU, \(\mathbb{I}, \mathbb{U}, A B S T O L, D, E, M\),


CHARACTER (LEN=1)::RANGE,ORDER
\(\mathbb{N} T E G E R:: N, \mathbb{L}, \mathbb{U}, \mathrm{M}, \mathrm{N}\) SPLIT, \(\mathbb{N} F \mathrm{O}\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: \(\mathbb{B L O C K}, \operatorname{ISPLIT}, \mathbb{I N}\) ORK
REAL (8) ::VL,VU,ABSTOL
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D , E, W, W ORK

SU BROUT \(\mathbb{N} E\) STEBZ_64 (RANGE, ORDER, N, VL, VU, \(\mathbb{I}, ~ \mathbb{U}, ~ A B S T O L, D, E, M\), N SPLIT, W, \(\mathbb{B} L O C K, I S P L I T,[W O R K],[\mathbb{W} O R K],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::RANGE,ORDER
\(\mathbb{N}\) TEGER (8) :: \(N, \mathbb{I}, \mathbb{U}, \mathrm{M}, \mathrm{N}\) SPL \(\mathbb{I}, \mathbb{I N F O}\)
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{M} \operatorname{ENSION}(:):: \mathbb{B L O C K}, \operatorname{ISPL} \mathbb{I}, \mathbb{I}\) ORK
REAL (8) ::VL,VU,ABSTOL
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D , E, W ,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void dstebz (char range, char order, intn, double vl, double vu, int il, int in, double abstol, double *d, double *e, int*m, int *nsplit, double *w, int *iblock, int *isplit, int *info);
void dstebz_64 (char range, charorder, long n, double vl, double vu, long il, long iu, double abstol, double \({ }^{*} \mathrm{~d}\), double \({ }^{*} \mathrm{e}\), long \({ }_{\mathrm{m}}\), long \({ }^{\text {nsplit, }}\) double \({ }_{\mathrm{w}}\), long *iblock, long *isplit, long *info);

\section*{PURPOSE}
dstebz com putes the eigenvalues of a sym metric tridiagonal \(m\) atrix T. The userm ay ask for alleigenvalues, alleigenvalues in the half-open interval \(N L, V U]\), or the \(\Pi\)-th through \(\mathbb{I U}\) th eigenvalues.

To avoid overflow, the m atrix m ustbe scaled so that its largestelem ent is no greater than overflow ** (1/2) * underflow ** (1/4) in absolute value, and forgreatest accuracy, itshould notbe m uch sm aller than that.

See W . K ahan "A ccurate Eigenvalues of a Sym m etric TridiagonalM atrix", ReportC S41, C om puterScience D ept., Stanford U niversity, July 21, 1966.

\section*{ARGUMENTS}

RANGE (input)
= A ': ("A ll") alleigenvalues w illbe found.
= V ': ("V alue") alleigenvalues in the half-open interval \(\mathrm{VL}, \mathrm{V}\) U ] w illbe found. = ' I ': ("Index") the \(\mathbb{I}\)-th through \(\mathbb{I U}\)-th eigenvalues (of the entire m atrix) w illbe found.

ORDER (input)
= B ': ("By B lock") the eigenvalues will be grouped by split-off block (see \(\mathbb{B L O C K}\), ISPLIT) and ordered from sm allest to largest w ithin the block. = E ': ("Entire m atrix") the eigenvalues for the entire m atrix w illbe ordered from sm allest to largest.

N (input) The order of the tridiagonalm atrix \(\mathrm{T} . \mathrm{N}>=0\).

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvahues. Eigenvalues less than or equal to \(V L\), or greater than VU, will not be retumed. VL < VU. N ot referenced ifRANGE=A'or 'I'.

VU (input)
See the description of L .

IL (input)
If RA N GE= ' I ', the indices (in ascending order) of the smallest and largest eigenvahues to be retumed. \(1<=\mathbb{L}<=\mathbb{U}<=N\), if \(N>0\); \(\mathbb{H}=1\) and \(\mathbb{U}=0\) if \(N=0\). \(N\) otreferenced ifRANGE \(=\) A'or V'.

IU (input)
See the description of \(\Pi\).

ABSTOL (input)
The absolute tolerance for the eigenvalues. A n eigenvalue (or cluster) is considered to be located if ithas been determ ined to lie in an intervalw hose w idth is A BSTOL or less. IfA BSTOL is less than orequal to zero, then \(U L P *|\mid w i l l\) be used, w here \(I \mid m\) eans the 1 -norm of \(T\).

E igenvalues w illbe com puted m ost accurately w hen ABSTOL is set to tw ige the underflow threshold 2*SLAM CH (S ), notzero.

D (input) The \(n\) diagonalelem ents of the tridiagonal \(m\) atrix T.

E (input) The \((n-1)\) off-diagonalelem ents of the tridiagonal m atrix T.

M (output)

The actual num berofeigenvahues found. \(0<=\mathrm{M}\) <= N. (Se also the description of \(\mathbb{N} F O=2,3\).)

NSPLIT (output)
The num ber of diagonalblocks in the m atrix T. 1
<= NSPLIT <= N .

W (output)
On exit, the firstM elem ents of \(W\) will contain the eigenvalues. (SSTEBZ may use the rem aining N M elem ents as w orkspace.)

BBLOCK (output)
A teach row /collm n jw here \(E()\) is zero or sm all, the m atrix T is considered to split into ablock diagonalm atrix. On exit, if \(\mathbb{N F O}=0, \mathbb{B L O C K}\) (i) specifies to which block (from 1 to the num ber of blocks) the eigenvalue W (i) belongs. (SSTEBZ may use the rem aining N M elem ents as w orkspace.)

ISPLIT (output)
The splitting points, atw hich \(T\) breaks up into subm atrices. The first subm atrix consists of row s/columns 1 to ISPLIT (1), the second of row s/colum ns ISPL IT (1)+1 through ISPLIT (2), etc., and the NSPLIT-th consists of row s/colum ns ISPLIT \((\mathbb{N}\) SPLIT-1) +1 through ISPLIT \((\mathbb{N} S P L I T)=N\). (Only the firstN SPLIT elem entsw ill actually be used, but since the user cannot know a prioriw hat value NSPLIT w ill have, N w ordsm ust be reserved for ISPLIT.)

W ORK (w orkspace)
dim ension ( \(4 * \mathrm{~N}\) )
IN ORK (w orkspace)
dim ension \((3 * N)\)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue
> 0: som e orall of the eigenvalues failed to converge or
w ere not com puted:
\(=1\) or 3 : B isection failed to converge for som e eigenvalues; these eigenvalues are flagged by a negative block num ber. The effect is that the eigenvalues \(m\) ay notbe as accurate as the absolute and relative tolerances. This is generally caused
by unexpectedly inaccurate arithm etic. \(=2\) or 3 :
RANGE=I'only:N otallof the eigenvalues \(\mathbb{I}: \mathbb{U}\) w ere found.
Effect: \(M<\mathbb{U}+1-\mathbb{I}\)
C ause: non-m onotonic arithm etic, causing the Sturm sequence to be non-m onotonic. Cure: recalculate, using \(R\) A N G E= A ', and pick
outeigenvalues \(\mathbb{I L}: \mathbb{Z U} .=4: \quad\) RANGE= I', and the
G ershgorin interval initially used w as too sm all.
N o eigenvahues w ere com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dstedc - com pute alleigenvalues and, optionally, eigenvectors of a sym \(m\) etric tridiagonalm atrix using the divide and conquerm ethod

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSTEDC (COMPZ,N,D,E,Z,LD Z,W ORK,LW ORK,IN ORK,LIN ORK,}
INFO)
CHARACTER * 1 COMPZ

```

```

INTEGER IN ORK (*)
DOUBLE PRECISION D (*),E (*),Z (LD Z,*),W ORK (*)
SUBROUT\mathbb{NEDSTEDC_64 (COMPZ,N,D,E,Z,LDZ,W ORK,LW ORK,IN ORK,}
LIN ORK,\mathbb{NFO)}
CHARACTER * 1 COMPZ
\mathbb{NTEGER*8N,LD Z,LW ORK,LIN ORK,INFO}
INTEGER*8 IN ORK (*)
DOUBLE PRECISIOND (*),E (*),Z (LD Z,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE STEDC COMPZ,N,D,E,Z, [LD Z], [W ORK], [LW ORK ], [IW ORK ], \(\left[\begin{array}{ll}\mathbb{N} & \mathrm{ORK}],[\mathbb{N F O}])\end{array}\right.\)

CHARACTER (LEN=1) ::COM PZ
\(\mathbb{N}\) TEGER ::N,LDZ,LW ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
REAL (8),D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK
REAL (8), D IM ENSION (:,:) :: Z

SU BROU TINE STEDC_64 (COM PZ,N,D,E,Z, [LD Z], [W ORK ], [LW ORK], [IW ORK ], \(\left[\begin{array}{l}\mathbb{N} \\ \text { ORK }],[\mathbb{N} F O])\end{array}\right.\)

CHARACTER (LEN=1): :COM PZ
\(\mathbb{N}\) TEGER (8) ::N,LD Z,LW ORK, LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
REAL (8),D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::Z

\section*{C INTERFACE}
\#include < sunperfh>
void dstedc (char com pz, intn, double *d, double *e, double *z, int ldz, int*info);
void dstedc_64 (char com pz, long n, double *d, double *e, double *z, long ldz, long *info);

\section*{PURPOSE}
dstedc com putes alleigenvalues and, optionally, eigenvectors of a sym \(m\) etric tridiagonalm atrix using the divide and conquerm ethod. The eigenvectors of a full or band real sym metric \(m\) atrix can also be found ifSSY TRD orSSPTRD or SSBTRD hasbeen used to reduce this \(m\) atrix to tridiagonal form.

This code \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. It w illw ork on \(m\) achines \(w\) ith a guard digit in add/subtract, or on those binary machines w thout guard digits which subtract like the C ray X \(-\mathrm{M} P\), C ray \(Y\) M P , C ray C-90, or C ray-2. Itcould conœívably fail on hexadecim al or decin al machines w thout guard digits, butw e know of none. Se SLAED 3 for details.

\section*{ARGUMENTS}

COMPZ (input)
\(=\mathrm{N}\) ': C om pute eigenvalues only .
= I': C om pute eigenvectors of tridiagonalm atrix also.
\(=\mathrm{V}\) : C om pute eigenvectors of original dense symmetric \(m\) atrix also. On entry, \(Z\) contains the orthogonalm atrix used to reduce the original \(m\) atrix to tridiagonal form .

N (input) The dim ension of the sym \(m\) etric tridiagonalm atrix.
\(\mathrm{N}>=0\).

D (input/output)
On entry, the diagonalelem ents of the tridiagonal m atrix. On exit, if \(\mathbb{N} F O=0\), the eigenvalues in ascending order.

E (input/output)
O n entry, the subdiagonalelem ents of the tridiagonalm atrix. On exit, E hasbeen destroyed.

Z (input) On entry, if COMPZ = V', then Z contains the orthogonal m atrix used in the reduction to tridiagonal form. On exit, if \(\mathbb{I N F O}=0\), then if \(C O M P Z\) \(=\mathrm{V}\) ', Z contains the orthonorm aleigenvectors of the original sym \(m\) etric \(m\) atrix, and if \(C O M P Z=I\) ', \(Z\) contains the orthonorm al eigenvectors of the sym \(m\) etric tridiagonalm atrix. If \(\mathrm{COMPZ}=\mathrm{N}\) ', then Z is not referenced.

LD Z (input)
The leading din ension of the array Z . LD \(\mathrm{Z}>=1\).
If eigenvectors are desired, then LD Z \(>=\mathrm{max}(1, \mathrm{~N})\).
W ORK (w orkspace)
dim ension (LW ORK)Onexit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim allW ORK.

LW ORK (input)
The dim ension of the aray \(W\) ORK. IfCOMPZ \(=N^{\prime}\) orN <= 1 then LW ORK m ustbe at least1. If COM PZ \(=\mathrm{V}\) 'and \(\mathrm{N}>1\) then LW ORK m ustbe at least ( \(1+\) \(3 * \mathrm{~N}+2{ }^{*} \mathrm{~N} * \lg \mathrm{~N}+3 * \mathrm{~N} * * 2\) ), where \(\lg (\mathrm{N})=\mathrm{sm}\) allest integerk such that \(2 * * k>=\mathrm{N}\). If \(\mathrm{COM} \operatorname{PZ}=\) 'I' and N > 1 then LW ORK m ustbe at least ( \(1+\) \(4 * \mathrm{~N}+\mathrm{N} * * 2\) ).

IfLW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IN ORK (w orkspace/output)
On exit, if \(\mathbb{N F} F=0, \mathbb{I N}\) ORK (1) retums the optim al LIN ORK.

LIV ORK (input)
The dim ension of the array \(\mathbb{I N} O R K\). IfCOMPZ \(=N^{\prime}\) or \(N\) <= 1 then LIW ORK m ustbe at least 1. If

COMPZ \(=V\) 'and \(N>1\) then \(L \mathbb{I V}\) ORK m ustbe at least

then LIN ORK must.be at least ( \(3+5{ }^{*} \mathrm{~N}\) ).

If LIV ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim al size of
the \(\mathbb{I V}\) ORK array, retums this value as the first entry of the \(\mathbb{I V}\) ORK amay, and no errorm essage related to LIN ORK is issued by XERBLA.
\(\mathbb{I N} F O\) (output)
= 0: successfulexit.
\(<0\) : if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue.
>0: The algonithm failed to com pute an eigenvalue while w orking on the subm atrix lying in row s and colum \(n s \mathbb{N} F O / \mathbb{N}+1\) ) through \(m\) od ( \(\mathbb{N} F O, N+1\) ).

\section*{FURTHER DETAILS}

B ased on contributions by
JeffR utter, C om puter Science D ívision, U niversity of C alifomia
at B erkeley, U SA
M odified by Francoise T isseur, U niversity of Tennessee.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
```

dstegr- (a) Com pute T-sigm a_i= L_iD_i L_i^T, such that
L_iD_iL_i^T is a relatively robustrepresentation

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DSTEGR(JOBZ,RANGE,N,D,E,VL,VU, U, IU,ABSTOL,M,W ,}
Z,LDZ,ISUPPZ,W ORK,LW ORK,IN ORK,LIW ORK,\mathbb{NFO)}

```
```

CHARACTER * 1 JOBZ,RANGE
\mathbb{N}TEGERN,\mathbb{L},\mathbb{U},M,LDZ,LW ORK,L\mathbb{N ORK,\mathbb{NFO}}\mathbf{M}\mathrm{ , L}
INTEGER ISUPPZ (*), IN ORK (*)
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION D (*),E (*),W (*),Z (LD Z ,*),W ORK (*)

```
SU BROUTINE DSTEGR_64 (JOBZ,RANGE,N,D,E,VL,VU, IL, \(\mathbb{I}, A B S T O L, M\),
    W, Z,LDZ, ISUPPZ,W ORK,LW ORK, IV ORK,LIN ORK, \(\mathbb{N} F O\) )
CHARACTER * 1 JOBZ,RANGE
\(\mathbb{N} T E G E R * 8 N, \mathbb{I}, \mathbb{U}, M, L D Z, L W O R K, L \mathbb{N} O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \operatorname{ISUPPZ}\) ( \(\left.^{*}\right)\), \(\mathbb{I W}\) ORK ( \({ }^{*}\) )
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION D (*), E (*), W (*), Z (LD Z , \(\left.{ }^{\star}\right), \mathrm{W} O R K(*)\)

\section*{F95 INTERFACE}

SU BROUTINE STEGR (JOBZ,RANGE, \(\mathbb{N}], D, E, V L, V U, \mathbb{I}, \mathbb{U}, A B S T O L, M\), W , Z, [LD Z], ISUPPZ, [W ORK ], [LW ORK ], [IN ORK ], [LINORK], [ \(\mathbb{N} F O])\)

CHARACTER (LEN=1): : JOBZ,RANGE
\(\mathbb{N} T E G E R:: N, \mathbb{L}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W\) ORK,LIN ORK, \(\mathbb{N} F \mathrm{O}\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S \mathbb{O N}(:)::\) ISUPPZ, \(\mathbb{I N}\) ORK
REAL (8) ::VL,VU,ABSTOL

REAL (8),D \(\mathbb{M}\) ENSION (:) ::D ,E,W,W ORK
REAL (8),D IM ENSION (:,:) ::Z
SU BROUTINE STEGR_64 (JOBZ,RANGE, \(\mathbb{N}], D, E, V L, V U, \mathbb{I}, \mathbb{U}, A B S T O L\),


CHARACTER (LEN=1): : JOBZ,RANGE
\(\mathbb{N}\) TEGER (8) ::N, \(\mathbb{I}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W\) ORK,LIN ORK, \(\mathbb{N} F \mathrm{O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) :: ISUPPZ, \(\mathbb{I N}\) ORK
REAL (8) ::VL,VU,ABSTOL
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D ,E,W ,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) :: Z

\section*{C INTERFACE}
\#include <sunperfh>
void dstegr(char j̀jbz, char range, intn, double *d, double *e, double vl, double vu, int il, int iú, double abstol, int *m, double *w , double *z, int ldz, int *isuppz, int *info);
void dstegr_64 (char jobz, char range, long n, double *d, double *e, double vl, double vu, long il, long iu, double abstol, long \({ }^{*} \mathrm{~m}\), double \({ }^{\mathrm{w}}{ }_{\mathrm{w}}\), double \(\mathrm{*}_{\mathrm{z}}\), long ldz, long *isuppz, long *info);

\section*{PURPOSE}
dstegrb) Com pute the eigenvalues, lam bda_j, of L_i D_i L_i^T to high relative accuracy by the dqds algorithm,
(c) If there is a cluster of close eigenvalues, "choose" sigm a_i close to the cluster, and go to step (a),
(d) G iven the approxim ate eigenvalue lam bda_jofL_i D_i L_i^T, com pute the corresponding eigenvectorby form ing a rank-revealing tw isted factorization. The desired accuracy of the output can be specified by the inputparam eterA BSTO L.

Form ore details, see "A new O ( \(\mathrm{n}^{\wedge} 2\) ) algorithm for the sym m etric tridiagonal eigenvalue/eigenvector problem ", by Inder\#̈̈tD hillon, C om puterScience D ìision TechnicalR eport N o.U CB C SD -97-971, U C Berkeley, M ay 1997.

N ote 1 : Curently SSTEGR is only setup to find A LL the \(n\) eigenvalues and eigenvectors of \(T\) in \(0\left(n^{\wedge} 2\right)\) tim e N ote 2 :Cumently the routine SSTE \(\mathbb{N}\) is called when an appropriate sigm a_i cannot be chosen in step (c) above.

SSTE \(\mathbb{I N}\) invokes m odified Gram -Schm idt when eigenvalues are close.
N ote 3 :SSTEGR w orks only on \(m\) achines which follow ieec-754 floating-point standard in their handling of infinities and NaN s. N orm alexecution of SSTEGR \(m\) ay create \(N a N s\) and infinities and hence \(m\) ay abort due to a floating point exception in environm ents w hich do not conform to the ieee standard.

\section*{ARGUMENTS}

JOBZ (input)
= N ': C om pute eigenvalues only;
\(=\mathrm{V}\) : C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvalues w illibe found.
= V : alleigenvalues in the half-open interval
( L L, VU ] will be found. = I': the \(\mathbb{I}\)-th through
IU -th eigenvaluesw illube found.

N (input) The order of the m atrix. \(\mathrm{N}>=0\).
D (input/output)
O n entry, the n diagonalelem ents of the tridiagonalm atrix T.On exit, D is overw ritten.

E (input/output)
O \(n\) entry, the ( \(n-1\) ) subdiagonal elem ents of the tridiagonal m atrix T in elem ents 1 to \(\mathrm{N}-1\) of E ; \(\mathrm{E}(\mathbb{N})\) need notbe set. On exit, E is overw rilten.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRA N GE = A' 'or I'.

VU (input)
Se the description of V L .

II (input)
IfRA N G E= I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{H}<=\mathbb{U}<=N\), ifn \(>0\); \(\mathbb{H}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N ot referenced ifRANGE= A 'or V'.

IU (input)
See the description of II.

ABSTOL (input)
The absolute error tolerance for the eigenvalues/eigenvectors. \(\mathbb{F} J \mathrm{OBZ}=\mathrm{V}\) ', the eigenvalues and eigenvectors outputhave residual norm s bounded by ABSTOL, and the dotproducts betw een different eigenvectors are bounded by ABSTOL. If ABSTOL is less than N *EPS*|T \(\mid\), then N *EPS*|T|w illbe used in its place, where EPS is the \(m\) achine precision and \(F\) is the 1 -norm of the tridiagonalm atrix. The eigenvalues are com puted to an accuracy ofEPS* 1 |imespective of A BSTOL . If high relative accuracy is im portant, setA B STO L to DLAM CH (Safem inim um '). See Barlow and Dem mel "C om puting A ccurate Eigensystem s of Scaled D iagonally D om inantM atrioes", LA PA CK W orking N ote \#7 for a discussion of \(w\) hich \(m\) atrioes define their eigenvalues to high relative accuracy .

M (output)
The total num berofeigenvalues found. \(0<=\mathrm{M}\) <= N . IfRANGE \(=A \prime, \mathrm{M}=\mathrm{N}\), and ifRANGE \(=\mathrm{I}^{\prime}, \mathrm{M}=\) \(\mathbb{I U}-\mathbb{H}+1\).

W (output)
The firstM elem ents contain the selected eigenvalues in ascending order.
\(Z\) (input) If \(J O B Z=V^{\prime}\), then if \(\mathbb{N F O}=0\), the first \(M\) colum ns of \(Z\) contain the orthonorm aleigenvectors of them atrix T comesponding to the selected eigenvalues, \(w\) ith the i-th colum \(n\) of \(Z\) holding the eigenvector associated w ith W (i). If \(\mathrm{JO} \mathrm{BZ}=\mathrm{N}\) ', then \(Z\) is not referenced. N ote: the userm ust ensure that at leastm ax ( \(1, M\) ) colum ns are supplied in the array \(Z\); ifRANGE = V', the exact value of M is not know n in advance and an upperbound m ust be used.

LD \(Z\) (input)
The leading dim ension of the array \(Z . L D Z>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD Z >= \(\mathrm{max}(1, N)\).

ISU PPZ (output)
The support of the eigenvectors in \(Z\), i.e., the indices indicating the nonzero elem ents in \(Z\). The i-th eigenvector is nonzero only in elem ents ISU PPZ (2*i-1 ) through ISU PPZ (2*i).

W ORK (w orkspace)

On exit, if \(\mathbb{N F O}=0, W\) ORK ( 1 ) retums the optim al (and minim al) LW ORK .

\section*{LW ORK (input)}

The dimension of the aray W ORK. LW ORK >= \(\max (1,18 * N)\)
IfLW O RK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK aray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IW ORK (w orkspace/output)
On exit, if \(\mathbb{N} F O=0, \mathbb{I W}\) ORK (1) retums the optim al LIN ORK.

LIV ORK (input)
The dim ension of the aray \(\mathbb{I W}\) ORK. L \(\mathbb{I W}\) ORK >= \(\max (1,10 \star N)\)

IfLIW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the \(\mathbb{I W}\) ORK array, retums this value as the first entry of the \(\mathbb{I W}\) ORK array, and no errorm essage related to \(L \mathbb{I W}\) ORK is issued by X ERBLA.

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N F O}=1\), intemalemor in SLARRE, if \(\mathbb{N} F O=2\), intemalemor in SLARRV.

\section*{FURTHER DETAILS}

B ased on contributions by
Inder屰D hillon, \(\mathbb{B M}\) A \(\operatorname{lm}\) aden, U SA
O sniM arques, LBNL N ER SC , U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dstein - com pute the eigenvectors of a real sym \(m\) etric tridiagonal \(m\) atrix \(T\) comesponding to specified eigenvalues, using inverse iteration

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSTE\mathbb{N}\mathbb{N},D,E,M,W,\mathbb{BLOCK,ISPLIT,Z,LD Z,W ORK,IN ORK,}}\mathbf{~},\textrm{L},\textrm{L}
\mathbb{FA}|,\mathbb{NNOO}
\mathbb{N TEGER N,M,LD Z, IN FO}
\mathbb{NTEGER \mathbb{BLOCK (*),ISPLIT (*), IN ORK (*),\mathbb{FA IL (*)}}\mathbf{(*)}}\mathbf{(*)}
DOUBLE PRECISION D (*),E (*),W (*),Z (LD Z ,*),W ORK (*)
SU BROUTINE DSTE IN_64 N,D ,E,M,W,\mathbb{BLOCK,ISPLIT,Z,LD Z,W ORK,}
IN ORK,\mathbb{FA}\mathbb{I},\mathbb{N}FO)
\mathbb{NTEGER*8N,M,LD Z,INFO}
\mathbb{NTEGER*8 \mathbb{BLOCK}(*),ISPLIT (*), IN ORK (*),\mathbb{FA IL (*)}}\mathbf{(*)}
DOUBLE PRECISION D (*),E (*),W (*),Z (LD Z ,*),WORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE STE \(\mathbb{N} \mathbb{N}, D, E, M, W, \mathbb{B L O C K}, \operatorname{ISPLIT}, \mathrm{Z},[L D Z],[W O R K]\), [ \(\mathbb{I V}\) ORK ], \(\mathbb{F A} \mathbb{I},[\mathbb{N} F O])\)
\(\mathbb{N} T E G E R:: N, M, L D Z, \mathbb{N F O}\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{B L O C K}, \operatorname{ISPLIT}, \mathbb{I}\) ORK, \(\mathbb{F} A \mathbb{I}\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D , E, W , W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::Z

SU BROUTINE STE \(\mathbb{N} \_64 \mathbb{N}, D, E, M, W, \mathbb{B L O C K}, \operatorname{ISPL} \mathbb{I}, \mathrm{Z},[\operatorname{LD} Z],[\mathbb{W}\) ORK ], [ \(\mathbb{I N}\) ORK], \(\mathbb{F A} \mathbb{I},[\mathbb{N} F O]\) )
\(\mathbb{N} T E G E R(8):: N, M, L D Z, \mathbb{N F O}\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{B L O C K}, \operatorname{ISPLIT}, \mathbb{I N} O R K, \mathbb{F A} \mathbb{L}\) REAL (8),D \(\mathbb{M}\) ENSION (:) ::D ,E,W ,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::Z

\section*{C INTERFACE}
\#include <sunperfh>
void dstein (intn, double *d, double *e, intm, double \({ }^{*}\), int *iblock, int *isplit, double * \(z\), int ldz, int
*ifail, int*info);
void dstein_64 (long n, double *d, double *e, long m, double
* \(_{\text {w }}\), long *iblock, long *isplit, double *z, long
ldz, long *ifail, long *info);

\section*{PURPOSE}
dstein com putes the eigenvectors of a realsym \(m\) etric tridiagonal m atrix T corresponding to specified eigenvalues, using inverse iteration.

The m axim um num ber of terations allow ed foreach eigenvector is specified by an intemal param eterM A X ITS (currently set to 5).

\section*{ARGUMENTS}

N (input) The order of the \(m\) atrix. \(\mathrm{N}>=0\).

D (input) The \(n\) diagonalelem ents of the tridiagonal \(m\) atrix
T.

E (input) The ( \(\mathrm{n}-1\) ) subdiagonalelem ents of the tridiagonal \(m\) atrix \(T\), in elem ents 1 to \(N-1\). \(E(\mathbb{N})\) need notbe set.

M (input) The num ber of eigenvectors to be found. \(0<=\mathrm{M}<=\) N .

W (input) The firstM elem ents ofW contain the eigenvalues for which eigenvectors are to be com puted. The eigenvalues should be grouped by split-off block and ordered from smallest to largestw ithin the block. (The output anay \(W\) from SSTEBZ w ith ORDER = B'is expected here.)

BLOCK (input)
The subm atrix indiges associated with the corresponding eigenvalues in \(W\); \(\mathbb{B L O C K}(i)=1\) if eigenvalue \(W\) (i) belongs to the first subm atrix from the top, \(=2\) ifW (i) belongs to the second subm atrix, etc. (The output array \(\mathbb{B L O C K}\) from SSTEBZ is expected here.)

\section*{ISPLIT (input)}

The splitting points, atw hich \(T\) breaks up into subm atrices. The first subm atrix consists of row s/columns 1 to ISPLIT ( 1 ), the second of row s/Colum ns ISPLIT ( 1 )+1 through ISPLIT (2), etc. (The outputaray ISPLIT from SSTEBZ is expected here.)
Z (output)
The com puted eigenvectors. The eigenvector associated w ith the eigenvalue \(W\) (i) is stored in the \(i-t h\) colum \(n\) of \(Z\). A ny vectorw hich fails to converge is set to its current iterate afterM AX ITS iterations.

LD \(Z\) (input)
The leading dim ension of the aray \(Z\). LD \(Z \quad>=\) \(\max (1, N)\).

W ORK (w orkspace)
dim ension ( \(5 * \mathrm{~N}\) )
IV ORK (w orkspace)
dim ension (N)
FFA II (output)
On norm alexit, allelem ents of \(\mathbb{F} A \mathbb{I}\) are zero. If one orm ore eigenvectors fail to converge after
M AXITS iterations, then their indices are stored in array \(\mathbb{F A} \mathbb{I}\).
\(\mathbb{I N F O}\) (output)
\(=0\) : successfulexit.
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N} F O=i\), then \(i\) eigenvectors failed to converge in M AX ITS iterations. Their indiges are stored in array \(\mathbb{F}\) A II.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsteqr - com pute alleigenvalues and, optionally, eigenvectors of a sym \(m\) etric tridiagonalm atrix using the im plicit \(Q L\) orQ R m ethod

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSTEQR(COMPZ,N,D,E,Z,LDZ,W ORK,NNFO)}
CHARACTER * 1 COMPZ
\mathbb{NTEGER N,LD Z,INFO}
DOUBLE PRECISION D (*),E (*),Z (LD Z,*),W ORK (*)
SUBROUT\mathbb{NEDSTEQR_64(COMPZ,N,D,E,Z,LD Z,W ORK,NNFO)}
CHARACTER * 1 COMPZ
\mathbb{NTEGER*8N,LD Z,NNFO}
DOUBLE PRECISION D (*),E (*),Z (LD Z ,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE STEQR COMPZ,N,D,E,Z, [LD Z], [W ORK], [NFO])
CHARACTER (LEN=1) ::COM PZ
\(\mathbb{N}\) TEGER :: N,LD Z, \(\mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK
REAL (8),D IM ENSION (:,:) ::Z
SU BROUTINE STEQR_64 (COM PZ,N,D,E,Z,[LD Z], [W ORK ], [ \(\mathbb{N} F O\) ])
CHARACTER (LEN=1) ::COMPZ
\(\mathbb{N}\) TEGER (8) :: N, LD Z, \(\mathbb{N}\) FO
REAL (8),D \(\mathbb{M}\) ENSION (:) :: D ,E,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::Z

\section*{C INTERFACE}
\#include <sunperfh>
void dsteqr(char com pz, intn, double *d, double *e, double
* \(z\), int ldz, int *info);
void dsteqr_64 (charcom pz, long n, double *d, double *e, double *z, long ldz, long *info);

\section*{PURPOSE}
dsteqr com putes alleigenvalues and, optionally, eigenvectors of a sym \(m\) etric tridiagonalm atrix using the im pliciti \(Q L\) orQ R m ethod. The eigenvectors of a fullorband sym m etric \(m\) atrix can also be found ifSSY TRD orSSPTRD orSSBTRD has been used to reduce thism atrix to tridiagonal form .

\section*{ARGUMENTS}

COMPZ (input)
\(=\mathrm{N}^{\prime}:\) C om pute eigenvalues only .
\(=\mathrm{V}\) : Com pute eigenvalues and eigenvectors of the original sym \(m\) etric \(m\) atrix. On entry, \(Z \mathrm{~m}\) ust contain the orthogonalm atrix used to reduce the originalm atrix to tridiagonal form . = I': C om pute eigenvalues and eigenvectors of the tridiagonal m atrix. Z is initialized to the identity \(m\) atrix.

N (input) The order of the m atrix. \(\mathrm{N}>=0\).
D (input/output)
On entry, the diagonalelem ents of the tridiagonal m atrix. On exit, if \(\mathbb{N} F O=0\), the eigenvalues in ascending order.

E (input/output)
O \(n\) entry, the ( \(n-1\) ) subdiagonal elem ents of the tridiagonal matrix. On exit, E hasbeen destroyed.

Z (input) On entry, if \(\mathrm{COMPZ}=\mathrm{V}\) ', then Z contains the orthogonal m atrix used in the reduction to tridiagonal form. On exit, if \(\mathbb{N F O}=0\), then if COM PZ \(=\mathrm{V}^{\prime}, \mathrm{Z}\) contains the orthonorm aleigenvectors of the original sym \(m\) etric \(m\) atrix, and if \(C O M P Z=\) 'I',
\(Z\) contains the orthonorm al eigenvectors of the sym \(m\) etric tridiagonalm atrix. If COM PZ \(=N^{\prime}\) ', then \(Z\) is not referenced.

LD \(Z\) (input)
The leading dim ension of the array Z . LD \(\mathrm{Z}>=1\), and if eigenvectors are desired, then LD Z >= \(\max (1, N)\).

W ORK (w orkspace)
dim ension (max ( \(1,2 \star \mathrm{~N}-2\) )) If \(\mathrm{COMPZ}=\mathrm{N}\) ', then W ORK is not referenced.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
>0: the algorithm has failed to find all the eigenvalues in a total of \(30 *\) N terations; if \(\mathbb{N}\) FO
\(=i\), then ielem ents of \(E\) have not converged to zero; on exit, \(D\) and \(E\) contain the elem ents of a
sym \(m\) etric tridiagonalm atrix which is orthogonally
sim ilar to the originalm atrix.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsterf -com pute alleigenvalues of a sym \(m\) etric tridiagonal \(m\) atrix using the Pal-w alkerK ahan variant of the \(Q L\) or \(Q R\) algorithm

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSTERF(N,D,E,INFO)}
\mathbb{NTEGER N, INFO}
DOUBLE PRECISION D (*),E (*)
SUBROUT\mathbb{NEDSTERF_64 N,D,E,INFO)}

```

```

DOUBLE PRECISION D (*),E (*)
F95 INTERFACE

```

```

\mathbb{NTEGER ::N,\mathbb{NFO}}\mathbf{N}=0
REAL (8),D IM ENSION (:) ::D ,E
SUBROUT\mathbb{NE STERF_64 (N ],D ,E,[NNFO])}
\mathbb{NTEGER (8)::N,\mathbb{NFO}}\mathbf{~}\mathrm{ ( }
REAL (8),D IM ENSION (:) ::D ,E

```

\section*{C INTERFACE}
```

\#include <sunperfh>
void dsterf(intn, double *d, double *e, int*info);

```

\section*{PURPOSE}
dsterf com putes alleigenvalues of a sym m etric tridiagonal \(m\) atrix using the Pal-W alkerK ahan variantof the \(Q L\) orQR algorithm .

\section*{ARGUMENTS}

N (input) The order of the m atrix. \(\mathrm{N}>=0\).
D (input/output)
O n entry, the \(n\) diagonalelem ents of the tridiagonalm atrix. On exit, if \(\mathbb{N F O}=0\), the eigenvalues in ascending order.

E (input/output)
O \(n\) entry, the ( \(n-1\) ) subdiagonal elem ents of the tridiagonal matrix. On exit, E hasbeen destroyed.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum enthad an illegalvalue
> 0: the algorithm failed to find all of the eigenvalues in a total of 30*N iterations; if \(\mathbb{N}\) FO
\(=i\), then ielem ents of \(E\) have not converged to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dstev -com pute alleigenvalues and, optionally, eigenvectors of a realsym m etric tridiagonalm atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DSTEV (OBZ,N,D IA G,OFFD,Z,LD Z,W ORK,INFO )}
CHARACTER * 1 JOBZ
\mathbb{NTEGER N,LD Z,INFO}
DOUBLE PRECISION D IAG (*),OFFD (*),Z (LD Z,*),W ORK (*)

```

```

CHARACTER * 1 JOBZ
INTEGER*8N,LDZ,INFO
DOUBLE PRECISION D IAG (*),OFFD (*),Z (LD Z,*),W ORK (*)

```

\section*{F95 INTERFACE}

CHARACTER (LEN=1) ::JOBZ
\(\mathbb{N} T E G E R:: N, L D Z, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::D IA G , OFFD ,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::Z
SUBROUTINESTEV_64 (OOBZ,N,D IAG,OFFD, Z, [LD Z ], [W ORK ], [NFO ])
CHARACTER (LEN=1)::JOBZ
\(\mathbb{N} T E G E R(8):: N, L D Z, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::D IA G,OFFD ,W ORK
REAL (8),D IM ENSION (:,:) ::Z

\section*{C INTERFACE}
\#include <sunperfh>
void dstev (char j.jbz, intn, double *diag, double *offd, double * \(z\), int ldz, int *info);
void dstev_64 (char jobz, long n, double *diag, double *offd, double *z, long ldz, long *info);

\section*{PURPOSE}
dstev com putes alleigenvalues and, optionally , eigenvectors of a realsym m etric tridiagonalm atrix A.

\section*{ARGUMENTS}

JO B Z (input)
\(=\mathrm{N}^{\prime}:\) C om pute eigenvalues only;
\(=\mathrm{V}\) ': C om pute eigenvalues and eigenvectors.

N (input) The order of them atrix. \(\mathrm{N}>=0\).

D IA G (input/output)
O n entry, the \(n\) diagonalelem ents of the tridiago-
nal matrix A. On exit, if \(\mathbb{N} F O=0\), the eigenvalues in ascending order.

OFFD (input/output)
O n entry, the \((n-1)\) subdiagonal elem ents of the tridiagonal \(m\) atrix \(A\), stored in elem ents 1 to \(N-1\) of OFFD ; OFFD \(\mathbb{N}\) ) need notbe set, but is used by the routine. On exit, the contents of OFFD are destroyed.
\(Z\) (input) If \(\mathrm{OOB}=\mathrm{V}^{\prime}\), then if \(\mathbb{N} F O=0, \mathrm{Z}\) contains the orthonorm aleigenvectors of the m atrix \(A, w\) th the i-th colum \(n\) of \(Z\) holding the eigenvector associated w ith D IA G (i). If \(\mathrm{OOBZ}=\mathrm{N}\) ', then Z is not referenced.

LD Z (input)
The leading \(d\) im ension of the array Z. LD Z \(>=1\), and if \(\mathrm{OBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{max}(1, N)\).

W ORK (w orkspace)
If \(\mathrm{OBBZ}=\mathrm{N}^{\prime}\) ', W ORK is notreferenced.
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=i\), the algorithm failed to converge; i off-diagonal elem ents of OFFD did not converge to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dstevd -com pute alleigenvalues and, optionally, eigenvectors of a real sym \(m\) etric tridiagonalm atrix

\section*{SYNOPSIS}

```

    \mathbb{NFO)}
    CHARACTER * 1 JOBZ
\mathbb{NTEGER N,LD Z,LW ORK,LIN ORK,INFO}
INTEGER IN ORK (*)
DOUBLE PRECISION D (*),E (*),Z (LD Z,*),W ORK (*)
SU BROUT\mathbb{NE DSTEVD_64(JO BZ,N,D ,E,Z,LD Z,W ORK,LW ORK,IN ORK,}
LIN ORK,INFO)
CHARACTER * 1 JOBZ
\mathbb{NTEGER*8N,LD Z,LW ORK,LIW ORK,INFO}
INTEGER*8 IN ORK (*)
DOUBLE PRECISION D (*),E (*),Z (LD Z ,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE STEVD (JOBZ,N,D,E,Z, [LD Z], [W ORK ], [LW ORK ], [IW ORK ], \(\left[\begin{array}{l}\mathbb{N} \\ \text { ORK ] } \\ [\mathbb{N} F O])\end{array}\right.\)

CHARACTER (LEN=1)::JOBZ
\(\mathbb{N} T E G E R:: N, L D Z, L W\) ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
REAL (8),D \(\mathbb{I}\) ENSION (:) ::D ,E,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) :: Z
SU BROUTINE STEVD_64 (JOBZ,N,D,E,Z, [LDZ], [W ORK ], [LW ORK], [IN ORK ],
\([\mathrm{L} \mathbb{N} O R K],[\mathbb{N} F O])\)

CHARACTER (LEN=1) :: JOBZ
\(\mathbb{N}\) TEGER (8) :: \(N\), LD Z, LW ORK, LIW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{I W}\) ORK
REAL (8), D \(\mathbb{M}\) ENSION (:) :: D , E, W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::Z

\section*{C INTERFACE}
\#include <sunperfh>
void dstevd (char j̀jbz, intn, double *d, double *e, double * \(z\), int ldz, int *info);
void dstevd_64 (char jobz, long n, double *d, double *e, double *z, long ldz, long *info);

\section*{PURPOSE}
dstevd com putes alleigenvahues and, optionally, eigenvectors of a realsym \(m\) etric tridiagonalm atrix. If eigenvectors are desired, it uses a divide and conquer algorithm .

The divide and conqueralgorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on \(m\) achines w th a guard digit in add/subtract, or on those binary \(m\) achines \(w\) thout guard digits w hich subtract like the \(C\) ray \(X-M P, C\) ray \(Y \neq M P, C\) ray \(C-90\), orC ray-2. It could conceivably fail on hexadecim al or decim al \(m\) achines \(w\) ithout guard digits, butw e know of none.

\section*{ARGUMENTS}

JO BZ (input)
\(=\mathrm{N}\) ': C om pute eigenvalues only;
\(=\mathrm{V}\) : C om pute eigenvahues and eigenvectors.

N (input) The order of the m atrix. \(\mathrm{N}>=0\).
D (input/output)
O n entry, the n diagonalelem ents of the tridiagonal \(m\) atrix \(A\). On exit, if \(\mathbb{N F O}=0\), the eigenvalues in ascending order.

E (input/output)
O \(n\) entry, the ( \(n-1\) ) subdiagonal elem ents of the tridiagonal \(m\) atrix A, stored in elem ents 1 to \(N-1\) of \(E\); \(E \mathbb{N}\) ) need notbe set, but is used by the
routine. On exit, the contents ofe are destroyed.
\(Z\) (input) If \(J O B Z=V '\), then if \(\mathbb{N} F O=0, Z\) contains the orthonorm aleigenvectors of the \(m\) atrix \(A, w\) th the \(i\)-th colum \(n\) of \(Z\) holding the eigenvector associated w th D (i). If JOBZ \(=N^{\prime}\) ', then \(Z\) is not referenced.

LD \(Z\) (input)
The leading din ension of the amay Z . LD \(\mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD Z >= \(\max (1, N)\).
W ORK (w orkspace)
dim ension ( \(\mathbb{L} W\) ORK)On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim alLW O RK .

LW ORK (input)
The dim ension of the array \(W\) ORK. If \(J O B Z=N^{\prime}\) or \(\mathrm{N}<=1\) then LW ORK m ustbe at least1. If JO B Z \(=\mathrm{V}\) 'and \(\mathrm{N}>1\) then LW ORK m ust.be at least ( \(1+\) \(4 * N+N * * 2)\).

If LW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of theW ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IN ORK (w orkspace/output)
On exit, if \(\mathbb{N} F O=0, \mathbb{I V} O R K(1)\) retums the optim al LIN ORK.

LIV ORK (input)
The dim ension of the array \(\mathbb{I N} O R K\). If \(J O B Z=N^{\prime}\) orN <= 1 then LIW ORK m ustbe at least1. If JO B Z \(=V\) 'and \(N>1\) then \(L \mathbb{I}\) ORK must be at least \(3+5 * N\).

If LIV ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the IV ORK array, retums this value as the first entry of the \(\mathbb{I N}\) ORK amray, and no errorm essage related to \(L \mathbb{I N} O R K\) is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the i-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=\mathrm{i}\), the algorithm failed to con-
verge; i off-diagonalelem ents ofe did not con-
verge to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dstevr - com pute selected eigenvalues and, optionally, eigenvectors of a realsym \(m\) etric tridiagonalm atrix \(T\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DSTEVR(JOBZ,RANGE,N,D,E,VL,VU, IL,IU,ABSTOL,M,W,}
Z,LDZ,ISUPPZ,W ORK,LW ORK,IN ORK,LIW ORK,\mathbb{NFO)}

```
```

CHARACTER * 1 JOBZ,RANGE
\mathbb{N}TEGERN,\mathbb{L},\mathbb{U},M,LDZ,LW ORK,L\mathbb{N ORK,\mathbb{NFO}}\mathbf{M}\mathrm{ , L}
INTEGER ISUPPZ (*), IN ORK (*)
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION D (*),E (*),W (*),Z (LD Z,*),W ORK (*)

```
SU BROUTINE D STEVR_64 (JOBZ,RANGE,N,D,E,VL,VU, \(\mathbb{I}, \mathbb{U}, A B S T O L, M\),
    W, Z,LD Z, ISUPPZ,W ORK,LW ORK, IV ORK,LIW ORK, \(\mathbb{N} F O\) )

\section*{CHARACTER * 1 JOBZ,RANGE}
\(\mathbb{N} T E G E R * 8 N, \mathbb{L}, \mathbb{U}, M, L D Z, L W O R K, L \mathbb{N} O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \operatorname{ISUPPZ}\) (*), \(\mathbb{I N}\) ORK (*)
DOUBLE PRECISION VL,VU, ABSTOL
DOUBLE PRECISION D (*), E (*), W (*), Z (LD Z , \(\left.{ }^{\star}\right), \mathrm{W} O R K(*)\)

\section*{F95 INTERFACE}

SU BROUTINE STEVR ( \(\operatorname{OOBZ}, R A N G E, \mathbb{N}], D, E, V L, V U, \mathbb{I}, \mathbb{U}, A B S T O L, M\), W , Z, [LD Z], ISUPPZ, [W ORK ], [LW ORK ], [IN ORK ], [LINORK], [ \(\mathbb{N} F O])\)

CHARACTER (LEN=1)::JOBZ,RANGE
\(\mathbb{N} T E G E R:: N, \mathbb{L}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W\) ORK,LIN ORK, \(\mathbb{N} F \mathrm{O}\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S \mathbb{O N}(:)::\) ISUPPZ, \(\mathbb{I V}\) ORK
REAL (8) ::VL,VU,ABSTOL

REAL (8), D \(\mathbb{M}\) ENSION (:) ::D ,E,W ,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::Z
SU BROUTINE STEVR_64 (JOBZ,RANGE, \(\mathbb{N}], D, E, V L, V U, \mathbb{I}, \mathbb{U}, A B S T O L\),


CHARACTER (LEN=1): : JOBZ,RANGE
\(\mathbb{N}\) TEGER (8) ::N, \(\mathbb{L}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W\) ORK, LIN ORK, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) :: ISUPPZ, \(\mathbb{I N}\) ORK
REAL (8) ::VL,VU,ABSTOL
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D ,E,W ,W ORK
REAL (8),D IM ENSION (:,:) ::Z

\section*{C INTERFACE}
\#include <sunperfh>
void dstevr(char jojbz, char range, intn, double *d, double
*e, double vl , double vu, int il, int iul, double abstol, int *m, double * w , double \({ }^{\text {* }}\), int ldz, int *isuppz, int*info);
void dstevr_64 (char jobz, char range, long n, double *d, double *e, double vl, double vu, long il, long in, double abstol, long *m , double *w , double *z, long ldz, long *isuppz, long *info);

\section*{PURPOSE}
dstevr com putes selected eigenvalues and, optionally, eigenvectors of a realsym \(m\) etric tridiagonalm atrix T. Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

W heneverpossible, SSTEV R calls SSTEGR to com pute the eigenspectrum using Relatively Robust Representations. SSTEGR com putes eigenvalues by the dqds algorithm, while orthogonaleigenvectors are com puted from various "good" L D L^T representations (also known as Relatively Robust Representations). G ram -Schm idtorthogonalization is avoided as far as possible. M ore specifically, the various steps of the algorithm are as follow s.For the i-th unreduced block ofT,
(a) Com pute \(T\)-sigm a_i= L_iD_iL_i^T, such that L_i D_iL_i^T
is a relatively robust representation,
(b) C om pute the eigenvalues, lam bda_jof of_i D _i L_i^T to high
relative accuracy by the dqds algorithm,
(c) If there is a cluster of close eigenvahues, "choose"
sigm a_i
close to the cluster, and go to step (a),
(d) G iven the approxim ate eigenvalue lam bda_jofL_i D_i L_i^T,
com pute the comesponding eigenvectorby form ing a rank-revealing tw isted factorization. The desired accuracy of the output can be specified by the inputparam eterABSTOL.

Form ore details, see "A new \(\mathrm{O}\left(\mathrm{n}^{\wedge} 2\right)\) algorithm for the sym m etric tridiagonal eigenvalue/eigenvector problem ", by Inder \#̈tD hillon, C om puter Science D ìvision TechnicalReport No.UCB/C SD -97-971, UC Berkeley, M ay 1997.

N ote 1 :SSTEVR calls SSTEGR when the full spectrum is requested on \(m\) achines \(w\) hich conform to the iee-754 floating pointstandard. SSTEVR callsSSTEBZ and SSTE IN on non-ieee m achines and when partial spectrum requests are \(m\) ade.

N orm alexecution of SSTEGR m ay create NaNs and infinities and hence \(m\) ay abort due to a floating point exception in environm ents which do nothandle N aN s and infinities in the ieee standard defaultm anner.

\section*{ARGUMENTS}

JOBZ (input)
\(=\mathrm{N}\) ': C om pute eigenvalues only;
= V : C om pute eigenvalues and eigenvectors.
RANGE (input)
= A : alleigenvalues \(w\) illbe found.
= V : alleigenvahues in the half-open interval
( \(\mathrm{L}, \mathrm{V} \mathrm{U}]\) will be found. = 'I': the \(\mathbb{I}\)-th through
\(\mathbb{I U}\)-th eigenvaluesw illlbe found.

N (input) The order of the m atrix. \(\mathrm{N}>=0\).

D (input/output)
O n entry, the n diagonalelem ents of the tridiagonal \(m\) atrix A. On exit, D m ay be multiplied by a constant factor chosen to avoid over/underflow in com puting the eigenvalues.

E (input/output)
O \(n\) entry, the ( \(n-1\) ) subdiagonal elem ents of the tridiagonal m atrix A in elem ents 1 to \(\mathrm{N}-1\) ofe;

E(N) need notbe set. On exit, E may be multiplied by a constant factor chosen to avoid over/underflow in com puting the eigenvahues.

\section*{VL (input)}

IfRANGE=V ', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE=A'or I'.

VU (input)
See the description of V L .

II (input)
IfRA NGE= 'I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{Z}<=\mathbb{U}<=\mathrm{N}\), if \(\mathrm{N}>0\); \(\mathbb{H}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N ot referenced ifRANGE \(=\) A'or V'.
\(\mathbb{U}\) (input)
See the description of II.

ABSTOL (input)
The absolute error tolerance for the eigenvalues.
A \(n\) approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b] of w idth less than orequal to

ABSTOL + EPS * \(\max (|k|, \mid)\) ),
\(w\) here EPS is the m achine precision. IfA BSTOL is less than or equalto zero, then EPS* \(\mid\) | w illbe used in its place, where \(\mid T\) is the 1 -norm of the tridiagonal \(m\) atrix obtained by reducing A to tridiagonal form .

See "C om puting Sm allSingularV ahues of B idiagonal M atrices w ith G uaranteed H igh Relative A ccuracy," by Dem m eland K ahan, LA PA CK W orking N ote \#3.

If high relative accuracy is im portant, setA BSTO L to SLAM CH (Safe minim um '). D oing so will guarantee thateigenvalues are com puted to high relative accuracy when possible in future releases. The current code does not make any guarantees about high relative accuracy, but future releases w ill. See J. B arlow and J. D em m el, "C om puting A ccurate E igensystem s of Scaled D iagonally D om inantM atrioes", LA PA CK W orking N ote \#7, for a discussion of \(w\) hich \(m\) atrices define their
eigenvalues to high relative accuracy .

M (output)
The total num berofeigenvalues found. \(0<=\mathrm{M}\) <=
N . IfRANGE \(=A^{\prime}, \mathrm{M}=\mathrm{N}\), and ifRANGE \(=\mathrm{I}^{\prime}, \mathrm{M}=\) \(\mathbb{U}-\mathbb{U}+1\).

W (output)
The firstM elem ents contain the selected eigenvalues in ascending order.
\(Z\) (input) If \(\mathcal{J O B Z}=\mathrm{V}^{\prime}\), then if \(\mathbb{N F O}=0\), the first \(M\) colum ns of \(Z\) contain the orthonorm aleigenvectors of them atrix A comesponding to the selected eigenvalues, w ith the i -th colum n of \(Z\) holding the eigenvector associated \(w\) ith \(W\) (i). \(N\) ote: the user \(m\) ust ensure that at leastm ax ( \(1, \mathrm{M}\) ) colum ns are supplied in the array \(Z\); ifRANGE = \(V\) ', the exact value of \(M\) is not know \(n\) in advance and an upper bound \(m\) ustbe used.

LD Z (input)
The leading dim ension of the array \(\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD \(Z>=\mathrm{max}(1, \mathrm{~N})\).

\section*{ISU PPZ (output)}

The support of the eigenvectors in \(Z\), i.e., the indices indicating the nonzero elem ents in \(Z\). The \(i\)-th eigenvector is nonzero only in elem ents ISU PPZ (2*i-1 ) through ISU PPZ (2*i).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al (andminim al) LW ORK .

LW ORK (input)
The dim ension of the array \(W\) ORK. LW ORK \(>=20 * N\).
If LW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IV ORK (w orkspace/output)
On exit, if \(\mathbb{N} F O=0, \mathbb{I N}\) ORK (1) retums the optim al (andminim al) LINORK.

LIN ORK (input)
The dim ension of the anay \(\mathbb{I N}\) ORK. LIN ORK \(>=10 * N\).

If LIW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the \(\mathbb{I V}\) ORK anray, retums this value as the first entry of the \(\mathbb{I W}\) ORK array, and no errorm essage related to \(L \mathbb{I W}\) ORK is issued by XERBLA.
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue
> 0: Intemalemor

\section*{FURTHER DETAILS}

B ased on contributions by
Inder]̈̈t D hillon, \(\mathbb{B M}\) A \(\operatorname{lm}\) aden, U SA
O sniM arques, LBNL N ER SC , U SA
K en Stanley, C om puterScience D ivision, U niversity of C alifomia at B erkeley, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dstevx - com pute selected eigenvalues and, optionally,
eigenvectors of a realsym m etric tridiagonalm atrix A

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSTEVX (OBZ,RANGE,N,D IAG,OFFD,VL,VU, IL, IU,ABTOL,}
NFOUND,W,Z,LDZ,WORK,INORK2,\mathbb{FA}\mathbb{L},\mathbb{N}FO)
CHARACTER * 1 JOBZ,RANGE
\mathbb{NTEGERN,}\mathbb{N},\mathbb{U},NFOUND,LDZ,\mathbb{NFO}
\mathbb{NTEGER IN ORK2 (*),\mathbb{FA}\mathbb{L}(*)}
DOUBLE PRECISION VL,VU,ABTOL
DOUBLE PRECISION DIAG (*),OFFD (*),W (*),Z (LD Z,*),W ORK (*)
SUBROUTINEDSTEVX_64(JOBZ,RANGE,N,DIAG,OFFD,VL,VU,元,IU,
ABTOL,NFOUND,W,Z,LDZ,W ORK,IN ORK 2,\mathbb{FA}|,\mathbb{NFO)}
CHARACTER * 1 JOBZ,RANGE
\mathbb{NTEGER*8N,\mathbb{N},\mathbb{U},NFOUND,LD Z,INFO}
\mathbb{NTEGER*8 IN ORK2 (*), \mathbb{FA LH (*)}}\mathbf{(*)}
DOUBLE PRECISION VL,VU,ABTOL
DOUBLE PRECISION DIAG (*),OFFD (*),W (*),Z (LD Z,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE STEVX (JOBZ,RANGE,N,DIAG,OFFD,VL,VU, \(\mathbb{I}, \mathbb{U}, A B T O L\), NFOUND, W, Z, [LD Z], [W ORK], [IW ORK 2], \(\mathbb{F A} \mathbb{I},[\mathbb{N} F O\) ])

CHARACTER (LEN=1)::JOBZ,RANGE
\(\mathbb{N} T E G E R:: N, \mathbb{L}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{I W}\) ORK2, \(\mathbb{F A} \mathbb{I}\)
REAL (8) ::VL,VU,ABTOL
REAL (8),D \(\mathbb{I}\) ENSION (:) ::D IA G,OFFD ,W ,W ORK

SUBROUTINE STEVX_64 (DOBZ,RANGE,N,DIAG,OFFD,VL,VU, \(\mathbb{I}, \mathbb{I U}\), ABTOL,NFOUND,W,Z,[LDZ], \(\mathbb{W} O R K],[\mathbb{W} O R K 2], \mathbb{F A} \mathbb{I},[\mathbb{N F O}])\)

CHARACTER (LEN=1): : OBZ,RANGE
\(\mathbb{N}\) TEGER (8) :: N , \(\mathbb{I}, \mathbb{I U}, \mathrm{NFOUND}, \mathrm{LD} Z, \mathbb{N}\) FO
\(\mathbb{N}\) TEGER (8), D \(\mathbb{M} E N S I O N(:):: \mathbb{I N} O R K 2, \mathbb{F} A \mathbb{I}\)
REAL (8) ::VL, VU, ABTOL
REAL (8), D \(\mathbb{M}\) ENSION (:) :: D \(\mathbb{I A G}, O F F D, W, W\) ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::Z

\section*{C INTERFACE}
\#include < sunperfh>
void dstevx (char jobz, char range, intn, double *diag, double *offd, double vl, double vu, int il, intin, double abtol, int *nfound, double * w , double \({ }^{*}\) z, int \(l d z\), int *ifail, int *info);
void dstevx_64 (char jobz, char range, long n, double *diag, double *offd, double vl, double vu, long il, long ìu, double abtol, long *nfound, double *w, double *z, long ldz, long *ifail, long *info);

\section*{PURPOSE}
dstevx com putes selected eigenvalues and, optionally, eigenvectors of a realsym \(m\) etric tridiagonalm atrix A. Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

\section*{ARGUMENTS}

JO BZ (input)
\(=\mathrm{N}\) : C om pute eigenvalues only;
\(=\mathrm{V}\) ': C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvalues w illbe found.
\(=\mathrm{V}\) ': alleigenvalues in the half-open interval
(VL, VU] w ill be found. = I': the IL th through IU th eigenvalues w ill.be found.

N (input) The order of the m atrix. \(\mathrm{N}>=0\).

D IA G (input/output)

O n entry, the n diagonalelem ents of the tridiagonal \(m\) atrix A. On exit, D IA G m ay be multiplied by a constant factor chosen to avoid over/undenflow in com puting the eigenvalues.

OFFD (input/output)
O \(n\) entry, the ( \(n-1\) ) subdiagonal elem ents of the tridiagonalm atrix \(A\) in elem ents 1 to \(N-1\) of FFD ; OFFD \(\mathbb{N}\) ) need notbe set. On exit, OFFD may be m ultiplied by a constant factor chosen to avoid over/underflow in com puting the eigenvalues.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE = A 'or I'.

VU (input)
See the description of V L .
II (input)
IfRA N G E= 'I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{U}<=\mathbb{U}<=N\), if \(N>0\); \(\mathbb{H}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N ot referenced ifRANGE \(=\) A 'or V'.

IU (input)
See the description of II .
ABTOL (input)
The absolute error tolerance for the eigenvalues. A \(n\) approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b] of w idth less than orequal to

ABTOL + EPS * \(\max (\nmid,||\),\() ,\)
where EPS is them achine precision. If ABTOL is less than or equal to zero, then EPS* \(\mid\) | w illbe used in its place, where \(F \mid\) is the 1 -norm of the tridiagonalm atrix.

E igenvalues w illbe com puted m ostaccurately when ABTOL is set to tw ice the underflow threshold 2*SLAM CH (S ), notzero. If this routine retums w ith \(\mathbb{N}\) FO \(>0\), indicating that som e eigenvectors did not converge, try setting ABTOL to \(2 *\) SLAM CH ( S ).

See "C om puting Sm allSingularV ahes ofB idiagonal
\(M\) atrices w th G uaranteed H igh Relative A ccuracy," by Dem m eland \(K\) ahan, LA PA CK W orking \(N\) ote \#3.

NFOUND (output)
The total num ber of eigenvalues found. \(0<=\) NFOUND <= N. IfRANGE = A',NFOUND = N, and if RANGE \(=I ', N F O U N D=\mathbb{U}-\mathbb{H}+1\).

W (output)
The firstNFOUND elem ents contain the selected eigenvalues in ascending order.

Z (input) If \(\mathrm{JOBZ}=\mathrm{V}^{\prime}\), then if \(\mathbb{N} F O=0\), the first NFOUND colum ns of \(Z\) contain the orthonorm aleigenvectors of them atrix A comesponding to the selected eigenvalues, \(w\) ith the i-th colum \(n\) of \(Z\) holding the eigenvector associated w ith \(W\) (i). If an eigenvector fails to converge ( \(\mathbb{N}\) FO \(>0\) ), then that colum \(n\) of \(Z\) contains the latest approxim ation to the eigenvector, and the index of the eigenvector is retumed in \(\mathbb{F A} I I\). If \(\mathrm{JOBZ}=\mathrm{N}\) ', then Z is not referenced. N ote: the userm ustensure that at leastm ax (1,NFOUND) colum ns are supplied in the array \(Z\); ifRANGE = \(V\) ', the exactvalue ofNFOUND is not know \(n\) in advance and an upperbound \(m\) ust.be used.

LD Z (input)
The leading dim ension of the array \(\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=V^{\prime}\), LD \(Z>=\max (1, N)\).

W ORK (w orkspace)
dim ension ( \(5 * \mathrm{~N}\) )

IW ORK 2 (w orkspace)
FFII (output)
If \(\mathrm{JOBZ}=\mathrm{V}\) ', then if \(\mathbb{N F O}=0\), the first NFOUND
elem ents of \(\mathbb{F A} I I\) are zero. If \(\mathbb{N F O}>0\), then
FAII contains the indices of the eigenvectors
that failed to converge. If \(\mathrm{JOB}=\mathrm{N}^{\prime}\), then
IFA II is not referenced.
\(\mathbb{N} F O\) (output)
= 0 : successfinlexit
<0: if \(\mathbb{N N}\) FO \(=-\) i, the i-th argum enthad an illegalvalue
> 0 : if \(\mathbb{N F O}=\) i, then ieigenvectors failed to converge. Their indioes are stored in array
ㅍFAI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dstsv - com pute the solution to a system of linearequations
\(A * X=B\) where \(A\) is a sym \(m\) etric tridiagonalm atrix

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSTSV N,NRHS,L,D,SUBL,B,LDB, \mathbb{PNV,INFO)}}\mathbf{N},\textrm{N},\textrm{N}
\mathbb{NTEGER N,NRHS,LDB,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
\mathbb{NTEGER \mathbb{PIV (*)}}\mp@subsup{}{(}{*})
D OU BLE PRECISION L (*),D (*),SUBL (*),B (LD B ,*)
SUBROUT\mathbb{NEDSTSV_64 N,NRHS,L,D,SUBL,B,LDB,\mathbb{P}\mathbb{N},\mathbb{N}FO)}\=()
\mathbb{NTEGER*8N,NRHS,LDB,INFO}
\mathbb{NTEGER*8 \mathbb{PIV (*)}}\mp@subsup{}{(}{*})
DOUBLE PRECISION L (*),D (*),SUBL (*),B (LDB,*)

```
F95 INTERFACE
    SU BROUTINE STSV \(\mathbb{N}, N R H S, L, D, S U B L, B,[L D B], \mathbb{P} \mathbb{I},[\mathbb{N} F O])\)
    \(\mathbb{N}\) TEGER ::N,NRHS,LDB, \(\mathbb{N}\) FO
    \(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}\)
    REAL (8),D \(\mathbb{I}\) ENSION (:) ::L,D ,SUBL
    REAL (8),D IM ENSION (:,:) ::B
    SUBROUTINESTSV_64 \(\mathbb{N}, N R H S, L, D, S U B L, B,[L D B], \mathbb{P} \mathbb{I},[\mathbb{N} F O])\)
    \(\mathbb{N}\) TEGER (8) :: N,NRHS,LD B, \(\mathbb{N} F O\)
    \(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION(:) :: \(\mathbb{P} \mathbb{V}\)
    REAL (8),D IM ENSION (:) :: L, D ,SUBL
    REAL (8),D IM ENSION (:,:) ::B

\section*{C INTERFACE}
\#include <sunperfh>
void dstsv (intn, intnins, double *l, double *d, double
*subl, double *b, int ldb, int *íiv , int *info);
void dstsv_64 (long n, long nrhs, double *l, double *d, double *subl, double *b, long ldb, long *ípiv, long *info);

\section*{PURPOSE}
dstsv com putes the solution to a system of linear equations \(A * X=B\) where \(A\) is a sym m etric tridiagonalm atrix.

\section*{ARGUMENTS}

N (input) \(\mathbb{N}\) TEGER
The order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides in B.

L (input/output)
REAL array, dim ension \(\mathbb{N}\) )
O n entry, the n-1 subdiagonalelem ents of the tridiagonal m atrix A. On exit, part of the factorization of A .

D (input/output)
REA L aray, dim ension \(\mathbb{N}\) )
O n entry, the \(n\) diagonalelem ents of the tridiagonalm atrix A. On exit, the n diagonalelem ents of the diagonalm atrix D from the factorization of \(A\).

SUBL (output)
REA L anay, dim ension \(\mathbb{N}\) )
On exit, part of the factorization ofA.

B (input/output)
The colum ns ofB contain the righthand sides.

LD B (input)
The leading dim ension ofB as specified in a type orD \(\mathbb{I M}\) ENSION statem ent.
\(\mathbb{P} \mathbb{I} V\) (output)
\(\mathbb{N}\) TEGER array, dim ension \(\mathbb{N}\) )
On exit, the pivot indices of the factorization.
\(\mathbb{I N F O}\) (output)
\(\mathbb{N}\) TEGER
= 0: successfulexit
\(<0:\) if \(\mathbb{N N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N} F O=i, D(k, k)\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular and division by zero w illoccur if it is used to solve a system of equations.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dstrif-com pute the factorization of a sym \(m\) etric tridiagonalm atrix A

\section*{SYNOPSIS}

\(\mathbb{N}\) TEGER \(N, \mathbb{I N F O}\)
\(\mathbb{N} T E G E R \mathbb{P} \mathbb{V}\left({ }^{*}\right)\)
D OUBLE PRECISIONL (*), D (*), SUBL (*)
SUBROUTINEDSTTRF_64 \(\mathbb{N}, L, D, S U B L, \mathbb{P} \mathbb{V}, \mathbb{N} F O)\)
\(\mathbb{N}\) TEGER*8 \(\mathrm{N}, \mathbb{N} F \mathrm{O}\)
\(\mathbb{N} \operatorname{TEGER} * 8 \mathbb{P} \mathbb{I}\left({ }^{\star}\right)\)
DOUBLE PRECISIONL (*), D (*), SUBL (*)

\section*{F95 INTERFACE}

SUBROUTINE STTRF ( \(\mathbb{N}], L, D, S U B L, \mathbb{P} \mathbb{I},[\mathbb{N} F O]\) )
\(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::L,D ,SUBL
SUBROUTINE STTRF_64 (N ],L,D ,SUBL, \(\mathbb{P} \mathbb{I}\), \([\mathbb{N} F O]\) )
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION(:):: \(\mathbb{P} \mathbb{V}\)
REAL (8), D IM ENSION (:) ::L,D ,SUBL

\section*{C INTERFACE}
\#include <sunperfh>
void dstrff(intn, double *l, double *d, double *subl, int *ípív, int *info);
void dsttrf_64 (long n, double *l, double *d, double *subl, long *ípiv, long *info);

\section*{PURPOSE}
dsttrf com putes the factorization of a com plex H erm titian tridiagonalm atrix A .

\section*{ARGUMENTS}

N (input) \(\mathbb{N}\) TEGER
The order of them atrix \(A . N>=0\).

L (input/output)
REA L anay, dim ension \(\mathbb{N}\) )
O n entry, the n-1 subdiagonalelem ents of the tridiagonal m atrix A. On exit, part of the factorization of A .

D (input/output)
REA L anay, dim ension \((\mathbb{N})\)
O n entry, the \(n\) diagonalelem ents of the tridiagonalm atrix A. On exit, the \(n\) diagonalelem ents of the diagonalm atrix D from the L *D *L**H factorization of A.

SUBL (output)
REA L array, dim ension \(\mathbb{N}\) )
On exit, part of the factorization ofA.
\(\mathbb{P} \mathbb{I V}\) (output)
\(\mathbb{N}\) TEGER array, dim ension \((\mathbb{N})\)
On exit, the pivot indices of the factorization.
\(\mathbb{N}\) FO (output)
IN TEGER
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an ille-
galvahue
\(>0:\) if \(\mathbb{N} F O=i, D(k, k)\) is exactly zero. The
factorization has been com pleted, but the block
diagonalm atrix D is exactly singular and division
by zero w illoccur if it is used to solve a system
of equations.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsturs - com putes the solution to a real system of linear equations \(A * X=B\)

\section*{SYNOPSIS}

```

\mathbb{NTEGER N,NRHS,LDB,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
\mathbb{NTEGER \mathbb{PIV (*)}}\mp@subsup{}{(}{*})
D OU BLE PREC ISION L (*),D (*),SUBL (*),B (LD B ,*)
SUBROUTINEDSTTRS_64 N,NRHS,L,D,SUBL,B,LDB,\mathbb{P}\mathbb{N},\mathbb{N}FO)
\mathbb{NTEGER*8N,NRHS,LDB,INFO}
\mathbb{NTEGER** \mathbb{PNV (*)}}\mathbf{*}\mathrm{ ( )}
DOUBLE PRECISION L (*),D (*),SUBL (*),B (LDB,*)

```
F95 INTERFACE
    SUBROUTINE STTRS \(\mathbb{N}, N R H S, L, D, S U B L, B,[L D B], \mathbb{P} \mathbb{N},[\mathbb{N} F O])\)
    \(\mathbb{N}\) TEGER ::N,NRHS,LDB, \(\mathbb{N}\) FO
    \(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}\)
    REAL (8),D \(\mathbb{I}\) ENSION (:) ::L,D ,SUBL
    REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::B
    SU BROUTINE STTRS_64 \(\mathbb{N}, N R H S, L, D, S U B L, B,[L D B], \mathbb{P} \mathbb{I},[\mathbb{N} F O])\)
    \(\mathbb{N}\) TEGER (8) :: N,NRHS,LDB, \(\mathbb{N} F O\)
    \(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION(:) :: \(\mathbb{P} \mathbb{I}\)
    REAL (8), D \(\mathbb{I M}\) ENSION (:) :: L, D ,SUBL
    REAL (8),D IM ENSION (:,:) ::B

\section*{C INTERFACE}
\#include <sunperfh>
void dsttrs (intn, intnrhs, double *l, double *d, double
*subl, double *b, int ldb, int *íqiv, int *info);
void dsttrs_64 (long n, long nihs, double *l, double *d, dou-
ble *subl, double *b, long ldb, long *ípiv, long
*info);

\section*{PURPOSE}
disttrs com putes the solution to a real system of linear equations \(A * X=B\), where \(A\) is an \(N\) boy \(-N\) symm etric tridiagonalm atrix and X and B are N -by-N R H S m atrioes.

\section*{ARGUMENTS}

N (input) \(\mathbb{N}\) TEGER
The order of them atrix \(A . N>=0\).

NRHS (input)
\(\mathbb{N}\) TEGER
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B.NRH S >=0.
\(L\) (input) REAL aray, dim ension \((\mathbb{N}-1)\)
O n entry, the subdiagonalelem ents ofLL and D D.

D (input) REAL array, dim ension \(\mathbb{N}\) )
O n entry, the diagonalelem ents ofD D .

SUBL (input)
REA L array, dim ension (N-2)
O n entry, the second subdiagonalelem ents of LL .

B (input/output)
REA L array, dim ension
(LDB, NRHS) On entry, the N boy NRHS right hand side \(m\) atrix \(B\). On exit, if \(\mathbb{N F O}=0\), the N -byNRHS solution m atrix X .

LD B (input)
IN TEGER
The leading dim ension of the aray B. LD B >= \(\max (1, N)\)

IPIV (output)
\(\mathbb{N}\) TEGER array, dim ension \(\mathbb{N}\) )
D etails of the interchanges and block pivot. If \(\mathbb{P} \mathbb{V}(\mathbb{K})>0,1\) by 1 pìvot, and if \(\mathbb{P} \mathbb{V}(\mathbb{K})=K+1\) an interchange done; If \(\mathbb{P} \mathbb{I V}(\mathbb{K})<0,2\) by 2
pivot, no interchange required.
\(\mathbb{I N F O}\) (output)
\(\mathbb{N}\) TEGER
= 0: successfulexit
\(<0\) : if \(\mathbb{N} F O=-k\), the \(k\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsw ap -E xchange vectors \(x\) and \(y\).

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSW AP N,X,\mathbb{NCX,Y,}\mathbb{NCY)}}\mathbf{N}=()
\mathbb{NTEGERN,}\mathbb{NCX,\mathbb{NCY}}\mathbf{T}\mathrm{ \}
DOUBLE PRECISION X (*),Y (*)
SUBROUTINEDSW AP_64 N,X,\mathbb{NCX,Y, NNCY)}
INTEGER*8N,\mathbb{NCX,INCY}
DOUBLE PRECISION X (*),Y (*)

```

\section*{F95 INTERFACE}

SU BROUTINE SW AP ( \(\mathbb{N}], X,[\mathbb{N} C X], Y,[\mathbb{N} C Y])\)
\(\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y\)
REAL (8), D \(\mathbb{I M}\) ENSION (:) :: X,Y
SU BROUTINE SW AP_64 ( \(\mathbb{N}], X,[\mathbb{N} C X], Y,[\mathbb{N} C Y])\)
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{I N C X}, \mathbb{N} C Y\)
REAL (8), D \(\mathbb{I M}\) ENSION (:) :: X,Y

\section*{C INTERFACE}
\#include <sunperfh>
void dsw ap (intn, double *x, int incx, double * \(y\), int incy);
void dsw ap_64 (long n, double *x, long incx, double *y, long incy);

\section*{PURPOSE}
dsw ap Exchange \(x\) and \(y\) w here \(x\) and \(y\) are \(n\)-vectors.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.
X (input/output)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C X))\). On entry, the increm ented array \(X\) m ust contain the vector \(x\). On exit, the \(y\) vector.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X\) m ustnotbe zero. U nchanged on exit.

Y (input/output)
(1+(n-1)*abs( \(\mathbb{N} C Y)\) ). On entry, the increm ented array \(Y\) m ust contain the vectory. On exit, the x vector.
\(\mathbb{N} C Y\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of Y. \(\mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsycon -estim ate the reciprocal of the condition num ber (in the 1 -norm ) of a realsym \(m\) etric \(m\) atrix \(A\) using the factorization \(A=U * D * U * * T\) orA \(=L * D * L * * T\) com puted by SSY TRF

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NEDSYCON (UPLO,N,A,LDA, \mathbb{PIVOT,ANORM,RCOND,W ORK,}}\mathbf{N},\textrm{A},\textrm{A}
IN ORK2,INFO)
CHARACTER * 1 UPLO
NNTEGERN,LDA,}\mathbb{N}F
\mathbb{NTEGER \mathbb{PIVOT (*), IN ORK 2 (*)}}\mathbf{(})
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION A (LDA,*),W ORK (*)
SU BROUT\mathbb{NEDSYCON_64 (UPLO,N,A,LDA, PIVOT,ANORM,RCOND,W ORK,}
IN ORK2,INFO)
CHARACTER * 1 UPLO
\mathbb{N TEGER*8N,LDA,}\mathbb{N}FO
\mathbb{NTEGER*8 \mathbb{PIVOT (*), IN ORK2 (*)}}\mathbf{(*)}
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION A (LDA,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE SYCON (UPLO,N,A, [LDA], \(\mathbb{P} \mathbb{I V O T}, A N O R M, R C O N D,[W O R K]\), [ \(\mathbb{I W}\) ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1)::UPLO
\(\mathbb{N}\) TEGER ::N,LDA, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V} O T, \mathbb{I N}\) ORK2
REAL (8) ::ANORM,RCOND

SUBROUTINE SYCON_64 (UPLO,N,A, [LDA], \(\mathbb{P} \mathbb{I} O\) OT,ANORM,RCOND, \(\mathbb{W} O R K]\), [IW ORK2], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER (8) :: N , LDA, \(\mathbb{N}\) FO
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T, \mathbb{I W}\) ORK 2
REAL (8) ::ANORM,RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include < sunperfh>
void dsycon (charuple, int \(n\), double *a, int lda, int
*ipivot, double anorm, double *rcond, int *info);
void dsycon_64 (charuplo, long n, double *a, long lda, long *ipívot, double anorm, double *rcond, long *info);

\section*{PURPOSE}
dsycon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a realsym \(m\) etric \(m\) atrix A using the factorization \(A=U * D * U * * T\) orA \(=L * D * L * * T\) com puted by SSY TRF.

A \(n\) estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND \(=1 /(A N O R M *\) norm (inv (A))).

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) : : Upper triangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\);
\(=L^{\prime}:\) Low er triangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\).

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input) The block diagonalm atrix D and the m ultipliers used to obtain the factorU orL as com puted by SSY TRF .

LDA (input)
The leading dim ension of the array A. LD A >=
\(\max (1, \mathbb{N})\).
PIVOT (input)
D etails of the interchanges and the block structure ofD as determ ined by SSY TRF.

ANORM (input)
The 1-norm of the originalm atrix A.
RCOND (output)
The reciprocal of the condition number of the
\(m\) atrix \(A\), com puted as RCOND \(=1 /(A N O R M * A \mathbb{N} V N M)\), where \(A \mathbb{N} V N M\) is an estim ate of the 1 -nom of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension \((2 * N)\)
IN ORK 2 (w orkspace)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsyev - com pute alleigenvalues and, optionally, eigenvectors of a realsym \(m\) etricm atrix A

\section*{SYNOPSIS}

```

CHARACTER * 1 JOBZ,UPLO
INTEGERN,LDA,LDWORK,\mathbb{NFO}
DOUBLE PRECISION A (LDA,*),W (*),W ORK (*)
SU BROUT\mathbb{NE DSYEV_64(JOBZ,UPLO,N,A,LDA,W ,W ORK,LDW ORK, INFO )}
CHARACTER * 1 JOBZ,UPLO
INTEGER*8N,LDA,LDW ORK, INFO
DOUBLE PRECISION A (LDA,*),W (*),WORK(*)

```

\section*{F95 INTERFACE}

SUBROUTINE SYEV (JOBZ, UPLO,N,A, [LDA ],W, [W ORK], [LDW ORK], [NFO ])
CHARACTER (LEN=1): : JOBZ, UPLO
\(\mathbb{N} T E G E R:: N, L D A, L D W\) ORK, \(\mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A
SUBROUTINE SYEV_64 (OOBZ,UPLO,N,A, [LDA],W, [W ORK ], [LDW ORK ], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1): : JOBZ, UPLO
\(\mathbb{N}\) TEGER (8) :: N,LDA,LDW ORK, \(\mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (: : : : ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dsyev (char jobz, char uplo, intn, double *a, int lda, double *w , int *info);
void dsyev_64 (char jobzz, charuplo, long n, double *a, long lda, double *w , long *info);

\section*{PURPOSE}
dsyev com putes alleigenvalues and, optionally, eigenvectors of a realsym m etric m atrix A.

\section*{ARGUMENTS}

JOBZ (input)
\(=\mathrm{N}^{\prime}\) : C om pute eigenvalues only;
\(=\mathrm{V}:\) : C om pute eigenvalues and eigenvectors.

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low er triangle ofA is stored.
N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input/output)
On entry, the symm etricm atrix A. If UPLO = U', the leading \(\mathrm{N}-\) by -N upper triangularpartofA contains the upper triangular part of the \(m\) atrix \(A\). If UPLO = L', the leading N -by -N low er triangular partofA contains the low er triangular part of the \(m\) atrix \(A\). On exit, if \(J O B Z=V\) ', then if \(\mathbb{N} F O=0, A\) contains the orthonorm al eigenvectors of them atrix \(A\). If \(J O B Z=N\) ', then on exit the low er triangle (if \(\mathrm{PLO}=\mathrm{L}\) ) or the upper triangle (if \(\mathrm{UPLO}=\mathrm{U}\) ) of A , including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, \mathbb{N})\).

W (output)
If \(\mathbb{N} F O=0\), the eigenvalues in ascending order.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length of the amray W ORK. LDW ORK >= \(\mathrm{max}(1,3 \star \mathrm{~N}-1)\). For optim alefficiency, LDW ORK \(>=\) \((\mathbb{N B}+2)^{\star} \mathrm{N}\), where NB is the blocksize for SSY TRD retumed by ILAENV.

IfLD W ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i-\) th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=\) i, the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dsyevd - com pute alleigenvalues and, optionally, eigenvectors of a realsym \(m\) etricm atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSYEVD (JOBZ,UPLO,N,A,LDA,W ,W ORK,LW ORK,IN ORK,}
LIN ORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGERN,LDA,LW ORK,LIN ORK,INFO}
INTEGER IN ORK (*)
DOUBLE PRECISION A (LDA,*),W (*),W ORK (*)

```

```

    L\mathbb{IN ORK, \mathbb{NFO)}}\mathbf{}\mathrm{ )}
    CHARACTER * 1 JOBZ,UPLO
\mathbb{N}TEGER*8N,LDA,LW ORK,LIN ORK,\mathbb{NFO}
INTEGER*8 IN ORK (*)
DOUBLE PRECISION A (LDA,*),W (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE SYEVD (JOBZ,UPLO ,N,A, [LDA ],W , [W ORK ], [LW ORK], [IW ORK ], \(\left[\begin{array}{ll}\mathbb{N} & O R K],[\mathbb{N} F O])\end{array}\right.\)

CHARACTER (LEN=1)::JOBZ, UPLO
\(\mathbb{N} T E G E R:: N, L D A, L W\) ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL (8),D \(\mathbb{I M}\) ENSION (:) ::W ,W ORK
REAL (8), D \(\mathbb{M}\) ENSIO N (:,:) ::A

SU BROUTINE SYEVD_64 (OBZ,UPLO,N,A, [LDA],W, [W ORK], [LW ORK], [IW ORK ], [LIV ORK ], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1)::JOBZ,UPLO
\(\mathbb{N}\) TEGER ( 8 ) :: N, LDA, LW ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
REAL (8),D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include < sunperfh>
void dsyevd (char j̀jbz, char uplo, intn, double *a, int lda, double *W , int*info);
void dsyevd_64 (char jobz, charuplo, long n, double *a, long lda, double *w , long *info);

\section*{PURPOSE}
dsyevd com putes alleigenvalues and, optionally, eigenvectors of a real symm etric matrix A. Ifeigenvectors are desired, ituses a divide and conqueralgorithm .

The divide and conquer algorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on \(m\) achines w ith a guard digit in add/subtract, or on those binary \(m\) achines \(w\) thout guard digits w hich subtract like the \(C\) ray X M P , C ray Y M P , C ray C-90, orC ray-2. Itcould conceivably fail on hexadecim al or decim al \(m\) achines \(w\) ithout guard digits, butw e know of none.

Because of large use ofBLAS of level3, SSY EV D needs N**2 m ore w orkspace than SSY EVX .

\section*{ARGUMENTS}

JOBZ (input)
= N ': C om pute eigenvalues only;
\(=\mathrm{V}\) : C om pute eigenvalues and eigenvectors.

UPLO (input)
= U : : U pper triangle ofA is stored;
= L': Low er triangle ofA is stored.
N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)

On entry, the symm etric \(m\) atrix \(A\). If \(\mathrm{PLO}=\mathrm{U}\) ', the leading N -by -N uppertriangularpart of A contains the upper triangular part of the \(m\) atrix \(A\). If U PLO = L', the leading \(N\) by -N low er triangular partofA contains the low er triangular part of the \(m\) atrix \(A\). On exit, if \(J O B Z=V\) ', then if \(\mathbb{N} F O=0, A\) contains the orthonorm al eigenvectors of them atrix \(A\). If \(J O B Z=N\) ', then on exit the low er triangle (if \(\mathrm{PLLO}=\mathrm{L}\) ) or the upper triangle (if \(\mathrm{UPLO}=\mathrm{U}\) ) of A , including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

W (output)
If \(\mathbb{N} F O=0\), the eigenvalues in ascending order.

\section*{W ORK (w orkspace)}
dim ension (LW ORK)On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim alLW ORK.

\section*{LW ORK (input)}

The dim ension of the array W ORK. If \(\mathrm{N}<=1\), LW ORK must be at least1. If \(\mathrm{JOBZ}=\mathrm{N}\) 'and \(\mathrm{N}>\) 1, LW ORK mustbe at least \(2{ }^{*} \mathrm{~N}+1\). If \(\mathrm{JOBZ}=\mathrm{V}^{\prime}\) and \(\mathrm{N}>1\), LW ORK must be atleast \(1+6 \star \mathrm{~N}+\) \(2 \star \mathrm{~N} * * 2\).

If LW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

IV ORK (w orkspace/output)
On exit, if \(\mathbb{N} F O=0, \mathbb{I N}\) O RK (1) retums the optim al
LIV ORK.
LIV ORK (input)
The dim ension of the array \(\mathbb{I N}\) ORK. If \(\mathrm{N}<=1\), LIN ORK mustbe at least1. If JOBZ \(=\mathrm{N}\) 'and \(\mathrm{N}>\) \(1, L \mathbb{I}\) ORK m ustbe at least 1. If \(\mathrm{JOBZ}=\mathrm{V}^{\prime}\) and \(\mathrm{N}>1, \mathrm{LIV}\) ORK must.be at least3 \(+5 * \mathrm{~N}\).

If LIV ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the IW ORK array, retums this value as the first entry of the \(\mathbb{I W}\) ORK array, and no errorm essage
related to LIN ORK is issued by XERB LA.
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N} F O=\) i, the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

\section*{FURTHER DETAILS}

B ased on contributions by
JeffR utter, C om puter Science D ìvision, U niversity of C alifomia
atB erkeley, U SA
M odified by Francoise \(T\) isseur, U niversity of Tennessee.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dsyevr - com pute selected eigenvalues and, optionally, eigenvectors of a realsym \(m\) etric tridiagonalm atrix \(T\)

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE DSYEVR(JOBZ,RANGE,UPLO,N,A,LDA,VL,VU,IL,\mathbb{U,}}\mathbf{N},

```

CHARACTER * 1 JOBZ,RANGE, UPLO
\(\mathbb{N}\) TEGER \(N, L D A, \mathbb{I}, \mathbb{U}, M, L D Z, L W O R K, L \mathbb{I N} O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R \operatorname{ISUPPZ}(*), \mathbb{I N}\) ORK ( \({ }^{*}\) )
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION A (LDA , *), W (*) , Z (LD Z , *), W ORK (*)
SU BROUTINEDSYEVR_64 (JOBZ,RANGE,UPLO,N,A,LDA,VL,VU, \(\mathbb{I}, \mathbb{I U}\),
    ABSTOL,M,W,Z,LDZ,ISUPPZ,WORK,LWORK, IN ORK,LIN ORK, \(\mathbb{N} F O\) )
```

CHARACTER * 1 JOBZ,RANGE,UPLO

```
\(\mathbb{N} T E G E R * 8 N, L D A, \mathbb{L}, \mathbb{U}, M, L D Z, L W O R K, L \mathbb{I N} O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \operatorname{ISUPPZ}(*), \mathbb{I N}\) ORK ( \({ }^{*}\) )
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION A (LDA ,*), W (*), Z (LD Z, \(\left.{ }^{\star}\right), \mathrm{W} O R K(*)\)

\section*{F95 INTERFACE}

SU BROUTINE SYEVR (JOBZ,RANGE,UPLO, \(\mathbb{N}], A,[L D A], V L, V U, \mathbb{L}, \mathbb{U}\), ABSTOL,M,W,Z,[LDZ], ISUPPZ, [W ORK], [LW ORK], [IN ORK], [LINORK], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1):: OBZ,RANGE,UPLO
\(\mathbb{N}\) TEGER :: \(N, L D A, \mathbb{L}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W\) ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: ISU PPZ, IN ORK

REAL (8) ::VL,VU,ABSTOL
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A, Z
SUBROUTINE SYEVR_64 (JOBZ,RANGE,UPLO, \(\mathbb{N}], A,[L D A], V L, V U, \mathbb{I}, \mathbb{I U}\), ABSTOL,M,W,Z,[LD Z], ISUPPZ, [W ORK ], [LW ORK], [IW ORK], [LIW ORK], [ \(\mathbb{N} F \mathrm{O}\) ])

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
\(\mathbb{N} T E G E R(8):: N, L D A, \mathbb{I}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W\) ORK,LIW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{I S U P P Z , ~} \mathbb{I}\) ORK
REAL (8) ::VL,VU,ABSTOL
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
REAL (8),D IM ENSION (:,:)::A,Z

\section*{C INTERFACE}
\#include <sunperfh>
void dsyevr(char jobz, char range, charuplo, intn, double
*a, int lda, double vl, double vu, int il, int in, double abstol, int \({ }^{\mathrm{m}} \mathrm{m}\), double \({ }^{\mathrm{W}}\), double \({ }^{\mathrm{z}}\), int ldz, int*isuppz, int*info);
void dsyevr_64 (char jobz, char range, char uplo, long n, double *a, long lda, double vl, double vu, long \(i 1\), long in, double abstol, long \({ }^{*}\) m, double \({ }^{*}\) w, double *z, long ldz, long *isuppz, long *info);

\section*{PURPOSE}
dsyevr com putes selected eigenvalues and, optionally, eigenvectors of a realsym \(m\) etric tridiagonalm atrix \(T\). Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

W heneverpossible, SSY EVR calls SSTEGR to com pute the eigenspectrum using Relatively Robust Representations. SSTEGR com putes eigenvalues by the dqds algorithm, while orthogonaleigenvectors are com puted from various "good" L D \(L^{\wedge} \mathrm{T}\) representations (also known as Relatively Robust Representations). G ram -Schm idtorthogonalization is avoided as far as possible. M ore specifically, the various steps of the algorithm are as follow s. For the \(i\)-th unreduced block oft,
(a) Com pute \(T\)-sigm a_i= L_iD_iL_i^T, such that L_i D_iL_i^T
is a relatively robust representation,
(b) C om pute the eigenvalues, lam bda_j, of L_i D_i L_i^T to high
relative accuracy by the dqds algorithm,
(c) If there is a cluster of close eigenvalues, "choose"
sigm a_i
close to the cluster, and go to step (a),
(d) G iven the approxim ate eigenvalue lam bda_jofL_i D_i L_i^T, com pute the corresponding eigenvectorby form ing a rank-revealing tw isted factorization.
The desired accuracy of the output can be specified by the inputparam eterABSTOL.

Form ore details, see "A new O ( \(n^{\wedge} 2\) ) algorithm for the sym \(m\) etric tridiagonal eigenvahue/eigenvector problem ", by Inder屰D hillon, \(C\) om puter Science D ivision TechnicalR eport N o.U CB /C SD -97-971, U C Berkeley, M ay 1997.
N ote 1 :SSYEVR calls SSTEGR when the full spectum is requested on \(m\) achines \(w\) hich conform to the ieee-754 floating pointstandard. SSYEVR calls SSTEBZ and SSTE \(\mathbb{N}\) on non-ieee \(m\) achines and
when partialspectrum requests are m ade.

N orm alexecution of SSTEGR m ay create NaNs and infinities and hence \(m\) ay abort due to a floating pointexception in environm ents which do nothandle N aN s and infinities in the ieee standard defaultm anner.

\section*{ARGUMENTS}

JOBZ (input)
\(=\mathrm{N}\) : C om pute eigenvalues only;
\(=\mathrm{V}\) ': C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvalues w illbe found.
\(=\mathrm{V}\) ::alleigenvalues in the half-open interval
( \(\mathrm{L}, \mathrm{V} \mathrm{U}] \mathrm{w}\) ill be found. = I ': the II th through
IU th eigenvahes w illbe found.

UPLO (input)
\(=\mathrm{U}\) ': Upper triangle ofA is stored;
\(=\mathbb{L}\) ': Low er triangle of A is stored.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the symm etric \(m\) atrix A. If UPLO \(=U '\), the leading N -by N uppertriangularpartof u contains the upper triangular part of the \(m\) atrix \(A\).

If UPLO \(=\mathrm{L}\) ', the leading N -by N low er triangular partofA contains the low er triangular part of the matrix A. On exit, the low ertriangle (if \(\mathrm{UPLO}=\mathrm{L}\) ) or the uppertriangle (if \(\mathrm{UPLO}=\mathrm{U}\) ) of A , including the diagonal, is destroyed.

LDA (input)
The leading dim ension of the amay A. LDA >= \(\max (1, \mathbb{N})\).

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE=A'or I'.

VU (input)
See the description of V L .

II (input)
If RA NGE= I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{L}<=\mathbb{U}<=N\), if \(N>0\); \(\mathbb{L}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N ot referenced ifRANGE= A 'or V'.

IU (input)
See the description of II.

ABSTOL (input)
The absolute error tolerance for the eigenvalues. A \(n\) approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b] of w idth less than orequal to

ABSTOL + EPS * max ( \(\mid\) |, \(\mid\) |),
where EPS is them achine precision. IfABSTOL is less than or equalto zero, then EPS* \(\mid\) |w illbe used in its place, where \(T\) is the 1 -nom of the tridiagonal \(m\) atrix obtained by reducing \(A\) to tridiagonal form .

See "C om puting Sm allSingularV alues of \(B\) idiagonal M atrices with G uaranteed H igh Relative A ccuracy," by D em meland K ahan, LA PA CK W orking N ote \#3.

If high relative accuracy is im portant, setA B STO L to SLAM CH (Safe minim um' ). D oing so will guarantee thateigenvalues are com puted to high relative accuracy when possible in future
releases. The cument code does not \(m\) ake any guarantees abouthigh relative accuracy, but funutre releasesw ill. See J.B arlow and J. Demmel, "C om puting A ccurate E igensystem s of Scaled D iagonally D om inantM atrices", LA PA CK W orking N ote \#7, for a discussion of which \(m\) atrices define their eigenvalues to high relative accuracy .

M (output)
The total num berofeigenvalues found. \(0<=\mathrm{M}\) <= N . IfRANGE \(=A \prime, \mathrm{M}=\mathrm{N}\), and ifRANGE \(=\mathrm{I}^{\prime}, \mathrm{M}=\) \(\mathbb{U}-\mathbb{L}+1\).

W (output)
The firstM elem ents contain the selected eigenvalues in ascending order.
\(Z\) (input) If \(\mathcal{O B Z}=\mathrm{V}\) ', then if \(\mathbb{N F O}=0\), the first M colum ns of \(Z\) contain the orthonorm aleigenvectors of them atrix A comesponding to the selected eigenvalues, \(w\) ith the \(i\)-th colum \(n\) of \(Z\) holding the eigenvector associated w ith W (i). If \(\mathrm{JOBZ}=\mathrm{N}\) ', then \(Z\) is not referenced. N ote: the userm ust ensure that at leastm ax ( \(1, \mathrm{M}\) ) colum ns are supplied in the amay \(Z\); ifRANGE = V', the exact value of M is not know n in advance and an upperbound m ust be used.

LD Z (input)
The leading din ension of the array \(\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{max}(1, N)\).

ISU PPZ (output)
The support of the eigenvectors in \(Z\), ie., the indices indicating the nonzero elem ents in Z . The i-th eigenvector is nonzero only in elem ents \(\operatorname{ISUPPZ}(2 \star i-1)\) through \(\operatorname{ISU} \operatorname{PPZ}(2 * i)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array W ORK. LW ORK >= \(\max (1,26 * N)\). For optim al efficiency, LW ORK >= (NB+6)*N, where NB is the m ax of the blocksize for SSY TRD and SORM TR retumed by \(\mathbb{L} A E N V\).

If LW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of
the W ORK amray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA .
IV ORK (w orkspace/output)
On exit, if \(\mathbb{N} F O=0, \mathbb{I N}\) O RK (1) retums the optim al LW ORK.

LIV ORK (input)
The dim ension of the array \(\mathbb{I N}\) ORK. LIV ORK >= \(\max (1,10 \star N)\).

If LIV ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the \(\mathbb{I W}\) ORK array, retums this value as the first entry of the IV ORK array, and no errorm essage related to LIN ORK is issued by XERBLA.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue
> 0: Intemalerror

\section*{FURTHER DETAILS}

B ased on contributions by
Inder\#̈̈D hillon, \(\mathbb{B M}\) A \(\operatorname{lm}\) aden, U SA
O sniM arques, LBNL LNERSC ,U SA
K en Stanley, C om puterScience D ivision, U niversity of C alifomia atB erkeley, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsyevx - com pute selected eigenvahues and, optionally, eigenvectors of a realsym \(m\) etric \(m\) atrix \(A\)

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NEDSYEVX (JOBZ,RANGE,UPLO,N,A,LDA,VL,VU,\mathbb{L},\mathbb{U},},N,
ABTOL,NFOUND,W ,Z,LDZ,W ORK,LDW ORK,IN ORK2,\mathbb{FA}|,\mathbb{N}FO)
CHARACTER * 1 JOBZ,RANGE,UPLO
\mathbb{NTEGERN,LDA,\mathbb{I},\mathbb{U},NFOUND,LDZ,LDW ORK,\mathbb{NFO}}\mathbf{N}\mathrm{ , L}
\mathbb{NTEGER IN ORK2 (*),\mathbb{FA L (*)}}\mathbf{(*)}
DOUBLE PRECISION VL,VU,ABTOL
DOUBLE PRECISION A (LDA ,*),W (*),Z (LD Z,*),W ORK (*)
SUBROUTINEDSYEVX_64(JOBZ,RANGE,UPLO,N,A,LDA,VL,VU,IL,\mathbb{U},

```

CHARACTER * 1 JOBZ,RANGE,UPLO
\(\mathbb{N}\) TEGER* \(8 \mathrm{~N}, \mathrm{LDA}, \mathbb{L}, \mathbb{U}, N F O U N D, L D Z, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{I V}\) ORK 2 (*), \(\mathbb{F} A \mathbb{L}\) (*)
DOUBLE PRECISION VL,VU,ABTOL
DOUBLE PRECISION A (LDA , *), W (*), Z (LD Z, *), W ORK (*)

\section*{F95 INTERFACE}
 ABTOL,NFOUND,W,Z, [LDZ], [W ORK], [LDW ORK], [IW ORK2], IFA \(\mathbb{I}\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1):: OB B, RANGE,UPLO
\(\mathbb{N} T E G E R:: N, L D A, \mathbb{L}, \mathbb{U}, N F O U N D, L D Z, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION (:) :: \(\mathbb{I W}\) ORK2, \(\mathbb{F A} \mathbb{I}\)
REAL (8) ::VL,VU,ABTOL

REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
REAL (8), D IM ENSION (:,:) ::A, Z

SU BROUTINE SYEVX_64 (OBB , RANGE, UPLO, N, A, [LDA ],VL,VU, \(\mathbb{I}, \mathbb{U}\),
 [ \(\mathbb{N} \mathrm{FO}]\) )

CHARACTER ( \(L E N=1\) ) : : OBZ, RANGE, UPLO
\(\mathbb{N}\) TEGER (8) :: N , LDA \(, \mathbb{I}, \mathbb{I}, N\) FOUND , LD Z , LD W ORK , \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{I M} \operatorname{ENSION}(:):: \mathbb{I N} O R K 2, \mathbb{F} A \mathbb{I}\)
REAL (8) :: VL,VU, ABTOL
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
REAL (8), D \(\mathbb{I M}\) ENSION (:,:) ::A, Z

\section*{C INTERFACE}
\#include <sunperfh>
void dsyevx (char jobz, char range, charuple, intn, double *a, int lda, double vl, double vu, int il, intiu, double abtol, int *nfound, double *W, double *z, int \(1 d z\), int *ifail, int *info);
void dsyevx_64 (char jobz, char range, char uplo, long n, double *a, long lda, double vl, double vu, long il, long iu, double abtol, long *nfound, double ** , double *z, long ldz, long *ifail, long *info);

\section*{PURPOSE}
dsyevx com putes selected eigenvalues and, optionally , eigenvectors of a real symm etric \(m\) atrix A. Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

\section*{ARGUMENTS}

JOBZ (input)
\(=\mathrm{N}^{\prime}:\) C om pute eigenvalues only;
\(=\mathrm{V}\) ': C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvalues w illbe found.
\(=\mathrm{V}\) : alleigenvalues in the half-open interval
( \(\mathrm{VL}, \mathrm{V} \mathrm{U}]\) w ill be found. = I ': the \(\Pi\)-th through
\(\mathbb{I U}\)-th eigenvalues w ill.be found.

UPLO (input)
\(=\mathrm{U}\) ': Upper triangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the symm etric \(m\) atrix \(A\). If \(\operatorname{PLO}=U '\), the leading N -by N uppertriangularpartof A contains the upper triangular part of the \(m\) atrix \(A\). If UPLO = L', the leading N -by N low er triangular partofA contains the low er triangular part of the \(m\) atrix \(A\). On exit, the low ertriangle (if \(\mathrm{UPLO}=\mathrm{L}\) ) or the uppertriangle (if \(\mathrm{UPLO}=\mathrm{U}\) ) of A , including the diagonal, is destroyed.
LDA (input)
The leading dim ension of the array A. LD A >= \(\max (1, \mathbb{N})\).

VL (input)
IfRANGE=V', the low er and upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE = A 'or I'.

VU (input)
See the description of V L .

II (input)
IfRA NGE= I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}\), if \(\mathrm{N}>0\); \(\mathbb{L}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N ot referenced ifRANGE \(=\) A 'or V'.
\(\mathbb{I U}\) (input)
See the description of II.

ABTOL (input)
The absolute error tolerance for the eigenvalues. A \(n\) approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b]
of w idth less than orequal to
ABTOL + EPS * max ( \(\mathfrak{k}|, \mathrm{p}|)\),
where EPS is the m achine precision. If ABTOL is less than orequal to zero, then EPS* \(\mid\) | w illbe used in its place, where \(T\) | is the 1 -norm of the tridiagonal \(m\) atrix obtained by reducing \(A\) to tridiagonal form .

E igenvalues w illbe com putedm ostaccurately when

ABTOL is set to tw ice the underflow threshold \(2 *\) SLAM CH (S ), notzero. If this routine retums w ith \(\mathbb{N}\) FO \(>0\), indicating that som e eigenvectors did not converge, try setting ABTOL to \(2 *\) SLAM CH (S ).

See "C om puting Sm allSingularV ahues of B idiagonal M atrices w ith G uaranteed H igh Relative A ccuracy," by D em m eland K ahan, LA PA CK W orking N ote \#3. NFOUND (output)

The total num ber of eigenvalues found. \(0<=\) NFOUND <= N. IfRANGE = A',NFOUND = N, and if RANGE \(=I^{\prime}\), NFOUND \(=\mathbb{U}-\mathbb{L}+1\).

W (output)
On norm alexit, the firstN FOUND elem ents contain the selected eigenvalues in ascending order.
\(Z\) (input) If \(J O B Z=V^{\prime}\), then if \(\mathbb{N F O}=0\), the first \(N F O U N D\) colum ns of \(Z\) contain the orthonorm aleigenvectors of them atrix A comesponding to the selected eigenvalues, \(w\) ith the i-th colum \(n\) of \(Z\) holding the eigenvector associated with \(W\) (i). If an eigenvector fails to converge, then that colum n of \(Z\) contains the latest approxim ation to the eigenvector, and the index of the eigenvector is retumed in FAII. If \(J 0 B Z=N\) ', then \(Z\) is not referenced. \(N\) ote: the user must ensure that at least \(m\) ax ( 1, NFO UND ) colum ns are supplied in the array \(Z\); if RANGE = V', the exactvalue ofNFOUND is not know \(n\) in advance and an upperbound \(m\) ust.be used.

LD Z (input)
The leading dim ension of the array \(Z . L D Z>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{max}(1, N)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length of the array \(W\) ORK. LDW ORK >= \(\max (1,8 * N)\). For optim al efficiency, LDW ORK >= ( \(\mathrm{N} B+3)^{*} \mathrm{~N}\), where \(N B\) is the \(m\) ax of the blocksize for SSY TRD and SORM TR retumed by IUAENV.

If LD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

IV ORK 2 (w orkspace)
ㅍFAII (output)
If \(\mathrm{OBZ}=\mathrm{V}^{\prime}\), then if \(\mathbb{N F O}=0\), the first NFOUND
elem ents of \(\mathbb{F A} I I\) are zero. If \(\mathbb{N} F O>0\), then
IFA II contains the indices of the eigenvectors
that failed to converge. If \(\mathrm{OBZ}=\mathrm{N}\) ', then
\(\mathbb{F} A \mathbb{I}\) is notreferenced.

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\) th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=\) i, then ieigenvectors failed to converge. Their indices are stored in anay 팦.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsygs2 - reduce a real sym m etric-definite generalized eigenproblem to standard form

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NEDSYGS2(TTYPE,UPLO,N,A,LDA,B,LDB,INFO )}
CHARACTER * 1 UPLO
INTEGER ITYPE,N,LDA,LDB,INFO
DOUBLE PRECISION A (LDA,*),B (LDB,*)
SU BROUT\mathbb{NE DSYGS2_64(TTYPE,UPLO,N,A ,LDA,B,LD B,INFO )}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8 ITYPE,N,LDA,LDB, INFO}
DOUBLE PRECISION A (LDA,*),B(LDB,*)

```
F95 INTERFACE
    SU BROUTINE SYGS2 (TTYPE, UPLO ,N,A, [LDA ], B, [LDB], [NFO])
    CHARACTER (LEN=1) ::UPLO
    \(\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D A, L D B, \mathbb{N} F O\)
    REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B
    SU BROUTINE SYGS2_64 (TTYPE, UPLO ,N,A, [LDA ], B, [LDB], [NFO ])
    CHARACTER (LEN=1) ::UPLO
    \(\mathbb{N} T E G E R(8):: \mathbb{I T Y P E , N , L D A , L D B , ~} \mathbb{N} F O\)
    REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B
C INTERFACE
    \#include <sunperfh>
void dsygs2 (int itype, charuple, intn, double *a, int lda, double *b, int ldb, int *info);
void dsygs2_64 (long itype, charuplo, long n, double *a, long lda, double *b, long ldb, long *info);

\section*{PURPOSE}
dsygs2 reduces a realsym m etric-definite generalized eigenproblem to standard form .

If ITYPE \(=1\), the problem is \(A * x=\) lam bda*B \({ }^{*} x_{\text {, }}\) and \(A\) is overw ritten by inv (U)*A *inv (U) orinv (L)*A *inv (L) If ITYPE \(=2\) or 3 , the problem is \(A * B * x=\) lam bda* \(x\) or \(B * A * x=1 a m\) bda* \(x\), and \(A\) is overw ritten by \(U * A * U\) `orL \({ }^{*} A * L\). B m usthave been previously factorized as \(U\) * \(U\) or \(L \star L^{\prime}\) by SPOTRF.

\section*{ARGUMENTS}

ITYPE (input)
\(=1:\) com pute inv (U) \()\) A *inv \((\mathrm{U})\) orinv (L)*A *inv (L);
\(=2\) or 3 : com pute \(U\) *A *U 'orL *A *L.

UPLO (input)
Specifies w hether the upper or low er triangular
part of the sym \(m\) etric \(m\) atrix A is stored, and how
B has been factorized. = U ': Upper triangular
= LL : Low er triangular

N (input) The order of the m atriges A and \(\mathrm{B} . \mathrm{N}>=0\).

A (input/output)
O n entry, the sym m etric m atrix A. If UPLO = U', the leading \(n\) by \(n\) upper triangularpart of A contains the upper triangular part of the \(m\) atrix \(A\), and the strictly low er triangularpart of A is not referenced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading n by n low er triangularpart ofA contains the low ertriangularpart of the \(m\) atrix \(A\), and the strictly upper triangularpart of A is not referenced.

On exit, if \(\mathbb{N F O}=0\), the transform ed \(m\) atrix, stored in the sam e form at as A.

\section*{LD A (input)}

The leading dim ension of the array A. LDA >= \(\max (1, N)\).
\(B\) (input) The triangular factor from the C holesky factorization ofB, as retumed by SPO TRF.

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).
\(\mathbb{I N} F O\) (output)
= 0: successfulexit.
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvahue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsygst-reduce a realsym \(m\) etric-definite generalized eigenproblem to standard form

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSYGST (TTYPE,UPLO,N,A,LDA,B,LDB, NNFO)}
CHARACTER * 1 UPLO
INTEGER ITYPE,N,LDA,LDB,INFO
DOUBLE PRECISION A (LDA,*),B (LDB,*)
SU BROUT\mathbb{NEDSYGST_64(ITYPE,UPLO,N,A ,LDA,B,LD B, INFO )}
CHARACTER * 1 UPLO
INTEGER*8 ITYPE,N,LDA,LDB,INFO
DOUBLE PRECISION A (LDA,*),B(LDB,*)

```
F95 INTERFACE
    SU BROUTINE SYGST (TTYPE, UPLO,N,A, [LDA], B, [LDB], [INFO])
    CHARACTER (LEN=1) ::UPLO
    \(\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D A, L D B, \mathbb{N} F O\)
    REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B
    SU BROUTINE SYGST_64 (ITYPE, UPLO ,N,A, [LDA ],B, [LD B], [ \(\mathbb{N} F \mathrm{FO}]\) )
    CHARACTER (LEN=1) ::UPLO
    \(\mathbb{N}\) TEGER (8) :: \(\mathbb{T} Y\) PE, \(N, L D A, L D B, \mathbb{N F O}\)
    REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B
C INTERFACE
    \#include <sunperfh>
void dsygst(int itype, charuplo, intn, double *a, int lda, double *b, int ldlo, int *info);
void dsygst 64 (long itype, charuple, long n, double *a, long lda, double *b, long ldb, long *info);

\section*{PURPOSE}
dsygst reduces a real sym \(m\) etric-definite generalized eigenproblem to standard form .

If ITY PE \(=1\), the problem is \(A * x=\) lam bda*B \({ }^{*} \mathrm{X}\), and \(A\) is overw ritten by inv \((U * * T) * A * \operatorname{inv}(U)\) or \(\operatorname{inv}(\amalg) * A\) *inv ( \(\ddagger * * T)\)
If ITYPE \(=2\) or 3 , the problem is \(A * B *_{x}=\) lam bda* x or \(B *_{A} *_{X}=\operatorname{lam}\) bda* \(x\), and \(A\) is overw ritten by \(U * A * U * * T\) or L**T*A*L。

B m usthave been previously factorized as U ** T * U or \(\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\) by SPO TRF.

\section*{ARGUMENTS}

ITYPE (input)
\(=1:\) compute \(\quad \operatorname{inv}(\mathrm{U} * * T) \star A * \operatorname{inv}(\mathrm{U})\) or
\(\operatorname{inv}(\amalg) \star A\) *inv \((\amalg * * T)\);
\(=2\) or 3 : com pute \(\mathrm{U} * \mathrm{~A} * \mathrm{U} * * \mathrm{~T}\) orL \({ }^{* *} \mathrm{~T} * \mathrm{~A} * \mathrm{~L}\).

UPLO (input)
\(=U^{\prime}\) : Uppertriangle of \(A\) is stored and \(B\) is factored as \(\mathrm{U} * * \mathrm{~T} * \mathrm{U} ;=\mathrm{L}:\) : Low er triangle of A stored and B is factored as \(\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\).

N (input) The order of the \(m\) atriges \(A\) and \(B . N>=0\).

A (input/output)
O n entry, the sym m etric m atrix A. If \(\mathrm{ULO}=\mathrm{U}\) ', the leading N boy N uppertriangularpantofA contains the upper triangular part of the \(m\) atrix \(A\), and the strictly low er triangularpart of A is not referenced. If UPLO \(=\mathrm{L}\) ', the leading N -by- N low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of A is not referenced.

On exit, if \(\mathbb{N F O}=0\), the transform ed matrix,
stored in the sam e form at as A .

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

B (input) The triangular factor from the Cholesky factorization ofB , as retumed by SPO TRF .

LD B (input)
The leading dim ension of the aray \(\mathrm{B} . \operatorname{LDB}>=\) \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i-\) th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dsygv - com pute all the eigenvalues, and optionally, the
eigenvectors of a real generalized sym $m$ etric-definite eigen-
problem, of the form $A * x=(\operatorname{lam} . b d a) * B * x, A * B x=(l a m . b d a) * x$, or
B *A *X=(lam bda)*x

```

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE DSYGV (TTYPE,JOBZ,UPLO,N,A,LDA,B,LDB,W ,W ORK,}
LDW ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGER ITYPE,N,LDA,LDB,LDW ORK, NNFO}
DOUBLE PRECISION A (LDA,*),B(LDB,*),W (*),W ORK (*)
SU BROUT\mathbb{NE DSYGV_64 (TTYPE,JOBZ,UPLO ,N,A LDA ,B,LDB,W ,W ORK,}
LDWORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGER*8 ITYPE,N,LDA,LDB,LDW ORK,INFO}
DOUBLE PRECISION A (LDA,*),B (LDB,*),W (*),W ORK (*)

```

\section*{F95 INTERFACE}
 [LDW ORK ], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1)::JOBZ,UPLO
\(\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D A, L D B, L D W O R K, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A, B
SU BROUTINE SYGV_64 (TTYPE, JOBZ, UPLO ,N,A, [LDA ], B, [LDB],W, [W ORK], [LDW ORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1):: DBZ, UPLO
\(\mathbb{N}\) TEGER (8) :: ITYPE,N,LDA,LDB,LDWORK, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void dsygv (int itype, char jंbz, charuplo, int n, double
*a, int lda, double *b, int ldlo, double *W , int
*info);
void dsygv_64 (long itype, char j̀bz, charuplo, long n, double *a, long lda, double *b, long ldlb, double *w , long *info);

\section*{PURPOSE}
dsygv com putes all the eigenvalues, and optionally, the eigenvectors of a realgeneralized sym \(m\) etric-definite eigenproblem, of the form \(A * x=\left(\operatorname{lam}\right.\) bda) \({ }^{*}{ }^{*} x, A * B x=(\operatorname{lam} b d a){ }^{*} x\), or \(B{ }^{*} A{ }^{*} x=(\operatorname{lam} . b d a){ }^{*} x\). H ere \(A\) and \(B\) are assum ed to be sym m etric and \(B\) is also
positive definite.

\section*{ARGUMENTS}

ITYPE (input)
Specifies the problem type to be solved:
\(=1: A{ }^{*} \mathrm{X}=(\operatorname{lam} . \mathrm{bda}){ }^{\star} \mathrm{B}{ }^{*} \mathrm{X}\)
\(=2: A * B *_{x}=\left(\operatorname{lam}\right.\) bda) \({ }^{*} \mathrm{x}\)
\(=3: B{ }^{*} A{ }^{*} \mathrm{X}=(\operatorname{lam} \mathrm{bda}){ }^{*} \mathrm{X}\)

JOBZ (input)
\(=\mathrm{N}^{\prime}:\) C om pute eigenvalues only;
\(=\mathrm{V}\) ': C om pute eigenvalues and eigenvectors.

UPLO (input)
\(=U\) ': U ppertriangles of \(A\) and \(B\) are stored;
\(=\mathrm{L}\) ': Low ertriangles of \(A\) and \(B\) are stored.

N (input) The order of them atrioes A and \(\mathrm{B} . \mathrm{N}>=0\).

A (input/output)
On entry, the sym \(m\) etric \(m\) atrix A. If UPLO \(=U '\), the leading N -by N uppertriangularpart of A con-
tains the upper triangular part of the \(m\) atrix \(A\). If U PLO = L', the leading N -by N low er triangular partofA contains the low er triangular part of the m atrix \(A\).

On exit, if \(J O B Z=V^{\prime}\), then if \(\mathbb{N F O}=0\), \(A\) contains the \(m\) atrix \(Z\) ofeigenvectors. The eigenvectors are norm alized as follow s: if ITYPE \(=1\) or \(2, Z * * T * B * Z=I\); if \(T T Y P E=3, Z * * T * \operatorname{inv}(B) * Z=I\). If \(\mathrm{JOBZ}=\mathrm{N}\) ', then on exit the uppertriangle (if \(\mathrm{U} P L O=\mathrm{U}\) ) or the low er triangle (if \(\mathrm{U} P L O=\mathrm{L}\) ) of A , including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, \mathbb{N})\).

B (input/output)
O \(n\) entry, the sym \(m\) etric positive definite \(m\) atrix \(B\). If \(U P L O=U\) ', the leading \(N\) by \(N\) uppertriangularpartofB contains the upper triangular part of the m atrix B. IfU PLO = L', the leading N toy-N lowertriangularpart of B contains the low er triangularpart of the \(m\) atrix \(B\).

On exit, if \(\mathbb{N} F O<=N\), the part of \(B\) containing the \(m\) atrix is overw rilten by the triangular factor \(U\) orL from the Cholesky factorization \(B=U * * T * U\) or \(B=L *\) \(\mathrm{L}^{* *} \mathrm{~T}\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

W (output)
If \(\mathbb{N}\) FO \(=0\), the eigenvalues in ascending order.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length of the array \(W\) ORK. LDW ORK >= \(\mathrm{max}(1,3 \star \mathrm{~N}-1)\). For optim alefficiency, LDW ORK >= \((N B+2) * N\), where \(N B\) is the blocksize for SSY TRD retumed by ILAENV .

If LD W ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim al size of the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue
>0: SPOTRF orSSYEV retumed an emorcode:
\(<=\mathrm{N}\) : if \(\mathbb{N} F \mathrm{O}=\mathrm{i}, \mathrm{SSYEV}\) failed to converge; i off-diagonal elements of an interm ediate tridiagonal form did not converge to zero; > N : if \(\mathbb{N} F O=N+i\), for \(1<=i<=N\), then the leading m inoroforderiofB is not positive definite. The factorization ofB could notbe com pleted and no eigenvalues oreigenvectors w ere com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
```

dsygvd -com pute all the eigenvalues, and optionally, the
eigenvectors of a realgeneralized sym m etric-definite eigen-
problem, of the form A *x= (lam bda)*B *x, A *B x= (lam bda)*x, or
B *A *X= (lam boda) *}\textrm{X

```

\section*{SYNOPSIS}
```

SU BROUTINE DSYGVD (ITYPE,NOBZ,UPLO,N,A,LDA,B,LDB,W,W ORK,
LW ORK,\mathbb{IN ORK,LIN ORK,INFO)}
CHARACTER * 1 JOBZ,UPLO

```

```

INTEGER 䟡ORK (*)
DOUBLE PRECISION A (LDA ,*),B (LDB,*),W (*),W ORK (*)
SUBROUT\mathbb{NE DSYGVD_64(TTYPE,JOBZ,UPLO,N,A,LDA,B,LDB,W,W ORK,}

```

CHARACTER * 1 JOBZ, UPLO
\(\mathbb{N} T E G E R * 8 \mathbb{I T} Y P E, N, L D A, L D B, L W O R K, L \mathbb{I W} O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{I N}\) ORK ( \(\left.{ }^{( }\right)\)
DOUBLE PRECISION A (LDA ,*), B (LDB,*),W (*),WORK (*)

\section*{F95 INTERFACE}

SU BROUTINE SYGVD (TTYPE, JOBZ, UPLO, \(\mathbb{N}], A,[L D A], B,[L D B], W,[W O R K]\), [LW ORK], [ \(\mathbb{W}\) ORK], [LIN ORK ], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1): : JOBZ, UPLO
\(\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D A, L D B, L W\) ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ,W ORK

REAL (8), D \(\mathbb{M}\) ENSION (: : : : : A, B
SU BROUTINE SYGVD_64 (TTYPE, JOBZ,UPLO, \(\mathbb{N}], A,[L D A], B,[L D B], W\), [W ORK ], [LW ORK], [IW ORK ], [LIN ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1)::JOBZ,UPLO
\(\mathbb{N} T E G E R(8)::\) ITYPE,N,LDA, LDB,LW ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
REAL (8), D IM ENSION (:,:) ::A , B

\section*{C INTERFACE}
\#include <sunperfh>
void dsygvd (int itype, char jobz, charuplo, int n, double
*a, int lda, double *b, int ldb, double *w , int
*info);
void dsygvd_64 (long itype, char jobz, char uplo, long n, double *a, long lda, double *b, long ldb, double *W, long *info);

\section*{PURPOSE}
dsygvd com putes all the eigenvalues, and optionally, the eigenvectors of a realgeneralized sym \(m\) etric-definite eigenproblem, of the form \(A * x=(\operatorname{lam} . b d a) * B * x, A * B x=(\operatorname{lam} . b d a){ }^{*} x\), or \(B *^{A} *^{x}=\left(\operatorname{lam}\right.\) bda) \({ }^{*} x\). Here \(A\) and \(B\) are assum ed to be sym \(m\) etric and \(B\) is also positive definite. If eigenvectors are desired, it uses a divide and conquer algorithm .

The divide and conquer algorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on \(m\) achines w th a guard digit in add/subtract, or on those binary \(m\) achines \(w\) thout guard digits w hich subtract like the \(C\) ray \(X-M P, C\) ray \(Y \neq M P, C\) ray C-90, orC ray-2. It could conceivably fail on hexadecim al or decim al machines w thout guard digits, butw e know of none.

\section*{ARGUMENTS}

ITYPE (input)
Specifies the problem type to be solved:
\(=1: A{ }^{*} \mathrm{x}=\left(\operatorname{lam}\right.\) bda)\({ }^{\mathrm{B}} \mathrm{B}^{*} \mathrm{x}\)
\(=2: \mathrm{A} * \mathrm{~B} * \mathrm{x}=(\operatorname{lam} \mathrm{bda}){ }^{*} \mathrm{x}\)
\(=3: B * A * x=\left(l a m\right.\) bda) \({ }^{*} \mathrm{X}\)

JOBZ (input)
\(=N^{\prime}:\) C om pute eigenvalues only;
\(=\mathrm{V}\) : C om pute eigenvalues and eigenvectors.

UPLO (input)
\(=\mathrm{U}\) ': U ppertriangles of \(A\) and \(B\) are stored;
\(=\mathrm{L}\) ': Low ertriangles of \(A\) and \(B\) are stored.
N (input) The order of the m atrices A and \(\mathrm{B} . \mathrm{N}>=0\).
A (input/output)
On entry, the sym m etric \(m\) atrix \(A\). If \(U P L O=U '\), the leading \(\mathrm{N}-\) by -N uppertriangularpartofA contains the upper triangular part of the \(m\) atrix \(A\). If UPLO = L', the leading N by -N low er triangular partofA contains the low er triangular part of the \(m\) atrix \(A\).

On exit, if \(\mathrm{JOBZ}=\mathrm{V}\) ', then if \(\mathbb{N F O}=0\), A contains the \(m\) atrix \(Z\) ofeigenvectors. The eigenvectors are norm alized as follow s: if ITYPE = 1 or 2, \(\mathrm{Z} * * \mathrm{~T} * \mathrm{~B} * \mathrm{Z}=\mathrm{I}\); if \(\mathrm{IT} Y \mathrm{PE}=3, \mathrm{Z} * * \mathrm{~T} * \operatorname{inv}(\mathrm{~B}) * \mathrm{Z}=\mathrm{I}\). If \(\mathrm{JOBZ}=\mathrm{N}\) ', then on exit the upper triangle (if \(\mathrm{UPLO}=\mathrm{U}\) ) or the low er triangle (if \(\mathrm{UPLO}=\mathrm{L}\) ) of A, including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, \mathbb{N})\).

B (input/output)
On entry, the symm etric \(m\) atrix \(B\). If \(\operatorname{PLO}=U\) ', the leading N -by N uppertriangularpartofB contains the upper triangular part of the \(m\) atrix \(B\). If U PLO = L', the leading N by -N low er triangular partofB contains the low er triangular part of the \(m\) atrix \(B\).

On exit, if \(\mathbb{N} F O<=N\), the part of \(B\) containing the \(m\) atrix is overw rilten by the triangular factor \(U\) orL from the C holesky factorization \(B=U * * T * U\) orB \(=\mathrm{L}\) * \(\mathrm{L} * * \mathrm{~T}\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

W (output)
If \(\mathbb{N} F O=0\), the eigenvalues in ascending order.

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. If \(N<=1\), LW ORK \(>=1\). If \(\mathrm{OBZ}=\mathrm{N}^{\prime}\) and \(\mathrm{N}>1\), LW ORK \(>=\) \(2 \star \mathrm{~N}+1\). If \(\mathrm{OBZ}=\mathrm{V}^{\prime}\) and \(\mathrm{N}>1\),LW ORK \(>=1+6 * \mathrm{~N}\) \(+2 \star \mathrm{~N} * * 2\) 。

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IW ORK (w orkspace/output)
On exit, if \(\mathbb{N} F O=0, \mathbb{I N}\) ORK (1) retums the optim al LIW ORK.

LIV ORK (input)
The dim ension of the amay \(\mathbb{I W} O R K\). If \(\mathrm{N}<=1\), LIW ORK >=1. If \(O B Z=N\) 'andN \(>1\),LIW ORK \(>=\) 1. If \(\mathrm{OBZ}=\mathrm{V}\) 'and \(\mathrm{N}>1\), LIN \(\mathrm{ORK}>=3+5 \star \mathrm{~N}\).

IfLIW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the \(\mathbb{I W}\) ORK array, retums this value as the first entry of the \(\mathbb{I W}\) ORK aray, and no errorm essage related to \(L \mathbb{I} W\) ORK is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N}\) FO \(=-i\), the \(i\) th argum ent had an illegalvalue
> 0: SPO TRF orSSY EVD retumed an errorcode:
\(<=\mathrm{N}:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{SSYEVD}\) failed to converge; i offf-diagonalelem ents of an interm ediate tridiagonal form did notconverge to zero; > N : if \(\mathbb{N N F O}\) \(=N+i\), for \(1<=i<=N\), then the leading \(m\) inor oforderiofB is not positive definite. The factorization of \(B\) could notbe com pleted and no eigenvalues or eigenvectors w ere com puted.

\section*{FURTHER DETAILS}

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv. of K entucky, USA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
```

dsygvx - com pute selected eigenvalues, and optionally,
eigenvectors of a realgeneralized symm etric-definite eigen-
problem, of the form A *x= (lam bda)*B *x, A *B x= (lam bda)*x, or
B *A *X= (lam boda) *}\textrm{X

```

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE DSYGVX (TTYPE,JOBZ,RANGE,UPLO,N,A,LDA,B,LDB,VL,}
VU,\mathbb{I},\mathbb{U},ABSTOL,M,W,Z,LD Z,W ORK,LW ORK,IN ORK,\mathbb{FA}|,
\mathbb{NFO)}
CHARACTER * 1 JOBZ,RANGE,UPLO
\mathbb{NTEGER ITYPE,N,LDA,LDB,\mathbb{L},\mathbb{U},M,LD Z,LW ORK,\mathbb{NFO}}\mathbf{M}\mathrm{ , L',}
INTEGER IN ORK (*),\mathbb{FA}|(*)
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION A (LDA,*),B (LDB,*),W (*),Z (LD Z,*),W ORK (*)
SUBROUT\mathbb{NE DSYGVX_64 (TYYPE,JOBZ,RANGE,UPLO,N,A,LDA,B,LDB,VL,}
VU,\mathbb{L},\mathbb{U},ABSTOL,M,W,Z,LD Z,W ORK,LW ORK,IN ORK,\mathbb{FA}\mathbb{L},
\mathbb{NFO)}

```
CHARACTER * 1 JOBZ,RANGE,UPLO
\(\mathbb{N} T E G E R * 8 \mathbb{T} Y P E, N, L D A, L D B, \mathbb{L}, \mathbb{U}, M, L D Z, L W\) ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{I N} O R K(*), \mathbb{F A} \mathbb{I}(*)\)
DOUBLE PRECISION VL,VU, ABSTOL
DOUBLE PRECISION A (LDA, *), B (LDB, *), W (*), Z (LD Z, *), W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE SYGVX (TTYPE, JOBZ,RANGE,UPLO, \(\mathbb{N}], A,[L D A], B,[L D B]\),
 \(\mathbb{F A} \mathbb{L},[\mathbb{N F O}])\)

CHARACTER (LEN=1) :: OBZ,RANGE, UPLO
\(\mathbb{N} T E G E R:: \mathbb{I T}\) YE, \(N, L D A, L D B, \mathbb{L}, \mathbb{U}, M, L D Z, L W O R K, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{I M} E N S I O N(:):: \mathbb{I W} O R K, \mathbb{F A} \mathbb{I}\)
REAL (8) :: VL,VU,ABSTOL
REAL (8), D IM ENSION (:) ::W ,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (: : : ) : : A , B , Z

SU BROU T INE SY GVX_64 (TTYPE, JOBZ,RANGE,UPLO, \(\mathbb{N}], A,[L D A], B,[L D B]\), \(\mathrm{VL}, \mathrm{VU}, \mathbb{I}, \mathbb{I}, \mathrm{ABSTOL}, \mathrm{M}, \mathrm{W}, \mathrm{Z},[\mathrm{LD} \mathrm{Z}],[\mathrm{W} O R K],[\mathrm{LW} O R K],[\mathbb{I W} O R K]\), \(\mathbb{F A} \mathbb{I},[\mathbb{N F O}])\)

CHARACTER (LEN=1) :: OBZ,RANGE,UPLO
\(\mathbb{N}\) TEGER (8) :: ITYPE,N,LDA,LDB, II, \(\mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W O R K\),
\(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{I N O R K}\), \(\mathbb{F} A \mathbb{I}\)
REAL (8) ::VL,VU,ABSTOL
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A , B , Z

\section*{C INTERFACE}
\#include <sunperfh>
void dsygvx (int itype, char jंbz, char range, charuplo, int n , double *a, int lda, double *b, int ldb, double vl , double vu, int il, intiu, double abstol, int
*m , double *w , double * z , int \(1 d \mathrm{z}\), int *ifail, int *info);
void dsygvx_64 (long type, char jंbz, char range, charuple, long n, double *a, long lda, double *b, long ldb, double vl, double vu, long il, long iu, double abstol, long *m, double *w, double *z, long ldz, long *ifail, long *info);

\section*{PURPOSE}
dsygvx com putes selected eigenvalues, and optionally , eigenvectors of a realgeneralized sym \(m\) etric-definite eigenproblem, of the form \(A * x=\left(l a m\right.\) bda) \({ }^{*} B * x, A * B x=(l a m \text { bda })^{*} x\), or \(B{ }^{\star} A{ }^{*} X=(\operatorname{lam} \text { bda })^{*} x\). \(H\) ere \(A\) and \(B\) are assum ed to be sym \(m\) etric and \(B\) is also posilive definite. Eigenvalues and eigenvectors can be selected by specifying either a range of values or range of indices forthe desired eigenvalues.

\section*{ARGUMENTS}

\section*{ITYPE (input)}

Specifies the problem type to be solved:
\(=1: A{ }^{*} \mathrm{X}=(\operatorname{lam} \mathrm{bda}){ }^{\mathrm{B}}{ }^{*} \mathrm{X}\)
\(=2: \mathrm{A} * \mathrm{~B} * \mathrm{X}=\left(\operatorname{lam}\right.\) bda) \({ }^{*} \mathrm{x}\)
\(=3: B * A * x=\left(\operatorname{lam}\right.\) bda) \({ }^{*} \mathrm{x}\)
JOBZ (input)
\(=\mathrm{N}^{\prime}\) : C om pute eigenvalues only;
\(=\mathrm{V}:\) : C om pute eigenvalues and eigenvectors.
RANGE (input)
= A ': alleigenvalues w ill.be found.
= V : alleigenvahues in the half-open interval ( \(\mathrm{L}, \mathrm{V}, \mathrm{J}]\) will be found. = I ': the I -th through \(\mathbb{I U}\)-th eigenvaluesw illlbe found.

UPLO (input)
= U ': U ppertriangle of A and B are stored;
= L': Low ertriangle of A and B are stored.
N (input) The order of the \(m\) atrix pencil \((A, B) . N>=0\).
A (input/output)
On entry, the symm etric m atrix \(A\). If UPLO = U', the leading N -by -N uppertriangularpart of A contains the upper triangular part of the \(m\) atrix \(A\). If UPLO \(=\mathrm{L}\) ', the leading N -by N low er triangular partofA contains the low er triangular part of them atrix A.

On exit, the low er triangle (ifU PLO = L) or the upper triangle (if \(\mathrm{PLO}=\mathrm{U}\) ) ofA, including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

B (input/output)
On entry, the symm etric \(m\) atrix \(B\). If \(U P L O=U\) ', the leading N -by N uppertriangular partofB contains the upper triangularpart of the \(m\) atrix \(B\). If UPLO = L', the leading N -by -N low er triangular partofB contains the low er triangular part of the matrix B .

On exit, if \(\mathbb{N F O}<=N\), the part of \(B\) containing the \(m\) atrix is overw rilten by the triangular factor U orL from the Cholesky factorization \(\mathrm{B}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(B=L{ }^{\star} L^{\star *}{ }^{\mathrm{T}}\).

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. \(\mathrm{VL}<\mathrm{VU}\). N ot referenced ifRANGE=A 'or 'I'.

VU (input)
See the description ofV L .
IU (input)
IfRA NGE= I', the indiges (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{Z}<=\mathbb{U}<=\mathrm{N}\), if \(\mathrm{N}>0\); \(\mathbb{U}=1\) and \(\mathbb{U}=0\) if \(N=0\). \(N\) otreferenced ifRANGE \(=\) A'or V'.

IU (input)
See the description of II.

ABSTOL (input)
The absolute emortolerance forthe eigenvalues.
A \(n\) approxim ate eigenvalue is accepted as converged \(w\) hen it is determ ined to lie in an interval [a,b] of w idth less than orequal to
\(A B S T O L+E P S * \max (|a|,||\),\() ,\)
where EPS is them achine precision. IfA BSTOL is less than or equal to zero, then EPS*|I w illbe used in its place, w here \(T\) | is the 1 -norm of the tridiagonal \(m\) atrix obtained by reducing \(A\) to tridiagonalform.

E igenvahues w ill.be com puted m ostaccurately w hen ABSTOL is set to tw ige the underflow threshold 2*D LAM CH (S ), notzero. Ifthis routine retums w ith \(\mathbb{N} \mathrm{FO}>0\), indicating thatsom e eigenvectors did not converge, try setting ABSTOL to \(2 *\) SLAM CH (S ).

M (output)
The totalnum ber ofeigenvalues found. \(0<=\mathrm{M}<=\) N . IfRANGE=A', M=N, and ifRANGE= 'I'M = \(\mathbb{U}-\mathbb{L}+1\) 。

W (output)
O n norm alexit, the firstM elem ents contain the selected eigenvahues in ascending order.
\(Z\) (input) If \(\mathrm{OBB}=\mathrm{N}^{\prime}\), then Z is notreferenced. If OBZ \(=V\) ', then if \(\mathbb{N} F O=0\), the firstM colum ns of \(Z\) contain the orthonorm aleigenvectors of the \(m\) atrix A corresponding to the selected eigenvalues, w ith the i-th colum \(n\) of \(Z\) holding the eigenvector associated w ith W (i). The eigenvectors are norm alized as follow s: if ITYPE \(=1\) or \(2, Z * * T * B * Z=I\); if \(\operatorname{ITYPE}=3, Z * * T * \operatorname{inv}(B) * Z=I\).
If an eigenvector fails to converge, then that colum \(n\) of \(Z\) contains the latestapproxim ation to the eigenvector, and the index of the eigenvector is retumed in \(\mathbb{F} A \mathbb{I}\). N ote: the userm ustensure that at leastm ax ( \(1, M\) ) colum ns are supplied in the amay \(Z\); ifRANGE = V', the exactvalue ofM is notknow \(n\) in advance and an upper bound m ust be used.

LD Z (input)
The leading dim ension of the array \(\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1\), and if \(\mathrm{OBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{max}(1, N)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK ( 1 ) retums the optim al LW ORK.

\section*{LW ORK (input)}

The length of the array \(W\) ORK. LW ORK >= \(\mathrm{max}(1,8 * \mathrm{~N})\). For optim al efficiency, LW ORK >= \((\mathbb{N B}+3) \star \mathrm{N}\), where NB is the blocksize for SSY TRD retumed by \(\amalg \mathrm{LA} E \mathrm{EN}\).

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IW ORK (w orkspace)
dim ension ( \(5 \star \mathrm{~N}\) )

FAII (output)
If \(O B Z=V^{\prime}\), then if \(\mathbb{N F O}=0\), the firstM ele\(m\) ents of \(\mathbb{F} A \mathbb{I}\) are zero. If \(\mathbb{N} F O>0\), then \(\mathbb{F} A \mathbb{H}\) contains the indices of the eigenvectors that failed to converge. If \(\mathrm{OOBZ}=\mathrm{N}\) ', then \(\mathbb{F} A \mathbb{I}\) is notreferenced.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
<0: if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an illegalvahue
> 0: SPO TRF orSSY EVX retumed an errorcode:
\(<=N:\) if \(\mathbb{N} F O=i, S S Y E V X\) failed to converge; i eigenvectors failed to converge. Their indices are stored in array \(\mathbb{F A} \mathbb{I} .>N\) : if \(\mathbb{N} F O=N+\) \(i\), for \(1<=i<=N\), then the leading \(m\) inor of orderiofB is notpositive definite. The factorization of B could not be com pleted and no eigenvalues or eigenvectors w ere com puted.

\section*{FURTHER DETAILS}

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv. of K entucky, USA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsymm -perform one of the \(m\) atrix-m atrix operations \(C:=\) alpha*A *B + beta*C orC := alpha*B*A + beta*C

\section*{SYNOPSIS}
```

SU BROUTINEDSYMM(SDE,UPLO,M,N,ALPHA,A,LDA,B,LDB,BETA,C,
LD C )
CHARACTER * 1SDEE,UPLO
INTEGERM,N,LDA,LDB,LDC
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION A (LDA,*),B (LDB,*),C (LDC ,*)
SU BROUTINE D SYMM _64 (S\mathbb{D E,UPLO ,M ,N,ALPHA,A ,LDA,B,LD B,BETA,C,}
LD C)
CHARACTER * 1 SIDE,UPLO
INTEGER*8M,N,LDA,LDB,LDC
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION A (LDA,*),B (LDB,*),C (LDC ,*)

```

\section*{F95 INTERFACE}

SU BROUTINE SYMM (SDE, UPLO, \(\mathbb{M}\) ], \(\mathbb{N}], A L P H A, A,[L D A], B,[L D B]\), BETA, C, [LDC])

CHARACTER (LEN=1) ::SDE,UPLO
\(\mathbb{N} T E G E R:: M, N, L D A, L D B, L D C\)
REAL (8) ::ALPHA,BETA
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A, B , C
SUBROUTINE SYMM_64 (SDE,UPLO, \(\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A], B,[L D B]\), BETA, C, [LDC])

CHARACTER ( \(几 E N=1\) ) : : SDE E , UPLO
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{LD} A, L D B, L D C\)
REAL (8) ::ALPHA, BETA
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A, B, C

\section*{C INTERFACE}
\#include <sunperfh>
void dsym m (char side, char uplo, intm , intn, double alpha, double *a, int lda, double *b, intldlo, double beta, double *c, int ldc);
void dsym m _64 (charside, charuplo, long m, long n, double alpha, double *a, long lda, double *b, long ldb, double beta, double *c, long ldc);

\section*{PURPOSE}
dsymm perform sone of the \(m\) atrix \(m\) atrix operations \(C:=\) alpha*A *B + beta* C or \(\mathrm{C}:=\) alpha*B *A + beta* C where alpha and beta are scalars, \(A\) is a sym \(m\) etric \(m\) atrix and \(B\) and \(C\) are \(m\) by \(n m\) atrices.

\section*{ARGUMENTS}

SID E (input)
O n entry, SIDE specifiesw hether the sym metric \(m\) atrix A appears on the leftorright in the operation as follow s:
\(S \mathbb{D E}=\) L'or I' \(C:=\) a耳pha*A *B + beta* \(C\),

SIDE = R 'or 'r' C : alpha*B *A + beta*C ,

U nchanged on exit.

\section*{UPLO (input)}

On entry, UPLO specifies whether the upper or lower triangular part of the symm etric \(m\) atrix \(A\) is to be referenced as follow \(s\) :
\(\mathrm{UPLO}=\mathrm{U}\) 'or L ' Only the upper triangularpart of the sym \(m\) etric \(m\) atrix is to be referenced.
\(\mathrm{UPLO}=\mathrm{L}\) 'or \(\mathrm{I}^{\prime}\) ' O nly the low ertriangularpart of the sym \(m\) etric \(m\) atrix is to be referenced.

U nchanged on exit.
M (input)
O \(n\) entry, M specifies the num ber of row sof the \(m\) atrix \(C . M\) M \(=0\). U nchanged on exit.

N (input)
O n entry, N specifies the num ber of colum ns of the \(m\) atrix \(C . N>=0\). Unchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.
A (input)
DOUBLE PRECISION array ofD \(\mathbb{M}\) ENSION (LDA, ka ), where ka is \(m\) when \(S \mathbb{D} E=\mathbb{L}\) 'or \(I^{\prime}\) and is \(n\)
otherw ise.
Before entry with SDDE \(=\mathrm{L}\) 'or 1 ', the \(m\) by
m part of the aray A mustcontain the sym\(m\) etric \(m\) atrix, such thatw hen UPLO \(=U\) 'or 4 ', the leading \(m\) by \(m\) uppertriangular part of the array A mustcontain the upper triangular part of the symm etric \(m\) atrix and the strictly lower triangularpart of A is not referenced, and \(w\) hen UPLO = L' or \({ }^{1}\) ', the leading \(m\) by \(m\) low er triangularpart of the array A m ust contain the lower triangular part of the sym \(m\) etric \(m\) atrix and the strictly upper triangular partof \(A\) is not referenced.

Before entry with \(S D E=R\) 'or \(r\) ', the \(n\) by \(n\) part of the aray A mustcontain the sym\(m\) etric \(m\) atrix, such thatw hen UPLO \(=U\) 'or L ', the leading \(n\) by \(n\) uppertriangularpart of the aray A mustcontain the upper triangular part of the symm etric \(m\) atrix and the strictly lower triangularpart of \(A\) is not referenced, and when UPLO = L' or 1 ', the leading \(n\) by \(n\) low er triangularpart of the array A must contain the lower triangular part of the sym \(m\) etric \(m\) atrix and the strictly upper triangular part of A is not referenced.

U nchanged on exit.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen
\(S \mathbb{D} E=\mathbb{L}\) 'or 1 ' then LD \(A>=m a x(1, m)\), otherw ise LDA \(>=\mathrm{max}(1, \mathrm{n})\). U nchanged on exit.

B (input)
D OUBLE PREC ISION array ofD \(\mathbb{I M}\) ENSION (LDB, n ).
Before entry, the leading \(m\) by \(n\) partof the array \(B m\) ustcontain the \(m\) atrix \(B\). Unchanged on exit.

LD B (input)
On entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program. LD B \(>=\max (1, m)\). U nchanged on exit.
BETA (input)
O n entry, BETA specifies the scalar beta. W hen
BETA is supplied as zero then C need notbe set on input. U nchanged on exit.

C (input/output)
DOUBLE PRECISION anay ofD \(\mathbb{I M}\) ENSION (LDC, n ).
Before entry, the leading \(m\) by \(n\) partof the array \(\mathrm{C} m\) ust contain the m atrix C , except when beta is zero, in which case \(C\) need notbe set on entry. On exit, the array \(C\) is overw rilten by the \(m\) by \(n\) updated \(m\) atrix.

LD C (input)
On entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program. LD C \(>=m a x(1, m)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsym \(v\)-perform the \(m\) atrix-vectoroperation \(y:=a l p h a * A * x\)
+ beta*y

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSYMV (UPLO,N,ALPHA,A,LDA,X, INCX,BETA,Y, INCY)}
CHARACTER * 1 UPLO
INTEGERN,LDA, INCX,INCY
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION A (LDA ,*),X (*),Y (*)
SU BROUT\mathbb{NEDSYM V_64 (UPLO,N,ALPHA,A,LDA,X, INCX,BETA,Y,INCY)}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N,LDA,INCX,}\mathbb{N}CY
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION A (LDA,*),X (*),Y (*)

```

\section*{F95 INTERFACE}

SU BROUTINE SYM V (UPLO, \(\mathbb{N}], A L P H A, A,[L D A], X,[\mathbb{N C X}], B E T A, Y,[\mathbb{N C Y}])\)
CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathrm{LDA}, \mathbb{N} C X, \mathbb{N} C Y\)
REAL (8) ::ALPHA,BETA
REAL (8), D IM ENSION (:) :: X,Y
REAL (8),D IM ENSION (:,:) ::A
SU BROUTINE SYMV_64 (UPLO, \(\mathbb{N}], A L P H A, A,[L D A], X,[\mathbb{N C X}], B E T A, Y\), [ \(\mathbb{N} C Y\) ])

CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R(8):: N, L D A, \mathbb{N C X}, \mathbb{N} C Y\)
REAL (8) ::A LPHA,BETA
REAL (8),D IM ENSION (:) :: X,Y
REAL (8),D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dsym v (charuplo, intn, double alpha, double *a, int lda, double *x, int incx, double beta, double *y, int incy);
void dsym v_64 (charuplo, long n, double alpha, double *a, long lda, double *x, long incx, double beta, double *y, long incy);

\section*{PURPOSE}
dsym \(v\) perform \(s\) the \(m\) atrix-vector operation \(y:=a l p h a * A * x+\) beta* y , w here alpha and beta are scalars, x and y are n ele\(m\) ent vectors and \(A\) is an \(n\) by \(n\) sym \(m\) etric \(m\) atrix.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the array \(A\) is to be referenced as follow s:

U PLO = U 'or L ' Only the upper triangularpart ofA is to be referenced.

UPLO = L 'or I' O nly the low ertriangularpart ofA is to be referenced.

U nchanged on exit.

N (input)
On entry, N specifies the order of the m atrix A . \(\mathrm{N}>=0\). U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha.
U nchanged on exit.
A (input)
Before entry with UPLO = U 'or L ', the leading
n by n upper triangularpart of the array A m ust contain the upper triangular part of the sym \(m\) etric \(m\) atrix and the strictly low ertriangularpartofA is not referenced. Before entry w ith UPLO = L' or '1', the leading \(n\) by \(n\) low er triangularpart of the array A m ust contain the low er triangular part of the sym \(m\) etric \(m\) atrix and the strictly uppertriangularpart of \(A\) is not referenced. U nchanged on exit.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD A >= \(\max (1, n)\). U nchanged on exit.

X (input)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C X))\). Before entry, the increm ented array \(X\) must contain the \(n\) elem ent vectorx. U nchanged on exit.
\(\mathbb{N} C X\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX . \(\mathbb{N} C X<>0\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then Y need notbe seton input. U nchanged on exit.

Y (input/output)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))\). B efore entry, the increm ented array \(Y\) m ust contain the \(n\) elem ent vectory. On exit, \(Y\) is overw rilten by the updated vectory.
\(\mathbb{N C C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(. \mathbb{N} C Y<>0\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsyr- perform the symmetric rank 1 operation A := alpha*x*x'+A

\section*{SYNOPSIS}
```

SUBROUTINE DSYR(UPLO,N,ALPHA,X,INCX,A,LDA)
CHARACTER * 1 UPLO
\mathbb{NTEGERN,INCX,LDA}
DOUBLE PRECISION ALPHA
DOUBLE PRECISION X (*),A (LDA,*)
SU BROUT\mathbb{NEDSYR_64 (UPLO,N,ALPHA,X,INCX,A,LDA)}
CHARACTER * 1 UPLO
INTEGER*8N,}\mathbb{N}CX,LD
DOUBLE PRECISION ALPHA
DOUBLE PRECISION X (*),A (LDA,*)

```

\section*{F95 INTERFACE}

SU BROUTINE SYR (UPLO, \(\mathbb{N}], A L P H A, X,[\mathbb{N} C X], A,[L D A])\)

CHARACTER (LEN=1)::UPLO
\(\mathbb{N}\) TEGER ::N, \(\mathbb{N} C X, L D A\)
REAL (8) ::A LPHA
REAL (8),D \(\mathbb{I M}\) ENSION (:) ::X
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A

SU BROUTINE SYR_64 (UPLO, \(\mathbb{N}], A L P H A, X,[\mathbb{N C X}], A,[L D A])\)
CHARACTER (LEN=1) ::UPLO
\(\mathbb{N} T E G E R(8):: N, \mathbb{N C X}\), LDA

REAL (8) ::A LPHA
REAL (8),D \(\mathbb{M}\) ENSION (:) ::X
REAL (8),D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dsyr(charuplo, intn, double alpha, double *x, int incx, double *a, int lda);
void dsyr_64 (charuplo, long n, double alpha, double *x, long incx, double *a, long lda);

\section*{PURPOSE}
dsyrperform \(s\) the sym \(m\) etric rank 1 operation \(A:=a l p h a * x^{\star} x^{\prime}\)
\(+A\), where alpha is a realscalar, x is an n elem entvector and \(A\) is an \(n\) by \(n\) sym \(m\) etric \(m\) atrix.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the array \(A\) is to be referenced as follow s:

UPLO = U 'or L' Only the upper triangularpart ofA is to be referenced.

UPLO = L 'or I' O nly the low ertriangularpart ofA is to be referenced.

U nchanged on exit.

N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix \(A\). \(\mathrm{N}>=0\). U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
\((1+(n-1) * a b s(\mathbb{N} C X))\). Before entry, the increm ented aray \(X\) must contain the \(n\) elem ent vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX . \(\mathbb{N} C X<>0\). U nchanged on exit.

A (input/output)
Before entry w ith UPLO = U 'or 4 ', the leading \(n\) by \(n\) uppertriangularpart of the array A m ust contain the uppertriangularpart of the sym \(m\) etric \(m\) atrix and the strictly low er triangularpart of \(A\) is not referenced. O n exit, the upper triangular part of the array A is overw ritten by the upper triangularpart of the updated \(m\) atrix. Before entry w ith UPLO = L 'or I', the leading \(n\) by \(n\) low er triangularpart of the array A m ust contain the low er triangularpart of the sym \(m\) etric \(m\) atrix and the strictly upper triangularpart of A is not referenced. On exit, the low er triangularpart of the array \(A\) is overw rilten by the low er triangular part of the updated \(m\) atrix.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD \(A>=\) \(\max (1, n)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsyr2-perform the symmetric rank 2 operation \(A:=\) alpha*x*y'+ alpha*y*x'+ A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSYR2(UPLO,N,ALPHA,X, INCX,Y,INCY,A,LDA)}
CHARACTER * 1 UPLO
INTEGERN,INCX,\mathbb{NCY,LDA}
DOUBLE PRECISION ALPHA
DOUBLE PRECISION X (*),Y (*),A (LDA,*)
SU BROUT\mathbb{NE DSYR2_64 (UPLO,N,A LPHA,X, INCX,Y, INCY,A,LDA)}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N, NNCX,}\mathbb{NCY,LDA}
DOUBLE PRECISION ALPHA
D OUBLE PRECISION X (*),Y (*),A (LDA,*)

```

\section*{F95 INTERFACE}

SU BROUTINE SYR2 (UPLO, \(\mathbb{N}], A L P H A, X,[\mathbb{N C X}], Y,[\mathbb{N C Y}], A,[L D A])\)
CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R:: N, \mathbb{N C X}, \mathbb{N} C Y, L D A\)
REAL (8) ::A LPHA
REAL (8), D IM ENSION (:) ::X,Y
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A
SU BROUTINE SYR2_64 (UPLO, \(\mathbb{N}], A L P H A, X,[\mathbb{N C X}], Y,[\mathbb{N C Y}], A,[L D A])\)
CHARACTER (LEN=1) ::UPLO
\(\mathbb{N} T E G E R(8):: N, \mathbb{N} C X, \mathbb{N} C Y, L D A\)

REAL (8) ::A LPHA
REAL (8),D IM ENSION (:) :: X,Y
REAL (8),D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dsyr2 (charuplo, intn, double alpha, double *x, int incx, double *y, int incy, double *a, intlda);
void dsyn2_64 (charupl,, long n, double alpha, double *x, long incx, double *y, long incy, double *a, long lda);

\section*{PURPOSE}
dsyr2 performs the symmetric rank 2 operation \(\mathrm{A}:=\) alpha*x*y' + alpha*y*x'+A, where alpha is a scalar, \(x\) and \(y\) are \(n\) elem ent vectors and \(A\) is an \(n\) by \(n\) sym \(m\) etric \(m\) atrix.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the array \(A\) is to be referenced as follow s:

UPLO = U 'or L' Only the upper triangularpart ofA is to be referenced.

UPLO = L'or I' O nly the low ertriangularpart ofA is to be referenced.

U nchanged on exit.
N (input)
O n entry, N specifies the order of the m atrix A . \(\mathrm{N}>=0\). U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
\((1+(n-1) * a b s(\mathbb{N} C X))\). Before entry, the increm ented array \(X\) must contain the \(n\) elem ent vectorx. Unchanged on exit.
\(\mathbb{I N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X<>0\). U nchanged on exit.

Y (input)
\((1+(n-1) * a b s(\mathbb{N} C Y))\). Before entry, the increm ented aray \(Y \mathrm{~m}\) ust contain the n elem ent vectory. U nchanged on exit.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents ofY. \(\mathbb{N} C Y\) <> 0 . U nchanged on exit.
A (input/output)
Before entry w ith UPLO = U 'or L ', the leading \(n\) by \(n\) upper triangular part of the array A m ust contain the upper triangular part of the sym \(m\) etric \(m\) atrix and the strictly low er triangularpartofA is not referenced. On exit, the upper triangular part of the array A is overw rilten by the upper triangularpart of the updated \(m\) atrix. Before entry with UPLO = L'or I', the leading \(n\) by \(n\) low er triangularpart of the array A m ust contain the low er triangularpart of the sym \(m\) etric \(m\) atrix and the strictly uppertriangularpartofA is not referenced. On exit, the low er triangularpart of the array \(A\) is overw ritten by the low er triangular part of the updated \(m\) atrix.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A >= \(m a x(1, n)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsyr \(2 k\)-perform one of the sym \(m\) etric rank \(2 k\) operations \(C\) \(:=\) alpha*A *B' + alpha*B*A ' + beta*C orC \(:=\) alpha*A *B + alpha*B *A + beta*C

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE DSYR2K (UPLO,TRANSA,N,K,ALPHA,A,LDA,B,LDB,BETA,C,}
LD C)
CHARACTER * 1 UPLO,TRANSA
INTEGERN,K,LDA,LDB,LDC
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION A (LDA,*),B (LDB,*),C (LDC ,*)
SUBROUT\mathbb{NEDSYR2K_64 (UPLO,TRANSA,N,K,ALPHA,A,LDA,B,LDB,BETA,}
C,LDC)

```
CHARACTER * 1 UPLO, TRANSA
\(\mathbb{N}\) TEGER* \(8 \mathrm{~N}, \mathrm{~K}, \operatorname{LDA}\), LD B , LDC
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION A (LDA, \()^{*}\), B (LDB,*), C (LDC ,*)

\section*{F95 INTERFACE}

SU BROUTINE SYR2K ©PLO, [TRANSA], \(\mathbb{N}],[K], A L P H A, A,[L D A], B,[L D B]\), BETA, C, [LDC])

CHARACTER (LEN=1) ::UPLO,TRANSA
\(\mathbb{N} T E G E R:: N, K, L D A, L D B, L D C\)
REAL (8) ::ALPHA,BETA
REAL (8), D IM ENSION (:,:) ::A,B,C
SU BROUTINE SYR2K_64 (UPLO, [TRANSA], \(\mathbb{N}],[K], A L P H A, A,[L D A], B\),
[LD B],BETA, C, [LD C ])

CHARACTER (LEN=1) ::UPLO,TRANSA
\(\mathbb{N} T E G E R(8):: N, K, L D A, L D B, L D C\)
REAL (8) ::ALPHA,BETA
REAL (8), D \(\mathbb{M}\) ENSION (: : : : ::A, B, C

\section*{C INTERFACE}
\#include <sunperfh>
void dsyr2k (charuplo, chartransa, int \(n\), int \(k\), double alpha, double *a, int lda, double *b, int ldb, double beta, double *c, int ldc);
void dsy_2k_64 (charuplo, char transa, long n, long k, double alpha, double *a, long lda, double *b, long ldb, double beta, double *C, long ldc);

\section*{PURPOSE}
dsy 22 k perform s one of the sym \(m\) etric rank 2 k operations \(\mathrm{C}:=\) alpha*A *B' + alpha*B*A'+ beta*C or C : alpha*A *B + alpha*B *A + beta*C w here alpha and beta are scalars, C is an \(n\) by \(n\) symmetric \(m\) atrix and \(A\) and \(B\) are \(n\) by \(k\) \(m\) atrices in the first case and \(k\) by \(n m\) atrices in the second case.

\section*{ARGUMENTS}

\section*{UPLO (input)}

On entry, UPLO specifies whether the upper
or lower triangular part of the array \(C\) is
to be referenced as follow s:

UPLO = U'or L' Only the upper triangular partof \(C\) is to be referenced.

UPLO = L'or I' Only the lower triangular partof C is to be referenced.

U nchanged on exit.
TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=\mathrm{N}^{\prime}\) or \(\mathrm{h}^{\prime} \mathrm{C}:=\) alpha*A *B '+ alpha*B*A '
+ beta*C.

TRANSA = T'ort' \(\mathrm{C}:=\) alpha*A *B + alpha*B *A + beta*C.

TRANSA \(=\) C 'or C' \(C:=\) alpha*A *B + alpha*B *A + beta*C .

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

N (input)
O n entry, \(N\) specifies the order of the \(m\) atrix C. N m ust.be at least zero. U nchanged on exit.
\(K\) (input)
On entry w ith TRANSA = N 'or h', K specifies the num ber of colum ns of the \(m\) atrioes \(A\) and \(B\), and on entry \(w\) ith TRANSA \(=T\) 'or t'or \(C^{\prime}\) or \(k^{\prime}, \mathrm{K}\) specifies the num ber of row sof the \(m\) atrices \(A\) and \(B\). \(K\) must be at least zero. U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha.
U nchanged on exit.

A (input)
D OUBLE PRECISION aray ofD \(\mathbb{M}\) ENSION (LDA, ka ), where ka isk when TRANSA = N 'or h ', and is
n otherw ise. Before entry w ith \(\mathrm{TRANSA}=\mathrm{N}^{\prime}\) or
h', the leading \(n\) by k partof the amay A
m ustcontain the m atrix \(A\), otherw ise the leading
k by n partof the aray \(A\) mustcontain the \(m\) atrix A. U nchanged on exit.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program.
W hen TRANSA \(=N\) 'or h'then LDA must be at least \(\max (1, n)\), otherw ise LD A m ustbe at least \(\max (1, k)\). U nchanged on exit.

B (input)
D OUBLE PRECISION aray ofD \(\mathbb{M} E N S I O N(L D B, k b)\), where kb isk when TRANSA = N 'or h', and is
n otherw ise. Before entry w ith TRANSA \(=\mathrm{N}^{\prime}\) or
\(h\) ', the leading \(n\) by k part of the anray \(B\) \(m\) ust contain the \(m\) atrix \(B\), otherw ise the leading k by n part of the aray \(B \mathrm{~m}\) ustcontain the \(m\) atrix B. U nchanged on exit.

LD B (input)
O n entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program.
\(W\) hen TRANSA \(=N\) 'or \(h\) 'then LDB must be at least \(\max (1, n)\), otherw ise LD B m ustbe at least \(\max (1, k)\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. U nchanged on exit.

C (input/output)
D OUBLE PRECISION aray ofD \(\mathbb{M} E N S I O N \quad(L D C, n)\).

Before entry with UPLO = U 'or G ', the leading \(n\) by \(n\) upper triangularpart of the array \(C\) \(m\) ustcontain the upper triangular part of the sym \(m\) etric \(m\) atrix and the strictly low er triangularpartofC is not referenced. On exit, the upper triangularpart of the array \(C\) is overw ritten by the upper triangularpart of the updated \(m\) atrix.

Before entry w ith UPLO = L 'or I', the leading \(n\) by \(n\) low er triangular part of the array \(C\) \(m\) ustcontain the low er triangular part of the sym \(m\) etric \(m\) atrix and the strictly uppertriangularpartof C is not referenced. On exit, the low er triangularpart of the array \(C\) is overw ritten by the low er triangularpart of the updated \(m\) atrix.

LD C (input)
O n entry, LD C specifies the first dim ension of C as declared in the calling (sub) program.
LD C m ust be at leastm ax ( \(1, \mathrm{n}\) ). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsyrfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is sym \(m\) etric indefintite, and provides emorbounds and backw ard error estim ates for the solution

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DSYRFS (UPLO,N,NRHS,A,LDA,AF,LDAF, \mathbb{P IVOT,B,LDB,X,}}\mathbf{N},\textrm{L}
LDX,FERR,BERR,W ORK,W ORK 2, INFO)
CHARACTER * 1 UPLO
INTEGERN,NRHS,LDA,LDAF,LDB,LDX,}\mathbb{N}F
INTEGER \mathbb{PIVOT (*),W ORK2(*)}
DOUBLE PRECISION A (LDA ,*),AF (LDAF,*), B (LDB,*), X (LDX,*),
FERR (*),BERR (*),W ORK (*)
SU BROUT\mathbb{NE DSYRFS_64 (UPLO,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,B,LDB,}}\mathbf{N},\textrm{L}
X,LDX,FERR,BERR,W ORK,W ORK 2,INFO)

```
CHARACTER * 1 UPLO
\(\mathbb{N}\) TEGER*8N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER * \(8 \mathbb{P} \mathbb{I V O T}(*), W\) ORK 2 ( )
D OUBLE PRECISION A (LDA, *), AF (LDAF,*), B (LDB,\(\star), ~ X(L D X, *)\),
FERR (*), BERR (*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUT INE SYRFS (UPLO,N,NRHS,A, [LDA],AF, [LDAF], \(\mathbb{P} I V O T, B,[L D B]\), X, [LDX],FERR,BERR, [W ORK], [W ORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V O T}, \mathrm{W} O R K 2\)

REAL (8), D \(\mathbb{I}\) ENSION (:) :: FERR,BERR,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (: :) : : A , AF, B , X

SU BROUT \(\mathbb{N} E S Y R F S \_64(U P L O, N, N R H S, A,[L D A], A F,[L D A F], \mathbb{P} \mathbb{I} O T, B\), \([\) [LD \(], X,[\llbracket D X], F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N F O}])\)

CHARACTER (LEN=1) :: UPLO
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T\), W ORK 2
REAL (8), D \(\mathbb{M} E N S I O N(:):: F E R R, B E R R, W\) ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) :: A , AF, B, X

\section*{C INTERFACE}
\#include <sunperfh>
void dsyrfs (char uplo, intn, intnrhs, double *a, int lda, double *af, int ldaf, int *ipivot, double *b, int ldb, double *x, int ldx, double *ferr, double *berr, int *info);
void dsyrfs_64 (charuplo, long n, long nrhs, double *a, long lda, double *af, long ldaf, long *ípivot, double
*b, long ldb, double *x, long ldx, double *ferr, double *berr, long *info);

\section*{PURPOSE}
dsyrfs im proves the com puted solution to a system of linear equations w hen the coefficientm atrix is sym \(m\) etric indefinte, and provides emrorbounds and backw ard error estim ates forthe solution.

\section*{ARGUMENTS}

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.
N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the m atrices \(B\) and X. NRH \(S>=0\).

A (input) The symm etric \(m\) atrix \(A\). If \(U P L O=U\) ', the leading \(N\) by -N uppertriangularpart ofA contains the upper triangularpart of the \(m\) atrix \(A\), and the strictly low ertriangular partofA is not refer-
enced. If UPLO = L', the leading N -by-N lower triangularpartofA contains the low er triangular part of the \(m\) atrix A, and the strictly upper triangularpart of A is not referenced.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

AF (input)
The factored form of them atrix A. AF contains the block diagonal matrix D and themultipliers used to obtain the factor \(U\) or \(L\) from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) as com puted by SSY TRF.

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, N)\).
\(\mathbb{P I V O T}\) (input)
D etails of the interchanges and the block structure ofD as determ ined by SSY TRF.
\(B\) (input) The righthand side m atrix \(B\).
LD B (input)
The leading dim ension of the array \(B\). LD B >= \(\max (1, N)\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by SSY TRS. On exit, the im proved solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the array X . LD X >= \(\max (1, N)\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X(\mathcal{j})\) the \(j\) th column of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{H})\), FERR ( \()\) ) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{D})-X\) TRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{)}\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

BERR (output)

The com ponentw ise relative backw ard error of each solution vector \(X(\mathcal{)}\) (ie., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{(}\) ) an exactsolution).

W ORK (w orkspace)
dim ension ( \(3 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsyuk -perform one of the sym \(m\) etric rank \(k\) operations \(C\) : alpha*A *A ' beta*C orC : alpha*A *A + beta*C

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSYRK (UPLO,TRANSA,N,K,ALPHA,A,LDA,BETA,C,LDC)}
CHARACTER * 1 UPLO,TRANSA
INTEGERN,K,LDA,LDC
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION A (LDA,*),C (LDC,*)
SUBROUTINE DSYRK_64(UPLO,TRANSA,N,K,ALPHA,A,LDA,BETA,C,LDC)
CHARACTER * 1 UPLO,TRANSA
\mathbb{NTEGER*8N,K,LDA,LDC}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION A (LDA,*),C (LDC,*)

```

\section*{F95 INTERFACE}

SU BROUTINE SYRK (UPLO, [TRANSA], \(\mathbb{N}],[K], A L P H A, A,[L D A], B E T A, C\), [LD C])

CHARACTER (LEN=1) ::UPLO,TRANSA
\(\mathbb{N} T E G E R:: N, K, L D A, L D C\)
REAL (8) ::ALPHA,BETA
REAL (8), D IM ENSION (:,:) ::A, C
SU BROUTINE SYRK_64 (UPLO, [TRANSA], \(\mathbb{N}],[K], A L P H A, A,[L D A], B E T A\), C, (LD C ])

CHARACTER (LEN=1) ::UPLO,TRANSA
\(\mathbb{N} T E G E R(8):: N, K, L D A, L D C\)
REAL (8) ::ALPHA,BETA
REAL (8), D \(\mathbb{M}\) ENSION (:,:) :: A , C

\section*{C INTERFACE}
\#include <sunperfh>
void dsyrk (charuplo, chartransa, int \(n\), int \(k\), double alpha, double *a, int lda, double beta, double *C, int \(1 d \mathrm{~d})\);
void dsyik_64 (charuple, chartransa, long n, long k, double alpha, double *a, long lda, double beta, double \({ }^{*}\) C, long ldc);

\section*{PURPOSE}
dsyrk perform s one of the sym \(m\) etric rank \(k\) operations \(C:=\) alpha*A *A '+ beta*C orC \(:=\) alpha*A *A + beta*C where alpha and beta are scalars, \(C\) is an \(n\) by \(n\) sym \(m\) etric \(m\) atrix and \(A\) is an \(n\) by \(k m\) atrix in the first case and a \(k\) by \(n\) \(m\) atrix in the second case.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the upper
or low er triangular part of the amay \(C\) is
to be referenced as follow s:
\(U P L O=U\) 'or \(G^{\prime}\) Only the upper triangular part of \(C\) is to be referenced.

UPLO = L'or I' Only the lower triangular part of \(C\) is to be referenced.

U nchanged on exit.

TRANSA (input)
O n entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} \mathrm{C}:=\) alpha*A *A ' + beta*C.

TRANSA \(=\) T'ort' \(\mathrm{C}:=\) alpha*A *A + beta*C.

TRANSA \(=\) C 'or \(C^{\prime} C:=\) alpha*A *A + beta*C.

U nchanged on exit.
TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

N (input)
On entry, \(N\) specifies the order of the \(m\) atrix \(C\). N m ustbe at least zero. U nchanged on exit.

K (input)
On entry w ith TRANSA \(=N\) 'or \(h\) ', \(K\) specifies the number of columns of the matrix \(A\), and on entry with TRANSA = 'T'or t'or C' or \(\mathrm{E}^{\prime}\), \(K\) specifies the num ber of row sof the m atrix \(\mathrm{A} . \mathrm{K}\) m ustibe at least zero. U nchanged on exit.
A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
D OUBLE PRECISION array ofD \(\mathbb{I M} E N S I O N(L D A, k a)\),
where ka isk when TRANSA = N 'or h', and is
n otherw ise. Before entry w ith TRANSA \(=\mathrm{N}\) ' or h ', the leading n by k part of the array \(A\) \(m\) ust contain the \(m\) atrix \(A\), otherw ise the leading k by n partof the aray A mustcontain the m atrix A. U nchanged on exit.

LDA (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program.
W hen TRANSA \(=\mathrm{N}\) 'or h 'then LDA must be at least \(\max (1, n)\), otherw ise LDA \(m\) ustbe at least \(\max (1, k)\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. U nchanged on exit.

C (input/output)
D OUBLE PRECISION amay ofD \(\mathbb{M}\) ENSION (LDC,n).

Before entry w ith UPLO = U 'or l ', the leading \(n\) by \(n\) uppertriangularpart of the array \(C\) \(m\) ustcontain the upper triangular part of the sym \(m\) etric \(m\) atrix and the strictly low ertriangularpartofC is not referenced. On exit, the upper triangularpart of the array \(C\) is overw ritten by the upper triangularpart of the updated
m atrix.

Before entry w ith UPLO = L 'or I', the leading \(n\) by \(n\) low ertriangular part of the anray \(C\) m ust contain the low er triangular part of the sym \(m\) etric \(m\) atrix and the strictly upper triangularpartof C is not referenced. On exit, the low er triangularpart of the amay \(C\) is overw ritten by the low er triangularpart of the updated \(m\) atrix.

LD C (input)
O n entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program. LD C must be at leastm ax ( \(1, \mathrm{n}\) ). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsysv -com pute the solution to a real system of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SU BROUTINEDSYSV (UPLO,N,NRHS,A,LDA, \mathbb{PIV,B,LDB,W ORK,LW ORK,}
\mathbb{NFO)}
CHARACTER * 1UPLO
INTEGERN,NRHS,LDA,LDB,LW ORK,INFO
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(})
DOUBLE PRECISION A (LDA,*),B (LDB,*),W ORK(*)
SU BROUTINE DSY SV_64 (UPLO ,N,NRHS,A,LDA, \mathbb{PIV ,B,LDB,W ORK,LW ORK,}
\mathbb{NFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N,NRHS,LDA,LDB,LW ORK,INFO}
\mathbb{NTEGER*8 \mathbb{PIV (*)}}\mp@subsup{}{(}{*})
DOUBLE PRECISION A (LDA,*),B (LDB,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE SYSV (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I}, B,[L D B],[W\) ORK], [LW ORK], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER ::N,NRHS,LDA,LDB,LW ORK, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I}\)
REAL (8), D IM ENSION (:) ::W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B
SU BROUTINE SYSV_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I}, B,[L D B],[W\) ORK],
\[
[\mathrm{LW} O R K],[\mathbb{N} F O])
\]

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDA,LDB,LW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{V}\)
REAL (8), D IM ENSION (:) ::W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A , B

\section*{C INTERFACE}
\#include <sunperfh>
void dsysv (charuplo, intn, intnrhs, double *a, int lda, int *ipiviv, double *b, int ldb, int *info);
void dsysv_64 (charuplo, long n, long nrhs, double *a, long lda, long *ipiv, double *b, long ldb, long *info);

\section*{PURPOSE}
dsysv com putes the solution to a realsystem of linearequations
\(A * X=B\), where \(A\) is an \(N\) boy-N symm etric \(m\) atrix and \(X\) and B are N -by-N RH S m atrices.

The diagonalpivoting \(m\) ethod is used to factorA as
\(A=U * D * U * * T\), if \(U P L O=U\) ', or
\(A=L * D * L * T\), if \(U P L O=L '\),
where \(U\) (orL) is a productof perm utation and unit upper (low er) triangular \(m\) atrioes, and \(D\) is sym \(m\) etric and block diagonalw ith 1 -by-1 and 2 -by-2 diagonalblocks. The factored form of \(A\) is then used to solve the system of equations \(A * X=B\).

\section*{ARGUMENTS}

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low er triangle ofA is stored.
N (input) The num ber of linearequations, i.e., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS \(>=0\).

A (input/output)
On entry, the sym m etric \(m\) atrix \(A\). If \(\mathrm{PLO}=\mathrm{U}\) ',
the leading N -by N uppertriangularpantofA contains the upper triangular part of the \(m\) atrix \(A\), and the strictly low er triangularpart of \(A\) is not referenced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading N -by -N low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly uppertriangularpart of A is notreferenced.

O n exit, if \(\mathbb{N F O}=0\), the block diagonalm atrix D and the \(m\) ultipliers used to obtain the factor \(U\) or L from the factorization \(A=U * D * U * * T\) or \(A=\) L*D *L**T as com puted by SSY TRF .
LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

IPIV (output)
D etails of the interchanges and the block structure ofD, as determ ined by SSY TRF. If \(\mathbb{P} \mathbb{I V}(k)>\) 0 , then row \(s\) and colum ns \(k\) and \(\mathbb{P} \mathbb{I V}(k)\) w ere interchanged, and \(\mathrm{D}(\mathrm{k}, \mathrm{k})\) is a 1 -by-1 diagonalblock. If \(U P L O=U\) 'and \(\mathbb{P} \mathbb{I V}(k)=\mathbb{P} \mathbb{I} V(k-1)<0\), then row \(s\) and columns \(k-1\) and \(-\mathbb{P} \mathbb{I V}(k)\) were interchanged and \(D(k-1 *, k-1: k)\) is a 2 -by -2 diagonal block. IfU PLO \(=\mathbb{L}\) 'and \(\mathbb{P} \mathbb{I V}(k)=\mathbb{P} \mathbb{I V}(k+1)<0\), then row \(s\) and colum nsk+1 and \(-\mathbb{P} \mathbb{I V}(k)\) w ere interchanged and \(D(k: k+1, k: k+1)\) is a 2 -by-2 diagonal block.

B (input/output)
O n entry, the N boy-N RHS righthand side m atrix B. On exit, if \(\mathbb{N F O}=0\), the N boy \(-\mathrm{NRH} S\) solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The length ofW ORK. LW ORK >=1, and forbestperform ance LW ORK \(>=N * N B, w h e r e N B\) is the optim al blocksize forSSY TRF .

IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{N N F O}=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=\mathrm{i}, \mathrm{D}(i, i)\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, so the solution could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsysvx -use the diagonalpivoting factorization to com pute the solution to a realsystem of linear equations A * \(X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDSYSVX (FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,B,}}\mathbf{N},\textrm{N},\textrm{N}
LDB,X,LDX,RCOND,FERR,BERR,W ORK,LDW ORK,W ORK 2,INFO)
CHARACTER * 1FACT,UPLO
INTEGER N,NRHS,LDA,LDAF,LDB,LDX,LDW ORK,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*),W ORK2(*)}
DOUBLE PRECISION RCOND
D OU BLE PRECISION A (LDA,*),AF (LDAF,*), B (LDB,*), X (LDX,*),
FERR (*),BERR (*),W ORK (*)
SUBROUT\mathbb{NEDSYSVX_64(FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,}}\mathbf{N},\mp@code{N},\mp@code{N}
B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,LDW ORK,W ORK 2,\mathbb{NFO)}

```
CHARACTER * 1 FACT,UPLO
\(\mathbb{N}\) TEGER*8N,NRHS,LDA,LDAF,LDB,LDX,LDW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{V} O T(\star), W\) ORK 2 ( \(\left.{ }^{( }\right)\)
DOUBLE PRECISION RCOND
D OU BLE PRECISION A (LDA, *), AF (LDAF,*), B (LDB ,*), X (LDX,*),
FERR (*), BERR (*), W ORK (*)

\section*{F95 INTERFACE}

SUBROUTINESYSVX \(\mathbb{F A C T}, \mathrm{UPLO}, \mathrm{N}, \mathrm{NRHS}, \mathrm{A},[L D A], A F,[L D A F], \mathbb{P} \mathbb{I} O T\), B, [LDB], X, [LDX],RCOND,FERR,BERR, [W ORK], [LDW ORK], [W ORK2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::FACT,UPLO
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{V} O T, W\) ORK 2
REAL (8) :: RCOND
REAL (8), D \(\mathbb{M} E N S I O N(:):: F E R R, B E R R, W\) ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A \(, A F, B, X\)

SU BROUTINE SYSVX_64 (FACT, UPLO, N,NRHS,A, [LDA],AF, [LDAF], \(\mathbb{P} \mathbb{I} O T, B,[L D B], X,[L D X], R C O N D, F E R R, B E R R,[W O R K],[L D W O R K]\), [W ORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::FACT, UPLO
\(\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D A F, L D B, L D X, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}, W\) ORK 2
REAL (8) :: RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,AF,B,X

\section*{C INTERFACE}
\#include <sunperfh>
void dsysvx (char fact, charuple, intn, int nrhs, double *a, int lda, double *af, intldaf, int *ipivot, double *b, int ldlo, double *x, int ldx, double *rcond, double *ferr, double *berr, int *info);
void dsysvx_64 (char fact, charuplo, long n, long nrhs, double *a, long lda, double *af, long ldaf, long *ípivot, double *b, long ldb, double *x, long ldx, double *rcond, double *ferr, double *berr, long *info);

\section*{PURPOSE}
dsysvx uses the diagonalpivoting factorization to com pute the solution to a realsystem of linear equations \(A * X=B\), \(w\) here \(A\) is an \(N\) boy \(N\) sym \(m\) etric \(m\) atrix and \(X\) and \(B\) are \(N\) boyN R H S m atrices.

E rrorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT \(=N\) ', the diagonalpivoting \(m\) ethod is used to factorA.
The form of the factorization is
\(A=U * D * U * * T\), if \(U P L O=U '\) or
\(A=L * D * L * * T\), if \(U P L O=L \prime\)
where \(U\) (orL) is a productofperm utation and unitupper
(low er)
triangularm atrices, and \(D\) is sym \(m\) etric and block diago-

\section*{nalw th}

1-by-1 and 2-by-2 diagonalblocks.
2. If som eD \((i, i)=0\), so thatD is exactly singular, then the routine
retums w ith \(\mathbb{N F O}=\) i. O therw ise, the factored form of A is used
to estim ate the condition num ber of the \(m\) atrix \(A\). If the reciprocal of the condition num ber is less than \(m\) achine precision,
\(\mathbb{N}\) FO \(=\mathrm{N}+1\) is retumed as a waming, but the routine stillgoes on
to solve for \(X\) and com pute error bounds as described below .
3.The system ofequations is solyed for \(X\) using the factored form of A.
4. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for it.

\section*{ARGUMENTS}

FACT (input)
Specifies w hether ornot the factored form of A has been supplied on entry. = \(\mathrm{F}^{\prime}:\) O n entry, \(\mathrm{A} F\) and \(\mathbb{P I V O T}\) contain the factored form of A. AF and \(\mathbb{P} \mathbb{I V}\) O T w illnotbe m odified. \(=\mathrm{N}\) ': Them atrix A w illbe copied to A F and factored.

UPLO (input)
= U ': U pper triangle ofA is stored;
\(=L^{\prime}\) : Low er triangle of \(A\) is stored.
N (input) The num ber of linear equations, i.e., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the m atrices B and X. NRHS >=0.

A (input) The symm etricm atrix \(A\). If UPLO \(=U\) ', the leading \(N\) by -N uppertriangularpartofA contains the upper triangularpart of the \(m\) atrix \(A\), and the strictly low ertriangularpartofA is not refer-
enced. If UPLO = L', the leading N -by-N lower triangularpartofA contains the low er triangular part of the \(m\) atrix A, and the strictly upper triangularpart ofA is not referenced.

LDA (input)
The leading dim ension of the anay A. LDA >= \(\max (1, N)\).

AF (input/output)
If FACT = F ', then \(A F\) is an inputargum ent and on entry contains the block diagonalm atrix D and the m ultipliers used to obtain the factor \(U\) orL from the factorization \(A=U * D * U * * T\) orA \(=L * D * L * * T\) as com puted by SSY TRF .
If FA C T = N ', then AF is an output argum ent and on exit retums the block diagonalm atrix \(D\) and the multipliers used to obtain the factorU or L from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\).

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, N)\).

IPIVOT (inputoroutput)
If FACT = \(\mathrm{F}^{\prime}\), then \(\mathbb{P} \mathbb{I} O T\) is an input argum ent and on entry contains details of the interchanges and the block structure of D, as determ ined by SSY TRF. If \(\mathbb{P} \operatorname{IV} O T(k)>0\), then row \(s\) and colum ns \(k\) and \(\mathbb{P} \mathbb{I V O T}(k)\) were interchanged and \(D(k, k)\) is a 1 -by-1 diagonal block. If UPLO \(=U '\) and \(\mathbb{P} \mathbb{I} O T(k)=\mathbb{P} \mathbb{I V} O T(k-1)<0\), then row sand colum ns \(\mathrm{k}-1\) and \(-\mathbb{P} \mathbb{I V O T}(\mathrm{k})\) were interchanged and \(\mathrm{D}(\mathrm{k}-\) \(1 \mathrm{k}, \mathrm{k}-1 \mathrm{k}\) ) is a 2 -by-2 diagonalblock. IfUPLO = L 'and \(\mathbb{P} \mathbb{I V} O T(k)=\mathbb{P} \mathbb{I V} O T(k+1)<0\), then row \(s\) and colum nsk+1 and \(-\mathbb{P} \operatorname{IV} O T(k)\) were interchanged and D \(k: k+1, k: k+1)\) is a 2 -by-2 diagonalblock.

If \(\mathrm{FACT}=\mathrm{N}\) ', then \(\mathbb{P} \mathbb{I V O T}\) is an output argum ent and on exitcontains details of the interchanges and the block structure of D, as determ ined by SSY TRF .

B (input) The N by -N R H S righthand side m atrix B .

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, \mathbb{N})\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=\mathrm{N}+1\), the N -by-NRH S solution
\(m\) atrix \(X\).

LD X (input)
The leading dim ension of the anay X . LD X >= \(\max (1, N)\).

RCOND (output)
The estim ate of the reciprocal condition num ber of the matrix A. IfRCOND is less than them achine precision (in particular, if RCOND \(=0\) ), the \(m\) atrix is singular to working precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO > 0.

\section*{FERR (output)}

The estim ated forw ard emrorbound for each solution vector \(X()\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{H}), \operatorname{FERR}(\mathcal{)})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{D})-X\) TRU \(E\) ) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector X (i) (i.e., the sm allest relative change in any elem entof \(A\) or \(B\) thatm akes \(X(\mathcal{J})\) an exactsolution).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length ofW ORK. LDW ORK >= 3*N , and for best perform ance LDW ORK \(>=N * N B\), where \(N B\) is the optim alblocksize forSSY TRF.

If LDW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N F O}=i\), and \(i\) is
\(<=\mathrm{N}: \mathrm{D}(i, i)\) is exactly zero. The factorization
has been completed but the factorD is exactly singular, so the solution and error bounds could not be com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1\) : D is nonsingular, butRCOND is less than \(m\) achine precision, \(m\) eaning that the \(m\) atrix is singular to w orking precision. N evertheless, the solution and error bounds are com puted because there are a num ber of situationsw here the com puted solution can bem ore accurate than the value ofRC O N D w ould suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dsytd2 -reduce a realsym \(m\) etric \(m\) atrix A to sym \(m\) etric tridiagonal form T by an orthogonal sim ilarity transform ation

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE DSYTD 2(UPLO,N,A,LDA,D,E,TAU, INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGERN,LDA,INFO}
DOUBLE PRECISION A (LDA,*),D (*),E (*),TAU(*)
SU BROUT\mathbb{NE DSYTD 2_64(UPLO ,N,A ,LDA,D ,E,TAU ,INFO)}
CHARACTER * 1UPLO
INTEGER*8N,LDA,INFO
DOUBLE PRECISION A (LDA,*),D (*),E (*),TAU (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE SY TD 2 (UPLO ,N,A, [LDA ],D ,E,TAU, [ $\mathbb{N F F O}$ ])
CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R:: N, L D A, \mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) :: D,E,TAU
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::A
SU BROUTINE SYTD 2_64 (UPLO, N,A, [LDA ],D ,E,TAU, [ $\mathbb{N} F O$ ])
CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER (8) :: N, LDA, $\mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::D ,E,TAU
REAL (8),D IM ENSION (:,:) ::A

```

\section*{C INTERFACE}
\#include <sunperfh>
void dsytd2 (charuple, intn, double *a, int lda, double *d, double *e, double *tau, int *info);
void dsytd2_64 (charuplo, long n, double *a, long lda, double *d, double *e, double *tau, long *info);

\section*{PURPOSE}
dsytd2 reduces a real sym \(m\) etric \(m\) atrix A to sym \(m\) etric tridiagonal form \(T\) by an orthogonal sim ilarity transform ation: \(Q^{\prime}\) * \(A * Q=T\).

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the upper or low er triangular part of the sym \(m\) etricm atrix A is stored:
= U ': U ppertriangular
= L ': Low ertriangular
N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input) \(O n\) entry, the sym \(m\) etric \(m\) atrix \(A\). If \(U P L O=U\) ', the leading \(n-b y-n\) upper triangularpartofA contains the uppertriangular part of the \(m\) atrix \(A\), and the strictly low er triangularpartofA is not referenced. If UPLO = ' L ', the leading \(\mathrm{n}-\mathrm{by}-\mathrm{n}\) low er triangularpart ofA contains the low ertriangularpart of the matrix \(A\), and the strictly upper triangularpartofA is not referenced. On exit, if UPLO = U ', the diagonal and first superdiagonalofA are overw rilten by the comesponding elem ents of the tridiagonalm atrix \(T\), and the ele\(m\) ents above the first superdiagonal, w ith the amay TAU, represent the orthogonalm atrix \(Q\) as a product of elem entary reflectors; ifU PLO = L ', the diagonaland firstsubdiagonalofA are overw ritten by the corresponding elem ents of the tridiagonalm atrix \(T\), and the elem ents below the first subdiagonal, \(w\) ith the array TA \(U\), represent the orthogonalm atrix \(Q\) as a product of elem entary reflectors. See FurtherD etails.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, \mathbb{N})\).

D (output)
The diagonalelem ents of the tridiagonalm atrix T :
D \((i)=A(i, i)\).

E (output)
The off-diagonal elem ents of the tridiagonal \(m\) atrix \(T: E(i)=A(i, i+1)\) if \(U P L O=U ', E(i)=\) A \((\mathbf{i}+1, i)\) ifUPLO \(=L\).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{N N F O}=-i\), the \(i\)-th argum ent had an illegalvalue.

\section*{FURTHER DETAILS}

If \(U P L O=U\) ', the \(m\) atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(n-1) \ldots H(2) H(1) .
\]

Each H (i) has the form
\[
H(i)=I-\tan * v^{*} v^{\prime}
\]
where tau is a real scalar, and \(v\) is a realvectorw ith \(v(i+1 m)=0\) and \(v(i)=1 ; v(1: i-1)\) is stored on exitin A ( \(1:-1, i+1)\), and tau in TAU (i).

If \(U P L O=L\) ', the \(m\) atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(1) H(2) \ldots H(n-1) .
\]

Each H (i) has the form
\[
H(i)=I-\tan * v^{*} v^{\prime}
\]
\(w\) here tau is a real scalar, and \(v\) is a realvectorw ith \(\mathrm{v}(1: i)=0\) and \(\mathrm{v}(\mathrm{i}+1)=1 ; \mathrm{v}(\mathrm{i}+2 \mathrm{n})\) is stored on exit in A (i+2n,i), and tau in TAU (i).
```

exam ples w ith n = 5:

```
```

ifUPLO = U ': ifUPLO = L':
( d e v2 v3 v4 ) ( d
)
( d e v3 v4 ) ( e d
)
( d e v4 ) ( v1 e d
)
( d e ) ( v1 v2 e d
)
( d ) (v1 v2 v3 e d
)

```
where d and e denote diagonaland off-diagonal elem ents of \(T\), and videnotes an elem ent of the vectordefining \(H\) (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dsyt42 -com pute the factorization of a real symmetric
\(m\) atrix A using the Bunch K aufn an diagonalpivoting \(m\) ethod

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
\mathbb{NTEGERN,LDA,}\mathbb{NFO}
\mathbb{NTEGER P\mathbb{IV (*)}}\mathbf{*})
DOUBLE PRECISION A (LDA,*)

```

```

CHARACTER * 1 UPLO
INTEGER*8N,LDA,INFO
\mathbb{NTEGER*8 \mathbb{P IV (*)}}\mp@subsup{}{(}{*})
DOUBLE PRECISION A (LDA,*)

```

\section*{F95 INTERFACE}
```

SU BROU T $\mathbb{N E}$ EY TF2 (UPLO , $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V},[\mathbb{N} F O])$
CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER :: $\mathrm{N}, \mathrm{LDA}, \mathbb{N} F O$
$\mathbb{I N}$ TEGER,D $\mathbb{I M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V}$
REAL (8),D $\mathbb{M}$ ENSION (:,:) ::A
SU BROUTINE SY TF2_64 (UPLO, $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V},[\mathbb{N} F O])$
CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER (8) :: $\mathrm{N}, \mathrm{LD} A, \mathbb{N}$ FO

```

\section*{C INTERFACE}
\#include <sunperfh>
void dsytf2 (charuplo, intn, double *a, int lda, int *ịì int*info);
void dsytf2_64 (charuplo, long n, double *a, long lda, long *ipiv, long *info);

\section*{PURPOSE}
dsytff com putes the factorization of a realsym \(m\) etric \(m\) atrix A using the Bunch \(-K\) aufm an diagonalpivoting \(m\) ethod:
\[
A=U * D * U^{\prime} \text { or } A=L * D * L^{\prime}
\]
where U (orL) is a productof perm utation and unit upper (low er) triangularm atrioes, U 'is the transpose of U , and D is sym \(m\) etric and block diagonalw ith 1 -by -1 and 2 -by -2 diagonalblocks.

This is the unblocked version of the algorithm , calling Level2 BLAS.

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the upper or low er triangular part of the sym \(m\) etric \(m\) atrix \(A\) is stored:
\(=\mathrm{U}\) ': Uppertriangular
= L': Low ertriangular

N (input) The order of the matrix A. N \(>=0\).

A (input/output)
O \(n\) entry, the sym m etric \(m\) atrix A. If \(\mathrm{U} P \mathrm{O}=\mathrm{U}\) ', the leading \(n-b y-n\) upper triangularpartofA contains the upper triangularpart of the \(m\) atrix \(A\), and the strictly low er triangularpart of A is not referenced. If UPLO \(=\mathbb{L}\) ', the leading \(n-b y-n\) low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of \(A\) is notreferenced.

On exit, the block diagonalm atrix D and the mul
tipliers used to obtain the factorU orl (see below for further details).

LDA (input)
The leading dim ension of the anray A. LD A >= \(\max (1, \mathbb{N})\).

IPIV (output)
D etails of the interchanges and the block structure ofD. If \(\mathbb{P} \mathbb{I V}(k)>0\), then row \(s\) and colum ns \(k\) and \(\mathbb{P} \mathbb{I V}(k)\) were interchanged and \(D(k, k)\) is a 1 -by-1 diagonalblock. IfUPLO \(=U\) 'and \(\mathbb{P} \mathbb{I V}(k)\) \(=\mathbb{P} \mathbb{V}(k-1)<0\), then row \(s\) and colmm ns \(k-1\) and \(-\mathbb{P} \mathbb{V}(k)\) w ere interchanged and \(D(k-1 * k, k-1 *)\) is a 2 -by-2 diagonalblock. IfUPLO \(=\mathbb{L}\) 'and \(\mathbb{P} \mathbb{I V}(k)\)
\(=\mathbb{P} \mathbb{I}(k+1)<0\), then row sand colum ns \(k+1\) and \(-\mathbb{P} \mathbb{V}(k)\) w ere interchanged and \(D(k, k+1, k \mathrm{k}+1)\) is a 2-by-2 diagonalblock.
\(\mathbb{N F O}\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\mathrm{k}\), the k -th argum enthad an illegalvalue \(>0:\) if \(\mathbb{N} F O=k, D(k, k)\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, and division by zero w illoccur if it is used to solve a system of equations.

\section*{FURTHER DETAILS}

\section*{1-96 -B ased on m odifications by J.Lew is, Boeing Com puter}

\section*{Services}

Com pany
If U PLO \(=\mathrm{U}\) ', then \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}\) ', where
\(U=P(n) \star U(n) * \ldots * P(k) U(k) * \ldots\),
i.e., \(U\) is a productof term \(S P(k) * U(k)\), where \(k\) decreases from \(n\) to 1 in steps of 1 or2, and \(D\) is ablock diagonal \(m\) atrix \(w\) ith 1 -by -1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{V}(k)\), and \(U(k)\) is a unituppertriangularm atrix, such that if the diagonal block \(D(k)\) is of orders ( \(s=1\) or 2 ), then
\[
\begin{aligned}
& \text { ( I v 0 ) k-s } \\
& \mathrm{U}(\mathrm{k})=(0 \mathrm{I} 0) \mathrm{s} \\
& \text { ( } 0 \text { O I ) } \mathrm{n}-\mathrm{k} \\
& \mathrm{k}-\mathrm{s} \mathrm{~s} \mathrm{n}-\mathrm{k}
\end{aligned}
\]

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(1 k-\)
\(1, k)\). If \(s=2\), the upper triangle ofD ( \(k\) ) overw rites \(A(k-\) \(1, k-1)\), A \((k-1, k)\), and \(A(k, k)\), and \(v\) overw rites A ( 1 k- \(2, k-\) \(1 \mathrm{k})\).

If \(\operatorname{PLO}=\mathrm{L}\) ', then \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L}\) ', where
\(\mathrm{L}=\mathrm{P}(1) \star \mathrm{L}(1){ }^{*} \ldots * \mathrm{P}(k) \star \mathrm{L}(k)^{*} \ldots\),
i.e., \(L\) is a product of term \(S P(k) * L(k)\), where \(k\) increases from 1 to \(n\) in steps of 1 or 2, and \(D\) is a block diagonal \(m\) atrix \(w\) ith 1 -by -1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{V}(k)\), and \(L(k)\) is a unit low ertriangularm atrix, such that if the diagonal
block D (k) is of orders ( \(s=1\) or2), then
\[
\begin{aligned}
& \text { ( } \left.\begin{array}{llll}
\mathrm{I} & 0 & 0
\end{array}\right) \mathrm{k}-1 \\
& L(k)=\left(\begin{array}{lll}
0 & I & 0
\end{array}\right) s \\
& \text { ( } 0 \text { v I ) } n-k-s+1 \\
& \text { k-1 s n-k-s+1 }
\end{aligned}
\]

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites
\(A(k+1 n, k)\). If \(s=2\), the low er triangle ofD ( \(k\) ) overw rites A \((k, k), A(k+1, k)\), and \(A(k+1, k+1)\), and \(v\) overw rites A \((k+2 m, k k+1)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dsytrd -reduce a realsym \(m\) etric \(m\) atrix A to real sym \(m\) etric tridiagonal form T by an orthogonal sim ilarity transform ation

\section*{SYNOPSIS}

SU BROUTINEDSYTRD (UPLO,N,A,LDA,D,E,TAU,WORK,LWORK, \(\mathbb{N} F O\) )
CHARACTER * 1 UPLO
\(\mathbb{N}\) TEGER N,LDA,LW ORK, \(\mathbb{N} F O\)

SU BROUTINEDSYTRD_64(UPLO,N,A,LDA,D,E,TAU,W ORK,LWORK, \(\mathbb{N} F O\) )

CHARACTER * 1 UPLO
\(\mathbb{N} T E G E R * 8 N, L D A, L W O R K, \mathbb{N} F O\)
DOUBLE PRECISION A (LDA, *), D (*), E (*),TAU (*),WORK (*)

\section*{F95 INTERFACE}

SU BROUTINE SYTRD (UPLO,N,A, [LDA],D,E,TAU, [W ORK ], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N} T E G E R:: N, L D A, L W\) ORK, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D , E, TAU ,W ORK
REAL (8),D IM ENSION (:,:) ::A
SU BROUTINE SYTRD_64 (UPLO, N, A, [LDA ],D, E,TAU, [W ORK ], [LW ORK ], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER (8) :: N,LDA,LW ORK, \(\mathbb{N}\) FO

REAL (8),D \(\mathbb{M}\) ENSION (:) ::D ,E,TAU,W ORK
REAL (8),D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dsytrd (charuplo, intn, double *a, int lda, double *d, double *e, double *tau, int *info);
void dsytrd_64 (charuplo, long n, double *a, long lda, double *d, double *e, double *tau, long *info);

\section*{PURPOSE}
dsytrd reduces a real sym m etric m atrix A to real sym m etric tridiagonal form \(T\) by an orthogonal sim ilarity transform ation: \(\mathrm{Q} * * \mathrm{~T} * \mathrm{~A} * \mathrm{Q}=\mathrm{T}\).

\section*{ARGUMENTS}

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input) On entry, the symm etric m atrix A. If \(\mathrm{PLO}=\mathrm{U}\) ', the leading N -by N uppertriangularpartof A contains the uppertriangular part of the \(m\) atrix \(A\), and the strictly low er triangularpart of A is not referenced. If UPLO = L', the leading N -by-N low er triangularpartof \(A\) contains the low ertriangularpart of the matrix \(A\), and the strictly upper triangular partofA is not referenced. On exit, if \(\mathrm{UPLO}=\mathrm{U}\) ', the diagonal and first superdiagonalofA are overw rilten by the comesponding elem ents of the tridiagonalm atrix T , and the ele\(m\) ents above the first superdiagonal, with the array TAU, represent the orthogonalm atrix \(Q\) as a product of elem entary reflectors; if U PLO = L', the diagonaland firstsubdiagonalofA are overw ritten by the comesponding elem ents of the tridiagonalm atrix T , and the elem ents below the first subdiagonal, w ith the amay TAU, represent the orthogonalm atrix \(Q\) as a productofelem entary reflectors. See FurtherD etails.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

D (output)
The diagonalelem ents of the tridiagonalm atrix T :
D \((i)=A(i, i)\).

E (output)
The off-diagonal elem ents of the tridiagonal \(m\) atrix \(T: E(i)=A(i, i+1)\) if \(U P L O=U^{\prime}, E(i)=\) A \((\mathbf{i}+1, i)\) if \(\mathrm{UPLO}=\mathrm{L}\).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. LW ORK >=1. For optim um perform ance \(L W\) ORK \(>=N * N B\), where \(N B\) is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue

\section*{FURTHER DETAILS}

If U PLO \(=\mathrm{U}\) ', the m atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(n-1) \ldots H(2) H(1) .
\]

Each \(H\) (i) has the form
\[
H(i)=I-\tan * V^{*} v^{\prime}
\]
where tau is a real scalar, and \(v\) is a realvectorw ith \(v(i+1 m)=0\) and \(v(i)=1 ; v(1: i-1)\) is stored on exitin

A ( \(1: i-1, i+1)\), and tau in TAU (i).
IfU PLO \(=\mathrm{L}\) ', the m atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(1) H(2) \ldots H(n-1) .
\]

Each \(H\) (i) has the form
H (i) \(=I-\tan { }^{*} V^{*} V^{\prime}\)
where tau is a real scalar, and \(v\) is a realvectorw ith \(\mathrm{v}(1: i)=0\) and \(\mathrm{v}(i+1)=1 ; \mathrm{v}(i+2 \mathrm{n})\) is stored on exit in A (i+2m,i), and tau in TAU (i).
The contents of A on exitare illustrated by the follow ing exam plesw ith \(\mathrm{n}=5\) :
```

ifUPLO = U ': ifUPLO = L :

```
    ( d e v2 v3 v4 ) ( d
)
    ( d e v3 v4 ) ( e d
)
    ( d e v4 ) (v1 e d
)
    ( d e ) ( v1 v2 e d
)
( d ) (v1 v2 v3 e d
)
where \(d\) and e denote diagonal and off-diagonal elem ents of \(T\), and videnotes an elem entof the vectordefining \(H\) (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dsytrf-com pute the factorization of a real symm etric
\(m\) atrix \(A\) using the \(B\) unch \(-K\) aufn an diagonalpivoting \(m\) ethod

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
INTEGER N,LDA,LDW ORK,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION A (LDA,*),W ORK (*)

```

```

CHARACTER * 1 UPLO
INTEGER*8N,LDA,LDW ORK,INFO
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION A (LDA,*),W ORK (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE SY TRF (UPLO ,N,A, [LDA], $\mathbb{P} \mathbb{I V O T}, \mathbb{W}$ ORK ], [LDW ORK ], [ $\mathbb{N F O}])$
CHARACTER (LEN=1)::UPLO
$\mathbb{N} T E G E R:: N, L D A, L D W$ ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$
REAL (8),D IM ENSION (:) ::W ORK
REAL (8),D IM ENSION (:,:) ::A
SU BROUTINE SYTRF_64 (UPLO,N,A, [LDA], $\mathbb{P} \mathbb{I V O T},\left[\begin{array}{l}\text { W ORK ], [LDW ORK ], }\end{array}\right.$ [ $\mathbb{N}$ FO ])

```

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER (8) ::N,LDA,LDW ORK, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8),D IM ENSION (:) ::W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dsytrf(charuplo, int \(n\), double *a, int lda, int
*ịivot, int*info);
void dsytrf_64 (char uplo, long n, double *a, long lda, long
*ịíivot, long *info);

\section*{PURPOSE}
dsytrf com putes the factorization of a realsym \(m\) etric \(m\) atrix A using the Bunch \(-K\) aufm an diagonalpivoting \(m\) ethod. The form of the factorization is
\[
A=U * D * U * * T \text { or } A=L * D * L * * T
\]
where \(U\) (orL) is a product of perm utation and unit upper (low er) triangular \(m\) atrices, and \(D\) is sym \(m\) etric and block diagonalw ith 1 -by-1 and 2 -by-2 diagonalblocks.

This is the blocked version of the algorithm, calling Level 3 BLAS.

\section*{ARGUMENTS}

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input/output)
O n entry, the sym m etric m atrix A. If UPLO = U', the leading N -by -N uppertriangularpartofA contains the upper triangular part of the \(m\) atrix \(A\), and the strictly low ertriangularpart of A is not referenced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading N -by -N low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly
upper triangularpartofA is not referenced.

On exit, the block diagonalm atrix D and the multipliers used to obtain the factorU orL (see below for further details).

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

\section*{IPIVOT (output)}

D etails of the interchanges and the block structure of D. If \(\mathbb{P I V O T}(k)>0\), then row sand columnsk and \(\mathbb{P I V O T}(k)\) were interchanged and \(D(k, k)\) is a \(1-b y-1\) diagonalblock. If \(U P L O=U^{\prime}\) and \(\mathbb{P} \mathbb{V} O T(k)=\mathbb{P} \mathbb{V} O T(k-1)<0\), then row \(s\) and colum ns \(k-1\) and - \(\mathbb{P I V O T}(k)\) were interchanged and D ( \(k-1 * k, k-1 k)\) is a \(2-b y-2\) diagonal block. If UPLO \(=\mathrm{L}\) 'and \(\mathbb{P I V O T}(k)=\mathbb{P} \mathbb{I V O T}(k+1)<0\), then row sand colum ns \(k+1\) and \(-\mathbb{P} \mathbb{V} O T(k)\) were interchanged and \(D(k k+1, k k+1)\) is a \(2-b y-2\) diagonal block.

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, \mathrm{~W}\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length ofW ORK. LDW ORK >=1. Forbestperfor\(m\) ance LDW ORK >=N *NB, where NB is the block size retumed by \(\amalg A E N V\).

IfLDW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfinlexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{D}(\mathrm{i}, \mathrm{i})\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, and division by zero w illoccur if it is used to solve a system of equations.

\section*{FURTHER DETAILS}

If \(\mathrm{ULO}=\mathrm{U}\) ', then \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}\) ', where
\(U=P(n) \star U(n)^{\star} \ldots{ }^{\star} P(k) U(k)^{\star} \ldots\),
ie., \(U\) is a product ofterm \(\operatorname{sP}(k) * U(k)\), where \(k\) decreases from \(n\) to 1 in steps of 1 or 2 , and \(D\) is a block diagonal \(m\) atrix w ith 1 -by-1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I V O T}(\mathrm{k})\), and \(\mathrm{U}(\mathrm{k})\) is a unituppertriangularm atrix, such that if the diagonal block D (k) is of orders ( \(s=1\) or 2 ), then
\[
\begin{aligned}
& \text { ( I v 0 ) k-s } \\
& U(k)=\left(\begin{array}{lll}
0 & I
\end{array}\right) s \\
& \text { ( } 000 \text { I ) n-k } \\
& \mathrm{k}-\mathrm{s} \text { s n-k }
\end{aligned}
\]

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(1 k-\) \(1, k\) ). If \(s=2\), the upper triangle ofD \((k)\) overw rites \(A(k-\) \(1, k-1), A(k-1, k)\), and \(A(k, k)\), and \(V\) overw rites \(A(1 k-2, k-\) \(1 \mathrm{k})\).

If \(\mathrm{UPLO}=\mathrm{L}\) ', then \(A=\mathrm{L} * \mathrm{D} * \mathrm{~L}\) ', where
\(L=P(1) \star L(1)^{\star} \ldots * P(k) \star L(k) * \ldots\)
ie., \(L\) is a productofterm \(s P(k) * L(k)\), where \(k\) increases
from 1 to n in steps of 1 or 2 , and D is a block diagonal \(m\) atrix \(w\) th 1 -by -1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I V O T}(k)\), and \(L(k)\) is a unitlow ertriangularm atrix, such that if the diagonal
block \(D(k)\) is of orders ( \(s=1\) or 2 ), then
\[
\begin{aligned}
& \text { ( } \left.\begin{array}{llll}
I & 0 & 0
\end{array}\right) k-1 \\
& L(k)=\left(\begin{array}{lll}
0 & I & 0
\end{array}\right) s \\
& \text { ( } 0 \mathrm{~V} \text { I ) } \mathrm{n}-\mathrm{k}-\mathrm{s}+1 \\
& \mathrm{k}-1 \text { s } \mathrm{n}-\mathrm{k}-\mathrm{s}+1
\end{aligned}
\]

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(k+1 m, k)\). If \(s=2\), the low ertriangle ofD \((k)\) overw rites \(A(k, k), A(k+1, k)\), and \(A(k+1, k+1)\), and \(v\) overw rites A \((k+2 m, k: k+1)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dsytri-com pute the inverse of a real sym \(m\) etric indefinite m atrix A using the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\) L*D *L**T com puted by SSY TRF

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
NNTEGER N,LDA, INFO
\mathbb{NTEGER \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION A (LDA,*),W ORK (*)

```

```

CHARACTER * 1UPLO
\mathbb{NTEGER*8N,LDA,INFO}
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION A (LDA,*),W ORK (*)

```
F95 INTERFACE
    SU BROUTINE SY TRI(UPLO, N,A, [LDA ], \(\mathbb{P} \mathbb{I} O T,[\mathbb{O} O R K],[\mathbb{N F O}])\)
    CHARACTER (LEN=1) ::UPLO
    \(\mathbb{N}\) TEGER :: N,LDA, \(\mathbb{N} F O\)
    \(\mathbb{N}\) TEGER,D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
    REAL (8),D IM ENSION (:) ::W ORK
    REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A
    SU BROUTINE SY TRI_64 (UPLO,N,A, [LDA ], \(\mathbb{P} \mathbb{I V O T},[\mathbb{W}\) ORK ], [ \(\mathbb{N F O}\) ])
    CHARACTER (LEN=1)::UPLO
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} A, \mathbb{N}\) FO
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8), D \(\mathbb{I M}\) ENSION (:) ::W ORK
REAL (8), D \(\mathbb{I M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dsytri(charuplo, int \(n\), double *a, int lda, int *ipivot, int *info);
void dsytri_64 (charuplo, long n, double *a, long lda, long
*ipivot, long *info);

\section*{PURPOSE}
dsytricom putes the inverse of a real sym \(m\) etric indefinite \(m\) atrix \(A\) using the factorization \(A=U * D * U * * T\) orA \(=\) L *D *L **T com puted by SSY TRF.

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) ': Uppertriangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\);
\(=\mathrm{L}^{\prime}\) : Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\).

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the block diagonalm atrix D and the m ultipliers used to obtain the factorU orL as com puted by SSY TRF .

On exit, if \(\mathbb{N F O}=0\), the (sym m etric) inverse of the original m atrix. If \(\mathrm{UPLO}=\mathrm{U}\) ', the upper triangularpart of the inverse is form ed and the part ofA below the diagonal is notreferenced; if \(\mathrm{UPLO}=\mathrm{L}\) ' the low er triangular part of the inverse is formed and the partofA above the diagonal is not referenced.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).
\(\mathbb{P I V O T}\) (input)
D etails of the interchanges and the block structure ofD as determ ined by SSY TRF.

W ORK (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i, D(i, i)=0\); the \(m\) atrix is singular and its inverse could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dsytrs-solve a system of linearequationsA *X = B w ith a
realsymm etric m atrix A using the factorization A = U *D *U **T
orA = L*D *L**T com puted by SSY TRF

```

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
INTEGERN,NRHS,LDA,LDB,INFO
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION A (LDA ,*),B (LDB,*)

```

```

CHARACTER * 1 UPLO
\mathbb{NTEGER*8N,NRHS,LDA,LDB,INFO}
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION A (LDA,*),B(LDB,*)

```

\section*{F95 INTERFACE}

SU BROUTINE SYTRS (UPLO,N,NRHS,A, [LDA], \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LD} B],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER ::N,NRHS,LDA,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8), D \(\mathbb{M}\) ENSIO N (:,:) ::A,B
SU BROUTINE SYTRS_64 (UPLO,N,NRHS,A, [LDA ], \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}],[\mathbb{N} F O])\)
CHARACTER (LEN=1) ::UPLO
\(\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D B, \mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \mathrm{ENSION}(:):: \mathbb{P} \mathbb{I} O T\)
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A , B

\section*{C INTERFACE}
\#include <sunperfh>
void dsytrs (charuplo, intn, intnrs, double *a, int lda, int *ipivot, double *b, int ldlb, int *info);
void dsytrs_64 (charuplo, long n, long nrhs, double *a, long
lda, long *ipivot, double *b, long ldb, long
*info);

\section*{PURPOSE}
dsytur solves a system of linearequations \(A * X=B \quad w\) ith \(a\) real sym \(m\) etric \(m\) atrix \(A\) using the factorization \(A=U * D * U * * T\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by SSY TRF.

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) : : U pper triangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}\) **T;
\(=\mathbb{L}:\) Low ertriangular, form is \(A=L * D * L * * T\).

N (input) The order of them atrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS \(>=0\).

A (input) The block diagonalm atrix D and the multipliers
used to obtain the factorU orL as com puted by SSY TRF.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

IPIVOT (input)
D etails of the interchanges and the block structure ofD as determ ined by SSY TRF .

B (input/output)
O \(n\) entry, the righthand side m atrix B. On exit,
the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray B. LD B \(>=\) \(\max (1, N)\).

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtboon -estim ate the reciprocal of the condition num ber of a triangular band \(m\) atrix \(A\), in etherthe 1 -norm orthe infinity-norm

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDTBCON NORM,UPLO,D IAG,N,KD,A,LDA,RCOND,W ORK,}
W ORK2,\mathbb{NFO)}
CHARACTER * 1 NORM,UPLO,DIAG
\mathbb{NTEGER N,KD,LDA,}\mathbb{NNFO}
INTEGER W ORK2 (*)
DOUBLE PRECISION RCOND
D OUBLE PRECISION A (LDA,*),W ORK (*)
SUBROUT\mathbb{NEDTBCON_64 NORM,UPLO,DIAG,N,KD,A,LDA,RCOND,W ORK,}
WORK2, \mathbb{NFO)}

```
CHARACTER * 1 NORM, UPLO, DIAG
\(\mathbb{N}\) TEGER*8N,KD,LDA, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER *8 W ORK 2 (*)
DOUBLE PRECISION RCOND
D OUBLE PRECISION A (LDA,*),W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE TBCON \(\mathbb{N} O R M, U P L O, D \mathbb{I} G, N, K D, A,[L D A], R C O N D,[W O R K]\), [W ORK 2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::NORM,UPLO,DIAG
\(\mathbb{N}\) TEGER :: \(N, K D, L D A, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) ::W ORK2
REAL (8) :: RCOND

REAL (8), D IM ENSION (:) ::W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A
SU BROUTINE TBCON_64 \(\mathbb{N} O R M, U P L O, D \mathbb{A G}, N, K D, A,[L D A], R C O N D\), [W ORK], [W ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::NORM,UPLO,DIAG
\(\mathbb{N}\) TEGER (8) :: N, KD,LDA, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) ::W ORK2
REAL (8) ::RCOND
REAL (8),D IM ENSION (:) ::W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dtbcon (charnorm , charuplo, chardiag, intn, int kd, double *a, int lda, double *rcond, int *info);
void dtbcon_64 (charnorm , charuplo, chardiag, long n, long kd, double *a, long lda, double *rcond, long *info);

\section*{PURPOSE}
dtboon estim ates the reciprocal of the condition num berof a triangular band matrix \(A\), in either the 1 -norm or the infinity-norm.

The norm ofA is com puted and an estim ate is obtained for norm (inv (A )), then the reciprocal of the condition num ber is com puted as
```

RCOND = 1/(norm (A)* norm (inv(A))).

```

\section*{ARGUMENTS}
```

NORM (input)
Specifies w hether the 1-nom condition num ber or
the infinity-norm condition num ber is required:
= 1'orD': 1-nom;
= I':}\quad\mathrm{ Infinity-nom .
UPLO (input)
= U ': A is uppertriangular;
= LL ': A is low ertriangular.

```
D IA G (input)
    \(=N^{\prime}\) : A is non-unit triangular;
\(=\mathrm{U}\) ': A is unit triangular.
N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).
KD (input)
The num berof superdiagonals or subdiagonals of the triangularband \(m\) atrix A. KD \(>=0\).

A (input) The upper or low er triangular band \(m\) atrix A, stored in the firstkd+1 row sof the amay. The \(j\) th column ofA is stored in the \(j\) th column of the anay A as follow s: if UPLO = U',A (kd+1+i\(j, j)=A(i, 7)\) for \(\max (1, j k d)<=i<=j\); f UPLO \(=\) L', A \((1+i-j\rangle)=A(i, 7)\) for \(j<=i<=m\) in \((n, j+k d)\). IfD IA G = U', the diagonalelem ents of A are not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the amay A. LDA >= K D +1 .

\section*{RCOND (output)}

The reciprocal of the condition number of the \(m\) atrix \(A\), computed as RCOND \(=1 /(\) nom \((A)\) * norm (inv (A))).

W ORK (w orkspace)
dim ension ( \(3 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dttom v -perform one of the m atrix-vectoroperations \(\mathrm{x}:=\) \(A * x\), or \(x:=A * x\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DTBMV (UPLO,TRANSA,D IAG,N,K,A,LDA,Y, INCY)}
CHARACTER * 1UPLO,TRANSA,DIAG
\mathbb{NTEGERN,K,LDA,INCY}
DOUBLE PRECISION A (LDA,*),Y (*)
SUBROUT\mathbb{NE DTBM V_64 (UPLO,TRANSA,D IAG N,N,A ,LDA ,Y, INCY)}
CHARACTER * 1 UPLO,TRANSA,D IAG
\mathbb{NTEGER*8N,K,LDA,INCY}
DOUBLE PRECISION A (LDA,*),Y(*)

```
F95 INTERFACE
    SU BROUTINE TBMV (UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{N}], K, A,[L D A], Y,[\mathbb{N C Y}])\)
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
    \(\mathbb{N} T E G E R:: N, K, L D A, \mathbb{N} C Y\)
    REAL (8),D \(\mathbb{M}\) ENSION (:) ::Y
    REAL (8), D IM ENSION (:,:) ::A
    SU BROUT \(\mathbb{N} E\) TBM V_64 (UPLO, [TRANSA ], D \(\mathbb{A} G, \mathbb{N}], K, A,[L D A], Y\),
        [ \(\mathbb{N} C Y\) ])
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
    \(\mathbb{N}\) TEGER (8) ::N,K,LDA, \(\mathbb{N} C Y\)
    REAL (8), D \(\mathbb{M}\) ENSION (:) ::Y
    REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include < sunperfh>
void dtbm v (charuplo, chartransa, chardiag, intn, int k, double *a, int lda, double *y, int incy);
void dttom v_64 (charuplo, chartransa, char diag, long n, long \(k\), double *a, long lda, double *y, long incy);

\section*{PURPOSE}
dtbm \(v\) perform s one of the \(m\) atrix-vectoroperations \(x: A * x\), or \(x:=A\) * \(x\), where \(x\) is an \(n\) elem ent vectorand \(A\) is an \(n\) by \(n\) unit, or non-unit, upper or low er triangular band \(m\) atrix, \(w\) ith \((k+1)\) diagonals.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies w hether the \(m\) atrix is an upper or low er triangularm atrix as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or L ' \(A\) is an upper triangular \(m\) atrix.

UPLO = L' or I' A is a lower triangular \(m\) atrix.

U nchanged on exit.
TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) 'or \(h^{\prime} \mathrm{x}:=\mathrm{A} * \mathrm{x}\).
TRANSA \(=\) T'ort' \(x:=A * x\).

TRANSA \(=\) C'ort' \(\mathrm{x}:=\mathrm{A}\) * x .

U nchanged on exit.
TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)

O n entry, D IA G specifies w hether ornotA is unit triangular as follow \(s\) :

D \(\mathbb{A G G}=\mathrm{U}\) 'or \(\mathrm{L}^{\prime} \mathrm{A}\) is assum ed to be unit triangular.
\(D \mathbb{A} G=N\) 'or \(h\) ' \(A\) is notassumed to be unit triangular.

U nchanged on exit.

N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.
\(K\) (input)
On entry w th UPLO \(=U\) 'or L ', K specifies the num ber of super-diagonals of them atrix \(A\). On entry w ith UPLO = L' or I', K specifies the num ber of sub-diagonals of the \(m\) atrix \(A . K>=0\). U nchanged on exit.

A (input)
Before entry w th UPLO = U 'or G ', the leading ( \(k+1\) ) by \(n\) part of the array A m ust contain the upper triangularband part of the \(m\) atrix of coefficients, supplied colum \(n\) by colum \(n\), w ith the leading diagonal of the \(m\) atrix in row ( \(k+1\) ) of the anay, the firstsuper-diagonal starting at position 2 in row \(k\), and so on. The top leftk by \(k\) triangle of the amay A is not referenced. The follow ing program segm entw ill transfer an upper triangular band \(m\) atrix from conventional full \(m\) atrix storage to band storage:
\[
\begin{aligned}
& \text { DO } 20, J=1, N \\
& \text { M }=\mathrm{K}+1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{M} A X(1, \mathrm{~J}-\mathrm{K}), \mathrm{J} \\
& \text { A }(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \text { CONTINE } \\
& 20 \text { CONTINUE }
\end{aligned}
\]

Before entry w ith UPLO = L 'or I', the leading ( \(k+1\) ) by \(n\) part of the amay A \(m\) ust contain the low er triangularband part of the \(m\) atrix of coefficients, supplied colum n by colum n, w th the leading diagonal of the \(m\) atrix in row 1 of the array, the firstsub-diagonal starting atposition 1 in row 2 , and so on. The bottom right \(k\) by \(k\) triangle of the amay A is not referenced. The follow ing program segm entw ill transfer a low er
triangular band \(m\) atrix from conventional full \(m\) atrix storage to band storage:
\[
\begin{aligned}
& \text { DO } 20, J=1, N \\
& \mathrm{M}=1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{J}, \mathrm{M} \mathbb{N}(\mathrm{~N}, \mathrm{~J}+\mathrm{K}) \\
& \mathrm{A}(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \operatorname{CONTINUE} \\
& 20 \mathrm{CONTINUE}
\end{aligned}
\]
\(N\) ote thatw hen D \(\mathbb{A} G=U\) 'or L 'the elem ents of the array A comesponding to the diagonalelem ents of the \(m\) atrix are not referenced, but are assum ed to be unity. U nchanged on exit.
LDA (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD A \(>=\) ( \(\mathrm{k}+1\) ). U nchanged on exit.

Y (input/output)
\((1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) must contain the \(n\) elem ent vectorx. On exit, \(Y\) is overw rilten \(w\) th the tranform ed vector \(x\).
\(\mathbb{N C Y}\) (input)
O n entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of. \(\mathbb{N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtbrfs -provide errorbounds and backw ard error estim ates forthe solution to a system of linearequations \(w\) th a triangularband coefficientm atrix

\section*{SYNOPSIS}
```

SUBROUTINEDTBRFS (UPLO,TRANSA,DIAG,N,KD,NRHS,A,LDA,B,LDB,
X,LDX,FERR,BERR,W ORK,W ORK2,\mathbb{NFO)}

```
CHARACTER * 1 UPLO, TRANSA, D IA G
\(\mathbb{N}\) TEGER N,KD,NRHS,LDA,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGERWORK2(*)
DOUBLE PRECISION A (LDA,*), B (LDB,*), X (LDX,*), FERR (*),
BERR (*),W ORK (*)
SU BROUTINEDTBRFS_64 (UPLO, TRANSA, D \(\mathbb{I A G}, N, K D, N R H S, A, L D A, B\),
    LD \(B, X, L D X, F E R R, B E R R, W\) ORK,W ORK 2, \(\mathbb{N} F O\) )
CHARACTER * 1 UPLO, TRANSA, D IAG
\(\mathbb{N}\) TEGER*8N,KD,NRHS,LDA,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER *8 W ORK 2 ( \({ }^{\star}\) )
DOUBLE PRECISION A (LDA,*), B (LDB,*), X (LDX, \(\left.{ }^{\star}\right)\), FERR (*),
\(\operatorname{BERR}\) ( \()^{*}\), \(\mathrm{W} O \operatorname{OR}\) ( \({ }^{*}\) )

\section*{F95 INTERFACE}

SU BROUTINE TBRFS (UPLO, [TRANSA],D IAG,N,KD,NRHS,A, [LDA],B, [LD B ], X, [LDX ],FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
\(\mathbb{N} T E G E R:: N, K D, N R H S, L D A, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION (:) ::W ORK2
REAL (8),D IM ENSION (:) ::FERR,BERR,W ORK

REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A, B , X

SU BROUTINE TBRFS_64 (UPLO, [TRANSA ],D IA G,N, KD,NRHS,A, [LDA], \(B,[L D B], X,[L D X], F E R R, B E R R,[\mathbb{W} O R K],[W O R K 2],[\mathbb{N} F O])\)

CHARACTER ( \(L E N=1\) ) : : UPLO, TRANSA, D IA G
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{KD}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} \mathrm{A}, \mathrm{LD} \mathrm{B}, \mathrm{LD} \mathrm{X}, \mathbb{N}\) FO
\(\mathbb{N}\) TEGER (8), D \(\mathbb{I M} E N S I O N(:):: W O R K 2\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (: ::) : : A , B , X

\section*{C INTERFACE}
\#include <sunperfh>
void dtbrfs (charuplo, chartransa, chardiag, int \(n\), int
kd , int nins, double *a, int lda, double *b, int ldb, double *x, int ldx, double *ferr, double *berr, int *info);
void dtbrfś_64 (char uplo, chartransa, char diag, long n, long kd, long nihs, double *a, long lda, double *b, long ldb, double *x, long ldx, double *ferr, double *berr, long *info);

\section*{PURPOSE}
dtbrif provides errorbounds and backw ard error estim ates forthe solution to a system of linear equationsw ith a triangularband coefficientm atrix.

The solution \(m\) atrix \(X\) m ustbe com puted by STBTRS or some other \(m\) eans before entering this routine. STBRFS does not do iterative refinem entbecause doing so cannotim prove the backw ard error.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': A is upper triangular;
\(=\mathbb{L} ': A\) is low ertriangular.

TRANSA (input)
Specifies the form of the system ofequations:
\(=\mathrm{N}^{\prime}: A * X=B \quad\) (No transpose)
\(=T T^{\prime}: A * T * X=B \quad\) ( ranspose)
\(=C^{\prime}: A * * H * X=B \quad\) (C onjugate transpose \(=T\) ran -
spose)

TRANSA is defaulted to \(N\) 'forF95 \(\mathbb{I N}\) TERFACE .

D IA G (input)
\(=N^{\prime}: A\) is non-unittriangular;
\(=\mathrm{U}\) : A is unit triangular.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

KD (input)
The num ber of superdiagonals or subdiagonals of the triangularband m atrix A. KD \(>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the \(m\) atrices \(B\) and X. NRHS \(>=0\).

A (input) The upper or low er triangular band matrix A, stored in the firstkd+1 row s of the amay. The \(j\) th column of \(A\) is stored in the \(j\) th column of
the anray A as follow s: if \(\mathrm{UPLO}=\mathrm{U}, \mathrm{A}(\mathrm{kd}+1+\mathrm{i}\) \(j, j)=A(i, j)\) for \(\max (1, j k d)<=i<=j\) if \(U P L O=\) \(L^{\prime}, A(1+i-j, j=A(i, j)\) for \(j<=i<=m\) in \((n, j+k d)\). IfD IA G \(=\mathrm{U}\) ', the diagonalelem ents of A are not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the array A. LDA >= K D +1 .
\(B\) (input) The righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

X (input) The solution \(m\) atrix X .

LD X (input)
The leading dim ension of the anay X. LD X >= \(\max (1, N)\).

FERR (output)
The estim ated forw ard emorbound for each solution vector \(X\) ( 1\()\) (the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{J}), \operatorname{FERR}(\mathcal{I})\) is an estim ated upperbound forthem agnitude of the largest ele\(m\) ent in ( \(\mathrm{X}(\mathcal{1})-\mathrm{XTRUE}\) ) divided by the magnitude of the largestelem ent in X ( 7 ) . The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slightoverestim ate of the true emor.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector \(X\) ( \(\mathcal{j}\) ) (i.e., the sm allest relative change in any elem entofA orB thatm akes \(X(\mathcal{J})\) an exactsolution).

W ORK (w orkspace)
dim ension ( \(3 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{N N}\) FO = -i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtbsv - solve one of the system sofequations \(A * x=b\), or
A * \(\mathrm{x}=\mathrm{b}\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDTBSV (UPLO,TRANSA,D IAG,N,K,A,LDA,Y,INCY)}
CHARACTER * 1UPLO,TRANSA,DIAG
\mathbb{NTEGERN,K,LDA,INCY}
DOUBLE PRECISION A (LDA,*),Y (*)
SU BROUT\mathbb{NEDTBSV_64(UPLO,TRANSA,D IA G ,N ,K,A,LDA,Y, INCY)}
CHARACTER * 1UPLO,TRANSA,DIAG
\mathbb{NTEGER*8N,K,LDA, INCY}
DOUBLE PRECISION A (LDA,*),Y(*)

```
F95 INTERFACE
    SU BROUTINE TBSV (UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{N}], K, A,[L D A], Y,[\mathbb{N C Y}])\)
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
    \(\mathbb{N} T E G E R:: N, K, L D A, \mathbb{N} C Y\)
    REAL (8),D \(\mathbb{M}\) ENSION (:) ::Y
    REAL (8), D IM ENSION (:,:) ::A
    SU BROUTINE TBSV_64 (UPLO, [TRANSA ],D \(\mathbb{I} G, \mathbb{N}], K, A,[L D A], Y\),
        [ \(\mathbb{N} C Y\) ])
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
    \(\mathbb{N}\) TEGER (8) ::N,K,LDA, \(\mathbb{N} C Y\)
    REAL (8),D \(\mathbb{M}\) ENSION (:) ::Y
    REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dtbsv (charuplo, chartransa, chardiag, intn, int \(k\), double *a, int lda, double *y, int incy);
void dtbsv_64 (charuplo, chartransa, char diag, long n, long \(k\), double *a, long lda, double *y, long incy);

\section*{PURPOSE}
dtbsv solves one of the system sofequations \(A * x=b\), or \(A\) * \(x=b\), where \(b\) and \(x\) are \(n\) elem ent vectors and \(A\) is an \(n\) by \(n\) unit, or non-unit, upper or low er triangular band \(m\) atrix, \(w\) ith ( \(k+1\) ) diagonals.

N o test forsingularity or near-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies w hether the \(m\) atrix is an upper or low er triangularm atrix as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or \(\mathrm{L}^{\prime} \mathrm{A}\) is an upper triangular \(m\) atrix.

UPLO = L' or I' A is a lower triangular \(m\) atrix.

U nchanged on exit.

TRANSA (input)
O n entry, TRANSA specifies the equations to be solved as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} A * x=b\).
TRANSA \(=T^{\prime}\) or \(t^{\prime} A^{*} x=b\).

TRANSA \(=C^{\prime}\) ort' \(\mathrm{C}^{*} \mathrm{x}=\mathrm{b}\).

U nchanged on exit.

TRANSA is defaulted to N 'forF95 \(\mathbb{I N}\) TERFACE.

D IA G (input)
On entry, D IA G specifies w hether ornotA is unit triangular as follow s:

D \(\mathbb{I}\) G \(=U\) 'or \(\mathrm{l}^{\prime} A\) is assum ed to be unit triangular.

D IA G = N 'or h' A is notassumed to be unit triangular.

U nchanged on exit.

N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.

K (input)
On entry with UPLO \(=U\) 'or U ', \(K\) specifies the num ber of super-diagonals of them atrix \(A\). On entry w th UPLO = L' or I', K specifies the num ber of sub-diagonals of the \(m\) atrix \(A . K>=0\). U nchanged on exit.

A (input)
Before entry w ith UPLO = U 'or L ', the leading ( \(k+1\) ) by \(n\) part of the array A m ust contain the upper triangularband part of the \(m\) atrix of coefficients, supplied colum \(n\) by colum \(n\), w ith the leading diagonal of the \(m\) atrix in row ( \(k+1\) ) of the aray, the firstsuper-diagonalstarting at position 2 in row \(k\), and so on. The top leftk by \(k\) triangle of the array A is not referenced. The follow ing program segm entw illtransfer an upper triangular band \(m\) atrix from conventional full \(m\) atrix storage to band storage:

D O 20, J=1, N
\(\mathrm{M}=\mathrm{K}+1-\mathrm{J}\)
DO \(10, I=M A X(1, J-K), J\)
\(A(M+I, J)=m \operatorname{atrix}(I, J)\)
10 CONTINUE
20 CONTINUE
Before entry w ith UPLO = L 'or 1', the leading ( \(k+1\) ) by \(n\) part of the array A m ustcontain the low er triangularband part of the \(m\) atrix of coefficients, supplied column by colum n, w th the
leading diagonal of the m atrix in row 1 of the array, the firstsub-diagonalstarting atposition 1 in row 2 , and so on. The bottom right \(k\) by \(k\) triangle of the array A is not referenced. The follow ing program segm entw ill transfer a low er triangular band \(m\) atrix from conventional full \(m\) atrix storage to band storage:
\[
\begin{aligned}
& \text { DO } 20, \mathrm{~J}=1, \mathrm{~N} \\
& \mathrm{M}=1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{J}, \mathrm{M} \mathbb{N}(\mathrm{~N}, \mathrm{~J}+\mathrm{K}) \\
& \mathrm{A}(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \text { CONTINUE } \\
& 20 \mathrm{CONTINUE}
\end{aligned}
\]
\(N\) ote thatw hen D IA G = U 'or L'the elem ents of the amay A comesponding to the diagonalelem ents of the \(m\) atrix are not referenced, but are assum ed to be unity. U nchanged on exit.

LDA (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A >= ( \(k+1\) ). U nchanged on exit.

Y (input/output)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) must contain the \(n\) elem ent right-hand side vectorb. On exit, \(Y\) is overw ritten \(w\) ith the solution vector \(x\).
\(\mathbb{N} C Y\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y\). \(\mathbb{N C Y}<>0\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtbters - solve a triangular system of the form \(A * X=B\) orA \({ }^{* *}\) T * \(\mathrm{X}=\mathrm{B}\),

\section*{SYNOPSIS}
```

SUBROUTINEDTBTRS (UPLO,TRANSA,D IAG,N,KD,NRHS,A,LDA,B,LDB,
\mathbb{NFO)}
CHARACTER * 1 UPLO,TRANSA,DIAG
INTEGERN,KD,NRHS,LDA,LDB, NNFO
DOUBLE PRECISION A (LDA,*),B (LDB,*)
SU BROUT\mathbb{NE D TBTRS_64 (UPLO,TRANSA,D IA G ,N,KD,NRHS,A,LDA,B,}
LDB,\mathbb{NFO)}

```
CHARACTER * 1 UPLO, TRANSA, D IA G
\(\mathbb{N}\) TEGER*8N,KD,NRHS,LDA,LDB, \(\mathbb{N} F\) O
DOUBLE PRECISION A (LDA,*), B (LDB,*)

\section*{F95 INTERFACE}

SU BROUTINE TBTRS (UPLO, TRANSA, D IA G ,N,KD,NRHS,A, [LDA],B, [LDB], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G \(\mathbb{N} T E G E R:: N, K D, N R H S, L D A, L D B, \mathbb{N F O}\) REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B

SUBROUTINE TBTRS_64 (UPLO, TRANSA, D \(\mathbb{A} G, N, K D, N R H S, A,[L D A], B\), [LDB], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
\(\mathbb{N}\) TEGER (8) ::N,KD,NRHS,LDA, LDB, \(\mathbb{N} F O\)

REAL (8), D \(\mathbb{M} \operatorname{ENSION}(:,:\) ) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void dtbtrs (char uplo, char transa, chardiag, int n, int kd , int nins, double *a, int lda, double *b, int ldlo, int*info);
void dtbtrs_64 (charuplo, chartransa, char diag, long n, long kd, long næhs, double *a, long lda, double
*b, long ldb, long *info);

\section*{PURPOSE}
dtbtres solves a triangular system of the form
\(w\) here \(A\) is a triangularband \(m\) atrix of order \(N\), and \(B\) is an N boy NRHS m atrix. A check ism ade to verify thatA is nonsingular.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) : A is upper triangular;
\(=\mathbb{L}\) ': A is low ertriangular.

TRAN SA (input)
Specifies the form the system of equations:
\(=\mathrm{N}^{\prime}: \mathrm{A} * \mathrm{X}=\mathrm{B} \quad\) ( O otranspose)
\(=T: A * * T X=B \quad\) ( \(r\) ranspose)
\(=C\) : \(A * * H * X=B \quad\) (C onjugate transpose \(=T\) ran -
spose)

D IA G (input)
\(=\mathrm{N}\) : A is non-unit triangular;
\(=\mathrm{U}: \mathrm{A}\) is unit triangular.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

KD (input)
The num ber of superdiagonals or subdiagonals of the triangularband \(m\) atrix A. KD \(>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRH S \(>=0\).

A (input) The upper or low er triangular band matrix A, stored in the first kd+1 row sofA. The jth column ofA is stored in the \(j\) th column of the aray A as follow s: if UPLO = U',A (kd+1+i-j) = A \((i, j)\) form ax \((1, j k d)<=i<=\dot{j}\) if UPLO \(=\mathrm{L}\) ', \(A(1+i-j)=A(i, j)\) for \(j=i<=m\) in \((n, j+k d)\). If \(D \mathbb{A G}=U\) ', the diagonalelem ents of \(A\) are not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the array A. LDA >= K D +1.

B (input/output)
On entry, the righthand side m atrix B. On exit, if \(\mathbb{N} F O=0\), the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the anay B . LD B \(>=\) \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the \(i\)-th diagonalelem ent of \(A\) is zero, indicating that the \(m\) atrix is singular and the solutions \(X\) have notbeen com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dtgevc - com pute som e or all of the rightand/or left generalized eigenvectors of a pair of real upper triangular \(m\) atrices ( \(A, B\) )

\section*{SYNOPSIS}

SUBROUTINE DTGEVC (SDE,HOW MNY, SELECT,N,A,LDA,B,LDB,VL,LDVL, VR, LDVR, M M , M , W ORK , \(\mathbb{N F O}\) )

CHARACTER * 1 SIDE,HOWMNY
\(\mathbb{N} T E G E R N, L D A, L D B, L D V L, L D V R, M M, M, \mathbb{N F O}\) LOG ICAL SELECT (*)
D OUBLE PRECISION A (LDA, \(\left.{ }^{*}\right), B(L D B, \star), V L(L D V L, *), V R(L D V R, *)\), W ORK (*)

SU BROUTINEDTGEVC_64 (SDE,HOWMNY,SELECT,N,A,LDA,B,LDB,VL, LDVL, VR,LDVR,MM, M,WORK, \(\mathbb{N} F O)\)

CHARACTER * 1 SDE E HOW M NY
\(\mathbb{N} T E G E R * 8 N, L D A, L D B, L D V L, L D V R, M M, M, \mathbb{N} F O\) LOG ICAL*8 SELECT (*)
D OUBLE PRECISION A (LDA ,*), B (LDB,*), VL (LDVL,*),VR (LDVR,*), W ORK (*)

\section*{F95 INTERFACE}

SUBROUTINE TGEVC (SDE,HOW MNY,SELECT,N,A, [LDA],B, [LDB],VL,
\([\) [LD V ], VR, [LDVR], M M , M, [W ORK ], [ \(\mathbb{N F O}]\) )
CHARACTER (LEN=1)::SDE,HOW MNY
\(\mathbb{N} T E G E R:: N, L D A, L D B, L D V L, L D V R, M M, M, \mathbb{N} F O\)
LOG ICAL,D \(\mathbb{M}\) ENSION (:) ::SELECT

REAL (8), D \(\mathbb{I M}\) ENSION (:) ::W ORK
REAL (8), D \(\mathbb{M}\) ENSION (: ::) ::A , B , VL , VR

SU BROUTINE TGEVC_64 (SDE, HOW M NY, SELECT,N, A, [LDA ], B, [LD B ], VL, [LDVL],VR, [LDVR], MM,M, \([W\) ORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::SIDE,HOW MNY
\(\mathbb{N}\) TEGER (8) :: N , LDA , LD B , LDVL, LDVR , M M , M , \(\mathbb{N} F O\)
LOGICAL (8), D \(\mathbb{M}\) ENSION (:) :: SELECT
REAL (8), D \(\mathbb{I M}\) ENSION (:) ::W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A , B, VL, VR

\section*{C INTERFACE}
\#include <sunperfh>
void dtgevc (char side, charhow my, int *select, intn, double *a, int lda, double *b, int ldb, double *Vl, int ldvl, double *vr, int ldvr, int \(m m\), int *m, int*info);
void dtgevc_64 (charside, charhow m ny, long *select, long n, double *a, long lda, double *b, long ldb, double *Vl, long ldvl, double *vr, long ldvr, long mm, long *m, long *info);

\section*{PURPOSE}
dtgevc com putes som e orall of the right and/or left generalized eigenvectors of a pair of realuppertriangular \(m\) atrices \((A, B)\).

The rightgeneralized eigenvectorx and the leftgeneralized eigenvectory of ( \(A, B\) ) corresponding to a generalized eigenvalue w are defined by:
\[
(A-w B)^{*} x=0 \text { and } Y^{\star *} H *(A-w B)=0
\]
where \(y^{* * H}\) denotes the conjugate tranpose ofy.

If an eigenvalue \(w\) is determ ined by zero diagonal elem ents of both A and B, a unit vector is retumed as the corresponding eigenvector.

If alleigenvectors are requested, the routine \(m\) ay either retum the \(m\) atriges \(X\) and/or \(Y\) of rightor lefteigenvectors of \((A, B)\), or the products \(Z * X\) and/or \(Q * Y\), where \(Z\) and \(Q\) are input orthogonal \(m\) atrices. If \((A, B) w\) as obtained from the generalized real-Schur factorization of an originalpair of \(m\) atrices
\((A 0, B 0)=Q * A * Z * * H, Q * B * Z * * H)\),
then \(Z\) *X and \(Q\) *Y are the \(m\) atriges of right or lefteigenvectors of A.

A m ustbe block uppertriangular, w ith \(1-b y-1\) and \(2-b y-2\) diagonal blocks. C orresponding to each 2 -by-2 diagonal block is a com plex conjugate pair ofeigenvalues and eigenvectors; only one
eigenvector of the pair is com puted, nam ely the one corresponding to the eigenvalue \(w\) ith positive im aginary part.

\section*{ARGUMENTS}

SID E (input)
\(=R\) ': com pute righteigenvectors only;
\(=\mathrm{L}\) ': com pute lefteigenvectors only;
\(=\mathrm{B}\) ': com pute both rightand lefteigenvectors.

HOW M NY (input)
= 'A ': com pute all right and/or lefteigenvectors;
\(=\mathrm{B}\) ': com pute all right and/or lefteigenvectors, and backtransform them using the inputm atrices supplied in VR and/orVL; = S ': com pute selected right and/or lefteigenvectors, specified by the logicalaray SELECT.

SELECT (input)
If HOW M NY = S', SELECT specifies the eigenvectors to be com puted. If HOW M NY=A 'or B', SELECT is notreferenced. T o select the real eigenvector corresponding to the real eigenvalue w ( \()\), SELECT (J) m ustbe setto .TRUE. To select the com plex eigenvector corresponding to a com plex conjugate pairw ( 7 ) and w ( \(\mathfrak{j}+1\) ), either SELECT ( 7 ) orSELECT (j+1\()\) m ustbe setto .TRUE..

N (input) The order of the m atriges A and \(\mathrm{B} . \mathrm{N}>=0\).

A (input) The upperquasi-triangularm atrix A.

LD A (input)
The leading dim ension of array A. LD A \(>=\max (1\), N) 。

B (input) The uppertriangularm atrix B. IfA has a 2 -by -2 diagonal block, then the corresponding \(2-b y-2\) block ofB m ustbe diagonal w ith posilive ele\(m\) ents.

LD B (input)
The leading dimension of array B . LDB \(>=\) \(\max (1, N)\).

VL (input/output)
On entry, ifS \(\mathrm{D} E=\mathrm{L}^{\prime}\) or \(\mathrm{B}^{\prime}\) 'and HOW MNY = B',
VL must contain an N -by N m atrix Q (usually the orthogonalm atrix Q of leftSchurvectors retumed by SHGEQZ). On exit, ifS \(\mathbb{D} E=L^{\prime}\) 'or \(B^{\prime}\) 'VL contains: if HOW M NY = \(A\) ', them atrix \(Y\) of left eigenvectors of \((A, B)\); if HOW M NY = \(B\) ', the matrix Q *Y ; if HOW M NY = \(S^{\prime}\) ', the left eigenvectors of (A ,B ) specified by SELEC T, stored consecutively in the colum ns of V , in the same order as their eigenvalues. If \(S \mathbb{D} E=R \prime, V L\) is notreferenced.

A com plex eigenvector corresponding to a com plex eigenvalue is stored in tw o consecutive colum ns, the firstholding the realpart, and the second the im aginary part.

LDVL (input)
The leading dimension of array VL . LDVL >= \(\max (1, N)\) if \(S \mathbb{D} E=L\) 'or \(B^{\prime} ; L D V L>=1\) otherw ise.

VR (input/output)
On entry, ifS \(\mathrm{D}=\mathrm{R}\) 'or \(\mathrm{B}^{\prime}\) 'and HOW MNY = B ', VR m ust contain an N -by -N m atrix Q (usually the orthogonal matrix \(Z\) of right Schur vectors retumed by \(S H G E Q Z\) ). On exit, if \(S \mathbb{D} E=R\) 'or \(B\) ', VR contains: if HOW MNY = A', the matrix X of right eigenvectors of \((A, B)\); if HOW M NY \(=B\) ', the m atrix \(\mathrm{Z} * \mathrm{X}\); if H O W M NY = S ', the right eigenvectors of ( \(A, B\) ) specified by SELECT, stored consecutively in the colum \(n s\) ofVR, in the sam e order as their eigenvalues. If \(S \mathbb{D} E=\mathbb{L}, \mathrm{VR}\) is not referenced.

A com plex eigenvector corresponding to a com plex eigenvalue is stored in tw o consecutive colum ns, the firstholding the realpart and the second the im aginary part.

LDVR (input)
The leading dim ension of the array VR . LDVR >= \(\max (1, N)\) if \(S \mathbb{D} E=R\) 'or \(B^{\prime} ; L D V R>=1\) otherwise.

M M (input)
The num berof colum ns in the arrays VL and/or VR.
M M >= M .

M (output)
The num ber of colum ns in the arrays VL and/or VR actually used to store the eigenvectors. If
HOW M NY = A 'or B', M is set to N. Each selected real eigenvector occupies one colmm \(n\) and each selected com plex eigenvector occupies tw o colum ns.

W ORK (w orkspace)
dim ension (6*N )
\(\mathbb{I N} F O\) (output)
= 0: successfulexit.
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an illegalvalue.
> 0 : the 2 -by-2 block ( \(\mathbb{N}\) FO : \(\mathbb{N}\) FO +1) does nothave a com plex eigenvalue.

\section*{FURTHER DETAILS}

A llocation of w orkspace:

W ORK \((j)=1\)-norm of \(j\) th column of, above the diagonal
W ORK \((N+j)=1\)-norm of \(j\) th colum \(n\) ofB , above the diagonal
W ORK \((2 * \mathrm{~N}+1: 3 * \mathrm{~N})=\) realpartofeigenvector
W ORK ( \(3 * \mathrm{~N}+1: 4{ }^{*} \mathrm{~N}\) ) = im aginary partofeigenvector
W ORK ( \(4 * \mathrm{~N}+1: 5 * \mathrm{~N})\) = realpartofback-transform ed eigenvector
W ORK ( \(5 * \mathrm{~N}+1\) :6*N ) = im aginary part of back-transform ed eigenvector

Row w ise vs. colum nw ise solution m ethods:

Finding a generalized eigenvector consists basically of solving the singular triangular system
(A -w B) \(x=0\) (for right) or: (A -w B)**H \(y=0\)
(forleft)

Consider finding the i-th right eigenvector (assume all eigenvalues are real). The equation to be solved is: \(0=\operatorname{sum} C(j k) v(k)=\operatorname{sum} C(j k) v(k) \quad\) for \(j=i,\). ., 1
where \(C=(A-w B)\) (The com ponents \(v(i+1\) n \()\) are 0 .)

The "row w ise" m ethod is:
(1) \(\mathrm{v}(\mathrm{i}):=1\)
for \(j=i-1, \ldots, 1\) :
i
(2) com pute \(s=-\) sum \(C(j k) \vee(k)\) and \(\mathrm{k}=\mathrm{j}+1\)
(3) \(\vee(1)=s / C(j)\)

Step 2 is som etim es called the "dotproduct" step, since it is an innerproduct.betw een the jth row and the portion of the eigenvector that has been com puted so far.

The "colum nw ise" \(m\) ethod consists basically in doing the sum \(s\) forall the row \(s\) in parallel. A s each \(v(j)\) is com puted, the contribution ofv \((\mathcal{j})\) tim es the \(j\) th colum \(n\) of \(C\) is added to the partial sum s. Since FORTRAN anays are stored colum nw ise, this has the advantage that at each step, the elem ents of \(C\) that are accessed are adjacent to one another, w hereas w ith the row w ise \(m\) ethod, the elem ents accessed at a step are spaced LD A (and LD B ) w ords apart.

W hen finding lefteigenvectors, the \(m\) atrix in question is the transpose of the one in storage, so the row w isem ethod then actually accesses colum ns of A and B ateach step, and \(s o\) is the preferred \(m\) ethod.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dtgexc - reorder the generalized realS churdecom position of a real \(m\) atrix pair \((A, B)\) using an orthogonalequivalence transform ation \((A, B)=Q\) * \((A, B) * Z^{\prime}\),

\section*{SYNOPSIS}

SUBROUTINEDTGEXC \(\mathbb{N}\) ANTQ, \(W\) ANTZ, \(N, A, L D A, B, L D B, Q, L D Q, Z, L D Z\), \(\mathbb{F} S T, \mathbb{L} S T, W\) ORK,LWORK, \(\mathbb{N} F O\) )
\(\mathbb{N}\) TEGER N,LDA,LDB,LDQ,LD Z, \(\mathbb{F S T}, \mathbb{L} S T, L W O R K, \mathbb{N} F O\) LOG ICALW ANTQ, W ANTZ
DOUBLE PRECISION A (LDA,*), B (LDB,*), Q (LDQ,*), Z (LDZ,*), W ORK (*)

SU BROUTINEDTGEXC_64 \(\mathbb{N} A N T Q, W\) ANTZ,N,A,LDA,B,LDB,Q,LDQ,Z,LDZ, \(\mathbb{F} S T, \mathbb{L} S T, W\) ORK,LW ORK, \(\mathbb{N} F O\) )
\(\mathbb{N}\) TEGER*8N,LDA,LD B,LDQ,LD Z, \(\mathbb{F} S T, \mathbb{L} S T, L W\) ORK, \(\mathbb{N} F O\)
LOGICAL*8W ANTQ,WANTZ
DOUBLE PRECISION A (LDA,*), B (LDB,*), Q (LDQ,*), Z (LDZ,*), WORK (*)

\section*{F95 INTERFACE}

SUBROUTINE TGEXC (NANTQ,W ANTZ,N,A, [LDA],B,[LDB],Q,[LDQ],Z, [LD Z ], \(\mathbb{F S T}, \mathbb{L} S T,\left[\begin{array}{l}\text { O ORK ], [LW ORK ], [ } \mathbb{N F O}])\end{array}\right.\)
\(\mathbb{N} T E G E R::\) N,LDA,LDB,LDQ,LD Z, \(\mathbb{F S T}, \mathbb{L S T}, L W\) ORK, \(\mathbb{N} F O\)
LOGICAL ::W ANTQ,W ANTZ
REAL (8), D IM ENSION (:) ::W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A, \(B, Q, Z\)

SUBROUTINETGEXC_64 (NANTQ,WANTZ,N,A,[LDA],B,[LDB],Q,[LDQ],Z, [LD Z], \(\mathbb{F} S T, \mathbb{L} S T,\left[\begin{array}{l}\text { ORK ], [LW ORK ], [ } \mathbb{N F O} \text { ]) }\end{array}\right.\)

LOG ICAL (8) ::W ANTQ,W ANTZ
REAL (8),D IM ENSION (:) ::W ORK
REAL (8),D \(\mathbb{M}\) ENSION (: : : : ::A, B, \(Q, Z\)

\section*{C INTERFACE}
\#include < sunperfh>
void dtgexc (intw antq, intw antz, intn, double *a, int lda, double *b, int ldb, double *q, int ldq, double *z, int ldz, int *ifst, int *ilst, int *info);
void dtgexc_64 (long w anta, long w antz, long n, double *a, long lda, double *b, long ldb, double *q, long ldq, double *z, long ldz, long *ifst, long *ilst, long *info);

\section*{PURPOSE}
dtgexc reorders the generalized realS churdecom position of a real \(m\) atrix pair ( \(A, B\) ) using an orthogonalequivalence transform ation
so that the diagonalblock of ( \(A, B\) ) w ith row index \(\mathbb{F} S T\) is m oved to row IIST.
(A , B ) m ustbe in generalized realSchurcanonical form (as retumed by SGGES), i.e.A is block uppertriangularw ith \(1-b y-1\) and \(2-b y-2\) diagonalblocks. \(B\) is uppertriangular.

O ptionally, the m atrices Q and Z of generalized Schur vectors are updated.

Q (in) * A (in) * Z (in) \({ }^{\prime}=\mathrm{Q}\) (out) * A (out) * Z (out)'
Q (in) * B (in) * Z (in) \({ }^{\prime}=\mathrm{Q}\) (out) * \(\mathrm{B}(\) out \() ~ * Z(\text { out })^{\prime}\)

\section*{ARGUMENTS}

W ANTQ (input)
W ANTZ (input)
N (input) The order of the m atrices A and \(\mathrm{B} . \mathrm{N}>=0\).

A (input/output)

On entry, the m atrix A in generalized real Schur canonical form. On exit, the updated \(m\) atrix A, again in generalized realSchur canonical form .

LD A (input)
The leading dim ension of the aray A. LD A >= \(\max (1, N)\).

B (input/output)
On entry, the m atrix B in generalized real Schur canonical form ( \(A, B\) ). On exit, the updated \(m\) atrix \(B\), again in generalized realSchur canonical form ( \(A, B\) ).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

Q (input/output)
On entry, ifW ANTQ = TRUE, the orthogonalm atrix \(Q\). On exit, the updatedmatrix \(Q\). If W ANTQ \(=\) FALSE., Q is not referenced.

LD Q (input)
The leading dim ension of the array \(\mathrm{Q} . \mathrm{LD} \mathrm{Q}>=1\).
IfW ANTQ = TRUE., LDQ \(>=\mathrm{N}\).
Z (input/output)
On entry, ifW ANTZ = TRUE., the orthogonalm atrix Z . On exit, the updatedm atrix Z . If W ANTZ \(=\) FALSE., Z is not referenced.

LD \(Z\) (input)
The leading dim ension of the array \(\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1\).
IfW ANTZ = .TRUE, ,LD Z >=N.
IFST (input/output)
Specify the reordering of the diagonal blocks of ( \(A, B\) ). The block w th row index \(\mathbb{F} S T\) ism oved to row ILST, by a sequence of Sw apping betw een adjacentblocks. On exit, if IFST pointed on entry to the second row of a 2 -by- 2 block, it is changed to point to the first row ; ILST alw ays points to the firstrow of the block in its final position (which \(m\) ay differ from its inputvalue by +1 or \(-1) .1<=\mathbb{F} S T, \mathbb{L} S T<=N\).

\section*{IST (input/output)}

See the description of \(\mathbb{F S T}\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the amay W ORK.LW ORK >=4*N + 16.

IfLW O RK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of theW ORK amay, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N}\) FO (output)
=0: successfulexit.
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvahue.
=1: The transform ed m atrix pair ( \(A\), B) w ould be too far from generalized Schur form ; the problem is ill-conditioned. (A , B ) m ay have been partially reordered, and ILST points to the first row of the currentposition of the block being m oved.

\section*{FURTHER DETAILS}

B ased on contributions by
B o K agstrom and PeterPorom aa, D epartm ent of C om puting Science,

Um ea U niversity, S-901 87 U m ea, Sw eden.
[1] B . K agstrom ; A D irectM ethod forReordering Eigenvalues in the

G eneralized RealSchurForm ofa RegularM atrix Pair (A, B) , in

M S .M oonen etal (eds), LinearA lgebra forLarge Scale and

R eal-T im e A pplications, K luw erA cadem ic Publ. 1993, pp 195-218.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dtgsen - reorder the generalized realSchurdecom position of a real \(m\) atrix pair ( \(A, B\) ) (in term sofan orthonorm al equivalence trans-form ation \(Q\) '* \((A, B)\) * \(Z\) ), so that a selected cluster of eigenvalues appears in the leading diagonalblocks of the upperquasi-triangularm atrix A and the upper triangularB

\section*{SYNOPSIS}

SUBROUTINEDTGSEN (LOBB,WANTQ,W ANTZ, SELECT,N,A,LDA, B, LDB, A LPHAR,ALPHA I, BETA, \(Q, L D Q, Z, L D Z, M, P L, P R, D \mathbb{F}, W\) ORK, LW ORK, \(\mathbb{I W}\) ORK, LIW ORK, \(\mathbb{N} F O\) )
\(\mathbb{N} T E G E R \operatorname{LJ} B, N, L D A, L D B, L D Q, L D Z, M, L W O R K, L \mathbb{N} O R K, \mathbb{N} F O\) \(\mathbb{I N T E G E R} \mathbb{I N}\) ORK (*)
LOG ICALW ANTQ, W ANTZ
LOG ICAL SELECT (*)
DOUBLE PRECISION PL, PR
 \(\operatorname{BETA}(\star), \mathrm{Q}(\mathbb{L D} Q, \star), \mathrm{Z}(\mathbb{L D} Z, \star), \mathrm{D} \mathbb{F}\left({ }^{\star}\right), \mathrm{W} O R K\left({ }^{*}\right)\)

SU BROUTINEDTGSEN_64 (LDBB,W ANTQ,W ANTZ,SELECT,N,A,LDA,B,LDB, A LPHAR,ALPHAI,BETA, \(Q, L D Q, Z, L D Z, M, P L, P R, D \mathbb{F}, W\) ORK, LW ORK, \(\mathbb{I N}\) ORK,LIW ORK, \(\mathbb{N} F O\) )
\(\mathbb{N} T E G E R * 8\) IJO B , N, LDA , LD B, LDQ \(, ~ L D ~ Z, ~ M, ~ L W ~ O R K, ~ L I N ~ O R K, ~\) \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER *8 \(\mathbb{I N}\) ORK (*)
LOGICAL*8W ANTQ,WANTZ
LOGICAL*8SELECT (*)
D OUBLE PREC ISION PL, PR
 \(\operatorname{BETA}(*), Q(\mathbb{L D} Q, *), \mathrm{Z}(\mathrm{LD} Z, *), D \mathbb{F}(\star), \mathrm{WORK}(*)\)

\section*{F95 INTERFACE}

SU BROUT INE TG SEN (LOB B , W ANTQ,W ANTZ, SELECT, N, A, [LDA], B, [LD B], A LPHAR, ALPHAI, BETA, Q , [LDQ], Z, [LD Z],M,PL,PR, D \(\mathbb{F}\), [W ORK], \([L W\) ORK \(],[\mathbb{W} O R K],[L \mathbb{W} O R K],[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER :: IJOB,N,LDA,LDB,LDQ,LDZ, M, LW ORK, LIW ORK, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{I W}\) ORK
LOGICAL ::W ANTQ, W ANTZ
LOG ICAL, D \(\mathbb{I M} E N S I O N\) (:) :: SELECT
REAL (8) :: PL, PR
REAL (8), D \(\mathbb{M}\) ENSION (:) ::ALPHAR,ALPHAI,BETA,D \(\mathbb{F}, W\) ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) :: A , B, Q , Z
SU BROUTINE TG SEN_64 (LOB B W ANTQ, W ANTZ, SELECT ,N,A, [LDA ], B, [LD B], \(A L P H A R, A L P H A I, B E T A, Q,[L D Q], Z,[L D Z], M, P L, P R, D \mathbb{F},[W O R K]\),
\([\) [W ORK \(],[\mathbb{W}\) ORK \(],[\mathbb{I} \mathbb{V}\) ORK \(],[\mathbb{N} F O])\)
\(\mathbb{N} \operatorname{TEGER}(8):: \operatorname{IOB}, N, L D A, L D B, L D Q, L D Z, M, L W O R K, L \mathbb{N}\) ORK,
\(\mathbb{N}\) FO
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{M} \operatorname{ENSION(:)::\mathbb {IW}ORK}\)
LOGICAL (8) ::W ANTQ,W ANTZ
LOG ICAL (8), D \(\mathbb{M}\) ENSION (:) :: SELECT
REAL (8) ::PL, PR
REAL (8), D \(\mathbb{M} E N S I O N(:):: A L P H A R, A L P H A I, B E T A, D \mathbb{F}, W\) ORK
REAL (8), D \(\mathbb{M} \operatorname{ENSION}(:,:):: A, B, Q, Z\)

\section*{C INTERFACE}
\#include <sunperfh>
void dtgsen (intijob, intw antor, intw antz, int *select, int
n , double *a, int lda, double *b, int ldb, double
*alphar, double *alphai, double *beta, double *q, int ldq, double * z , int ldz, int *m, double *pl, double *pr, double *dif, int *info);
void dtgsen_64 (long ijob, long w anta, long w antz, long
*select, long n, double *a, long lda, double *b, long lalb, double *alphar, double *alphai, double
*beta, double *q, long ldq, double *z, long ldz, long *m , double *pl, double *pr, double *dif, long *info);

\section*{PURPOSE}
dtgsen reorders the generalized realSchur decom position of a real matrix pair (A, B) (in term sofan orthonorm al
equivalence trans-form ation \(Q^{\prime} *(A, B)\) * \(Z\) ), so that a selected cluster of eigenvalues appears in the leading diagonalblocks of the upperquasi-triangularm atrix A and the upper triangular B. The leading colum ns of \(Q\) and \(Z\) form orthonorm albases of the corresponding left and righteigenspaces (deflating subspaces). (A,B) m ustbe in generalized realSchurcanonical form (as retumed by SG GES), ie.A is block upper triangular w ith 1 -by-1 and 2 -by-2 diagonal blocks.B is upper triangular.

D TG SEN also com putes the generalized eigenvalues
\[
w(\mathcal{j})=\left(\operatorname{ALPHAR}(\mathcal{j})+i^{\star} \operatorname{ALPHAI}(\mathcal{j})\right) B E T A(\mathcal{j})
\]
of the reordered \(m\) atrix pair ( \(A, B\) ).
Optionally, D TGSEN com putes the estim ates of reciprocal condition num bers foreigenvalues and eigenspaces. These are D ifu [ \(A 11, B 11\) ), (A 22, B 22)] and D ifl[ (A 11, B11), (A 22, B22)], i.e. the separation ( \(s\) ) betw een the \(m\) atrix pairs ( \(A 11, B 11\) ) and (A 22,B22) that correspond to the selected cluster and the eigenvalues outside the cluster, resp., and norm sof "pro jections" onto left and right eigenspaces w r.t. the selected cluster in the ( 1,1 )-block.

\section*{ARGUMENTS}
```

INO B (input)
Specifies w hether condition num bers are required
for the cluster ofeigenvalues (PL and PR) orthe
deflating subspaces (D ifu and D ifl):
=0:Only reorderw r.t.SELEC T .N o extras.
=1:Reciprocal ofnorm s of "pro jections" onto left
and righteigenspaces w r.t. the selected cluster
(PL andPR). = 2:U pperbounds on D ifu and D ifl.
F-norm -based estim ate
(D \mathbb{F (1:2)).}
=3:Estim ate ofD ifu and D ifl.1-norm -based esti-
m}\mathrm{ ate
(D IF (1:2)). A bout5 tim es as expensive as INO B =
2. =4: Com pute PL,PR and D FF (ie.0,1 and 2
above): Econom ic version to get it all. =5: C om -
putePL,PR and D FF (i.e.0,1 and 3 above)

```
W ANTQ (input)

\section*{SELECT (input)}

SELEC T specifies the eigenvalues in the selected cluster. To select a real eigenvalue w ( \(\mathcal{j}\) ), SELECT ( \(j\) ) must be set to \(\mathrm{w}(\mathcal{j})\) and \(\mathrm{w}(j+1)\), corresponding to a 2 -by-2 diagonalblock, either SELECT ( \(\ddagger\) ) orSELECT ( \(j+1\) ) orboth \(m\) ust be set to either both included in the cluster or both excluded.

N (input) The order of the m atrioes A and \(\mathrm{B} . \mathrm{N}>=0\).
A (input/output)
On entry, the upper quasi-triangular \(m\) atrix \(A\), w ith (A, B) in generalized realSchur canonical form. On exit, A is overw rilten by the reordered \(m\) atrix A.

\section*{LD A (input)}

The leading dim ension of the aray A. LD A >= \(\max (1, N)\).

B (input/output)
O n entry, the uppertriangularm atrix \(B\), \(w\) th \(A\), B) in generalized realSchurcanonical form. On exit, \(B\) is overw ritten by the reordered \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, \mathbb{N})\).

ALPHAR (output)
On exit, (ALPHAR ( ) + ALPHAI ( ) *i) BETA ( ) , \(\dot{j} 1, \ldots, N, w i l l\) be the generalized eigenvalues. A LPHAR \((\mathcal{Z})+\) ALPHAI \((\boldsymbol{j} *\) iand BETA \((\mathcal{j}), \dot{于} 1, \ldots, N\) are the diagonals of the com plex Schur form \((S, T)\) that w ould result if the 2 -by-2 diagonalblocks of the real generalized Schur form of ( \(A, B\) ) w ere further reduced to triangular form using com plex unitary transform ations. If A LPHA I( \()\) is zero, then the \(j\) th eigenvalue is real; ifpositive, then the \(j\) th and ( \(j+1\) )-steigenvahues are a com plex con \(j \mu-\) gate pair, w ith A LPH A I(j+1) negative.

\section*{A LPH A I (output)}

See the description of A LPHAR.
BETA (output)
See the description of A LPHAR.
Q (input/output)

On entry, if \(W\) ANTQ = TRUE., \(Q\) is an \(N\)-by \(N\) \(m\) atrix. On exit, \(Q\) has been postm ultiplied by the left orthogonal transform ation \(m\) atrix which reorder ( \(A\), B); The leading \(M\) colum ns of f form orthonorm albases for the specified pair of left eigenspaces (deflating subspaces). If \(W\) ANTQ \(=\) FALSE., Q is notreferenced.

LD Q (input)
The leading dim ension of the array \(\mathrm{Q} . \mathrm{LD} Q>=1\); and ifW ANTQ = .TRUE.,LDQ >=N.

Z (input/output)
On entry, if \(W\) ANTZ \(=\).TRUE., \(Z\) is an \(N\) by \(-N\) \(m\) atrix. O n exit, Z has been postm ultiplied by the left orthogonal transform ation \(m\) atrix \(w\) hich reorder ( \(A\), B); The leading \(M\) colum ns of \(Z\) form orthonorm albases for the specified pair of left eigenspaces (deflating subspaces). If W ANTZ = FALSE., Z is notreferenced.

LD Z (input)
The leading dim ension of the array Z . LD \(\mathrm{Z}>=1\); If \(W A N T Z=. T R U E, \operatorname{LDZ}>=N\).

M (output)
The dim ension of the specified pair of left and right eigen-spaces (deflating subspaces). \(0<=\mathrm{M}\) \(<=N\).

PL (output)
If \(\mathrm{IJOB}=1,4\) or \(5, \mathrm{PL}, \mathrm{PR}\) are low er bounds on the reciprocal of the norm of "projections" onto leftand righteigenspaces w ith respect to the selected cluster. \(0<\mathrm{PL}, \mathrm{PR}<=1\). IfM \(=0\) orM \(=N, P L=P R=1\). IfIOOB \(=0,2\) or \(3, P L\) and \(P R\) are not referenced.

PR (output)
See the description ofPL .

D \(\mathbb{F}\) (output)
If IUO B >=2,D \(\mathbb{F}(1: 2)\) store the estim ates ofD ifu and \(D\) iff.
If \(\mathrm{IHO} B=2\) or \(4, D \mathbb{F}(1: 2)\) are F -norm Hoased upper bounds on
\(D\) ifu and \(D\) ifl. If \(I N O B=3\) or \(5, D \mathbb{F}(1: 2)\) are \(1-\)
norm -based estim ates ofD ifu and \(D\) ifl. \(\mathrm{If} M=0\)
orN , \(D \mathbb{F}(1: 2)=F-n o r m([A, B])\). If \(I J O B=0\) or \(1, D\) IF is notreferenced.

W ORK (w orkspace)
If \(\mathrm{IJOB}=0, \mathrm{~W} O R K\) is not referenced. O therw ise, on exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the amay \(W\) ORK. LW ORK \(>=4 * N+16\). If \(\mathrm{LJO} B=1,2\) or 4, LW ORK \(>=\mathrm{MAX}(4 * N+16,2 * \mathrm{M} * \mathbb{N}-\) \(\mathrm{M})\) ). If \(\mathrm{IJOB}=3\) or 5 , LW ORK \(>=\mathrm{MAX}(4 * \mathrm{~N}+16\), \(4 * M *(N-M))\).

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the \(W\) ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IV ORK (w orkspace/output)
If \(\mathrm{IJO} B=0, \mathbb{I V} O R K\) is not referenced. O therw ise, on exit, if \(\mathbb{N} F O=0, \mathbb{I V}\) ORK (1) retums the optim al LIN ORK.

LIV ORK (input)
The dim ension of the array \(\mathbb{I V}\) ORK.LIIN ORK >=1. If \(\mathrm{IJOB}=1,2\) or \(4, \mathrm{~L} \mathbb{I}\) ORK \(>=\mathrm{N}+6\). If IJOB \(=3\) or \(5, L \mathbb{N}\) ORK \(>=M A X(2 * M * \mathbb{N}+M), N+6)\).

If LIV ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the \(\mathbb{I V}\) ORK array, retums this value as the first entry of the \(\mathbb{I N}\) ORK array, and no errorm essage related to \(L \mathbb{I N} O R K\) is issued by XERBLA.
\(\mathbb{I N} F O\) (output)
=0: Successfiulexit.
\(<0:\) If \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegal value.
\(=1\) : Reordering of ( \(A, B\) ) failed because the transform ed \(m\) atrix pair ( \(A, B\) ) w ould be too far from generalized Schur form ; the problem is very ill-conditioned. (A, B) m ay have been partially reordered. If requested, 0 is retumed in D IF (*), \(P L\) and \(P R\).

\section*{FURTHER DETAILS}

D TG SEN first collects the selected eigenvalues by com puting orthogonal \(U\) and \(W\) thatm ove them to the top leftcomerof (A,B). In otherw ords, the selected eigenvalues are the
eigenvalues of (A11, B 11) in:
\[
\begin{gathered}
U *(A, B) * W=(A 11 A 12)(B 11 B 12) n 1 \\
(0 \text { A 22),(0 B22) n2 } \\
n 1 \mathrm{n} 2 \quad \mathrm{n} 1 \mathrm{n} 2
\end{gathered}
\]
w here \(\mathrm{N}=\mathrm{n} 1+\mathrm{n} 2\) and U ' m eans the transpose of U . The first n1 colum ns of \(U\) and \(W\) span the specified pair of left and righteigenspaces (deflating subspaces) of \((A, B)\).

If ( \(A, B\) ) has been obtained from the generalized real Schur decom position of a matrix pair ( \(C, D\) ) \(=Q\) * \((A, B) * Z\) ', then the reordered generalized realSchur form of ( \(C, D\) ) is given by
\[
(C, D)=(Q * U)^{\star}(U \star(A, B) * W) *(Z * W)^{\prime},
\]
and the firstn1 colum ns of Q * U and \(\mathrm{Z} * \mathrm{~W}\) span the comesponding deflating subspaces of ( \(C, D\) ) \(Q\) and \(Z\) store \(Q * U\) and Z *W, resp.).

N ote that if the selected eigenvalue is sufficiently illconditioned, then its value \(m\) ay differ significantly from its value before reordering.

The reciprocalcondition num bers of the left and right eigenspaces spanned by the firstn1 colum ns ofU and \(W\) (or \(\mathrm{Q} * \mathrm{U}\) and \(\mathrm{Z} * \mathrm{~W}) \mathrm{m}\) ay be retumed in \(\mathrm{D} \mathbb{F}(1: 2)\), corresponding to \(D\) ifu and \(D\) ifll, resp.

The D ifu and D iflare defined as:
ifu \([\) A 11, B11), (A 22, B22) \(]=\operatorname{sigm}\) am in ( Zu )
and
where sigm a-m in ( 2 u ) is the sm allest singular value of the ( \(2 *_{n} 1 *{ }_{n} 2\) )-by- \(\left(2 *_{n} 1 *_{n} 2\right.\) ) m atrix
\(\mathrm{u}=[\mathrm{kron}(\mathrm{In} 2, \mathrm{~A} 11)-\mathrm{kron}(\mathrm{A} 22\) ', In1) ]
[kron(In2,B11) kron (B22', In1)].
H ere, \(\operatorname{In} x\) is the identity \(m\) atrix of size \(n x\) and \(A 22\) 'is the transpose of A 22. kron ( \(X, Y\) ) is the \(K\) roneckerproduct betw een the \(m\) atrices \(X\) and \(Y\).

W hen D IF (2) is sm all, sm all changes in (A, B) can cause large changes in the deflating subspace. A \(n\) approxim ate (asym ptotic) bound on them axim um angularemor in the com puted deflating subspaces is PS * norm ( \((A, B)\) )/D \(\mathbb{F}(2)\),
where EPS is the \(m\) achine precision.

The reciprocal norm of the pro jectors on the left and right eigenspaces associated with (A 11, B 11) m ay be retumed in PL and PR. They are com puted as follow s. First we com pute \(L\) and \(R\) so that \(P\) * \((A, B){ }^{\star} Q\) is block diagonal, w here
\(=(I-\amalg) n 1 \quad Q=(I R) n 1\)
\[
\begin{array}{lll}
(0 \text { I) n2 } & \text { and } & (0 I) n 2 \\
\text { n1 n2 } & \text { n1 n2 }
\end{array}
\]
and ( \(L, R\) ) is the solution to the generalized Sylvester equation \(11 * \mathrm{R}-\mathrm{L} *\) A \(22=-\mathrm{A} 12\)

Then PL \(=(F \text {-norm }(L) * * 2+1)^{* *}(-1 / 2)\) and \(P R=(F-\) norm \((\mathbb{R}) * * 2+1) * *(-1 / 2)\). A n approxim ate (asym ptotic) bound on the average absolute error of the selected eigenvalues is PS * norm ( \((A, B)) / P L\).

There are also globalemorbounds which valid forperturbations up to a œertain restriction: A low erbound \((x)\) on the sm allest \(F\)-norm ( \(E, F\) ) forw hich an eigenvalue of (A 11, B 11) \(m\) ay \(m\) ove and coalesce \(w\) th an eigenvalue of (A 22,B22) under perturbation \((\mathbb{E})\), (i.e. \((A+E, B+F)\), is
\(\mathrm{x}=\)


A \(n\) approxim ate bound on \(x\) can be com puted from \(D \mathbb{F}(1: 2)\), PL and PR .

If \(y=(F-\) norm \((E, F) / x)<=1\), the angles betw een the per turbed (L', R) and unperturbed ( \(L, R\) ) left and right deflating subspaces associated \(w\) ith the selected cluster in the \((1,1)\)-blocks can be bounded as
max-angle \((L, L)<=\arctan (y * P L /(1-y *(1-P L *\) PL)** (1/2))
\(\max -\operatorname{angle}(R, R)<=\arctan (y * P R /(1-y *(1-P R *\)
\(\operatorname{PR}) * *(1 / 2))\)
See LA PA CK U ser's G uide section 4.11 or the follow ing references form ore inform ation.

N ote that if the default \(m\) ethod for com puting the Frobenius-norm - based estim ate D \(\mathbb{F}\) is not wanted (see SLA TDF), then the param eter ID \(\mathbb{F} \cdot \mathcal{B}\) (see below) should be changed from 3 to 4 (routine SLA TDF (LOB \(=2 \mathrm{w}\) illbe used)). See STG SY L form ore details.

B ased on contributions by
Bo K agstrom and PeterPorom aa, D epartm ent of Com puting Science,

Um ea U niversity, S-901 87 Um ea, Sw eden.

\section*{References}
[1] B .K agstrom ; A D irectM ethod forReordering Eigenvalues in the

G eneralized RealSchurForm of a RegularM atrix Pair (A, B), in

M S.M oonen etal (eds), LinearA lgebra for Large Scale and

Real-T im e A pplications, K luw erA cadem ic Publ. 1993, pp 195-218.
[2] B . K agstrom and P.Porom aa; C om puting Eigenspaces w th Specified E igenvalues of a R egularM atrix Pair ( \(A\), \(B\) ) and Condition

Estim ation: Theory, A lgorithm s and Softw are, ReportU M IN F - 94.04 , D epartm ent of Computing Science, Um ea

U niversity, S-901 87 Um ea, Sw eden, 1994.A lso as LA PA CK W orking

N ote 87. To appear in N um ericalA lgorithm \(\mathrm{s}, 1996\).
[3] B . K agstrom and P .Porom aa, LA PA C K -Style A lgorithm s and Softw are
for Solving the Generalized Sylvester Equation and Estim ating the Separation betw een RegularM atrix Pairs, ReportU M IN F 93 23,

D epartm entofC om puting Science, Um ea U niversity, S-901 87 Um ea,

Sw eden, D ecem ber 1993, Revised A pril 1994, A lso as LAPACK W orking

N ote 75. To appear in A CM Trans. on M ath. Softw are, V ol 22, N ○ 1,
1996.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtgsja - com pute the generalized singular value decom position (G SVD) of two realupper triangular (ortrapezoidal) \(m\) atrices \(A\) and \(B\)

\section*{SYNOPSIS}
```

SUBROUTINE DTGSJA (JOBU,JOBV,JOBQ,M,P,N,K,L,A,LDA,B,LDB,
TOLA,TOLB,ALPHA,BETA,U,LDU,V,LDV,Q,LDQ,W ORK,NCYCLE,
\mathbb{NFO)}

```
CHARACTER * 1 JOBU, \(0 \mathrm{OBV}, \mathrm{JOBQ}\)
\(\mathbb{N} T E G E R M, P, N, K, L, L D A, L D B, L D U, L D V, L D Q, N C Y C L E, \mathbb{N} F O\)
DOUBLE PRECISION TOLA, TOLB
DOUBLE PRECISION A (LDA,*), B (LDB,*), ALPHA (*), BETA (*),
U (LDU,*), V (LDV,*), Q (LDQ ,*), W ORK (*)
SU BROUTINE DTGSJA_64 (JOBU, \(\mathcal{J O B V}, \mathcal{J O Q}, \mathrm{M}, \mathrm{P}, \mathrm{N}, \mathrm{K}, \mathrm{L}, \mathrm{A}, \mathrm{LD} A, B, L D B\),
    TOLA,TOLB,ALPHA,BETA,U,LDU,V,LDV,Q,LDQ,W ORK,NCYCLE,
    \(\mathbb{N} F O\) )
CHARACTER * 1 JOBU, JOBV , JOBQ
\(\mathbb{N} T E G E R * 8 M, P, N, K, L, L D A, L D B, L D U, L D V, L D Q, N C Y C L E\),
\(\mathbb{N}\) FO
DOUBLE PRECISION TOLA, TOLB
DOUBLE PRECISION A (LDA, *), B (LDB,*), ALPHA (*), BETA (*),
U(LDU, \(\left.)^{*}\right), \mathrm{V}(\mathrm{LDV}, \star), \mathrm{Q}(\mathrm{LD} Q, \star), \mathrm{W} O R K\left({ }^{( }\right)\)

\section*{F95 INTERFACE}

SU BROUTINE TGSJA (JOBU, JOBV, JOBQ, M, \(\mathrm{P}, \mathrm{N}, \mathrm{K}, \mathrm{L}, \mathrm{A},[\mathrm{LDA}], \mathrm{B},[\mathrm{LDB}]\), TO LA, TOLB,ALPHA,BETA, U, [LDU],V, [LDV],Q, [LDQ], [W ORK], NCYCLE, [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) :: JOBU, \(0 \mathrm{OBV}, \mathrm{JOBQ}\)
\(\mathbb{N}\) TEGER ::M, \(\mathrm{P}, \mathrm{N}, \mathrm{K}, \mathrm{L}, \mathrm{LDA}, \mathrm{LD} B, L D \mathrm{U}, \mathrm{LDV}, \mathrm{LD} Q, \mathrm{NCYCLE}\),
\(\mathbb{N} F O\)
REAL (8) ::TOLA,TOLB
REAL (8),D \(\mathbb{M}\) ENSION (:) ::A LPHA,BETA,W ORK
REAL (8), D IM ENSION (:,:) ::A,B,U,V,Q

SU BROUTINE TGSJA_64 (JOBU, \(\mathrm{JOBV}, \mathcal{J O B Q}, \mathrm{M}, \mathrm{P}, \mathrm{N}, \mathrm{K}, \mathrm{L}, \mathrm{A},[\mathrm{LD} A], \mathrm{B}\), [LD B],TOLA,TOLB,ALPHA,BETA, U, [LDU],V, [LDV ], Q, [LDQ], [W ORK ],NCYCLE, [ \(\mathbb{N F F O}\) ])

CHARACTER (LEN=1) :: JOBU, NOBV , JOBQ
\(\mathbb{N}\) TEGER (8) ::M , P, N,K,L,LDA,LDB,LDU,LDV, LD Q, NCY-
\(C L E, \mathbb{I N} F O\)
REAL (8) ::TOLA,TOLB
REAL (8), D \(\mathbb{M}\) ENSION (:) ::ALPHA,BETA,W ORK
REAL (8),D IM ENSION (: : : : : A, B, U, V,Q

\section*{C INTERFACE}
\#include < sunperfh>
void dtgsja (char jobu, char jobv, char jobq, intm, int p, int n , int k , int l, double *a, int lda, double *b, int ldd, double tola, double tolb, double *alpha, double *beta, double *u, intldu, double * v , int ldv, double *q, int ldq, int *ncycle, int *info);

p, long \(n\), long k, long ll double *a, long lda, double *b, long ldb, double tola, double tolb, double *alpha, double *beta, double *u, long ldu, double *v, long ldv, double *q, long ldq, long *ncycle, long *info);

\section*{PURPOSE}
dtgsja com putes the generalized singular value decom position (GSVD) of tw o real upper triangular (or trapezoidal) \(m\) atrices \(A\) and \(B\).

On entry, it is assum ed thatm atrioes \(A\) and \(B\) have the follow ing form \(s\), which \(m\) ay be obtained by the preprocessing subroutine SG GSVP from a generalM by -N m atrix A and \(\mathrm{P}-\) by -N \(m\) atrix B :
```

        N-K- K L
    A = K (0 A12 A13) ifM K 乙 >= 0;
L (0 0 A23)
M K-工(0 0 0 )

```
```

            N-K-L L
    A = K (0 A12 A 13) ifM K L<0;
MK(0 0 A 23)

```
        N K K K
\(B=L\left(\begin{array}{lll}0 & 0 & B 13\end{array}\right)\)
    P-工 ( \(0 \quad 0 \quad 0 \quad\) )
where the K -by-K m atrix A 12 and L-by- m atrix B 13 are nonsingular upper triangular; A 23 is L-by- upper triangular if \(M-K->=0\), otherw ise \(A 23\) is \((M-K)\)-by -4 uppertrapezoidal.

On exit,
\[
U{ }^{*} A * Q=D 1 *(0 R), \quad V{ }^{*} B * Q=D 2 *(0 R)
\]
where \(U, V\) and \(Q\) are orthogonal matrioes, \(Z\) ' denotes the transpose of \(Z, R\) is a nonsingularupper triangularm atrix, and D 1 and D 2 are 'diagonal"m atrices, which are of the follow ing structures:

If \(M-\mathrm{K}->=0\),
```

            K L
    DI= K(IO)
L (0 C )
M K-工(0 0)

```
            K L
D \(2=\mathrm{L} \quad(0 \mathrm{~S})\)
    P - ( 0 0)
            NK- K L
( 0 R ) \(=\mathrm{K}\) ( 0 R11 R12) K
    L ( 0 O R22) L
w here
\[
\begin{aligned}
& C=\operatorname{diag}(A L P H A(K+1), \ldots, A L P H A(K+L)), \\
& S=\operatorname{diag}(\operatorname{BETA}(\mathbb{K}+1), \ldots, \operatorname{BETA}(\mathbb{K}+L)), \\
& C \star * 2+S \star \star 2=I .
\end{aligned}
\]
\(R\) is stored in \(A(1: K+L, N-K+1 \mathbb{N})\) on exit.

IfM K- < 0,

K M K K + L M
\(D 1=K\left(\begin{array}{lll}I & 0\end{array}\right)\)
\(M-K\left(\begin{array}{lll}(0 C O\end{array}\right)\)
```

            K M K K +L-M
    D2 = M K (0 S 0 )
K+L-M (0 0 I )
P-(0 0 0 )

```
            \(\mathrm{N} K \dashv \mathrm{~K} \quad \mathrm{M}\) K \(\mathrm{K}+\mathrm{L} \mathrm{M}\)
    \(M\) K (0 O R22 R23)
    \(\mathrm{K}+\mathrm{L} \mathrm{M}\) (0 0 (0 R33)
w here
\(C=\operatorname{diag}(A L P H A(K+1), \ldots, A L P H A(M))\),
\(S=\operatorname{diag}(B E T A(K+1), \ldots, B E T A(M))\),
\(C * * 2+S * * 2=I\).
\(R=(R 11 R 12 R 13)\) is stored in \(A(1 M, N\) K \(\mathrm{N}+1 \mathbb{N})\) and \(R 33\) is stored
( 0 R22R23)
in \(B(M-K+1: L, N+M-\longleftarrow+1 N)\) on exit.

The com putation of the orthogonal transform ation \(m\) atrices \(U\), \(V\) or \(Q\) is optional. These \(m\) atrices \(m\) ay eitherbe form ed explicitly, or they \(m\) ay be postm ultiplied into input \(m\) atrices \(\mathrm{U} 1, \mathrm{~V} 1\), or Q 1 .
STG SJA essentially uses a variant of \(K\) ogbetliantz algorithm to reduce \(m\) in ( \((, M-K)\)-by -4 triangular (ortrapezoidal) \(m\) atrix A 23 and \(\mathrm{L}-\mathrm{by}-\mathrm{m}\) atrix B 13 to the form :

U 1 *A 13*Q \(1=\mathrm{C} 1 * \mathrm{R} 1\); V 1 *B13*Q \(1=\mathrm{S} 1 * \mathrm{R} 1\), w here \(\mathrm{U} 1, \mathrm{~V} 1\) and Q 1 are orthogonalm atrix, and \(Z '\) is the transpose of Z. C1 and S1 are diagonalm atrices satisfying \(\mathrm{C} 1 * * 2+\mathrm{S} 1 * * 2=\mathrm{I}\),
and R1 is an \(\mathrm{L}-\mathrm{by}-\mathrm{L}\) nonsingularuppertriangularm atrix.

\section*{ARGUMENTS}
\(J 0 \mathrm{BU}\) (input)
\(=\mathrm{U}\) ': U m ustcontain an orthogonalm atrix U 1 on entry, and the product \(\mathrm{I} * \mathrm{U}\) is retumed; = I ': U is initialized to the unitm atrix, and the orthogonal matrix U is retumed; = N ': U is notcom puted.

JOBV (input)
\(=\mathrm{V}\) : V m ustcontain an orthogonalm atrix V 1 on entry, and the product V 1*V is retumed; = I ': V is initialized to the unitm atrix, and the onthogonal matrix \(V\) is retumed; \(=\mathrm{N}: \mathrm{V}\) is notcom puted.
\(J O B Q\) (input)
\(=\mathrm{Q}: \mathrm{Q}\) mustcontain an orthogonalm atrix Q 1 on entry, and the product \(\mathrm{Q} \mathrm{I}^{*} \mathrm{Q}\) is retumed; = \(\mathrm{I}^{\prime}: \mathrm{Q}\) is initialized to the unitm atrix, and the orthogonal matrix Q is retumed; \(=\mathrm{N}\) : Q is not com puted.

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
\(P\) (input) The num ber of row sof the \(m\) atrix \(B . P>=0\).
N (input) The num ber of collm ns of the \(m\) atrices \(A\) and \(B\). \(N\) \(>=0\).
\(K\) (input) \(K\) and \(L\) specify the subblocks in the input \(m\) atrices \(A\) and \(B\) :
\(\mathrm{A} 23=\mathrm{A}(\mathrm{K}+1 \mathbb{M} \mathbb{N}(\mathrm{~K}+\mathrm{L}, \mathrm{M}), \mathbb{N}-\mathrm{L}+1 \mathbb{N})\) and \(\mathrm{B} 13=\) \(B(1: L, N-\Psi+1 \mathbb{N})\) of \(A\) and \(B\),whose \(G S V D\) is going to be com puted by STG SJA. Se Further details.

L (input) See the description of K .
A (input/output)
On entry, the M -by -N matrix A. On exit, A \(\mathbb{N}\) \(K+1 \mathbb{N}, \mathbb{1} M \mathbb{N}(K+L, M)\) ) contains the triangular \(m\) atrix \(R\) orpartofR. See Punpose for details.

LDA (input)
The leading dim ension of the aray A. LD A >= max (1,M).

B (input/output)
On entry, the \(P\)-by \(-N\) m atrix B. On exit, if necessary, \(B(M+1: L, N+M-\Psi+1 \mathbb{N}\) ) contains a partofR . See Purpose for details.

LD B (input)
The leading dim ension of the array \(\mathrm{B} . \operatorname{LD} \mathrm{B}>=\) \(\max (1, P)\).

TO LA (input)
TO LA and TO LB are the convergence criteria for the Jacobi- K ogbetliantz tieration procedure. Generally, they are the sam e as used in the preprocessing step, say TOLA \(=\max (\mathrm{M}, N) \star\) norm (A) ) M A CHEPS, \(T O L B=\max (P, N){ }^{\star}\) norm (B)\()^{\star} \operatorname{MCHEPS}\).

TO LB (input)
See the description of TO LA .

A LPHA (output)
On exit, ALPHA and BETA contain the generalized singular value pairs of \(A\) and ; A LPHA \((1 K)=1\), \(\operatorname{BETA}(1: K)=0\), and ifM \(K-L>=0\), ALPHA \((\mathbb{K}+1 \mathbb{K}+\mathrm{L})\)
\(=\operatorname{diag}(\mathrm{C})\),
\(\operatorname{BETA}(\mathbb{K}+1 \mathrm{~K}+\mathrm{L})=\operatorname{diag}(\mathrm{S})\), or if \(\mathrm{M} \mathrm{K}-\mathrm{L}<0\),
A LPHA \((\mathbb{K}+1 \mathrm{M})=\mathrm{C}\), ALPHA \((\mathrm{M}+1 \mathrm{~K}+\mathrm{L})=0\)
BETA \((K+1 M)=S, B E T A(M+1: K+L)=1\). Furtherm ore, if \(K+L<N, A L P H A(K+L+1 \mathbb{N})=0\) and
BETA \((K+L+1 \mathbb{N})=0\).
BETA (output)
See the description of A LPH A .
U (input) On entry, if \(\mathrm{JOBU}=\mathrm{U}\) ', U m ustcontain a matrix
U 1 (usually the orthogonal matrix retumed by SGGSVP). On exit, if \(J 0 B U=\) ' \(I\) ', \(U\) contains the orthogonalm atrix U ; if \(\mathrm{OOBU}=\mathrm{U}\) ', U contains the product \(\mathrm{U} 1 * \mathrm{U}\). If \(\mathrm{JOBU}=\mathrm{N}\) ', U is notreferenced.

LD U (input)
The leading dim ension of the aray \(U\). LD U >= \(m\) ax \((1, M)\) if \(\mathcal{O B} B=U\) '; LD \(U>=1\) otherw ise.

V (input) On entry, if \(J \mathrm{O} B V=\mathrm{V}\) ', V m ustcontain a matrix V1 (usually the orthogonal \(m\) atrix retumed by SGGSVP). On exit, if \(\mathrm{JOBV}=\mathrm{I}^{\prime}, \mathrm{V}\) contains the orthogonalm atrix \(V\); if \(\mathrm{JOBV}=\mathrm{V}, \mathrm{V}\) contains the product \(\mathrm{V} 1 * \mathrm{~V}\). If \(\mathrm{JOBV}=\mathrm{N}, \mathrm{V}\) is notreferenced.

LD V (input)
The leading dim ension of the aray V. LDV >= \(\mathrm{max}(1, \mathrm{P})\) if \(\mathrm{JOBV}=\mathrm{V}\); LDV \(>=1\) otherw ise.
\(Q\) (input) \(O n\) entry, if \(J O B Q=Q\) ', \(Q\) mustcontain a matrix Q 1 (usually the orthogonal \(m\) atrix retumed by SGGSVP). On exit, if \(\mathcal{O B Q}=I^{\prime}, Q\) contains the orthogonalm atrix \(Q\); if \(J O B Q=Q ', Q\) contains the product \(\mathrm{Q} 1^{*} \mathrm{Q}\). If \(\mathrm{JOBQ}=\mathrm{N}^{\prime}, \mathrm{Q}\) is notreferenced.

LD Q (input)
The leading dim ension of the aray \(Q . L D Q>=\) \(\max (1, N)\) if \(J O B Q=Q ; L D Q>=1\) otherw ise.

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
NCYCLE (output)
The num ber of cycles required for convergence.
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-\) i, the i-th argum enthad an illegalvalue.
\(=1\) : the procedure does not converge after M A X IT cycles.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dtgsna - estim ate reciprocal condition num bers for specified eigenvalues and/or eigenvectors of a \(m\) atrix pair \((A, B)\) in generalized realSchurcanonical form (or of any \(m\) atrix pair \(\mathrm{Q} * \mathrm{~A} * \mathrm{Z}\) ', \(\mathrm{Q} * \mathrm{~B} * \mathrm{Z}\) I) w ith orthogonalm atrices Q and Z , where \(\mathrm{Z}^{\prime}\) denotes the transpose of \(Z\)

\section*{SYNOPSIS}
```

SUBROUTINE DTGSNA (JOB,HOW MNT,SELECT,N,A,LDA,B,LDB,VL,LDVL,
VR,LDVR,S,D\mathbb{F,MM,M,WORK,LWORK,INORK,INFO)}
CHARACTER * 1 JOB,HOW MNT
\mathbb{NTEGER N,LDA,LDB,LDVL,LDVR,MM ,M,LW ORK, INFO}
INTEGER IN ORK (*)
LOG ICAL SELECT (*)
D OUBLE PRECISION A (LDA,*),B (LDB,*),VL (LDVL,*),VR (LDVR,*),
S (*),D F (*),WORK (*)
SUBROUT\mathbb{NEDTGSNA_64(JOB,HOW MNT,SELECT,N,A,LDA,B,LDB,VL,}
LDVL,VR,LDVR,S,D\mathbb{F},MM,M,W ORK,LW ORK,\mathbb{N ORK,INFO)}
CHARACTER * 1 JOB,HOW MNT
INTEGER*8N,LDA,LD B,LDVL,LDVR,MM ,M ,LW ORK,\mathbb{NFO}
INTEGER*8 \mathbb{N ORK (*)}
LO G ICAL*8 SELECT (*)
D OUBLE PRECISION A (LDA,*),B (LDB,*),VL (LDVL,*),VR (LDVR,*),
S (*),D \mathbb{F (*),W ORK (*)}

```

\section*{F95 INTERFACE}

SU BROUTINE TGSNA (JOB,HOW MNT,SELECT, \(\mathbb{N}], A,[L D A], B,[L D B], V L\), \([L D V L], V R,[L D V R], S, D \mathbb{F}, M M, M,[W O R K],[L W O R K],[\mathbb{W} O R K]\),

CHARACTER ( \(4 E N=1\) ) : : JOB, HOW MNT
\(\mathbb{N}\) TEGER : \(: N\),LDA, LDB,LDVL,LDVR, M M, M, LW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{I N} O R K\)
LOG ICAL, D \(\mathbb{M}\) ENSION (:) :: SELECT
REAL (8), D \(\mathbb{M}\) ENSION (:) :: S, D \(\mathbb{F}, W\) ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A , B, VL, VR

SU BROUTINE TG SNA_64 (DBB,HOW M NT,SELECT, \(\mathbb{N}], A,[L D A], B,[L D B], V L\), \([[L D V L], V R,[L D V R], S, D \mathbb{F}, M M, M,[W O R K],[L W O R K],[\mathbb{W} O R K]\), [ \(\mathbb{N} \mathrm{FO}]\) )

CHARACTER (LEN=1) :: JOB,HOW MNT

\(\mathbb{N}\) TEGER (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{I W}\) ORK
LOGICAL (8), D \(\mathbb{M}\) ENSION (:) :: SELECT
REAL (8), D \(\mathbb{M}\) ENSION (:) :: S, D \(\mathbb{F}, W\) ORK
REAL (8), D \(\mathbb{M}\) ENSION (: ::) ::A , B , VL , VR

\section*{C INTERFACE}
\#include <sunperfh>
void dtgsna (char j’.b, charhow m nt, int *select, intn, double *a, intlda, double *b, int ldb, double *vl, int ldvl, double *Vr, int ldvr, double *s, double *dif, intm \(m\), int *m, int*info);
void dtgsna_64 (char jं.b, charhow m nt, long *select, long n, double *a, long lda, double *b, long ldb, double *vl, long ldvl, double *vr, long ldvr, double *s, double *dif, long m m , long *m , long *info);

\section*{PURPOSE}
dtgsna estim ates reciprocal condition num bers for specified eigenvalues and/or eigenvectors of a m atrix pair ( \(A, B\) ) in generalized realSchur canonicalform (or of any m atrix pair Q *A *Z', Q *B *Z \()\) w ith orthogonalm atrices Q and Z , w here \(\mathrm{Z}^{\prime}\) denotes the transpose of \(Z\).
( \(\mathrm{A}, \mathrm{B}\) ) m ustbe in generalized realSchur form (as retumed by SGGES), i.e.A is block uppertriangularw ith 1 toy-1 and 2-by-2 diagonalblocks.B is upper triangular.

\section*{ARGUMENTS}

\section*{JOB (input)}

Specifies w hether condition num bers are required
foreigenvalues (S) oreigenvectors (D \(\mathbb{F}\) ):
\(=\mathrm{E}\) ': foreigenvalues only (S);
\(=\mathrm{V}\) ': foreigenvectors only (D \(\mathbb{F})\);
= B ': forboth eigenvalues and eigenvectors ( S and D \(\mathbb{F}\) ).

HOW MNT (input)
= 'A ': com pute condition num bers for all eigenpairs;
= \(S^{\prime}\) : com pute condition num bers for selected eigenpairs specified by the array SELEC T .

\section*{SELECT (input)}

If HOW MNT = S', SELECT specifies the eigenpairs for which condition num bers are required. To select condition num bers for the eigenpair comesponding to a realeigenvalue w ( \(\mathcal{\nu}\), SELECT ( \()\) m ustibe set to .TRU E ..To selectcondition num bers corresponding to a complex conjugate pair of eigenvaluesw ( 7 ) and w ( \(\ddagger+1\) ), either SELECT ( 7 ) or SELECT ( \(j+1\) ) or both, mustbe setto .TRUE .. If HOW MNT = A', SELECT is notreferenced.

N (input) The order of the square \(m\) atrix pair ( \(\mathrm{A}, \mathrm{B}\) ). \(\mathrm{N} \quad>=\) 0.

A (input) The upperquasi-triangularm atrix \(A\) in the pair \((A, B)\).

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).
\(B\) (input) The upper triangularm atrix \(B\) in the pair \((A, B)\).

LD B (input)
The leading dim ension of the array \(B . L D B>=\) \(\max (1, N)\).

VL (input)
If \(\mathrm{JOB}=\mathrm{E}\) 'or B ', VL mustcontain left eigenvectors of \((A, B)\), comesponding to the eigenpairs specified by H OW M NT and SELEC T. The eigenvectors \(m\) ust be stored in consecutive colum ns of \(V \mathrm{~L}\), as retumed by STGEVC . If JOB \(=V\) ', \(V L\) is not referenced.

The leading dim ension of the array VL. LD VL \(>=1\). If \(\mathrm{JOB}=\mathrm{E}\) 'or \(\mathrm{B}^{\prime}\), LDVL \(>=\mathrm{N}\).

VR (input)
If \(\mathrm{JOB}=\mathrm{E}\) 'or B ', VR m ust contain right eigenvectors of ( \(A, B\) ), comesponding to the eigenpairs specified by HOW M NT and SELECT. The eigenvectors m ust be stored in consecutive colum ns ov VR, as retumed by \(S T G E V C\). If \(J O B=V ', V R\) is not referenced.
LDVR (input)
The leading dim ension of the array VR.LD VR >= 1 . If \(\mathrm{JOB}=\mathrm{E}\) 'or B ', LDVR \(>=\mathrm{N}\).

S (output)
If \(J O B=E\) ' or \(B\) ', the reciprocal condition num bers of the selected eigenvalues, stored in consecutive elem ents of the array. For a com plex conjugate pair of eigenvalues tw o consecutive ele\(m\) ents of \(S\) are set to the sam e value. Thus \(S(\mathcal{j})\), D \(\mathbb{F}(\mathcal{j})\), and the \(j\) th colum ns ofVL and VR all comespond to the sam e eigenpair (butnot in general the jth eigenpair, unless alleigenpairs are selected). If \(70 \mathrm{~B}=\mathrm{V}\) ', S is not referenced.

D \(\mathbb{F}\) (output)
If \(\mathrm{JO} \mathrm{B}=\mathrm{V}\) 'or B ', the estim ated reciprocal condition numbers of the selected eigenvectors, stored in consecutive elem ents of the array. For a complex eigenvector tw o consecutive elem ents of \(D \mathbb{F}\) are set to the sam e value. If the eigenvalues cannot be reordered to com pute \(D \mathbb{F}(\mathcal{I}), \mathrm{D} \mathbb{F}()\) is set to 0 ; this can only occurw hen the true value would be very sm allanyw ay. If \(\mathrm{JO} B=\mathrm{E}, \mathrm{D} \mathbb{F}\) is not referenced.

M M (input)
The num berof elem ents in the arrays \(S\) and \(D \mathbb{F} . M M\) \(>=\mathrm{M}\).

M (output)
The num berof elem ents of the arrays \(S\) and \(D \mathbb{F}\) used to store the specified condition num bers; for each selected realeigenvalue one elem ent is used, and for each selected com plex conjugate pair of eigenvalues, tw o elem ents are used. If HOW M NT = A', M is set to \(N\).

W ORK (w orkspace)
If \(\mathrm{JOB}=\mathrm{E}\) ', W ORK is not referenced. O therw ise,
on exit, if \(\mathbb{N F} F=0, W\) ORK (1) retums the optim al LW ORK.

\section*{LW ORK (input)}

The dim ension of the anay \(W\) ORK .LW ORK >= N. If JOB = V 'or B 'LW ORK >= 2*N * \(N+2\) ) +16 .
If LW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of theW ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IW ORK (w orkspace)
dim ension \((\mathbb{N}+6)\) If \(\mathcal{O B}=\mathrm{E}^{\prime}, \mathbb{I N} \mathrm{ORK}\) is not referenced.
\(\mathbb{N} F O\) (output)
=0: Successfulexit
\(<0:\) If \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegal value

\section*{FURTHER DETAILS}

The reciprocal of the condition num ber of a generalized eigenvalue \(w=(a, b)\) is defined as \((w)=(\mu A v|\star * 2+\mu \mathrm{Bv}| \star * 2)^{* *}(1 / 2) /\left(\right.\) norm \((u){ }^{*}\) norm (v))
\(w\) here \(u\) and \(v\) are the left and righteigenvectors of (A, B) comesponding to \(\mathrm{w} ;|z|\) denotes the absolute value of the com plex num ber, and norm (u) denotes the 2 -norm of the vector u. The pair ( \(a, b\) ) corresponds to an eigenvalue \(w=a \nless \vDash\) \(u\) Av/uBv) of the m atrix pair ( \(A, B\) ). If both \(a\) and \(b\) equal zero, then \((A B)\) is singular and \(S(I)=-1\) is retumed.

A n approxim ate errorbound on the chordal distance betw een the i-th computed generalized eigenvalue \(w\) and the comesponding exacteigenvalue lam bda is hord (w , lam bda) <= EPS * norm (A , B) /S (I)
where EPS is the \(m\) achine precision.

The reciprocal of the condition num ber D \(\mathbb{F}\) (i) of right eigenvector \(u\) and lefteigenvectorv comesponding to the generalized eigenvalue \(w\) is defined as follow s:
a) If the \(i\)-th eigenvalue \(w=(a, b)\) is real

Suppose U and V are orthogonal transform ations such that
\(U *(A, B) * V=(S, T)=(a *)(b *)\)
1
\[
(0 \mathrm{~S} 22),(0 \mathrm{~T} 22)
\]
n-1
\[
1 \mathrm{n}-1 \quad 1 \mathrm{n}-1
\]

Then the reciprocalcondition num berD \(\mathbb{F}(i)\) is
D ifl( \((\mathrm{a}, \mathrm{b}),(\mathrm{S} 22, \mathrm{~T} 22))=\operatorname{sigm} \mathrm{a}-\mathrm{m}\) in \((\mathrm{Z} 1)\),
where sigm a-m in (Zl) denotes the sm allest singular value of the
\(2(n-1)\)-by-2 (n-1) m atrix
```

Zl= [kron (a, In-1) kron (1,S22) ]
[kron(b, In-1) -kron(1,T22)].

```

H ere \(\mathrm{In}-1\) is the identily \(m\) atrix of size \(\mathrm{n}-1 . \operatorname{kron}(X, Y)\) is the
\(K\) roneckerproductbetw een the \(m\) atrices \(X\) and \(Y\).

N ote that if the defaultm ethod for com puting D IF (i) is w anted
(see SLA TDF), then the param eter D FPD RI (see below) should be
changed from 3 to 4 (routine SLATD F (LJO B \(=2 \mathrm{w}\) ill be used)).
See STG SY L form ore details.
b) If the \(i\)-th and (i+1)-th eigenvalues are com plex conjugate pair,

Suppose U and V are orthogonal transform ations such that
\(U^{*}(A, B) * V=(S, T)=(S 11 *)(T 11 *\)
) 2
\[
(0 \quad S 22),(0
\]

T22) \(n-2\)
\[
\begin{array}{llll}
2 & n-2 & n-2
\end{array}
\]
and (S11, T11) comesponds to the com plex conjugate eigenvalue
pair ( \(w\), con \(\dot{g}(w)\) ). There exist unitary \(m\) atrices \(U 1\) and V1 such
that
U 1 *S11*V1 = (s11 s12 ) and U 1 *T11*V \(1=(\mathrm{t} 11 \mathrm{t} 12\)
)
\[
\text { ( } 0 \text { s22) ( } 0 \text { t22 }
\]
)
w here the generalized eigenvalues \(\mathrm{w}=\mathrm{s} 11\) t11 and con \({ }^{g}(w)=s 22\) t 22 .

Then the reciprocalcondition num berD \(\mathbb{F}\) (i) is bounded by
\[
\mathrm{m} \text { in }(\mathrm{d} 1, \mathrm{~m} \text { ax }(1, \text { real(s11)/real(s22)|)*d2 ) }
\]
where, \(\mathrm{d} 1=\mathrm{D}\) ifl( \((\mathrm{s} 11, \mathrm{t} 11),(\mathrm{s} 22, \mathrm{t} 22))=\) sigm a-m in \((\mathrm{Z} 1)\), where
Z 1 is the com plex 2 -by -2 m atrix
```

Z1 = [s11 -s22]
[t11 -t22 ],

```

This is done by com puting (using realarithm etic) the roots of the characteristicalpolynom ialdet(Z1'* Z1 lam bda I),
where Z1'denotes the conjugate transpose of Z1 and \(\operatorname{det}(X)\) denotes the determ inantof \(X\).
and d2 is an upperbound on D ifl((S11,T11), (S22,T22)), ie.an
upperbound on sigm a-m in ( \(Z 2\) ), where \(Z 2\) is ( \(2 n-2\) )-by-( \(2 n-\) 2)
\[
\begin{aligned}
& \mathrm{Z} 2=[\mathrm{kron}(\mathrm{~S} 11 \text { ', In-2) } \mathrm{kron}(\mathrm{I}, \mathrm{~S} 22)] \\
& \text { [kron(T11', In-2) kron (12, T22) ] }
\end{aligned}
\]
\(N\) ote that if the default \(m\) ethod for com puting \(D\) IF is w anted (see
SLA TD F), then the param eterD IFD R I (see below ) should be changed
from 3 to 4 (routine SLA TDF (INO B \(=2\) w ill be used)). See STGSYL
form ore details.

For each eigenvaluekector specified by SE LEC T, D IF stores a Frobenius norm -based estim ate ofD ifl.

A \(n\) approxim ate errorbound forthe i-th com puted eigenvector VL (i) orVR (i) is given by

> EPS * nom (A, B) /D IF (i).

See ref. [2-3] form ore details and further references.

B ased on contributions by
Bo K agstrom and PeterPorom aa, D epartm ent of C om puting Science,

Um ea U niversity, S-901 87 Um ea, Sw eden.

\section*{References}
[1] B .K agstrom ; A D irectM ethod forReordering Eigenvalues in the

G eneralized RealSchurForm of a RegularM atrix Pair (A, B), in

M S.M oonen etal (eds), LinearA lgebra for Large Scale and

Real-T im e A pplications, K luw erA cadem ic Publ. 1993, pp 195-218.
[2] B . K agstrom and P.Porom aa; C om puting Eigenspaces w th Specified E igenvalues of a R egularM atrix Pair ( \(A\), \(B\) ) and Condition

Estim ation: Theory, A lgorithm s and Softw are, ReportUM IN F-94.04,D epartm ent of Computing Science, Umea

U niversity, S-901 87 Um ea, Sw eden, 1994.A lso as LA PA CK W orking

N ote 87. To appear in N um ericalA lgorithm s, 1996.
[3] B . K agstrom and P . Porom aa, LA PA C K -Style A lgorithm s and Softw are
for Solving the Generalized Sylvester Equation and Estim ating the Separation betw een RegularM atrix Pairs, ReportU M \(\mathbb{I N F}\) 93 23,

D epartm entofC om puting Science, Um ea U niversity, S-901 87 Um ea,

Sw eden, D ecem ber 1993, Revised A pril 1994, A lso as LAPACK W orking

N ote 75. To appear in A CM Trans. on M ath. Softw are, V ol 22,

No1, 1996.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dtgsyl-solve the generalized Sylvester equation

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE DTGSYL (TRANS,IDOB,M,N,A,LDA,B,LDB,C,LDC,D,LDD,}
E,LDE,F,LDF,SCALE,D \mathbb{F,W ORK,LW ORK,IN ORK,INFO)}

```
CHARACTER * 1 TRANS
\(\mathbb{N} T E G E R \operatorname{LOB}, M, N, L D A, L D B, L D C, L D D, L D E, L D F, L W O R K\),
\(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER \(\mathbb{I N}\) ORK (*)
DOUBLE PRECISION SCALE,D F
DOUBLE PRECISION A (LDA, \(\left.{ }^{\star}\right)\), B (LDB,\(\left.\star\right)\), C (LDC,\(\left.^{\star}\right)\), D (LDD , \(\left.{ }^{\star}\right)\),
\(\mathrm{E}\left(\mathrm{LDE},{ }^{\star}\right), \mathrm{F}(\mathrm{LD} \mathrm{F}, \star), \mathrm{W} O R \mathrm{~K}\left({ }^{( }\right)\)
SU BROUTINEDTGSYL_64 (TRANS, IDOB,M,N,A,LDA,B,LDB,C,LDC,D,
    LDD, E,LDE,F,LDF,SCALE,D \(\mathbb{F}, W\) ORK,LW ORK, \(\mathbb{I N} O R K, \mathbb{N} F O\) )
CHARACTER * 1 TRANS
\(\mathbb{N} T E G E R * 8\) IJOB, M, N,LDA, LDB, LD C, LDD ,LDE, LDF, LW ORK,
\(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER*8 \(\mathbb{I N}\) ORK (*)
DOUBLE PRECISION SCALE,D \(\mathbb{F}\)
DOUBLE PRECISION A (LDA,*), B (LDB,*), C (LDC , *), D (LDD , \(\left.{ }^{\star}\right)\),
E (LDE, \(\left.{ }^{\star}\right), \mathrm{F}(\mathrm{LD} F, \star), \mathrm{W} O R K(*)\)

\section*{F95 INTERFACE}

SUBROUTINE TGSYL (TRANS, IJOB, \(\mathbb{M}], \mathbb{N}], A,[L D A], B,[L D B], C,[L D C]\), D, [LDD ], E, [LDE], F, [LDF],SCALE,D \(\mathbb{F},\left[\begin{array}{l}\text { W ORK ], [LW ORK ], [IW ORK ], }\end{array}\right.\) [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::TRANS
\(\mathbb{N} T E G E R::\) IJOB, \(M, N, L D A, L D B, L D C, L D D, L D E, L D F, L W O R K\),
\(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL (8) :: SCALE,D \(\mathbb{F}\)
REAL (8),D IM ENSION (:) ::W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:)::A,B,C,D,E,F

SU BROUTINE TGSY L_64 (TRANS, IJO B, M ], \(\mathbb{N}], A,[L D A], B,[L D B], C\),
 [ \(\mathbb{I N}\) ORK], \([\mathbb{N F O}])\)

CHARACTER (LEN=1)::TRANS

LW ORK, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{I N}\) ORK
REAL (8) :: SCALE,D \(\mathbb{F}\)
REAL (8),D IM ENSION (:) ::W ORK
REAL (8),D IM ENSION (:,:) ::A,B,C,D,E,F

\section*{C INTERFACE}
\#include <sunperfh>
void dtgsyl(chartrans, int ijob, intm, intn, double *a, int lda, double *b, int ldb, double \({ }^{*}\) c, int ldc, double *d, int ldd, double *e, int lde, double *f, int ldf, double *scale, double *dif, int *info);
void dtgsyl_64 (chartrans, long ijob, long m, long n, double *a, long lda, double *b, long ldb, double *c, long ldc, double *d, long ldd, double *e, long lde, double *f, long ldf, double *scale, double *dif, long *info);

\section*{PURPOSE}
dtgsylsolves the generalized Sylvesterequation:
\(A * R-L * B=\) scale * C
\(D * R-L * E=\) scale \(* F\)
(1)
where \(R\) and \(L\) are unknow \(n m\)-by- \(n m\) atrices, ( \(A, D\) ), ( \(B, E\) ) and ( \(C, F\) ) are given \(m\) atrix pairs of size \(m-b y-m, n-b y-n\) and \(m\)-by- \(n\), respectively, with realentries. ( \(A, D\) ) and ( \(B, E\) ) m ust be in generalized (real) Schur canonical form, ie. A, \(B\) are upperquasi triangularand D, E are uppertriangular.

The solution \((\mathbb{R}, \mathrm{L})\) overw rites ( \(\mathrm{C}, \mathrm{F}\) ). \(0<=\mathrm{SCA} \mathrm{LE}<=1\) is an output scaling factor chosen to avoid overflow .

In \(m\) atrix notation (1) is equivalent to solve \(\mathrm{Zx}=\) scale b , where \(Z\) is defined as
\[
\begin{aligned}
Z= & {\left[\operatorname{kron}(\operatorname{In}, A) \operatorname{kron}\left(B^{\prime}, \operatorname{Im}\right)\right] } \\
& {\left[\operatorname{kron}(\operatorname{In}, D) \operatorname{kron}\left(E^{\prime}, \operatorname{Im}\right)\right] . }
\end{aligned}
\]

H ere \(\mathbb{k}\) is the identily \(m\) atrix of size \(k\) and X 'is the transpose of \(X\). kron ( \(X, Y\) ) is the \(K\) roneckerproduct.betw een the \(m\) atrices \(X\) and \(Y\).

If TRANS = T', STGSY L solves the transposed system Z "y = scale*b, which is equivalent to solve for \(R\) and \(L\) in
\[
\begin{aligned}
& A^{\prime} * R+D^{\prime *} L=\text { scale * C } \\
& R B^{\prime}+L E^{\prime}=\text { scale * }(F)
\end{aligned}
\]
(3)

This case (TRANS = T) is used to com pute an one-norm -based estim ate of \(D\) if \([(A, D),(B, E)]\), the separation betw een the \(m\) atrix pairs \((A, D)\) and \((B, E)\), using SLA CON.

If IJO B >= 1, STG SY L com putes a Frobenius norm -based esti\(m\) ate ofD if \([(A, D),(B, E)]\). That is, the reciprocal ofa low er
bound on the reciprocal of the sm allest singular value of \(Z\).
See [1-2] form ore inform ation.

This is a level3 BLA S algorithm .

\section*{ARGUMENTS}

TRANS (input)
= N ', solve the generalized Sylvester equation (1). = T', solve the transposed 'system (3).

IJOB (input)
Specifies whatkind of functionality to be per-
form ed. \(=0\) : solve (1) only.
\(=1\) : The functionality of 0 and 3 .
\(=2\) : The functionality of 0 and 4 .
\(=3: 0\) nly an estim ate ofD if \([(A, D),(B, E)]\) is computed. (look ahead strategy IJO B \(=1\) is used). \(=4: 0\) nly an estim ate ofD if \([(A, D),(B, E)]\) is computed. ( SGECON on sub-system s is used). N ot referenced if TRANS = \(T\) '.
\(M\) (input) The order of the \(m\) atrioes \(A\) and \(D\), and the row dim ension of the \(m\) atrices \(C, F, R\) and \(L\).
\(N\) (input) The order of the \(m\) atrices \(B\) and \(E\), and the \(c o l u m n\) dim ension of the \(m\) atrioes \(C, F, R\) and \(L\).

A (input) The upperquasitriangularm atrix A.

LDA (input)
The leading dim ension of the aray A. LDA >= \(m a x(1, M)\).
\(B\) (input) The upperquasitriangularm atrix \(B\).

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).
C (input/output)
On entry, \(C\) contains the right-hand-side of the first \(m\) atrix equation in (1) or (3). On exit, if IJOB \(=0,1\) or \(2, C\) has been overw ritten by the solution R. If \(\mathrm{IO} B=3\) or 4 and TRANS \(=N^{\prime}\), C holds \(R\), the solution achieved during the com putation of the \(D\) if-estim ate.

LD C (input)
The leading dim ension of the aray C. LD C >= max (1, M).

D (input) The upper triangularm atrix D.

\section*{LD D (input)}

The leading dim ension of the aray D. LDD >= \(m a x(1, M)\).

E (input) The upper triangularm atrix E .

LDE (input)
The leading dim ension of the aray E. LDE >= \(\max (1, N)\).

F (input/output)
On entry, F contains the right-hand-side of the second \(m\) atrix equation in (1) or (3). On exit, if IJO B \(=0,1\) or \(2, F\) has been overw ritten by the solution L. If IOB \(=3\) or 4 and TRANS \(=N^{\prime}, F\) holds L , the solution achieved during the com putation of the D if-estim ate.

LD F (input)
The leading dim ension of the array F. LDF >= max (1, M).

\section*{SCALE (output)}

On exitSCALE is the reciprocal of a low er bound of the reciprocal of the \(D\) if-function, ie. SCALE is an upper bound of \(D\) if \([(A, D)\), ( \(B, E)]=\) sigm a_m in \((Z)\), where \(Z\) as in \((2)\). If IJO \(B=0\) or TRANS = T', SCALE is not touched.
D \(\mathbb{F}\) (output)
On exitSCALE is the reciprocal of a low er bound
of the reciprocal of the \(D\) if-function, ie. SCALE
is an upper bound of \(D\) if \([(A, D),(B, E)]=\) sigm a_m in \((Z)\), where \(Z\) as in (2). If IJO \(B=0\) or TRANS = T', SCALE is nottouched.

\section*{W ORK (w orkspace)}

If \(\mathrm{IJOB}=0, \mathrm{~W} O R K\) is not referenced. O therw ise, on exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the aray \(W\) ORK.LW ORK \(>=1\). If \(\mathrm{IJOB}=1\) or 2 and TRANS \(=\mathrm{N}^{\prime}, \mathrm{LW}\) ORK \(>=2 * \mathrm{M} * \mathrm{~N}\).

If LW O RK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IW ORK (w orkspace)
dim ension \((M+N+2)\)
\(\mathbb{N}\) FO (output)
=0: successfulexit
\(<0:\) If \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegal
value.
\(>0\) : \((A, D)\) and \((B, E)\) have com \(m\) on orclose eigenvalues.

\section*{FURTHER DETAILS}
\(B\) ased on contributions by
Bo K agstrom and PeterPorom aa, D epartm ent of C om puting
Science,
Um ea U niversity, S-901 87 Um ea, Sw eden.
[1] B . K agstrom and P .Porom aa, LA PA C K -Style A lgorithm s and Softw are
for Solving the Generalized Sylvester Equation and Estim ating the

Separation betw een RegularM atrix Pairs, ReportU M IN F -

9323,
D epartm entofC om puting Science, Um ea U niversity, S-901 87 Um ea,

Sw eden, D ecem ber 1993, Revised A pril 1994, A lso as LAPACK W orking

N ote 75. To appear in A CM Trans. on M ath. Softw are, V ol 22,

No1, 1996.
[2] B . K agstrom , A Perturbation A nalysis of the G eneralized Sylvester

Equation (AR -LB,DR-LE ) = (C,F), SIAM J. M atrix A nal.

Appl, 15 (4):1045-1060,1994
[3] B . K agstrom and L. W estin, G eneralized Schur M ethods w th

C ondition Estim ators for Solving the G eneralized Sylvester

Equation, \(\mathbb{E E E}\) Transactions on A utom atic C ontrol, V ol. 34,No.7,

July 1989, pp 745-751.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtpcon -estim ate the reciprocal of the condition num ber of a packed triangular \(m\) atrix \(A\), in either the 1 -norm or the infinity-norm

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDTPCON NORM,UPLO,D IAG,N,A,RCOND,W ORK,W ORK2, INFO)}
CHARACTER * 1NORM,UPLO,DIAG
\mathbb{NTEGER N, INFO}
INTEGER W ORK2 (*)
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (*),W ORK (*)
SUBROUT\mathbb{NEDTPCON_64 NORM,UPLO,D IAG,N,A,RCOND,W ORK,W ORK2,}
INFO)
CHARACTER * 1 NORM,UPLO,DIAG
INTEGER*8N,INFO
\mathbb{NTEGER*8 W ORK2 (*)}
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE TPCON \(\mathbb{N} O R M, U P L O, D \mathbb{I} G, N, A, R C O N D,[\mathbb{W} O R K],[W O R K 2]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1)::NORM,UPLO,D IA G
\(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
\(\mathbb{I N} T E G E R, D \mathbb{I M}\) ENSION (:) ::W ORK2
REAL (8) :: RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::A ,W ORK

SU BROUTINE TPCON_64 \(\mathbb{N} O R M, U P L O, D \mathbb{I} G, N, A, R C O N D,[\mathbb{O} O R K],[W O R K 2]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1)::NORM,UPLO,DIAG
\(\mathbb{N}\) TEGER (8) ::N, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) ::W ORK2
REAL (8) :: RCOND
REAL (8),D \(\mathbb{I}\) ENSION (:) ::A,W ORK

\section*{C INTERFACE}
\#include < sunperfh>
void dtpcon (charnorm, charuplo, chardiag, int n, double
*a, double *rcond, int *info);
void dtpcon_64 (charnorm , char uple, chardiag, long n, double *a, double *roond, long *info);

\section*{PURPOSE}
dtpoon estim ates the reciprocal of the condition num berofa packed triangular \(m\) atrix \(A\), in either the 1 -nom or the infinity-norm .

The norm ofA is com puted and an estim ate is obtained for norm (inv (A )), then the reciprocal of the condition num ber is com puted as
\(\operatorname{RCOND}=1 /(\operatorname{norm}(A) *\) norm (inv (A))).

\section*{ARGUMENTS}

NORM (input)
Specifies whether the 1-norm condition number or the infinity-norm condition num ber is required:
= 1'or \(^{0}\) ': 1-nom ;
= I': Infinity-norm.
UPLO (input)
\(=\mathrm{U}\) : A is uppertriangular;
\(=\mathrm{L}\) ': A is low ertriangular.
D IA G (input)
\(=\mathrm{N}: \mathrm{A}\) is non-unit triangular;
\(=\mathrm{U}\) ': A is unit triangular.
N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input) The upper or low er triangular m atrix A, packed colum nw ise in a linear array. The jth colum n of A is stored in the aray A as follow s: if UPLO = \(U^{\prime}, A(i+(j-1) * j 2)=A(i, j)\) for \(1<=i<=j\); if \(\mathrm{UPLO}=\mathrm{L}, \mathrm{A}(i+(j-1) *(2 n-j) / 2)=A(i, j)\) for \(j=i<=n\). IfD \(\mathbb{A} G=U\) ', the diagonalelem ents of \(A\) are not referenced and are assum ed to be 1 .

RCOND (output)
The reciprocal of the condition num ber of the \(m\) atrix \(A\), computed as RCOND \(=1 /(\) noim (A) * norm (inv (A))).
W ORK (w orkspace)
dim ension \((3 * N)\)

W ORK 2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtpm \(v\)-perform one of the \(m\) atrix-vector operations \(x:=\) A*x, orx \(:=A * x\)

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE DTPMV (UPLO,TRANSA,D IAG ,N,A,Y, INCY)}
CHARACTER * 1 UPLO,TRANSA,DIAG
\mathbb{NTEGER N, INCY}
DOUBLE PRECISION A (*),Y (*)
SU BROUT\mathbb{NE DTPM V_64(UPLO,TRANSA,D IAG,N,A,Y, INCY)}
CHARACTER * 1 UPLO,TRANSA,DIAG
INTEGER*8N,\mathbb{NCY}
DOUBLE PRECISION A (*),Y (*)
F95 INTERFACE

```

```

    CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
    \mathbb{NTEGER ::N,\mathbb{NCY}}\mathbf{N}=\mp@code{N}
    REAL (8),D IM ENSION (:) ::A ,Y
    ```

```

    CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
    \mathbb{NTEGER (8)::N,\mathbb{NCY}}\mathbf{N}=\mp@code{N}
    REAL (8),D IM ENSION (:) ::A ,Y
    ```
C INTERFACE
    \#include < sunperfh>
void dtpm v (charuplo, chartransa, chardiag, intn, double *a, double *y, int incy);
void dtpm v_64 (charuplo, chartransa, char diag, long n, double *a, double *y, long incy);

\section*{PURPOSE}
dtpm \(v\) perform s one of the \(m\) atrix-vector operations \(x:=A * x\), or \(\mathrm{x}:=A{ }^{*} \mathrm{x}\), where x is an n elem entvectorand \(A\) is an \(n\) by \(n\) unit, ornon-unit, upper or low er triangular m atrix, supplied in packed form .

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whetherthem atrix is an upper or low er triangularm atrix as follow s:

UPLO = U'or 4 ' A is an upper triangular \(m\) atrix.

UPLO = L' or 'I' A is a lower triangular m atrix.

U nchanged on exit.
TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} \mathrm{x}:=\mathrm{A}{ }^{*} \mathrm{x}\).

TRANSA \(=\) T'ort' \(x:=A * x\).

TRANSA \(=\) C'or \(匕^{\prime} \mathrm{x}:=\mathrm{A} * \mathrm{x}\).
U nchanged on exit.

TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
On entry, D IA G specifies whether ornotA is unit triangular as follow s:

D \(\mathbb{A G G}=\mathrm{U}\) 'or \(\mathrm{L}^{\prime} \mathrm{A}\) is assum ed to be unit tri-
angular.
\(D \mathbb{A G}=N\) 'or \(h\) ' \(A\) is notassum ed to be unit triangular.

U nchanged on exit.

N (input)
O n entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.

A (input)
( \((n *(n+1)) / 2)\). Before entry with UPLO = \(U\) ' or \(L\) ', the amay A m ustcontain the upper triangularm atrix packed sequentially, colum \(n\) by colum \(n\), so thatA (1) contains a (1, 1), A (2) and \(A(3)\) contain \(a(1,2)\) and \(a(2,2)\) respectively, and so on. Before entry w ith UPLO = L' or I', the anray A m ust contain the low er triangular m atrix packed sequentially, colum \(n\) by colum \(n\), so thatA (1) contains a (1,1),A(2) and \(A(3)\) contain \(a(2,1)\) and \(a(3,1)\) respec tively, and so on. N ote thatw hen D IA G \(=U U^{\prime}\) or \(G\) ', the diagonal elem ents of A are notreferenced, butare assum ed to be unity. U nchanged on exit.

Y (input/output)
\((1+(n-1) * a b s(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) must contain the \(n\) elem ent vectorx. O n exit, \(Y\) is overw ritten \(w\) ith the tranform ed vector \(x\).
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(. \mathbb{N} C Y<>0\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtprfs -provide enrorbounds and backw ard error estim ates forthe solution to a system of linearequations \(w\) th a triangular packed coefficientm atrix

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE D TPRFS (UPLO,TRANSA,D IAG,N,NRHS,A,B,LDB,X,LDX,}
FERR,BERR,W ORK,W ORK2,INFO)

```
CHARACTER * 1 UPLO, TRANSA, D IA G
\(\mathbb{N}\) TEGERN,NRHS,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGERWORK2(*)
DOUBLE PRECISION A (*), B (LDB,\(\star), \mathrm{X}(\mathrm{LDX}, \star), \operatorname{FERR}(*), \operatorname{BERR}(*)\),
W ORK (*)
SU BROUTINEDTPRFS_64 (UPLO,TRANSA, DIAG,N,NRHS,A,B,LDB,X,LDX,
    FERR,BERR,W ORK,WORK2, \(\mathbb{N} F O\) )
CHARACTER * 1 UPLO, TRANSA, DIAG
\(\mathbb{N}\) TEGER*8N,NRHS,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER *8 W ORK 2 (*)

W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE TPRFS (UPLO, [TRANSA],D IA G,N,NRHS,A,B,[LDB],X,[LDX], FERR, BERR, [W ORK], [W ORK 2], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
\(\mathbb{I N}\) TEGER ::N,NRHS,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) ::W ORK2
REAL (8),D IM ENSION (:) ::A,FERR,BERR,W ORK

REAL (8), D \(\mathbb{I}\) ENSION (:,:) ::B,X

SU BROUTINE TPRFS_64 (UPLO, [TRANSA ],D IA G ,N,NRHS,A,B,[LDB],X, [LDX ],FERR,BERR, [WORK],[WORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER (8), D \(\mathbb{I M} E N S I O N(:):: W O R K 2\)
REAL (8), D \(\mathbb{M}\) ENSION (:) :: A, FERR,BERR,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (: : : ) : : B , X

\section*{C INTERFACE}
\#include <sunperfh>
void dtprfs (charuplo, chartransa, chardiag, int n, int nrhs, double *a, double *b, int ldb, double *x, int ldx, double *ferr, double *berr, int *info);
void dtpres_64 (charuplo, chartransa, char diag, long n, long nihs, double *a, double *b, long ldb, double \({ }^{*} x\), long ldx, double *ferr, double *berr, long *info);

\section*{PURPOSE}
dtprifs provides errorbounds and backw ard error estim ates forthe solution to a system of linear equations w ith a triangular packed coefficientm atrix.

The solution \(m\) atrix \(X\) m ustbe com puted by STPTRS or some other \(m\) eans before entering this routine. STPRFS does not do iterative refinem entbecause doing so cannotim prove the backw ard error.

\section*{ARGUMENTS}

\section*{UPLO (input)}
\(=U\) ': A is upper triangular;
= \({ }^{L}\) ': A is low er triangular.

TRANSA (input)
Specifies the form of the system of equations:
\(=\mathrm{N}^{\prime}: \mathrm{A} * \mathrm{X}=\mathrm{B} \quad\) (N o transpose)
\(=T\) ': \(A * * T * X=B \quad\) ( ranspose)
\(=C^{\prime}: A * * H * X=B \quad\) (C onjugate transpose \(=T\) ran spose)

TRANSA is defaulted to \(N\) 'forF95 \(\mathbb{I N}\) TERFACE.

D IA G (input)
\(=\mathrm{N}\) : A is non-unit triangular;
\(=\mathrm{U}\) ': A is unit triangular.

N (input) The orderof the matrix A. \(\mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the m atrices B and X. NRHS \(>=0\).

A (input) The upper or low er triangular m atrix A, packed colum nw ise in a linear array. The jth colum \(n\) of A is stored in the array A as follow s: if UPLO = \(U^{\prime}, A(i+(j-1) * j 2)=A(i, j)\) for \(1<=i<=j\); if UPLO = L', A ( \(\left.i+(j-1)^{*}(2 \star n-j) / 2\right)=A(i, j)\) for \(j=i<=n\). IfD \(\mathbb{A} G=U\) ', the diagonalelem ents of A are not referenced and are assum ed to be 1 .
\(B\) (input) The righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, N)\).

X (input) The solution \(m\) atrix X .

LD X (input)
The leading dim ension of the amay X . LD X >= \(\max (1, \mathbb{N})\).

\section*{FERR (output)}

The estim ated forw ard enrorbound for each solution vector \(X()\) ) the \(j\) th column of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\neg)\), FERR \((\neg)\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(\mathrm{X}(\mathcal{\nu})-\mathrm{X}\) TRU E ) divided by the \(m\) agnitude of the largestelem ent in \(X(j)\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

BERR (output)
The com ponentw ise relative backw ard error of each solution vector \(X(j)\) (ie., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension \((3 * N)\)

W ORK 2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N}\) FO (output)
= 0: successfiulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dtpsv - solve one of the system s ofequationsA *x = b, or

``` A * \(\mathrm{x}=\mathrm{b}\)

\section*{SYNOPSIS}
```

SUBROUTINEDTPSV (UPLO,TRANSA,D IAG,N,A,Y, INCY)
CHARACTER * 1 UPLO,TRANSA,DIAG
\mathbb{NTEGER N, INCY}
DOUBLE PRECISION A (*),Y (*)
SUBROUT\mathbb{NEDTPSV_64(UPLO,TRANSA,D IAG,N,A,Y,INCY)}
CHARACTER * 1 UPLO,TRANSA,DIAG
INTEGER*8N,\mathbb{NCY}
DOUBLE PRECISION A (*),Y (*)

```
F95 INTERFACE
    SU BROUTINE TPSV (UPLO, [TRANSA ], D \(\mathbb{A} G, \mathbb{N}], A, Y,[\mathbb{N C Y}])\)
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
    \(\mathbb{N} T E G E R:: N, \mathbb{N} C Y\)
    REAL (8),D IM ENSION (:) ::A , Y
    SU BROUTINE TPSV_64 (UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{N}], A, Y,[\mathbb{N} C Y])\)
    CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
    \(\mathbb{N} T E G E R(8):: N, \mathbb{N C Y}\)
    REAL (8), D IM ENSION (:) ::A, Y
C INTERFACE
    \#include <sunperfh>
void dtpsv (charuplo, chartransa, chardiag, intn, double *a, double *y, intincy);
void dtpsv_64 (charuple, chartransa, char diag, long n, double *a, double *y, long incy);

\section*{PURPOSE}
dtpsv solves one of the system s of equations \(A{ }^{*} x=b\), or \(A\) * \(x=b\), where \(b\) and \(x\) are \(n\) elem entvectors and \(A\) is an \(n\) by \(n\) unit, ornon-unit, upper or low er triangular m atrix, supplied in packed form .

N o testforsingularity or near-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.

\section*{ARGUMENTS}

UPLO (input)
O n entry, UPLO specifies w hether the \(m\) atrix is an upper or low er triangularm atrix as follow \(s\) :
\(\mathrm{UPLO}=\mathrm{U}\) 'or U ' \(A\) is an upper triangular m atrix.
\(\mathrm{UPLO}=\mathrm{L}\) ' or \(\mathrm{I}^{\prime}\) A is a lower triangular \(m\) atrix.

U nchanged on exit.

TRANSA (input)
On entry, TRAN SA specifies the equations to be solved as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} A{ }^{*} \mathrm{x}=\mathrm{b}\).

TRANSA = T'ort'A *x = b.

TRANSA \(=C^{\prime}\) or \(\mathrm{C}^{\prime} A{ }^{*} \mathrm{x}=\mathrm{b}\).

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)

On entry, D IA G specifies whether ornotA is unit triangular as follow s:

D \(\mathbb{A G G}=\mathrm{U}\) 'or \(\mathrm{L}^{\prime} \mathrm{A}\) is assum ed to be unit triangular.
\(D \mathbb{A} G=N\) 'or \(h\) ' \(A\) is notassumed to be unit triangular.

U nchanged on exit.

N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.
A (input)
\(\left(\left(\mathrm{n}^{*}(\mathrm{n}+1)\right) / 2\right)\). Before entry w th \(\mathrm{UPLO}=\) \(U\) ' or U ', the array A m ustcontain the upper triangularm atrix packed sequentially, colum \(n\) by colum n , so that A (1) contains a ( 1,1 ), A (2) and \(A(3)\) contain \(a(1,2)\) and \(a(2,2)\) respectively, and so on. Before entry w ith UPLO = \(\mathrm{L}^{\prime}\) or 1 ', the amay A m ust contain the low er triangular \(m\) atrix packed sequentially, colum \(n\) by colum n, so thatA (1) contains a (1,1), A (2) and \(A\) ( 3 ) contain a \((2,1)\) and a \((3,1)\) respectively, and so on. N ote thatw hen D IA G \(=\) U' or G ', the diagonal elem ents of A are notreferenced, but are assum ed to be unity. Unchanged on exit.

Y (input/output)
\((1+(n-1) * a b s(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y \mathrm{~m}\) ust contain the n elem ent righthand side vectorb. O \(n\) exit, \(Y\) is overw ritten \(w\) ith the solution vector \(x\).
\(\mathbb{N} C Y\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of. \(\mathbb{I N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dtptri-com pute the inverse of a real upper or low er triangularm atrix A stored in packed form at

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DTPTRI(UPLO,D IAG,N,A, INFO)}
CHARACTER * 1 UPLO,DIAG
\mathbb{NTEGER N, INFO}
DOUBLE PRECISION A (*)
SU BROUT\mathbb{NE DTPTRI_64(UPLO,D IA G ,N,A , INFO )}
CHARACTER * 1 UPLO,D IAG
INTEGER*8 N, INFO
DOUBLE PRECISION A (*)
F95 INTERFACE

```

```

CHARACTER (LEN=1) ::UPLO,D IAG
\mathbb{NTEGER ::N,\mathbb{NFO}}\mathbf{N}=0
REAL (8),D IM ENSION (:) ::A
SU BROUT\mathbb{NE TPTRI_64 (UPLO,D IA G ,N,A, [NNFO ])}
CHARACTER (LEN=1)::UPLO,D IAG
\mathbb{NTEGER (8)::N, INFO}
REAL (8),D IM ENSION (:) ::A

```
void dtptri(charuplo, chardiag, int n, double *a, int *info);
void dtptri_ 64 (charuple, chardiag, long n, double *a, long *info);

\section*{PURPOSE}
dtptricom putes the inverse of a realupper or low er triangularm atrix A stored in packed form at.

\section*{ARGUMENTS}
```

UPLO (input)
= U ':A is uppertriangular;
= IL':A is low ertriangular.

```

D IA G (input)
\(=\mathrm{N}: A\) is non-unittriangular;
\(=\mathrm{U}: \mathrm{A}\) is unit triangular.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the upper or low er triangularm atrix A, stored colum nw ise in a linearamay. The jth colum \(n\) of \(A\) is stored in the array \(A\) as follow \(s\) : if UPLO \(=U ', A(i+(j-1) * j 2)=A(i, j)\) for \(1<=i<=j\) ifUPLO \(=\mathbb{L}, A(i+(j 1) \star((2 * n-j) / 2)=\) \(A(i, j)\) for \(j<=i<=n\). See below for further details. On exit, the (triangular) inverse of the original \(m\) atrix, in the sam e packed storage format.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), \(A(i, i)\) is exactly zero. The triangular \(m\) atrix is singular and its inverse can notbe com puted.

\section*{FURTHER DETAILS}

A triangularm atrix A can be transferred to packed storage using one of the follow ing program segm ents:
\(\mathrm{UPLO}=\mathrm{U} ': \quad \mathrm{UPLO}=\mathrm{L}^{\prime}:\)
J \(=1\)
DO \(2 \mathrm{~J}=1\), N
\(\pi=1\)
DO \(2 \mathrm{~J}=1\), N
D○ \(1 \mathrm{I}=1\), J
DO \(1 \mathrm{I}=\mathrm{J}, \mathrm{N}\)
\(A(J C+I-1)=A(I, J) \quad A(J C+I-J)=\)
A (I, N)
\(\begin{array}{ll}1 & \text { CONTINUE } \\ \mathbb{C}=\mathrm{C}+\mathrm{J} & \mathbb{C O N T I N U E} \\ & \mathbb{C}=\mathrm{J}+\mathrm{N}-\mathrm{J}+\end{array}\)
1
2 CONTINUE \(\quad 2\) CONTINUE

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

dtptrs - solve a triangular system of the form A * X = B

``` orA \({ }^{* *}\) T * \(\mathrm{X}=\mathrm{B}\),

\section*{SYNOPSIS}
```

SU BROUTINE D TPTRS(UPLO,TRANSA,D IAG,N,NRHS,A,B,LD B,NNFO)
CHARACTER * 1 UPLO,TRANSA,D IAG
\mathbb{NTEGERN,NRHS,LDB,INFO}
DOUBLE PRECISION A (*),B (LDB,*)
SU BROUT\mathbb{NEDTPTRS_64(UPLO,TRANSA,D IAG,N,NRHS,A,B,LDB,INFO)}
CHARACTER * 1 UPLO,TRANSA,D IAG
\mathbb{NTEGER*8N,NRHS,LDB,INFO}
DOUBLE PRECISION A (*),B (LDB,*)

```
F95 INTERFACE
    SU BROUTINE TPTRS (UPLO,TRANSA,D \(\mathbb{A} G, N, N R H S, A, B,[L D B],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
    \(\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O\)
    REAL (8),D \(\mathbb{M}\) ENSION (:) ::A
    REAL (8),D IM ENSION (:,:) ::B
    SU BROUTINE TPTRS_64 (UPLO,TRANSA,D \(\mathbb{A} G, N, N R H S, A, B,[L D B],[\mathbb{N} F O])\)
    CHARACTER (LEN=1)::UPLO,TRANSA,D IA G
    \(\mathbb{N}\) TEGER (8) :: N,NRHS,LDB, \(\mathbb{N}\) FO
    REAL (8),D \(\mathbb{M}\) ENSION (:) ::A
    REAL (8),D IM ENSION (:,:) ::B

\section*{C INTERFACE}
\#include <sunperfh>
void dtptrs (char uplo, char transa, chardiag, int \(n\), int nrhs, double *a, double *b, int ldb, int *info);
void dtptrs_64 (charuplo, chartransa, char diag, long n, long nihs, double *a, double *b, long ldb, long *info);

\section*{PURPOSE}
dtpters solves a triangular system of the form
w here \(A\) is a triangularm atrix of order \(N\) stored in packed form at, and B is an N boy-NRH S m atrix. A check ism ade to verify thatA is nonsingular.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) : A is upper triangular;
\(=\mathbb{L}^{\prime}: A\) is low er triangular.

TRANSA (input)
Specifies the form of the system of equations:
\(=\mathrm{N}^{\prime}: \mathrm{A} * \mathrm{X}=\mathrm{B} \quad\) (Notranspose)
\(=T{ }^{\prime}: A * * T * X=B \quad\) ( ranspose)
\(=C^{\prime}: A * * H * X=B \quad\) (C onjugate transpose \(=T\) ran spose)

D IA G (input)
\(=\mathrm{N}^{\prime}: A\) is non-unittriangular;
\(=U\) ': A is unittriangular.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of right hand sides, i.e., the num ber
of colum ns of the m atrix B. NRHS \(>=0\).

A (input) The upper or low er triangular matrix A, packed colum nw ise in a linear amay. The jth colum n of
A is stored in the aray A as follow s: ifUPLO =
\(U ', A(i+(j-1) \star j 2)=A(i, j)\) for \(1<=i<=j\) if
\(\mathrm{UPLO}=\mathrm{L}, \mathrm{A}\left(i+(j-1)^{\star}\left(2{ }^{\star} \mathrm{n}-\mathrm{j}\right) / 2\right)=A(i, j)\) for
\(j=\mathrm{i}<=\mathrm{n}\) 。

B (input/output)
On entry, the righthand side m atrix B. On exit, if \(\mathbb{N} F O=0\), the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
<0: if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the \(i\)-th diagonalelem ent of \(A\) is zero, indicating that the \(m\) atrix is singular and the solutions \(X\) have notbeen com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtrans -transpose and scale source m atrix

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDTRANS (PLACE,SCALE,SOURCE,M,N,DEST)}
CHARACTER * 1 PLACE
\mathbb{NTEGERM,N}
DOUBLE PRECISION SCALE
DOUBLE PRECISION SOURCE (*),DEST (*)
SUBROUT\mathbb{NEDTRANS_64(PLACE,SCALE,SOURCE,M,N,DEST)}
CHARACTER * 1 PLACE
INTEGER*8M ,N
DOUBLE PRECISION SCALE
DOUBLE PRECISION SOURCE (*),DEST (*)

```

\section*{F95 INTERFACE}

SU BROUTINE TRANS ([PLACE],SCALE,SOURCE,M,N, DEST])
CHARACTER (LEN=1) ::PLACE
INTEGER :: M , N
REAL (8) :: SCALE
REAL (8),D \(\mathbb{M}\) ENSION (:) ::SOURCE,DEST

SU BROUTINE TRANS_64 (PLACE],SCALE,SOURCE,M,N, DEST])
CHARACTER (LEN=1) ::PLACE
\(\mathbb{N} T E G E R(8):: M, N\)
REAL (8) :: SCALE
REAL (8),D \(\mathbb{M}\) ENSION (:) ::SOURCE,DEST

\section*{C INTERFACE}
\#include <sunperfh>
void dtrans (charplace, double scale, double *source, intm , intn, double *dest);
void dtrans_64 (charplace, double scale, double *source, long \(m\), long \(n\), double *dest);

\section*{PURPOSE}
dtrans scales and transposes the sourcem atrix. The N \(2 \times \mathrm{N} 1\) result is w ritten into SO U RCE when PLACE = I'or 'i', and DEST when PLACE = 0 'or \(\mathrm{b}^{\prime}\).
PLACE = 'I'or \({ }^{1}\) ': SOURCE = SCALE * SOURCE'
PLACE = O'orb':DEST = SCALE * SOURCE'

\section*{ARGUMENTS}

PLACE (input)
Type of transpose. 'I'or i'for in-place, \(0^{\prime}\) or b'for out-of-place. 'T' is default.

SCALE (input)
Scale factor on the SO U RCE m atrix.
SOURCE (input/output)
\((M, N)\) on input. A ray of \((\mathbb{N}, M)\) on output if in-place transpose.
\(M\) (input)
N um ber of row s in the SO U RCE \(m\) atrix on input.
\(N\) (input)
\(N\) um ber of colum ns in the SO U RCE m atrix on input.
DEST (output)
Scaled and transposed SOURCE matrix if out-ofplace transpose. N ot referenced if in-place transpose.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtroon -estim ate the reciprocal of the condition num ber of a triangular \(m\) atrix \(A\), in either the 1 -norm or the infinity-norm

\section*{SYNOPSIS}
```

SUBROUTINEDTRCON NORM,UPLO,DIAG,N,A,LDA,RCOND,W ORK,W ORK 2,
\mathbb{NFO)}
CHARACTER * 1 NORM,UPLO,DIAG
INTEGER N,LDA,INFO
INTEGER W ORK2 (*)
DOUBLE PRECISION RCOND
DOUBLE PRECISION A (LDA ,*),W ORK (*)
SUBROUT\mathbb{NEDTRCON_64NORM,UPLO,DIAG,N,A,LDA,RCOND,W ORK,W ORK2,}
\mathbb{NFO)}
CHARACTER * 1 NORM,UPLO,DIAG
\mathbb{NTEGER*8N,LDA,INFO}
INTEGER*8 W ORK 2 (*)
DOUBLE PRECISION RCOND
DOU BLE PRECISION A (LDA,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE TRCON \(\mathbb{N} O R M, U P L O, D \mathbb{A} G, N, A,[L D A], R C O N D,[W O R K],[W O R K 2]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::NORM,UPLO,D IAG
\(\mathbb{N}\) TEGER ::N,LDA, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) ::W ORK2
REAL (8) :: RCOND

REAL (8), D IM ENSION (:) ::W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A
SU BROUTINE TRCON_64 NORM, UPLO,D \(\mathbb{A} G, N, A,[L D A], R C O N D,[W O R K]\), [W ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::NORM,UPLO,DIAG
\(\mathbb{N}\) TEGER (8) :: N, LDA, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) ::W ORK2
REAL (8) ::RCOND
REAL (8),D IM ENSION (:) ::W ORK
REAL (8), D \(\mathbb{M}\) ENSION (: : : : : A

\section*{C INTERFACE}
\#include <sunperfh>
void dtrcon (charnorm , charuple, chardiag, int n, double *a, int lda, double *rcond, int *info);
void dtrcon_64 (charnorm , char uplo, chardiag, long n, double *a, long lda, double *roond, long *info);

\section*{PURPOSE}
dtroon estim ates the reciprocal of the condition num ber of a triangular \(m\) atrix \(A\), in either the 1 -norm orthe infinitynorm .

The norm of \(A\) is com puted and an estim ate is obtained for norm (inv (A )), then the reciprocal of the condition num ber is com puted as
\(\operatorname{RCOND}=1 /(\operatorname{norm}(A) *\) norm (inv \((A)))\).

\section*{ARGUMENTS}

NORM (input)
Specifies w hether the 1-nom condition number or the infinity-norm condition num ber is required:
= 1 'or \(0^{\prime}\) : 1-nom ;
= I': Infinity-norm .
UPLO (input)
\(=\mathrm{U}: \mathrm{A}\) is uppertriangular;
= L ': A is low er triangular.
D IA G (input)
\(=N^{\prime}: A\) is non-unit triangular;
\(=\mathrm{U}\) ': A is unit triangular.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input) The triangularm atrix \(A\). If \(\mathrm{UPLO}=\mathrm{U}\) ', the leading N -by N upper triangularpart of the array A contains the upper triangular matrix, and the strictly low ertriangularpartofA is not referenced. If UPLO = L', the leading N -by N lower triangular part of the array A contains the low er triangularm atrix, and the strictly uppertriangular part ofA is not referenced. IfD \(\mathbb{I A} G=U\) ', the diagonalelem ents ofA are also not referenced and are assum ed to be 1 .
LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, \mathbb{N})\).

RCOND (output)
The reciprocal of the condition number of the \(m\) atrix \(A\), com puted as RCOND \(=1 /(\) norm (A) * norm (inv (A))).

W ORK (w orkspace)
dim ension ( \(3 *\) N )

W ORK 2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{I N}\) FO = -i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dtrevc - com pute som e or all of the right and/or lefteigen-
vectors of a real upperquasi-triangularm atrix \(T\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDTREVC (S\mathbb{DE,HOW MNY,SELECT,N,T,LDT,VL,LDVL,VR,}}\mathbf{N},\textrm{L},\textrm{L}
LDVR,MM,M,WORK,INFO)

```
CHARACTER * 1 SDE D ,HOW M NY
\(\mathbb{N} T E G E R N, L D T, L D V L, L D V R, M M, M, \mathbb{N} F O\)
LOG ICAL SELECT (*)
DOUBLE PRECISION T (LDT,*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
SUBROUTINEDTREVC_64 (SDE,HOW MNY,SELECT,N,T,LDT,VL,LDVL,VR,
    LDVR,MM,M,WORK, \(\mathbb{N} F O)\)
CHARACTER * 1 SIDE,HOW M NY
\(\mathbb{N}\) TEGER*8N,LDT,LDVL,LDVR,MM,M, \(\mathbb{N} F O\)
LOG ICAL*8 SELECT (*)
DOUBLE PRECISION T (LDT,*),VL (LDVL, \(), V R(L D V R, \star), W\) ORK ( \({ }^{\star}\) )

\section*{F95 INTERFACE}

SU BROUTINE TREVC (SDE,HOW MNY,SELECT,N,T, [LDT],VL, [LDVL],VR, \([L D V R], M M, M,[\mathbb{N} O R K],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::SDE,HOW MNY
\(\mathbb{N}\) TEGER :: N,LDT,LDVL,LDVR,MM,M, \(\mathbb{N} F O\)
LOG ICAL,D IM ENSION (:) ::SELECT
REAL (8),D IM ENSION (:) ::W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::T,VL,VR

SU BROUTINE TREVC_64 (SDE ,HOW M NY, SELECT,N,T, [LDT],VL, [LDVL], VR, [LDVR], MM, M, [WORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::SDE,HOW MNY
\(\mathbb{N}\) TEGER (8) :: N, LD T, LDVL, LDVR, M M , M , \(\mathbb{N}\) FO
LOGICAL (8), D IM ENSION (:) :: SELECT
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::T,VL,VR

\section*{C INTERFACE}
\#include <sunperfh>
void dtrevc (charside, charhow my, int *select, intn, double *t, int ldt, double *Vl, int ldvl, double *Vr, int ldvr, intm \(m\), int *m, int*info);
void dtrevc_64 (charside, charhow m ny, long *select, long n, double *t, long ldt, double *vl, long ldvl, double *Vr, long ldvr, long m m , long *m , long *info);

\section*{PURPOSE}
dtrevc com putes som e orall of the rightand/or left eigenvectors of a realupperquasi-triangularm atrix T.

The righteigenvectorx and the left eigenvector y of \(T\) corresponding to an eigenvalue w are defined by:
\[
\mathrm{T}^{\star} \mathrm{x}=\mathrm{w}^{\star} \mathrm{x}, \quad \mathrm{y}^{\star} \mathrm{T}=\mathrm{w}^{\star} \mathrm{y}^{\prime}
\]
w here y'denotes the conjugate transpose of the vectory.

If alleigenvectors are requested, the routine \(m\) ay either retum the \(m\) atrioes \(X\) and/or \(Y\) of rightor lefteigenvectors of \(T\), or the products \(Q * X\) and/or \(Q * Y\), where \(Q\) is an input orthogonal
\(m\) atrix. If \(T\) w as obtained from the real-Schur factorization of an originalm atrix \(A=Q * T * Q\) ', then \(Q * X\) and \(Q * Y\) are the \(m\) atrices of right or lefteigenvectors of \(A\).

T m ustbe in Schurcanonical form (as retumed by SH SEQR), thatis, block upper triangularw ith 1-boy-1 and \(2-b y-2\) diagonalblocks; each 2 -by-2 diagonal block has its diagonal elem ents equal and its off-diagonalelem ents of opposite sign. C omesponding to each 2 -by-2 diagonalblock is a com plex conjugate pair ofeigenvalues and eigenvectors; only one eigenvector of the pair is com puted, nam ely the one corresponding to the eigenvalue \(w\) ith posilive im aginary part.

\section*{ARGUMENTS}

STDE (input)
\(=\mathrm{R}\) : com pute righteigenvectors only;
= L ': com pute lefteigenvectors only;
\(=\mathrm{B}\) ': com pute both right and lefteigenvectors.

HOW MNY (input)
= A ': com pute all right and/or left eigenvec-
tors;
= B ': com pute all right and/or left eigenvec-
tors, and backtransform them using the input m atrices supplied in VR and/orV L; \(=\mathrm{S}\) ': com pute selected right and/or left eigenvectors, specified by the logicalamay SELEC T .

\section*{SELECT (input/output)}

If H OW M NY = S', SELEC T specifies the eigenvectors
to be com puted. IfHOW M NY = A 'or B', SELECT is
not referenced. To select the real eigenvector
corresponding to a realeigenvalue w ( \(\mathcal{j}\), SELECT ( \()\)
m ustibe set to TRUE.. To select the complex eigenvector comesponding to a com plex conjugate
pair w ( \(\mathcal{j}\) ) and w ( \(j+1\) ), either SELECT ( \(\mathcal{j}\) ) or
SELECT (j+1) must be setto TRUE.; then on exit
SELECT ( \(\mathfrak{j}\) ) is TRUE. and SELECT ( \(\mathfrak{j}+1\) ) is FALSE ..

N (input) The order of the m atrix \(\mathrm{T} \cdot \mathrm{N}>=0\).
T (input/output)
The upper quasi-triangular \(m\) atrix \(T\) in Schur canonicalform.

LD T (input)
The leading dim ension of the array T. LD T >= \(\max (1, \mathbb{N})\).

VL (input/output)
On entry, ifSDE = L'or B'and HOW M NY = B',
VL must contain an \(N\)-by N m atrix Q (usually the orthogonalm atrix \(Q\) of Schurvectors retumed by SH SEQR). On ex斗, if \(S \mathbb{D} E=\mathrm{L}\) 'or B',VL contains: if HOW M NY = A', the matrix Y of left eigenvectors of \(T\); VL has the sam equasi-low er triangular form as T'. If T ( \(i, i\) i) is a real eigenvalue, then the \(i\)-th columnVL (i) ofVL is its comesponding eigenvector. If \(T\) ( \(i: i+1, i: i+1\) ) is a 2 -by-2 block whose eigenvalues are complexconjugate eigenvalues of T , then VL (i)+sqıt(
1)*V L (i+1) is the com plex eigenvector comesponding to the eigenvalue \(w\) ith positive realpart. if HOW M NY = B', them atrix Q *Y ; ifHOW M NY = S', the lefteigenvectors of \(T\) specified by SELEC T, stored consecutively in the colum ns ofV \(L\), in the sam \(e\) order as their eigenvalues. A com plex eigenvector comesponding to a com plex eigenvalue is stored in tw o consecutive colum ns, the first holding the real part, and the second the im aginary part. If \(S \mathbb{D E}=\mathrm{R}, \mathrm{VL}\) is not referenced.

LD V L (input)
The leading dim ension of the aray VL. LDVL >= \(\max (1, N)\) if \(S \mathbb{D} E=L\) 'or \(B^{\prime} ;\) LDVL \(>=1\)
otherw ise.

VR (input/output)
On entry, if \(S \mathbb{D} E=R\) 'or \(B\) 'and HOW M NY = B', VR m ust contain an N -by -N m atrix Q (usually the orthogonalm atrix \(Q\) of Schurvectors retumed by SH SEQR). On exit, if \(S \mathbb{D} E=R\) 'or \(B\) ', \(V R\) contains: ifHOW MNY = A', the \(m\) atrix \(X\) of right eigenvectors of \(T\); VR has the sam equasi-upper triangular form as \(T\). If \(T(i, i)\) is a real eigenvalue, then the \(i\)-th colmmnVR (i) ofVR is its comesponding eigenvector. If \(T(i: i+1, i: i+1)\) is a 2 -by-2 block whose eigenvalues are complexconjugate eigenvalues of T, then VR (i)+sqrt(1)*VR(i+1) is the com plex eigenvector corresponding to the eigenvalue w th positive realpart. if HOW M NY = B', them atrix Q *X; if HOW MNY = S', the right eigenvectors of \(T\) specified by SELECT, stored consecutively in the colum ns of \(V R\), in the sam e order as their eigenvalues. A com plex eigenvector corresponding to a com plex eigenvahue is stored in tw o consecutive colum ns, the firstholding the real part and the second the im aginary part. If \(S \mathbb{D} E=\Sigma \prime, V R\) is not referenced.

LDVR (input)
The leading dim ension of the array VR. LDVR >= \(\max (1, N)\) if \(S \mathbb{D} E=R\) 'or B';LDVR >= 1 otherw ise.

M M (input)
The num berof colum ns in the arrays VL and/or VR. M M >=M.

M (output)
The num berof colum ns in the arrays VL and/or VR
actually used to store the eigenvectors. If HOW MNY = A 'or \(B\) ', M is set to N. Each selected real eigenvector occupies one colmm \(n\) and each selected com plex eigenvector occupies tw o colum ns.

W ORK (w orkspace)
dim ension ( \(3{ }^{*}\) N )
\(\mathbb{N}\) FO (output)
= 0 : successfinlexit
< 0 : if \(\mathbb{N F O}=-i\), the \(i\)-th argum ent had an illegal value

\section*{FURTHER DETAILS}

The algorithm used in this program is basically backw ard (forw ard) substitution, \(w\) th scaling to \(m\) ake the the code robustagainst possible overflow .

E ach eigenvector is norm alized so that the elem ent of largest \(m\) agnitude has \(m\) agnitude 1 ; here the \(m\) agnitude of a com plex num ber \((x, y)\) is taken to be \(|x|+|y|\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtrexc -reorderthe real Schur factorization of a real
\(m\) atrix \(A=Q * T * Q * * T\), so that the diagonalblock of \(T\) w th
row index \(\mathbb{F S T}\) ism oved to row \(\mathbb{L S T}\)

\section*{SYNOPSIS}

```

CHARACTER * 1 COMPQ

```

```

DOUBLE PRECISION T (LDT,*),Q (LDQ ,*),W ORK (*)
SU BROUTINE D TREXC_64 (COM PQ,N,T,LD T,Q,LDQ,FFST, ILST,W ORK,
INFO)
CHARACTER * 1 COMPQ
\mathbb{NTEGER*8N,LDT,LDQ,\mathbb{FST},\mathbb{LST,}\mathbb{N}FO}=0
DOUBLE PRECISION T (LDT,*),Q (LDQ ,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE TREXC (COMPQ,N,T, [LD T],Q, [LDQ], \(\mathbb{F} S T, \amalg S T,\left[\begin{array}{l}\text { O RK ], }\end{array}\right.\) [ \(\mathbb{N} F \mathrm{~F}\) ])

CHARACTER (LEN=1) ::COM PQ
\(\mathbb{N} T E G E R:: N, L D T, L D Q, \mathbb{F S T}, \mathbb{L} S T, \mathbb{N F O}\)
REAL (8),D IM ENSION (:) ::W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::T, Q
SU BROUTINE TREXC_64 (COMPQ,N,T,[LDT],Q,[LDQ], FST, ILST, [W ORK], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1): \(: C O M P Q\)
\(\mathbb{N} T E G E R(8):: N, L D T, L D Q, \mathbb{F S T}, \mathbb{L} S T, \mathbb{N} F O\)
REAL (8),D \(\mathbb{I M}\) ENSION (:) ::W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::T,Q

\section*{C INTERFACE}
\#include <sunperfh>
void dtrexc (char com pq, intn, double *t, int ldt, double
*q, int ldq, int *ifst, int *ilst, int *info);
void dtrexc_64 (charcom pq, long n, double *t, long ldt, double *q, long ldq, long *ifst, long *ilst, long
*info);

\section*{PURPOSE}
dtrexc reorders the real Schur factorization of a real \(m\) atrix \(A=Q * T * Q * * T\), so that the diagonalblock of \(T\) with row index \(\mathbb{F} S T\) ismoved to row \(\mathbb{H} S T\).

The realSchur form \(T\) is reordered by an orthogonal sim ilarity transform ation \(\mathrm{Z} * * \mathrm{~T} * \mathrm{~T} * \mathrm{Z}\), and optionally the m atrix Q of Schurvectors is updated by postm ultiplying itw ith \(Z\).

T m ustbe in Schurcanonical form (as retumed by SH SEQR), that is, block uppertriangularw ith 1 -by -1 and 2 -by- 2 diagonalblocks; each 2 -by- 2 diagonal block has its diagonal elem ents equal and its off-diagonalelem ents of opposite sign.

\section*{ARGUMENTS}

COM PQ (input)
\(=\mathrm{V}\) ': update the m atrix Q of Schurvectors;
= N ': do notupdate Q .

N (input) The order of the m atrix \(\mathrm{T} . \mathrm{N}>=0\).
T (input/output)
O \(n\) entry, the upperquasi-triangularm atrix \(T\), in Schur Schur canonical form. On exit, the reordered upper quasi-triangular \(m\) atrix, again in Schur canonical form .

LD T (input)
The leading dim ension of the anay T. LD T >= \(\max (1, N)\).

Q (input) \(O n\) entry, if \(C O M P Q=V\) ', the \(m\) atrix \(Q\) of Schur vectors. On exit, if COMPQ = V', Q has been postm ultiplied by the orthogonal transform ation \(m\) atrix \(Z\) which reorders \(T\). IfCOM \(P Q=N\) ', \(Q\) is not referenced.

LDQ (input)
The leading dim ension of the anay \(Q\). LDQ >= \(\max (1, \mathbb{N})\).

FST (input/output)
Specify the reordering of the diagonal blocks of
T. The block w th row index \(\mathbb{F S T}\) ism oved to row

LIST, by a sequence of transpositions betw een
adjacnt blocks. On exit, if \(\mathbb{F S T}\) pointed on entry to the second row of a 2 -by- 2 block, it is changed to point to the firstrow; ILST alw ays
points to the first row of the block in its final
position (w hich \(m\) ay differ from its input value by
+1 or -1 ). \(1<=\mathbb{F} S T<=N ; 1<=\mathbb{L} S T<=N\).
ㄴST (input/output)
See the description of IFST.
W ORK (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue
= 1: tw o adjacentblocks were too close to sw ap (the problem is very ill-conditioned); T m ay have been partially reordered, and ILST points to the first row of the currentposition of the block being \(m\) oved.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtrm \(m\)-perform one of the \(m\) atrix \(-m\) atrix operations \(B:=\) alpha*op (A ) *B , orB \(:=\) alpha*B *op (A )

\section*{SYNOPSIS}
```

SUBROUTINEDTRMM(S\mathbb{DE,UPLO,TRANSA,DIAG,M,N,ALPHA,A,LDA,B,}
LD B )
CHARACTER * 1SDE,UPLO,TRANSA,D IAG
INTEGER M ,N,LDA,LDB
DOUBLE PRECISION ALPHA
DOUBLE PRECISION A (LDA,*),B(LDB,*)

```

```

    LD B)
    ```
CHARACTER * 1 SDE, UPLO,TRANSA,D IAG
\(\mathbb{N}\) TEGER*8 M , N , LD A , LD B
DOUBLE PRECISION ALPHA
DOUBLE PRECISION A (LDA ,*), B (LDB,*)

\section*{F95 INTERFACE}

SU BROUTINE TRMM (SIDE,UPLO, [TRANSA ],D IA G, \(\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A]\), B, [LD B])

CHARACTER (LEN=1)::SDE,UPLO,TRANSA,D IAG
\(\mathbb{N} T E G E R:: M, N, L D A, L D B\)
REAL (8) :: ALPHA
REAL (8), D IM ENSION (:,:) ::A,B
SUBROUTINE TRMM_64 (SDEE,UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{M}], \mathbb{N}], A L P H A, A\), [LDA], B, [LDB])

CHARACTER ( \(L E N=1\) ) : : SDE E , UPLO, TRAN SA, D IA G
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{LDA}, \operatorname{LDB}\)
REAL (8) ::ALPHA
REAL (8), D \(\mathbb{M}\) ENSION (: : : ) :: A, B

\section*{C INTERFACE}
\#include <sunperfh>
void dtrm \(m\) (charside, charuplo, chartransa, chardiag, int m , int \(n\), double alpha, double *a, int lda, double *b, int ldb);
void dtrm m _64 (charside, charuplo, char transa, char diag, long \(m\), long \(n\), double alpha, double *a, long lda, double *b, long ldb);

\section*{PURPOSE}
dtrm \(m\) perform s one of the \(m\) atrix-m atrix operations \(B:=\) alpha*op ( \(A\) ) *B , orB \(:=\) alpha*B*op (A ) where alpha is a scalar, \(B\) is an \(m\) by \(n m\) atrix, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix and op (A ) is one of
\[
o p(A)=A \text { or } o p(A)=A^{\prime}
\]

\section*{ARGUMENTS}

SIDE (input)
On entry, SIDE specifies w hether op (A ) m ultiplies B from the leftor rightas follow s:
\(S \mathbb{D E}=\mathbb{L}\) 'or I' B := alpha*op (A )*B .
\(S \mathbb{D E}=\mathrm{R}^{\prime}\) or \(r^{\prime} \mathrm{B}:=\) alpha*B*op (A) .

U nchanged on exit.

UPLO (input)
O n entry, UPLO specifies w hether the m atrix A is an upper or low er triangularm atrix as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or U ' \(A\) is an upper triangular \(m\) atrix.
\(\mathrm{UPLO}=\mathrm{L}\) ' or \(\mathrm{I}^{\prime}\) A is a lower triangular \(m\) atrix.

U nchanged on exit.
TRANSA (input)
On entry, TRAN SA specifies the form ofop (A ) to be used in the \(m\) atrix \(m\) ultiplication as follow \(s\) :

TRANSA \(=N^{\prime}\) or \(h^{\prime}\) op(A) \(=A\).
TRANSA = T'or \(\mathrm{t}^{\prime}\) op(A) \(=\mathrm{A}^{\prime}\).

TRANSA = C'ort' op(A) = A'.

U nchanged on exit.

TRANSA is defaulted to N 'forF95 \(\mathbb{I N}\) TERFACE.
D IA G (input)
O n entry, D IA G specifies w hether or notA is unit triangular as follow s:

D \(\mathbb{A} G=U\) 'or \(u\) ' \(A\) is assum ed to be unit triangular.
\(D \mathbb{A} G=N\) 'or \(h^{\prime} A\) is notassum ed to be unit triangular.

U nchanged on exit.

M (input)
O \(n\) entry, M specifies the num ber of row s of B. M >= 0 . U nchanged on exit.
\(N\) (input)
O \(n\) entry, \(N\) specifies the num ber of colum ns of \(B\).
\(\mathrm{N}>=0\). U nchanged on exit.

A LPHA (input)
On entry, A LPH A specifies the scalar alpha.W hen alpha is zero then A is notreferenced and B need notbe setbefore entry. U nchanged on exit.

A (input)
DOUBLE PRECISION array OfD \(\mathbb{I M}\) ENSION (LDA, k), where \(k\) is \(m\) when \(S \mathbb{D E}=\mathbb{L}\) 'or \(I\) ' and is \(n\) when \(S D E=R\) 'or \(\mathrm{r}^{\prime}\).

Before entry \(w\) ith UPLO = U 'or L ', the leading \(k\) by \(k\) upper triangularpart of the array A \(m\) ustcontain the upper triangularm atrix and the strictly low ertriangularpartofA is not refer-
enced.

Before entry w ith UPLO = L 'or I', the leading \(k\) by \(k\) low ertriangularpart of the array \(A\) \(m\) ust contain the low ertriangularm atrix and the strictly uppertriangularpart of \(A\) is not referenced.

N ote thatw hen D IA G = U 'or L', the diagonal elem ents ofA are notreferenced either, butare assum ed to be one. U nchanged on exit.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen \(S \mathbb{D E}=' \mathrm{~L}\) 'or \(\mathrm{I}^{\prime}\) then LD \(A>=\mathrm{max}(1, m)\), when \(S \mathbb{D E}=R^{\prime}\) or \(r^{\prime}\) then LDA \(>=\max (1, n)\). U nchanged on exit.

B (input/output)
D OUBLE PRECISION amay ofD \(\mathbb{I}\) ENSION (LDB, n ).
Before entry, the leading \(m\) by \(n\) part of the array Bm ustcontain the \(m\) atrix B , and on exit is overw rilten by the transform ed \(m\) atrix.

LD B (input)
O n entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program.
LD B \(>=\max (1, m)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtrm \(v\)-perform one of the \(m\) atrix-vector operations \(x:=\) \(A * x\), or \(x: A * x\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDTRMV(UPLO,TRANSA,D IAG,N,A,LDA,Y, INCY)}
CHARACTER * 1UPLO,TRANSA,DIAG
INTEGERN,LDA,}\mathbb{N}C
DOUBLE PRECISION A (LDA,*),Y (*)
SU BROUT\mathbb{NE DTRM V_64(UPLO,TRANSA,D IAG ,N,A ,LDA,Y , INCY)}
CHARACTER * 1 UPLO,TRANSA,DIAG
\mathbb{NTEGER*8N,LDA,}\mathbb{N}CY
DOUBLE PRECISION A (LDA,*),Y (*)

```
F95 INTERFACE
    SU BROUTINE TRMV (UPLO, [TRANSA],D \(\mathbb{I A G}, \mathbb{N}], A,[L D A], Y,[\mathbb{N} C Y])\)
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
    \(\mathbb{N} T E G E R:: N, L D A, \mathbb{N} C Y\)
    REAL (8),D \(\mathbb{M}\) ENSION (:) :: Y
    REAL (8), D IM ENSION (:,:) ::A
    SU BROUTINE TRM V_64 (UPLO, [TRANSA ],D \(\mathbb{I A} G, \mathbb{N}], A,[L D A], Y,[\mathbb{N} C Y])\)
    CHARACTER (LEN=1)::UPLO,TRANSA,D IA G
    \(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} A, \mathbb{N} C Y\)
    REAL (8),D \(\mathbb{M}\) ENSION (:) ::Y
    REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dtm v (charuplo, chartransa, chardiag, intn, double
*a, int lda, double *y, intincy);
void dtrm v_64 (charuplo, chartransa, char diag, long n, double *a, long lda, double *y, long incy);

\section*{PURPOSE}
dtrm \(v\) perform s one of the \(m\) atrix-vectoroperations \(x: A{ }^{*} x\), or \(x:=A{ }^{*} x\), where \(x\) is an \(n\) elem entvector and \(A\) is an \(n\) by \(n\) unit, ornon-unit, upper or low er triangularm atrix.

\section*{ARGUMENTS}

UPLO (input)
O n entry, UPLO specifies w hether the \(m\) atrix is an upper or low er triangularm atrix as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or \(\mathrm{U}^{\prime} \mathrm{A}\) is an upper triangular \(m\) atrix.
\(\mathrm{UPLO}=\mathbb{L}\) ' or \(\mathrm{I}^{\prime}\) A is a lower triangular \(m\) atrix.

U nchanged on exit.

TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} x:=A * x\).

TRANSA \(=\) T'ort' \(x:=A * x\).

TRANSA \(=\) C'or \(C^{\prime} x:=A{ }^{*} x\).

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
O n entry, D IA G specifies w hether ornotA is unit triangular as follow s:

D IA G = U'or L ' \(A\) is assum ed to be unit triangular.
\(D \mathbb{A G}=N\) 'or \(h^{\prime} A\) is notassum ed to be unit triangular.

U nchanged on exit.

N (input)
On entry, N specifies the order of the m atrix A . \(\mathrm{N}>=0\). U nchanged on exit.

A (input)
Before entry w ith UPLO = U 'or L ', the leading \(n\) by \(n\) upper triangular part of the array A m ust contain the upper triangular \(m\) atrix and the strictly low er triangular part of A is not referenced. Before entry with UPLO = L 'or 1', the leading \(n\) by \(n\) low er triangularpart of the aray A m ustcontain the low er triangular \(m\) atrix and the strictly upper triangularpartofA is not referenced. N ote thatw hen \(\mathrm{D} \mathbb{I A G}=\mathrm{U}\) ' or L ', the diagonal elem ents of \(A\) are not referenced either, but are assum ed to be unity. U nchanged on exit.

LDA (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program . LD A \(>=\) \(\max (1, n)\). U nchanged on exit.

Y (input/output)
\((1+(n-1) * a b s(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) must contain the \(n\) elem ent vectorx. On exit, \(Y\) is overw rilten \(w\) ith the tranform ed vectorx.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y . \mathbb{N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtmis -provide errorbounds and backw ard error estim ates for the solution to a system of linear equations w ith a triangular coefficientm atrix

\section*{SYNOPSIS}
```

SUBROUTINE DTRRFS (UPLO,TRANSA,DIAG,N,NRHS,A,LDA,B,LDB,X,
LD X,FERR,BERR,W ORK,WORK2,INFO)

```
CHARACTER * 1 UPLO, TRANSA, DIAG
\(\mathbb{N}\) TEGER N,NRHS,LDA,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGERWORK2(*)
DOUBLE PRECISION A (LDA,*), B (LDB,*), X (LDX,*), FERR (*),
BERR (*),W ORK (*)
SU BROUTINE DTRRFS_64 (UPLO,TRANSA,D IAG,N,NRHS,A,LDA,B,LDB,X,
    LD \(\mathrm{X}, \mathrm{FERR}, \mathrm{BERR}, \mathrm{W}\) ORK,WORK2, \(\mathbb{N} F \mathrm{~F}\) )
CHARACTER * 1 UPLO, TRANSA, D IAG
\(\mathbb{N}\) TEGER*8N,NRHS,LDA,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER * 8 W ORK 2 ( \({ }^{\star}\) )
DOUBLE PRECISION A (LDA,*), B (LDB,*), X (LDX, \(\left.{ }^{\star}\right)\), FERR (*),
\(\operatorname{BERR}\) ( \()^{*}\), \(\mathrm{W} O \operatorname{OR}\) ( \({ }^{*}\) )

\section*{F95 INTERFACE}

SU BROUTINE TRRFS (UPLO, [TRANSA],D IA G,N,NRHS, A, [LDA ], B, [LDB], X, [LDX],FERR,BERR, [W ORK], [W ORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
\(\mathbb{N}\) TEGER :: N,NRHS,LDA,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) ::W ORK2
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK

REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A, B , X

SU BROUTINE TRRFS_64 (UPLO, [TRANSA ], D \(\mathbb{I A G}, N, N R H S, A,[L D A], B,[L D B]\), \(X,[L D X], F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N} F O])\)

CHARACTER ( \(L E N=1\) ) : : UPLO, TRANSA, D IA G
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDA,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER (8), D \(\mathbb{I M} E N S I O N(:):: W O R K 2\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A , B, X

\section*{C INTERFACE}
\#include <sunperfh>
void dtrnfs (charuplo, chartransa, chardiag, int n, int nrhs, double *a, int lda, double *b, int ldb, double *x, int ldx, double *ferr, double *berr, int *info);
void dtnfs_64 (charuplo, chartransa, char diag, long n, long nins, double *a, long lda, double *b, long ldly, double *x, long ldx, double *ferr, double *berr, long *info);

\section*{PURPOSE}
dtmes provides emorbounds and backw ard error estim ates forthe solution to a system of linear equations w ith a triangular coefficientm atrix.

The solution \(m\) atrix \(X\) m ustbe com puted by STRTRS or some other \(m\) eans before entering this routine. STRRFS does not do iterative refinem entbecause doing so cannotim prove the backw ard error.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}: \mathrm{A}\) is upper triangular;
\(=\mathbb{L}^{\prime}: A\) is low ertriangular.

TRANSA (input)
Specifies the form of the system ofequations:
\(=\mathrm{N}^{\prime}: A * X=B \quad\) (Notranspose)
\(=T T^{\prime}: A * T * X=B \quad\) ( ranspose)
\(=C^{\prime}: A * * H * X=B \quad\) (C onjugate transpose \(=T\) ran -
spose)

TRANSA is defaulted to \(N\) 'forF95 \(\mathbb{I N}\) TERFACE .

D IA G (input)
\(=\mathrm{N}\) : A is non-unit triangular;
\(=\mathrm{U}: \mathrm{A}\) is unit triangular.
N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the \(m\) atrices \(B\) and X. NRHS \(>=0\).
A (input) The triangularm atrix A . If \(\mathrm{PLO}=\mathrm{U}\) ', the leading N -by -N upper triangularpart of the array A contains the upper triangular \(m\) atrix, and the strictly low ertriangularpartofA is not referenced. IfU PLO = L', the leading N -by-N lower triangular part of the array A contains the low er triangularm atrix, and the strictly upper triangular part ofA is not referenced. IfD \(\mathbb{I A} G=U\) ', the diagonalelem ents ofA are also not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the anay A. LDA >= \(\max (1, \mathbb{N})\).
\(B\) (input) The righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, \mathbb{N})\).
\(X\) (input) The solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the array X . LD X >= \(\max (1, N)\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X()\) ) the \(j\) th column of the solution matrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{H}), \operatorname{FERR}(\mathcal{)})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(\mathrm{X}(\mathcal{\nu})-\mathrm{X}\) TRU E ) divided by the m agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

The com ponentw ise relative backw ard error of each solution vector \(X(\mathcal{)}\) (ie., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{(}\) ) an exactsolution).
W ORK (w orkspace)
dim ension ( \(3 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N F O}\) (output)
= 0 : successfiulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dtrisen -reorder the real Schur factorization of a real \(m\) atrix \(A=Q * T * Q * * T\), so that a selected cluster of eigenvalues appears in the leading diagonalblocks of the upper quasi-triangularm atrix \(T\),

\section*{SYNOPSIS}
```

SU BROUTINE DTRSEN (JOB,COMPQ,SELECT,N,T,LDT,Q,LDQ,W R,W I,M,
S,SEP,W ORK,LW ORK,INORK,LIN ORK,INFO)
CHARACTER * 1 JOB,COMPQ

```

```

INTEGER IN ORK (*)
LOG ICAL SELECT (*)
DOUBLE PRECISION S,SEP
DOUBLE PRECISION T (LDT,*),Q (LDQ ,*),W R (*),W I(*),W ORK (*)
SUBROUTINE DTRSEN_64(JOB,COMPQ,SELECT,N,T,LDT,Q,LDQ,W R,W I,
M,S,SEP,W ORK,LW ORK,IN ORK,LIN ORK,\mathbb{NFO)}

```
CHARACTER * \(1 \mathrm{JOB}, \mathrm{COMPQ}\)
\(\mathbb{N}\) TEGER*8N,LDT,LDQ,M,LW ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{I N}\) ORK (*)
LOG ICAL*8 SELECT (*)
DOUBLE PRECISION S,SEP


F95 INTERFACE
SU BROUTINE TRSEN (JOB,COM PQ,SELECT,N,T, [LDT],Q, [LDQ],W R,W I, \(\mathrm{M}, \mathrm{S}, \mathrm{SEP},[\mathbb{W}\) ORK \(]\), [LW ORK], [ \(\mathbb{W}\) ORK ], [LIWORK], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1) :: JOB,COMPQ
\(\mathbb{N}\) TEGER :: N,LDT,LDQ,M,LW ORK,LIN ORK, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
LOG ICAL,D IM ENSION (:) ::SELECT
REAL (8) :: S, SEP
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W R,W I,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:)::T,Q

SU BROUTINE TRSEN_64 (OB B,COM PQ,SELECT,N,T, [LD T],Q, [LDQ ],W R, W I, M , S, SEP, [W ORK ], [LW ORK ], [WW ORK ], [LIN ORK ], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) :: JOB,COM PQ
\(\mathbb{N}\) TEGER (8) :: N, LD T,LDQ,M,LW ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
LO G ICAL (8), D IM ENSIO N (:) :: SELECT
REAL (8) :: S, SEP
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W R,W I,W ORK
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::T,Q

\section*{C INTERFACE}
\#include <sunperfh>
void dtrsen (char j̀b, char com pq, int *select, intn, double
*t, int ldt, double *q, int ldq, double *w r , double *wi, int \({ }^{*}\) m, double \({ }^{\text {s }}\), double \({ }^{\text {sepp, int }}\) *info);
void dtrsen_64 (char jo.b, charcom pq, long *select, long n, double *t, long ldt, double *q, long ldq, double *w r, double *w i, long *m, double *s, double *sep, long *info);

\section*{PURPOSE}
dtrsen reorders the real Schur factorization of a real \(m\) atrix \(A=Q * T * Q * * T\), so that a selected chuster of eigenvalues appears in the leading diagonalblocks of the upper quasi-triangularm atrix \(T\), and the leading colum ns of \(Q\) form an orthonorm albasis of the corresponding right invariant subspace.

Optionally the routine com putes the reciprocal condition num bers of the cluster of eigenvalues and/or the invariant subspace.

T m ustbe in Schurcanonical form (as retumed by SH SEQR), that is, block uppertriangularw ith 1 -by -1 and 2 -by- 2 diagonalblocks; each 2-by-2 diagonal block has its diagonal elem nts equal and its off-diagonal elem ents of opposite sign.

\section*{ARGUMENTS}

JO B (input)
Specifies w hether condition num bers are required for the cluster ofeigenvalues ( S ) or the invariantsubspace (SEP):
= N ':none;
= E ': foreigenvalues only ( S );
\(=\mathrm{V}\) : for invariant subspace only (SEP);
= B ': forboth eigenvalues and invariant subspace ( S and SEP).

COMPQ (input)
= V ': update the m atrix Q ofSchurvectors;
\(=N\) ': do notupdate Q .
SELECT (input)
SELEC T specifies the eigenvalues in the selected cluster. To select a real eigenvalue w ( \(\mathcal{j}\), SELECT (j) must be set to w (i) and w (j+1), corresponding to a 2 -by-2 diagonalblock, either SELECT (i) orSELECT (j+1) orboth \(m\) ust be set to either both included in the cluster or both excluded.

N (input) The order of the m atrix \(\mathrm{T} . \mathrm{N}>=0\).
T (input/output)
O \(n\) entry, the upperquasi-triangularm atrix \(T\), in Schur canonical form. On exit, T is overw rilten by the reordered \(m\) atrix \(T\), again in Schur canonical form, w ith the selected eigenvalues in the leading diagonalblocks.

LD T (input)
The leading dim ension of the array T. LD T >= \(\max (1, \mathbb{N})\).

Q (input) \(O n\) entry, if \(C O M P Q=V\) ', the \(m\) atrix \(Q\) of Schur vectors. On exit, if COMPQ = V', Q hasbeen postm ultiplied by the orthogonal transform ation \(m\) atrix which reorders \(T\); the leading \(M\) colum ns of \(Q\) form an orthonorm al basis for the specified invariant subspace. If COM PQ \(=N^{\prime}\) ', Q is not referenced.

LD Q (input)

The leading dim ension of the array \(\mathrm{Q} . \mathrm{LD} \mathrm{Q}>=1\);
and if \(C O M P Q=V\) ', LD \(Q>=N\).
W R (output)
The realand im aginary parts, respectively, of the reordered eigenvalues of \(T\). The eigenvalues are stored in the sam e order as on the diagonal of \(T\), w th \(\mathrm{W} R(i)=T(i, i)\) and, if \(T(i: i+1, i: i+1)\) is a 2 -by-2 diagonalblock, W I(i) > 0 and W I(i+1) = -W I(i). N ote that if a com plex eigenvalue is sufficiently ill-conditioned, then its value may differ significantly from its value before reordering.

\section*{W I (output)}

See the description ofW R .
M (output)
The dim ension of the specified invariant subspace. \(0<=\mathrm{M}<=\mathrm{N}\).

S (output)
If \(J 0 B=E\) 'or \(B ', S\) is a lower bound on the reciprocal condition num ber for the selected clusterofeigenvalues. S cannot underestim ate the true reciprocal condition num berby \(m\) ore than a factor of sqrt \(\mathbb{N}\) ). If \(M=0\) or \(N, S=1\). If \(\mathcal{O D} B=\) N 'or V ', S is not referenced.

\section*{SEP (output)}

If \(\mathrm{JOB}=\mathrm{V}\) 'or B ', SEP is the estim ated reciprocal condition number of the specified invariant subspace. If \(M=0\) or \(N\), \(S E P=\) norm ( \(T\) ). If \(J O B=\) N 'or E ', SEP is notreferenced.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. If JOB \(=N\) ', LW ORK >= max ( \(1, N\) ); if \(J O B=E \prime\), LW ORK \(>=M * \mathbb{N}-M)\); if \(J O B=V\) 'or \(B\) ', LW ORK \(>=2 \star M *(N+M)\).

If LW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

IW ORK (w orkspace/output)
If \(\mathrm{OB}=\mathrm{N}\) 'or E ', \(\mathbb{I W}\) ORK is notreferenced.

LIW ORK (input)
The dim ension of the anay \(\mathbb{I V}\) ORK. If \(O B=N\) 'or E', LIW ORK >=1; if \(\mathrm{OBB}=\mathrm{V}\) 'or B',LIWORK \(>=\) \(M *(\mathbb{N}-\mathrm{M})\) 。

If LIW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the IV ORK amay, retums this value as the first entry of the \(\mathbb{I W}\) ORK aray, and no errorm essage related to \(\mathrm{L} \mathbb{I W}\) ORK is issued by X ERBLA.
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
= 1: reordering of T failed because some eigenvalues are too close to separate the problem is very ill-conditioned); T m ay have been partially reordered, and W R and W I contain the eigenvalues in the same order as in T; S and SEP (if requested) are set to zero.

\section*{FURTHER DETAILS}

STRSEN firstcollects the selected eigenvalues by com puting an orthogonaltransform ation \(Z\) to \(m\) ove them to the top left comer of T. In otherw ords, the selected eigenvalues are the eigenvalues of T11 in:
\[
\begin{gathered}
\mathrm{Z} * \mathrm{~T} * \mathrm{Z}=(\mathrm{T} 11 \mathrm{~T} 12) \mathrm{n} 1 \\
(0 \mathrm{~T} 22) \mathrm{n} 2 \\
\mathrm{n} 1 \mathrm{n} 2
\end{gathered}
\]
w here \(\mathrm{N}=\mathrm{n} 1+\mathrm{n} 2\) and Z ' m eans the transpose of Z . The first n1 colum ns of \(Z\) span the specified invariant subspace of \(T\).

If \(T\) has been obtained from the realSchur factorization of a \(m\) atrix \(A=Q * T * Q '\), then the reordered realSchur factorization of \(A\) is given by \(A=(Q * Z)^{*}\left(Z{ }^{*} T * Z\right)^{*}(Q * Z)\) ', and the first n 1 colum ns of Q *Z span the comesponding invariant subspace ofA .

The reciprocalcondition num ber of the average of the eigenvalues of T11 m ay be retumed in S.S lies betw een 0 (very badly conditioned) and 1 (very w ellconditioned). It is com puted as follow s. Firstw e com pute R so that
\[
\begin{gathered}
P=\left(\begin{array}{l}
\text { I R }) ~ n 1 ~ \\
(0 \quad 0) n 2 \\
n 1 n 2
\end{array}\right.
\end{gathered}
\]
is the pro jector on the invariant subspace associated w ith T11. R is the solution of the Sylvesterequation:
\[
\mathrm{T} 11 * \mathrm{R}-\mathrm{R} * \mathrm{~T} 22=\mathrm{T} 12 .
\]

LetF-norm M) denote the Frobenius-norm of \(M\) and 2 -norm \(M\) ) denote the tw o-norm ofM. Then \(S\) is com puted as the low er bound
\[
(1+\text { F-norm }(R) * * 2)^{* *}(-1 / 2)
\]
on the reciprocal of 2 -norm (P), the true reciprocal condition number. S cannotunderestim ate \(1 / 2\)-norm (P) by m ore than a factor of squt (N).

A \(n\) approxim ate errorbound forthe com puted average of the eigenvalues of T11 is
EPS * norm (T) /S
where EPS is the m achine precision.
The reciprocalcondition num ber of the right invariant subspace spanned by the firstn1 colum nsofZ (orofQ *Z) is retumed in SEP. SEP is defined as the separation of T11 and T 22 :
\[
\operatorname{sep}(T 11, T 22)=\text { sigm a-m in (C ) }
\]
where sigm a-m in (C) is the sm allest singularvalue of the \(\mathrm{n} 1{ }^{*} \mathrm{n} 2-b y-n 1 * n 2 \mathrm{~m}\) atrix
\[
C=\operatorname{kprod}(I(n 2), T 11)-\operatorname{kprod}(\text { transpose }(I 22), I(n 1))
\]

Im ) is an \(m\) by \(m\) identity \(m\) atrix, and kprod denotes the \(K\) ronecker product. \(W\) e estim ate sigm a-m in (C ) by the reciprocalof an estim ate of the 1 -norm of inverse (C). The true reciprocal 1-norm of inverse ( \(C\) ) cannot differ from sigm a\(m\) in (C) by \(m\) ore than a factor of sqrt (n1*n2).

W hen SEP is sm all, sm all changes in \(T\) can cause large changes in the invariantsubspace. A \(n\) approxim ate bound on the maxim um angularemor in the com puted right invariant subspace is

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtrsm -solve one of the matrix equations op (A )*X = alpha*B \(\operatorname{orX}\) *op ( A ) \(=\) alpha*B

\section*{SYNOPSIS}

```

    LD B )
    CHARACTER * 1SDEE,UPLO,TRANSA,D IAG
INTEGER M,N,LDA,LDB
DOUBLE PRECISION ALPHA
DOUBLE PRECISION A (LDA,*),B(LDB,*)
SU BROUT\mathbb{NE DTRSM _64 (SDE E,UPLO,TRANSA,D IA G ,M ,N,ALPHA,A ,LDA,B,}
LD B)

```
CHARACTER * 1 SDE , UPLO, TRANSA,D IA G
\(\mathbb{N} T E G E R * 8 \mathrm{M}, \mathrm{N}, \mathrm{LD} A, L D B\)
DOUBLE PRECISION ALPHA
D OUBLE PRECISION A (LDA ,*), B (LDB,*)

\section*{F95 INTERFACE}

SU BROUTINE TRSM (SDE, UPLO, [TRANSA ],D \(\mathbb{I A} G, \mathbb{M}], \mathbb{N}], A L P H A, A,[L D A]\), B, [LD B])

CHARACTER (LEN=1)::SDE,UPLO,TRANSA,D IAG
\(\mathbb{N} T E G E R:: M, N, L D A, L D B\)
REAL (8) :: ALPHA
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B
SU BROUTINE TRSM_64 (SDE,UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{M}], \mathbb{N}], A L P H A, A\), [LDA], B, [LDB])

CHARACTER ( \(L E N=1\) ) : : SDE E , UPLO, TRAN SA, D IA G
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{LDA}, \mathrm{LDB}\)
REAL (8) ::ALPHA
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A , B

\section*{C INTERFACE}
\#include <sunperfh>
void dtrsm (char side, charuplo, chartransa, chardiag, int m , int \(n\), double alpha, double *a, int lda, double
*b, int ldb);
void dtrsm _64 (char side, charuplo, chartransa, char diag, long \(m\), long \(n\), double alpha, double *a, long lda, double *b, long ldb);

\section*{PURPOSE}
dtrem solves one of the matrix equations op (A )*X = alpha*B, or \(\mathrm{X} *\) op ( A ) = alpha*B w here alpha is a scalar, X and \(B\) are \(m\) by \(n m\) atriges, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix and op (A ) is one of
\[
\text { op }(A)=A \text { or } o p(A)=A^{\prime}
\]

Them atrix X is overw rilten on B .

\section*{ARGUMENTS}

SID E (input)
O n entry, SID E specifiesw hetherop (A ) appears on the left or rightofX as follow s:

SDE = 'L or I' op (A ) *X = a pha*B.
\(S \mathbb{D} E=R\) 'or \(r^{\prime} X^{*} o p(A)=\) alpha*B.

U nchanged on exit.

UPLO (input)
O n entry, UPLO specifies w hether the m atrix \(A\) is an upper or low er triangularm atrix as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or \(\mathrm{L}^{\prime}\) ' \(A\) is an upper triangular m atrix .
\(\mathrm{UPLO}=\mathrm{L}\) ' or \(\mathrm{I}^{\prime}\) A is a lower triangular \(m\) atrix.

U nchanged on exit.

TRANSA (input)
On entry, TRAN SA specifies the form of op (A ) to be used in the \(m\) atrix \(m\) ultiplication as follow s:

TRANSA \(=N\) 'or \(h^{\prime}\) op (A) \(=A\).

TRANSA = T'or \(t^{\prime} \mathrm{op}(\mathrm{A})=\mathrm{A}\) '.

TRANSA = C'ort' op (A) =A'.

U nchanged on exit.
TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
O n entry, D IA G specifies w hether ornotA is unit triangular as follow s:

D \(\mathbb{A} G=U\) 'or \(U^{\prime} A\) is assum ed to be unit triangular.

D IA G \(=N\) 'or \(h^{\prime} A\) is notassum ed to be unit triangular.

U nchanged on exit.

M (input)
O \(n\) entry, \(M\) specifies the num ber of row s of \(B . M\) \(>=0\). U nchanged on exit.

N (input)
O n entry, \(N\) specifies the num ber of colum ns of \(B\). \(\mathrm{N}>=0\). U nchanged on exit.

ALPHA (input)
On entry, A LPH A specifies the scalar alpha. W hen alpha is zero then \(A\) is notreferenced and \(B\) need notbe setbefore entry. U nchanged on exit.

A (input)
D OUBLE PRECISION amay ofD \(\mathbb{I M} E N S I O N(L D A, k)\), where \(k\) is \(m\) when \(S \mathbb{D} E=\mathbb{L}\) 'or \(\mathbb{I}^{\prime}\) and is \(n\)
when \(S \mathbb{D} E=R\) 'or \(r^{\prime}\).
Before entry with UPLO = U 'or \(G\) ', the leading \(k\) by \(k\) uppertriangularpart of the aray A
\(m\) ustcontain the upper triangularm atrix and the
strictly low ertriangularpartofA is not refer-
enced.
Before entry with UPLO = L'or 1', the leading \(k\) by \(k\) low er triangularpart of the array \(A\) \(m\) ustcontain the low er triangularm atrix and the strictly uppertriangularpartofA is not referenced.
N ote thatw hen D IA G = U 'or ' U ', the diagonal elem ents of A are not referenced either, butare assum ed to be one. U nchanged on exit.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen
SDEE = 'L 'or I' then LD A >=max ( \(1, \mathrm{~m}\) ), when \(S \mathbb{D} E=R^{\prime}\) or \(\mathrm{r}^{\prime}\) then LDA \(>=\max (1, \mathrm{n})\). U nchanged on exit.

B (input/output)
D OUBLE PRECISION anay ofD \(\mathbb{I M}\) ENSION (LDB, n ).
Before entry, the leading \(m\) by \(n\) partof the array \(B\) must contain the righthand side \(m\) atrix \(B\), and on exit is overw ritten by the solution \(m\) atrix \(X\).

LD B (input)
On entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program. LD B \(>=m a x(1, m)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dtrsna -estim ate reciprocal condition num bers for specified eigenvalues and/or right eigenvectors of a real upper quasi-triangularm atrix T (orof any m atrix \(\mathrm{Q} * \mathrm{~T} * \mathrm{Q} * * \mathrm{~T}\) w ith Q orthogonal)

\section*{SYNOPSIS}
```

SU BROUTINE DTRSNA (JOB,HOW MNY,SELECT,N,T,LDT,VL,LDVL,VR,LDVR,
S,SEP,MM ,M ,W ORK,LDW ORK,W ORK1, NNFO)
CHARACTER * 1 OOB,HOW MNY
\mathbb{NTEGER N,LDT,LDVL,LDVR,MM ,M,LDW ORK,INFO}
\mathbb{NTEGERW ORK1(*)}
LOG ICAL SELECT (*)
DOUBLE PRECISION T (LDT,*), VL (LDVL,*), VR (LDVR,*), S (*),
SEP (*),W ORK (LDW ORK,*)
SUBROUT\mathbb{NEDTRSNA_64(JOB,HOW MNY,SELECT,N,T,LDT,VL,LDVL,VR,}
LDVR,S,SEP,MM ,M ,W ORK,LDW ORK,W ORK 1, INFO)
CHARACTER * 1 JOB,HOW MNY
NNTEGER*8N,LDT,LDVL,LDVR,MM ,M,LDW ORK,INFO
\mathbb{NTEGER*8 W ORK1 (*)}
LOG ICAL*8 SELECT (*)
DOUBLE PRECISION T (LDT,*), VL (LDVL,*), VR (LDVR,*), S (*),
SEP (*),W ORK (LDW ORK,*)

```

F95 INTERFACE
SU BROUTINE TRSNA (JOB,HOW MNY,SELECT,N,T, [LDT],VL, [LDVL],VR, [LDVR],S,SEP, MM,M,[WORK], [LDW ORK], [WORK1], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) :: JOB,HOW M NY
\(\mathbb{N}\) TEGER : : N, LD T, LDVL, LDVR, M M , M , LDW ORK , \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: W\) ORK1
LOGICAL,D \(\mathbb{I M} E N S I O N(:):\) SELECT
REAL (8), D \(\mathbb{M}\) ENSION (:) :: S, SEP
REAL (8), D \(\mathbb{M}\) ENSION (:,:) :: T, VL,VR,W ORK

SU BROUTINE TRSNA_64 (DOB,HOW M NY, SELECT, N, T, [LDT],VL, [LDVL],VR, [LDVR],S,SEP,MM,M,[WORK],[LDWORK],[WORK1],[ \(\mathbb{N F O}])\)

CHARACTER ( \(L E N=1\) ) : : JOB , HOW M NY
\(\mathbb{N}\) TEGER (8) :: N , LD T , LDVL, LDVR , M M , M, LDW ORK , \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{I M} \operatorname{ENSION}(:):: W\) ORK1
LOGICAL (8), D \(\mathbb{M}\) ENSION (:) :: SELECT
REAL (8), D \(\mathbb{M}\) ENSION (:) :: S, SEP
REAL (8), D \(\mathbb{M}\) ENSION (: : : : : : T, VL,VR,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void dtrsna (char job, charhow my, int *select, intn, double *t, int ldt, double *Vl, int ldvl, double *Vr, int ldvr, double *s, double *sep, intm m, int *m, int ldw ork, int *info);
void dtrsna_64 (char job, charhow m ny, long *select, long n, double *t, long ldt, double *vl, long ldvl, double
*Vr, long ldvr, double *s, double *sep, long mm, long *m, long ldw ork, long *info);

\section*{PURPOSE}
dtrsna estim ates reciprocalcondition num bers for specified eigenvalues and/or right eigenvectors of a real upper quasi-triangularm atrix T (or of any \(m\) atrix Q *T *Q **T w ith Q orthogonal).

T m ustbe in Schur canonical form (as retumed by SH SEQR), that is, block upper triangularw th 1-by-1 and 2 -by-2 diagonalblocks; each 2 -by -2 diagonal block has its diagonal elem ents equal and its off-diagonalelem ents of opposite sign.

\section*{ARGUMENTS}

JOB (input)
Specifies w hethercondition num bers are required foreigenvalues (S) oreigenvectors (SEP):
= E ': foreigenvalues only ( S );
= V ': foreigenvectors only (SEP);
= B ': forboth eigenvalues and eigenvectors ( S and SEP).

HOW MNY (input)
= A': com pute condition num bers for all eigenpairs;
\(=S^{\prime}\) : com pute condition num bers for selected eigenpairs specified by the amray SELEC T.

\section*{SELECT (input)}

If H OW M NY = S',SELECT specifies the eigenpairs for which condition numbers are required. To select condition num bers for the eigenpair corresponding to a realeigenvalue w ( \(\mathcal{\nu}\), SELECT ( ) m ustbe set to .TRUE .. To selectcondition num bers conresponding to a complex conjugate pair of eigenvaluesw ( 7 ) and w ( \(j+1\) ), either SELECT ( \(j\) ) or SELECT ( \(j+1\) ) or both, mustbe set to TRUE .. If HOW MNY = A', SELECT is notreferenced.

N (input) The order of the matrix \(\mathrm{T} \cdot \mathrm{N}>=0\).
T (input) The upper quasi-triangular \(m\) atrix \(T\), in Schur canonical form .

LD T (input)
The leading dim ension of the array T. LD T >= \(\max (1, N)\).

VL (input)
If \(\mathrm{JOB}=\mathrm{E}\) 'or B ', VL mustcontain left eigenvectors of \(T\) (orofany \(Q * T{ }^{*} Q * T\) w th \(Q\) orthogonal), comesponding to the eigenpairs specified by HOW M NY and SELECT. The eigenvectorsm ustbe stored in consecutive colum ns of \(V L\), as retumed by SHSEIN orSTREVC. If \(J O B=V\) ', VL is notreferenced.

LDVL (input)
The leading dim ension of the array VL. LD V L >=1; and if \(J 0 B=E\) 'or \(B ', L D V L>=N\).

VR (input)
If \(\mathrm{OB}=\mathrm{E}\) 'or B ', VR m ust contain right eigenvectors of \(T\) (orofany \(Q * T * Q * T\) w ith \(Q\) orthogonal), comesponding to the eigenpairs specified by HOW M NY and SELECT. The eigenvectors m ustbe stored in consecutive columns of \(V R\), as retumed by

SHSEIN orSTREVC. If \(J O B=V\) ', VR is notreferenced.

LDVR (input)
The leading dim ension of the amay VR. LDVR >=1; and if \(\mathrm{OB}=\mathrm{E}\) 'or \(\mathrm{B}^{\prime}, \mathrm{LDVR}>=\mathrm{N}\).

S (output)
If \(\mathrm{OB}=\mathrm{E}^{\prime}\) or \(\mathrm{B}^{\prime}\), the reciprocal condition num bers of the selected eigenvalues, stored in consecutive elem ents of the aray. For a com plex conjugate pair of eigenvalues two consecutive elem ents of \(S\) are set to the same value. Thus \(S(\mathcal{I}, \operatorname{SEP}(\mathcal{I})\), and the \(j\) th colum ns of VL and VR all correspond to the sam e eigenpair but not in general the jth eigenpair, unless alleigenpairs are selected). If \(\mathrm{OB}=V^{\prime}, \mathrm{S}\) is not referenced.

SEP (output)
If \(\mathrm{OB}=\mathrm{V}\) 'or B', the estim ated reciprocal condition numbers of the selected eigenvectors, stored in consecutive elem ents of the array. For a com plex eigenvector two consecutive elem ents of SEP are set to the sam e value. If the eigenvalues cannot be reordered to com pute \(\operatorname{SEP}(7)\), \(\operatorname{SEP}(7)\) is set to 0 ; this can only occurw hen the true value w ould be very sm allanyw ay. If \(\mathrm{JOB}=\mathrm{E}\) ', SEP is notreferenced.

M M (input)
The num berofelem ents in the arrays S (if \(\mathrm{OB}=\)
E' or B') and/orSEP (if \(\mathrm{OB}=\mathrm{V}\) 'or B).M M
\(>=M\).

M (output)
The num ber of elem ents of the arays \(S\) and/or SEP actually used to store the estim ated condition
num bers. If H O W M NY = 'A', M is setto \(N\).

W ORK (w orkspace)
dim ension (LD W ORK, \(N+1\) ) If \(O B=E \prime\), W ORK is not
referenced.

LDW ORK (input)
The leading dim ension of the array W ORK. LDW ORK \(>=1\); and if \(\mathrm{OB}=\mathrm{V}^{\prime}\) or \(\mathrm{B}^{\prime}, \mathrm{LDW}\) ORK \(>=\mathrm{N}\).

W ORK1 (w orkspace)
dim ension \((\mathbb{N})\) If \(O B=E\) ', W ORK 1 is not referenced.
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

The reciprocal of the condition num ber of an eigenvalue lam bda is defined as
\[
S(\operatorname{lam} \text { bda })=N^{*} u \mid /(\text { norm }(u) * \text { norm }(v))
\]
\(w\) here \(u\) and \(v\) are the right and left eigenvectors of \(T\) comesponding to lam bda; v 'denotes the conjugate-transpose of \(v\), and norm (u) denotes the Euclidean norm. These reciprocal condition num bers alw ays lie betw een zero (very badly conditioned) and one (very w ell conditioned). If \(\mathrm{n}=1\), S (lam bda) is defined to be 1.

A \(n\) approxim ate errorbound for a com puted eigenvalue \(W\) (i) is given by
EPS * norm (T) /S (i)
where EPS is the \(m\) achine precision.

The reciprocal of the condition num ber of the right eigenvectoru corresponding to lam bda is defined as follow s. Suppose
\[
\begin{gathered}
\mathrm{T}=(\operatorname{lam} \text { bda } \mathrm{c}) \\
\left(\begin{array}{ll}
\mathrm{O} & \mathrm{~T} 22
\end{array}\right)
\end{gathered}
\]

Then the reciprocalcondtion num ber is
SEP ( lam bda, T22 ) = sigm a-m in ( T22 -lam bda^I )
where sigm a-m in denotes the sm allest singular value. We approxim ate the sm allest singular value by the reciprocal of an estim ate of the one-norm of the inverse of T22 lam bda*I. If \(n=1\), SEP ( 1 ) is defined to be abs ( \((1,1)\) ).

A \(n\) approxim ate errorbound for a com puted right eigenvector VR (i) is given by
```

EPS * norm (T) /SEP (i)

```

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtrsv - solve one of the system sofequations \(A{ }^{*} x=b\), or A * \(\mathrm{x}=\mathrm{b}\)

\section*{SYNOPSIS}
```

SUBROUTINEDTRSV (UPLO,TRANSA,DIAG,N,A,LDA,Y,\mathbb{NCY)}
CHARACTER * 1UPLO,TRANSA,DIAG
INTEGERN,LDA,}\mathbb{N}C
DOUBLE PRECISION A (LDA,*),Y (*)
SU BROUT\mathbb{NE DTRSV_64(UPLO,TRANSA,D IAG,N,A,LDA,Y, INCY)}
CHARACTER * 1 UPLO,TRANSA,DIAG
\mathbb{NTEGER*8N,LDA,}\mathbb{N}CY
DOUBLE PRECISION A (LDA,*),Y (*)

```
F95 INTERFACE
    SU BROUTINE TRSV (UPLO, [TRANSA],D \(\mathbb{I A} G, \mathbb{N}], A,[L D A], Y,[\mathbb{N C Y}])\)
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
    \(\mathbb{N} T E G E R:: N, L D A, \mathbb{N} C Y\)
    REAL (8),D \(\mathbb{M}\) ENSION (:) :: Y
    REAL (8), D IM ENSION (:,:) ::A
    SU BROUTINE TRSV_64 (UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{N}], A,[L D A], Y,[\mathbb{N} C Y])\)
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
    \(\mathbb{N}\) TEGER (8) ::N,LDA, \(\mathbb{N} C Y\)
    REAL (8),D \(\mathbb{M}\) ENSION (:) ::Y
    REAL (8),D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void dtrsv (charuple, chartransa, chardiag, intn, double
*a, int lda, double *y, intincy);
void dtrsv_64 (charuplo, chartransa, char diag, long n, double *a, long lda, double *y, long incy);

\section*{PURPOSE}
dtrsv solves one of the system s ofequations \(A * x=b\), or \(A{ }^{*} x=b, w\) here \(b\) and \(x\) are \(n\) elem entvectors and \(A\) is an \(n\) by \(n\) unit, ornon-unit, upper or low er triangularm atrix. N o testforsingularity ornear-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.

\section*{ARGUMENTS}

UPLO (input)
O n entry, UPLO specifies w hether the \(m\) atrix is an upper or low er triangularm atrix as follow \(s\) :
\(\mathrm{UPLO}=\mathrm{U}\) 'or G ' \(A\) is an upper triangular m atrix.
\(\mathrm{UPLO}=\mathrm{L}\) ' or \(\mathrm{I}^{\prime}\) A is a lower triangular \(m\) atrix.

U nchanged on exit.

TRANSA (input)
On entry, TRANSA specifies the equations to be solved as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} A{ }^{*} x=b\).

TRANSA \(=T\) 'or \(t^{\prime} A{ }^{*} \mathrm{x}=\mathrm{b}\).

TRANSA \(=C^{\prime}\) ort' \(A^{*} x=b\).

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)

On entry, D IA G specifies whether ornotA is unit triangular as follow s:

D \(\mathbb{A G G}=\mathrm{U}\) 'or \(\mathrm{L}^{\prime} \mathrm{A}\) is assum ed to be unit triangular.
\(D \mathbb{A} G=N\) 'or \(h\) ' \(A\) is notassumed to be unit triangular.

U nchanged on exit.

N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.
A (input)
Before entry w ith UPLO = U 'or L ', the leading n by n upper triangularpart of the array A m ust contain the upper triangular \(m\) atrix and the strictly low ertriangularpartofA is not referenced. Before entry w ith UPLO = L' 'or I', the leading \(n\) by \(n\) low er triangularpart of the array A \(m\) ust contain the low ertriangularm atrix and the strictly uppertriangularpartofA is not referenced. N ote thatw hen D IAG \(=\mathrm{U}\) ' or L ', the diagonal elem ents of \(A\) are not referenced either, but are assum ed to be unity. U nchanged on exit.

LDA (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program . LD A \(>=\) \(m a x(1, n)\). U nchanged on exit.

Y (input/output)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) must contain the \(n\) elem ent right-hand side vectorb. 0 n exit, Y is overw ritten \(w\) th the solution vectorx.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents ofY. \(\mathbb{N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtrsyl-solve the realSylvesterm atrix equation

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE DTRSYL (TRANA,TRANB,ISGN,M,N,A,LDA,B,LDB,C,LDC,}
SCALE,INFO)
CHARACTER * 1 TRANA,TRANB
\mathbb{NTEGER ISGN,M,N,LDA,LDB,LDC,}\mathbb{N}FO
DOUBLE PRECISION SCALE
DOUBLE PRECISION A (LDA,*),B (LDB,*),C (LD C ,*)
SUBROUTINEDTRSYL_64 (TRANA,TRANB,ISGN,M,N,A,LDA,B,LDB,C,
LDC,SCALE, INFO)

```
CHARACTER * 1 TRANA, TRANB
\(\mathbb{N}\) TEGER*8 \(\operatorname{ISGN}, \mathrm{M}, \mathrm{N}, \mathrm{LD} A, L D B, L D C, \mathbb{N} F O\)
DOUBLE PRECISION SCALE
DOUBLE PRECISION A (LDA, \()\), B (LDB , \(\left.{ }^{\star}\right)\), C (LDC,\(\left.^{\star}\right)\)

\section*{F95 INTERFACE}

SU BROUTINE TRSYL (IRANA, TRANB, ISGN, \(M, N, A,[L D A], B,[L D B], C\), [LD C ],SCALE, [ \(\mathbb{N F F O}\) ])

CHARACTER (LEN=1) ::TRANA,TRANB
\(\mathbb{N} T E G E R:: \mathbb{I S G N}, \mathrm{M}, \mathrm{N}, L D A, L D B, L D C, \mathbb{N} F O\)
REAL (8) :: SCALE
REAL (8), D \(\mathbb{M}\) ENSION (:,:) ::A, B , C

SU BROUTINE TRSY L_64 (TRANA,TRANB,ISGN,M,N,A, [LDA],B,[LDB],C, [LD C ],SCALE, [ \(\mathbb{N F F O}\) ])

CHARACTER (LEN=1) ::TRANA,TRANB
\(\mathbb{N}\) TEGER (8) :: ISGN,M,N,LDA,LDB,LDC, \(\mathbb{N}\) FO
REAL (8) :: SCALE
REAL (8), D IM ENSION (:,:) ::A , B , C

\section*{C INTERFACE}
\#include <sunperfh>
void dtrsyl(char trana, char tranb, int isgn, intm, int n, double *a, int lda, double *b, int ldlb, double *c, int ldc, double *scale, int *info);
void dtrsyl_64 (chartrana, chartranb, long isgn, long m, long n, double *a, long lda, double *b, long ldb, double *c, long ldc, double *scale, long *info);

\section*{PURPOSE}
dtrsylsolves the realSylvesterm atrix equation:
\(o p(A) * X+X * o p(B)=s c a l e * C\) or \(\mathrm{op}(A) \star X-X\) * \(\mathrm{op}(B)=\) scale* \({ }^{*}\),
where op (A) \(=A\) or \(A * * T\), and \(A\) and \(B\) are both upper quasitriangular. \(A\) is \(M\)-by \(M\) and \(B\) is \(N\) by \(-N\); the righthand side C and the solution X are M -by -N ; and scale is an output scale factor, set <= 1 to avoid overflow in X .

A and B m ustbe in Schur canonical form (as retumed by SH SEQR ), that is, block uppertriangularw ith 1 -by-1 and 2-by-2 diagonalblocks; each 2-by-2 diagonal block has its diagonal elem ents equal and its off-diagonalelem ents of opposite sign.

\section*{ARGUMENTS}

TRANA (input)
Specifies the option op (A):
\(=N: \operatorname{op}(A)=A \quad\) \(\circ\) otranspose)
\(=T \mathrm{~T}: \mathrm{op}(\mathrm{A})=\mathrm{A} * * \mathrm{~T}\) ( T ranspose)
= C ': op (A) \(=\mathrm{A} * * \mathrm{H}\) (C onjugate transpose \(=\mathrm{T}\) ranspose)

TRANB (input)
Specifies the option op (B):
\(=N\) : op \((B)=B \quad(N \circ\) transpose)
\(=T\) ': op (B) = B**T (Transpose)
\(=C: o p(B)=B * * H \quad\) (Conjugate transpose \(=T\) ran spose)

ISGN (input)
Specifies the sign in the equation:
\(=+1\) : solve op (A )*X \(+X\) *op \((B)=s c a l e * C\)
\(=-1\) : solve op (A)*X \(-X\) *op \((B)=\) scale \({ }^{*} C\)
\(M\) (input) The order of the m atrix \(A\), and the num berof row s in the \(m\) atrioes X and \(\mathrm{C} . \mathrm{M}>=0\).

N (input) The order of the \(m\) atrix \(B\), and the num ber of colum ns in the m atrices X and \(\mathrm{C} . \mathrm{N}>=0\).

A (input) The upper quasi-triangular matrix A, in Schur canonical form.
LD A (input)
The leading dim ension of the aray A. LD A >= max (1, M).
\(B\) (input) The upper quasi-triangular \(m\) atrix \(B\), in Schur canonical form .

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, \mathbb{N})\).

C (input/output)
On entry, the M -by-N righthand side m atrix C. On exit, \(C\) is overw rilten by the solution \(m\) atrix \(X\).

LD C (input)
The leading dim ension of the anay C. LD C >= \(\max (1, M)\)

SCALE (output)
The scale factor, scale, set < = 1 to avoid overflow in \(X\).
\(\mathbb{I N} F \mathrm{O}\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue
\(=1\) : \(A\) and \(B\) have com \(m\) on or very close eigenvalues; perturbed values w ere used to solve the equation (but the \(m\) atrices \(A\) and \(B\) are unchanged).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtrti2 -com pute the inverse of a realupper or low er triangularm atrix

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDTRTI2(UPLO,D IAG,N,A LDA , INFO)}
CHARACTER * 1 UPLO,D IAG
\mathbb{NTEGERN,LDA,}\mathbb{N}FO
DOUBLE PRECISION A (LDA,*)
SU BROUT\mathbb{NE DTRTI2_64(UPLO,D IAG,N,A,LDA , INFO)}
CHARACTER * 1 UPLO,D IAG
\mathbb{NTEGER*8N,LDA,}\mathbb{N}FO
DOUBLE PRECISION A (LDA,*)
F95 INTERFACE
SUBROUT\mathbb{NE TRTI2 (UPLO,D IAG, N ],A, [LDA ],[NFO])}
CHARACTER (LEN=1) ::UPLO,D IA G
INTEGER ::N,LDA,\mathbb{NFO}
REAL (8),D IM ENSION (:,:) ::A
SUBROUT\mathbb{NE TRTI2_64(UPLO,DIAG, NN],A,[LDA],[INFO])}
CHARACTER (LEN=1) ::UPLO,D IAG
\mathbb{NTEGER (8) ::N,LDA,\mathbb{NFO}}0=1/2
REAL (8),D IM ENSION (:,:) ::A

```
C INTERFACE
    \#include <sunperfh>
void dtrti2 (charuplo, chardiag, intn, double *a, int lda, int*info);
void dtrti2_64 (charuple, chardiag, long n, double *a, long lda, long *info);

\section*{PURPOSE}
dtrti2 com putes the inverse of a realupper or low er triangularm atrix.

This is the Level 2 B LAS version of the algorithm .

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the m atrix A is upper or low er
triangular. = U ': U pper triangular
\(=\mathrm{L}\) ': Low ertriangular

D IA G (input)
Specifies w hether ornot the m atrix A is unittri-
angular. \(=\mathrm{N}\) ': N on-unittriangular
\(=\mathrm{U}\) ': Unittriangular

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the triangularm atrix A. If \(\mathrm{U} P \mathrm{O}=\mathrm{U}\) ', the leading \(n\) by \(n\) uppertriangularpart of the anray A contains the uppertriangularm atrix, and the strictly low er triangular part of A is not referenced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading n by n low er triangular part of the amray A contains the low ertriangularm atrix, and the strictly upper triangularpart ofA is not referenced. IfD IA G = U',the diagonal elem ents of A are also not referenced and are assum ed to be 1 .

On exit, the (triangular) inverse of the original \(m\) atrix, in the sam e storage form at.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

IN FO (output)
= 0: successfulexit
< 0: if \(\mathbb{N N}\) FO \(=-\mathrm{k}\), the k -th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dttri-com pute the inverse of a realupper or low er triangularm atrix A

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE DTRTRI(UPLO,D IA G,N,A,LDA, NNFO)}
CHARACTER * 1 UPLO,D IAG
\mathbb{NTEGERN,LDA,}\mathbb{N}FO
DOUBLE PRECISION A (LDA,*)
SUBROUT\mathbb{NE DTRTRI_64(UPLO,D IAG,N,A,LDA, INFO )}
CHARACTER * 1 UPLO,D IAG
INTEGER*8N,LDA,INFO
DOUBLE PRECISION A (LDA,*)

```
F95 INTERFACE
    SU BROUTINE TRTRI(UPLO, D \(\mathbb{A} G, N, A,[L D A],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO, D IA G
    \(\mathbb{N} T E G E R:: N, L D A, \mathbb{N F O}\)
    REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::A
    SU BROUTINE TRTRI_64 (UPLO, D \(\mathbb{A} G, N, A,[L D A],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO,D IA G
    \(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{LD} A, \mathbb{N} F O\)
    REAL (8),D \(\mathbb{M}\) ENSION (: : : : : A
C INTERFACE
    \#include <sunperfh>
void dtrtri(charuple, chardiag, intn, double *a, int lda, int*info);
void dtrtri_64 (charuplo, chardiag, long n, double *a, long lda, long *info);

\section*{PURPOSE}
dtrtricom putes the inverse of a realupper or low er triangularm atrix A.

This is the Level3 B LA S version of the algorithm .

\section*{ARGUMENTS}

UPLO (input)
= U : : A is uppertriangular;
= L': A is low er triangular.

D IA G (input)
\(=\mathrm{N}\) : A is non-unit triangular;
\(=U\) : A is unittriangular.

N (input) The order of the matrix A. \(\mathrm{N}>=0\).

A (input/output)
O n entry, the triangularm atrix A. If \(\mathrm{PLO}=\mathrm{U}\) ', the leading N -by N uppertriangularpart of the array A contains the upper triangularm atrix, and the strictly low er triangular partofA is not referenced. If UPLO = L', the leading N -by-N low er triangular part of the array A contains the low er triangularm atrix, and the strictly upper triangularpartofA is not referenced. IfD IA G = U', the diagonal elem ents of A are also not referenced and are assum ed to be 1 . O n exit, the (triangular) inverse of the original \(m\) atrix, in the sam e storage form at.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an ille-
galvalue
\(>0:\) if \(\mathbb{N} F O=i, A(i, i)\) is exactly zero. The triangular \(m\) atrix is singular and its inverse can notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dtrters - solve a triangular system of the form \(A * X=B\) orA \({ }^{* *}\) T * \(\mathrm{X}=\mathrm{B}\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDTRTRS (UPLO,TRANSA,DIAG,N,NRHS,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO,TRANSA,DIAG
INTEGERN,NRHS,LDA,LDB,INFO
DOUBLE PRECISION A (LDA,*),B (LDB,*)
SU BROUT\mathbb{NEDTRTRS_64 (UPLO,TRANSA,D IAG,N,NRHS,A,LDA,B,LDB,}
\mathbb{NFO)}
CHARACTER * 1UPLO,TRANSA,DIAG
\mathbb{NTEGER*8N,NRHS,LDA,LDB,INFO}
DOUBLE PRECISION A (LDA,*),B (LDB,*)

```

\section*{F95 INTERFACE}

SU BROUTINE TRTRS (UPLO, [TRANSA ], D IA G ,N,NRHS,A, [LDA ], B, [LD B ], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
\(\mathbb{N}\) TEGER ::N,NRHS,LDA,LDB, \(\mathbb{N} F O\)
REAL (8), D IM ENSION (:,:) ::A,B

SU BROUTINE TRTRS_64 (UPLO, [TRANSA ], D \(\mathbb{A} G, N, N R H S, A,[L D A], B,[L D B]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
\(\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D B, \mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (: ::) ::A, B

\section*{C INTERFACE}
\#include <sunperfh>
void dtrtrs (charuplo, chartransa, chardiag, int n, int nihs, double *a, int lda, double *b, int ldb, int *info);
void dtrtrs_64 (charuplo, chartransa, char diag, long n, long nihs, double *a, long lda, double *b, long ldb, long *info);

\section*{PURPOSE}
dtrtres solves a triangular system of the form
where \(A\) is a triangularm atrix of order \(N\), and \(B\) is an \(N\) -by-NRHS matrix. A check ism ade to verify thatA is nonsingular.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}: \mathrm{A}\) is uppertriangular;
= L': A is low ertriangular.
TRANSA (input)
Specifies the form of the system of equations:
\(=N\) : A * X = B N o transpose)
\(=T: A * * T * X=B \quad(T\) ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose \(=\mathrm{T}\) ranspose)

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
\(=\mathrm{N}: \mathrm{A}\) is non-unit triangular;
\(=\mathrm{U}: \mathrm{A}\) is unit triangular.
N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of them atrix B. NRHS \(>=0\).

A (input) The triangularm atrix A . If \(\mathrm{PLO}=\mathrm{U}\) ', the leading N -by -N upper triangularpart of the array A contains the upper triangular \(m\) atrix, and the
strictly low ertriangularpantofA is notreferenced. IfUPLO \(=\mathrm{L}\) ', the leading N -by N lower triangular part of the array A contains the low er triangularm atrix, and the strictly upper triangular part of \(A\) is notreferenced. IfD \(\mathbb{A} G=U '\), the diagonal elem ents ofA are also not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).
B (input/output)
On entry, the right hand side \(m\) atrix \(B\). On exit, if \(\mathbb{N} F O=0\), the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray \(\mathrm{B} . \mathrm{LDB}>=\) \(\max (1, N)\).

IN FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N ~ F O ~}=\) i, the \(i\)-th diagonalelem ent of \(A\) is zero, indicating that the \(m\) atrix is singular and the solutions X have notbeen com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dtzrqf - routine is deprecated and has been replaced by routine STZRZF

\section*{SYNOPSIS}
```

SUBROUTINEDTZRQF(M,N,A,LDA,TAU,\mathbb{NFO)}
INTEGERM,N,LDA,\mathbb{NFO}
DOUBLE PRECISION A (LDA,*),TAU (*)
SUBROUT\mathbb{NEDTZRQF_64M,N,A,LDA,TAU, INFO)}
INTEGER*8M,N,LDA, INFO
DOUBLE PRECISION A (LDA,*),TAU (*)
F95 INTERFACE
SU BROUT\mathbb{NE TZRQFM,N,A,[LDA],TAU, [\mathbb{NFO])}}\mathbf{M}\mathrm{ (TM}
\mathbb{NTEGER ::M,N,LDA, NNFO}
REAL (8),D IM ENSION (:) ::TAU
REAL (8),D IM ENSION (:,:) ::A
SUBROUT\mathbb{NE TZRQF_64M,N,A,[LDA ],TAU,[NFO ])}

```

```

    REAL (8),D IM ENSION (:) ::TAU
    REAL (8),D IM ENSION (:,:) ::A
    C INTERFACE
\#include <sunperfh>

```
void dtzrqf(intm , intn, double *a, intlda, double *tau, int*info);
void dtzrqf_64 (long m, long n, double *a, long lda, double
*tau, long *info);

\section*{PURPOSE}
dtzrqf routine is deprecated and has been replaced by routine STZRZF .

STZRQF reduces the M -by-N ( \(\mathrm{M}<=\mathrm{N}\) ) real upper trapezoidal \(m\) atrix \(A\) to upper triangular form by \(m\) eans of orthogonal transform ations.

The upper trapezoidalm atrix \(A\) is factored as
\(A=\left(\begin{array}{ll}R & 0\end{array}\right)^{*} Z\),
where Z is an N -by -N orthogonalm atrix and R is an M boy M upper triangularm atrix.

\section*{ARGUMENTS}

M (input) The num ber of row s of the matrix A. M >=0.

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=\mathrm{M}\).

A (input/output)
O \(n\) entry, the leading M łoy -N upper trapezoidal part of the array A m ustcontain the m atrix to be factorized. On exit, the leading \(M\) boy -M upper triangularpart of A contains the upper triangular \(m\) atrix \(R\), and elem ents \(M+1\) to \(N\) of the first \(M\) row s of \(A\), w ith the array \(T A U\), represent the orthogonalm atrix \(Z\) as a product of \(M\) elem entary reflectors.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, M)\).

TAU (output)
The scalar factors of the elem entary reflectors.

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an ille-

\section*{FURTHER DETAILS}

The factorization is obtained by H ouseholdersm ethod. The \(k\) th transform ation \(m\) atrix, \(Z(k)\), w hich is used to introduce zeros into the ( \(m-k+1\) )th row ofA, is given in the form
\[
\begin{gathered}
\mathrm{Z}(\mathrm{k})=\left(\begin{array}{ll}
\mathrm{I} & 0
\end{array}\right), \\
\left(\begin{array}{l}
\mathrm{O}(\mathrm{k})
\end{array}\right)
\end{gathered}
\]
where
\[
\begin{gathered}
\left.T(k)=I-\tan { }^{*} u(k) * u(k)\right), \quad u(k)=\left(\begin{array}{ll}
1
\end{array}\right), \\
\left(\begin{array}{l}
0
\end{array}\right) \\
(z(k))
\end{gathered}
\]
tau is a scalar and \(z(k)\) is an ( \(n-m\) ) elem ent vector. tau and \(\mathrm{z}(\mathrm{k})\) are chosen to annihilate the elem ents of the kth row of .

The scalartau is retumed in the \(k\) th elem entofTA \(U\) and the vectoru ( \(k\) ) in the \(k\) th row of \(A\), such that the elem ents of \(z(k)\) are in \(a(k, m+1), \ldots, a(k, n)\). The elem ents ofR are retumed in the upper triangularpartofA.
\(Z\) is given by
\[
Z=Z(1) * Z(2) * \ldots * Z(m) .
\]

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
dtzrzf-reduce the M -by -N ( \(\mathrm{M}<=\mathrm{N}\) ) real upper trapezoidal \(m\) atrix \(A\) to upper triangular form by \(m\) eans of orthogonal transform ations

\section*{SYNOPSIS}
\[
\text { SU BROUTINEDTZRZF } M, N, A, L D A, T A U, W O R K, L W O R K, \mathbb{N} F O)
\]
\(\mathbb{N}\) TEGERM,N,LDA,LWORK, \(\mathbb{N} F O\)
DOUBLE PRECISION A (LDA, \(\left.{ }^{\star}\right)\), TAU ( \({ }^{*}\) ), WORK ( \({ }^{*}\) )
SU BROUTINEDTZRZF_64M,N,A,LDA,TAU,WORK,LW ORK, \(\mathbb{N} F=\) )
\(\mathbb{N} T E G E R * 8 \mathrm{M}, \mathrm{N}, \mathrm{LD} A, L W\) ORK, \(\mathbb{N} F \mathrm{O}\)
DOUBLE PRECISION A (LDA, \(\left.{ }^{\star}\right)\),TAU (*),WORK (*)

\section*{F95 INTERFACE}

SU BROUTINE TZRZF (M ], \(\mathbb{N}], A,[L D A], T A U,[W O R K],[L W\) ORK ], [ \(\mathbb{N F O}])\)
\(\mathbb{N} T E G E R:: M, N, L D A, L W\) ORK, \(\mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL (8),D IM ENSION (:,:) ::A

SU BROUTINE TZRZF_64 (M ], \(\mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L W\) ORK ], \([\mathbb{N} F O])\)
\(\mathbb{N} T E G E R(8):: M, N, L D A, L W O R K, \mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL (8),D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include < sunperfh>
void dtzrzf(intm, intn, double *a, intlda, double *tau, int*info);
void dtzrzf_64 (long m, long n, double *a, long lda, double *tau, long *info);

\section*{PURPOSE}
dtzrzfreduces the M -by -N ( \(\mathrm{M}<=\mathrm{N}\) ) real upper trapezoidal \(m\) atrix \(A\) to upper triangular form by \(m\) eans of orthogonal transform ations.

The upper trapezoidalm atrix \(A\) is factored as
\[
A=\left(\begin{array}{ll}
R & 0
\end{array}\right) * Z,
\]
\(w\) here \(Z\) is an \(N\) boy \(-N\) orthogonalm atrix and \(R\) is an \(M\) boy \(-M\) upper triangularm atrix.

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of collm ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input/output)
On entry, the leading M -by -N upper trapezoidal part of the amay A m ust contain them atrix to be factorized. On exit, the leading M -by -M upper triangularpart ofA contains the upper triangular \(m\) atrix \(R\), and elem ents \(M+1\) to \(N\) of the first \(M\) row s of \(A\), w th the array TAU, represent the orthogonalm atrix Z as a product of M elem entary reflectors.

LD A (input)
The leading dim ension of the array A. LDA >= max (1, M).

TAU (output)
The scalar factors of the elem entary reflectors.
W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

The dimension of the array \(W\) ORK. LW ORK >= \(m\) ax \((1, M)\). Foroptim um perform anœ \(L W O R K>=M * N B\), w here NB is the optim alblocksize.

If LW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{N F O}=-\) i, the \(i\)-th argum ent had an illegal value

\section*{FURTHER DETAILS}

B ased on contributions by
A. Petitet, C om puterScience D ept., U niv . of Tenn., K noxville, U SA

The factorization is obtained by H ouseholdersm ethod. The \(k\) th transform ation \(m\) atrix, \(Z(k)\), which is used to introduce zeros into the ( \(m-k+1\) )th row ofA, is given in the form
\[
\begin{gathered}
\mathrm{Z}(\mathrm{k})=\left(\begin{array}{ll}
\mathrm{I} & 0
\end{array}\right), \\
\binom{\mathrm{O}}{\mathrm{~T}(\mathrm{k})}
\end{gathered}
\]
where
\[
\begin{gathered}
\left.T(k)=I-\tan { }^{*} u(k) * u(k)\right), \quad u(k)=\left(\begin{array}{ll}
1
\end{array}\right), \\
\left(\begin{array}{l}
0
\end{array}\right) \\
(z(k))
\end{gathered}
\]
tau is a scalar and \(z(k)\) is an ( \(n-m\) ) elem ent vector. tau and \(\mathrm{z}(\mathrm{k})\) are chosen to annihilate the elem ents of the kth row of .

The scalartau is retumed in the kth elem entofTAU and the vectoru ( \(k\) ) in the kth row of A, such that the elem ents of \(z(k)\) are in \(a(k, m+1), \ldots, a(k, n)\). The elem ents ofR are retumed in the upper triangularpartofA.
\(Z\) is given by
\[
Z=Z(1) * Z(2) * \ldots * Z(m) .
\]

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dvbrm m -variable block sparse row form atm atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEDVBRMM(TRANSA,MB,N,KB,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,BINDX,RPNTR,CPNTR,BPNTRB,BPNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,MB,N,KB,DESCRA (5),LDB,LDC,LW ORK}
INTEGER INDX(*),B\mathbb{NDX(*),RPNTR M B+1),CPNTR(KB+1),}
* BPNTRB MB),BPNTRE MB)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL (*),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NEDVBRMM_64(TRANSA,M B,N,KB,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,BINDX,RPNTR,CPNTR,BPNTRB,BPNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER*8 TRANSA,MB,N,KB,DESCRA (5),LDB,LDC,LW ORK}

```

```

* BPNTRB MB),BPNTREMB)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION VAL (*),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NEVBRMM (TRANSA,MB, N ],KB,ALPHA,DESCRA,}

* VAL, NNDX,BINDX,RPNTR,CPNTR,BPNTRB,BPNTRE,
* B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])
INTEGER TRANSA,MB,KB
\mathbb{NTEGER,D IM ENSION (:) :: DESCRA, INDX,B INDX}
NNTEGER,D IM ENSION (:) :: RPNTR,CPNTR,BPNTRB,BPNTRE
REAL*8 ALPHA,BETA
REAL*8,D IM ENSION (:) ::VAL

```

SUBROUTINEVBRMM_64 (TRANSA, MB, \(\mathbb{N}], K B, A L P H A, D E S C R A\), * VAL, \(\mathbb{N} D X, B \mathbb{N} D X, R P N T R, C P N T R, B P N T R B, B P N T R E\), * B, [LDB],BETA, C, [LDC], [WORK], [LW ORK])
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,KB
\(\mathbb{N} T E G E R * 8, D \mathbb{M}\) ENSION (:) :: DESCRA, \(\mathbb{N} D X, B \mathbb{N} D X\)
\(\mathbb{N} T E G E R * 8, D \mathbb{I} \operatorname{ENSION}(:)::\) RPNTR,CPNTR,BPNTRB,BPNTRE
REAL*8 ALPHA,BETA
REAL*8,D \(\mathbb{M}\) ENSION (:) ::VAL
REAL*8,D \(\mathbb{I}\) ENSION (: : : : : B, C

\section*{DESCRIPTION}
C <-alpha op (A) B + beta C
where A LPHA and BETA are scalar, \(C\) and \(B\) are \(m\) atrices, A is a m atrix represented in variableblock sparse row form at and op (A ) is one of
\(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{con} \dot{g}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRAN SA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix.
2 is equivalent to 1 if the \(m\) atrix is real.

M B \(\quad\) um berofblock row sin matrix A
N \(\quad\) Num berof colum ns in \(m\) atrix \(C\)

KB \(\quad\) um berofblock colum ns in m atrix A

A LPH A Scalarparam eter
DESCRA () D escriptor argum ent. Five elem ent integer array
DESCRA (1) m atrix structure
0 : general
1 : symmetric ( \(A=A\) )
2: Herm Aian ( \(\mathrm{A}=\mathrm{CONJG}\) (A))
3 :Triangular
4 : Skew (Anti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\operatorname{CONJG}(\mathrm{A})\) )

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 :non-unit
1 : unit
DESCRA (4) A ray base \(N\) OT \(\mathbb{M}\) PLEM ENTED)
0 :C C++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL ( scalar array of length NN Z consisting of the block entries of A w here each block entry is a dense rectangularm atrix stored colum \(n\) by colum \(n\).
NN Z is the total num berof pointentries in all nonzero block entries of am atrix A.
\(\mathbb{N} D \mathrm{X}\) () integer anray of length BNN Z +1 where BNNZ is the num berof block entries of a m atrix A such that the I-th elem entof \(\mathbb{N}\) D X [] points to the location in VAL of the \((1,1)\) elem ent of the I-th block entry.

B IND X () integer array of length BNN Z consisting of the block colum \(n\) indiges of the block entries of \(A\) where BNNZ is the num berblock entries of a m atrix A.

RPN TR 0) integer amay of length M B+1 such thatRPN TR (I) RPN TR (1) +1 is the row index of the firstpoint row in the \(I\)-th block
row .
RPN TR \(M B+1\) ) is set to \(M+\operatorname{RPN} \operatorname{TR}(1)\) where \(M\) is the num ber of row \(s\) in \(m\) atrix \(A\).
Thus, the num berof point row sin the I-th block row is RPNTR (I+1)RPNTR (I).

CPN TR 0 integer array of length \(K B+1\) such that CPN TR (J)-CPN TR (1)+1 is the colum \(n\) index of the firstpoint colum \(n\) in the \(J\) th block colum n. CPN TR ( \(K B+1\) ) is set to \(K+C P N T R(1)\) where \(K\) is the num ber of \(c o l u m n s\) in \(m\) atrix \(A\). Thus, the num ber of point \(\infty 0\) lum ns in the \(J\) th block colum n is CPNTR ( \(\mathrm{J}+1\) )-CPNTR ( \(J\) ).

BPNTRB () integer array of length \(M B\) such thatBPNTRB (I) BPNTRB (1) +1 points to location in B IND X of the first.block entry of the I-th block row of A.

BPNTRE 0 integer anay of length \(M B\) such that BPN TRE (I) BPNTRB (1) points to location in B \(\mathbb{N}\) D X of the lastblock entry of the I-th block row of A.

B 0 rectangular array w ith firstdin ension LD B .

LD B leading dim ension ofB

BETA Scalarparam eter
C 0 rectangular anray w ith firstdim ension LD C .

LD C leading dim ension ofC
W ORK () scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is not referenced in the cumentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at: http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1.For a generalm atrix (DESCRA (1)=0), array CPN TR can be different from RPNTR. Forallotherm atrix types, RPNTR \(m\) ustequalC CN TR and a single array can be passed forboth argum ents.
2. It is know \(n\) that there exists another representation of the variable block sparse row form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of six anay instead of the seven used in the current im plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each block row in the array \(B \mathbb{N} D X\) is used instead of two arrays BPNTRB and BPNTRE.To use the routine w th this kind of variable block sparse row form at the follow ing calling sequence should be used

SUBROUTINE SVBRMM (TRANSA,MB,N,KB,ALPHA,DESCRA,
* \(\quad V A L, \mathbb{N} D X, B \mathbb{N} D X, R P N T R, C P N T R, \mathbb{A}, \mathbb{A}(2)\),
* B,LDB,BETA, C,LDC,W ORK,LW ORK )

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
dvbrsm -variable block sparse row form at triangular solve

\section*{SYNOPSIS}
```

SUBROUTINE DVBRSM(TRANSA,M B,N,UNITD,DV,ALPHA,DESCRA,

* VAL,\mathbb{NDX,BINDX,RPNTR,CPNTR,BPNTRB,BPNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,MB,N,UNITD,DESCRA (5),LDB,LDC,LW ORK}
\mathbb{NTEGER }\mathbb{NDX(*),B\mathbb{NDX (*),RPNTR M B+1),CPNTR M B+1),}}\mathbf{(}),
* BPNTRB M B),BPNTRE M B)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION DV (*),VAL (*),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NEDVBRSM_64(TRANSA,M B,N,UNITD,DV,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,BINDX,RPNTR,CPNTR,BPNTRB,BPNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER*8 TRANSA,M B,N,UNTTD,DESCRA (5),LDB,LDC,LW ORK
\mathbb{NTEGER*8 INDX (*),B\mathbb{NDX (*),RPNTR M B+1),CPNTR M B+1),}}\mathbf{~}\mathrm{ ,},
* BPNTRB MB),BPNTRE MB)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION DV (*),VAL (*),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NEVBRSM (TRANSA,M B, N ],UNITD,DV,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,B}\mathbb{N}DX,RPNTR,CPNTR,BPNTRB,BPNTRE,
* B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])
INTEGER TRANSA,MB,UNITD

```

```

\mathbb{NTEGER,D IM ENSION (:) :: RPNTR,CPNTR,BPNTRB,BPNTRE}
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D IM ENSION(:) ::VAL,DV
DOUBLE PRECISION,D IM ENSION (:,:) :: B,C

```

SUBROUTINEVBRSM_64 (TRANSA, MB, \(\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A\),
* \(\quad \mathrm{B},[\) [DB], BETA \(, \mathrm{C},[\) [DC], \([\mathbb{W} O R K],[L W O R K])\)
\(\mathbb{N} T E G E R * 8\) TRANSA, M B , UN ITD
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O}(:):: \operatorname{DESCRA}, \mathbb{N} D X, B \mathbb{N} D X\)
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: R P N T R, C P N T R, B P N T R B, B P N T R E\)
DOUBLE PRECISION ALPHA,BETA
DOUBLE PRECISION,D \(\mathbb{M} E N S I O N(:):: V A L, D V\)
DOUBLE PRECISION ,D \(\mathbb{I M}\) ENSION (:, :) :: B , C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA C C <-ALPHA D op (A) B + BETA C } \\
& C<-A L P H A \text { op (A) D B + BETA C } \\
& \text { where A LPH A and BETA are scalar, C and B arem by n densem atrices, } \\
& D \text { is a block diagonalm atrix, A is a unit, ornon-unit, upper or } \\
& \text { low ertriangularm atrix represented in variable block sparse row } \\
& \text { form atand op (A ) is one of }
\end{aligned}
\]
op (A) \()=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}\left(c o n g\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicates \(m\) atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(\sin m\) atrix \(A\)

N \(\quad\) Num berof colum ns in matrix C

UN ITD Type of scaling:
1 : Identity \(m\) atrix (argum entD V [] is ignored)
2 : Scale on left (row block scaling)
3 : Scale on right (colum \(n\) block scaling)

DV () A rray containing the block entries of the block diagonalm atrix D. The size of the Jth block is RPN TR ( \(\mathrm{J}+1\) )-RPN TR (J) and each block containsm atrix entries stored colum n-m ajor. The total length of aray DV is given by the form ula:
sum over J from 1 to M B:

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem ent integer array
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) )
3 :Triangular
4 : Skew (Anti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CON}\) J ( A ) )
N ote: For the routine, DESCRA (1)=3 is only supported.
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) main diagonal type
0 : non-identity blocks on the \(m\) ain diagonal
1 : identily diagonalblock
2 : diagonalblocks are densem atrices
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED )
0 :C C++ com patible
1 :Fortran com patible
D ESCRA (5) repeated indices? NOT IM PLEM EN TED )
0 : unknown
1 :no repeated indices
VAL 0 scalar array of length NN Z consisting of the block entries ofA where each block entry is a dense rectangularm atrix stored colum \(n\) by colum \(n\).
NN Z is the total num berof pointentries in allnonzero block entries of a m atrix A.
\(\mathbb{I N}\) D X 0 integer array of length BNN Z +1 where BNN \(Z\) is the num ber block entries of a \(m\) atrix A such that the I-th elem ent of \(\mathbb{N} D \mathrm{X}[]\) points to the location in VAL of the \((1,1)\) elem ent of the I-th block entry.
\(B \mathbb{N} D\) X () integer array of length BNNZ consisting of the block colum \(n\) indioes of the block entries of A where BNN Z is the num berblock entries of a m atrix A. B lock colum n indices M U ST be sorted in increasing order foreach block row .

RPN TR 0) integer aray of length M B +1 such thatRPN TR (I) RPN TR (1) +1 is the row index of the firstpoint row in the I-th block
row.
RPN TR \(M B+1\) ) is set to \(M+R P N T R(1)\) where \(M\) is the num ber of row \(s\) in square triangularm atrix \(A\).

Thus, the num berof point row s in the I-th block row is RPNTR (I+1)RPNTR (I).

NOTE: For the cumentversion CPN TR m ustequalRPN TR and a single array can be passed forboth argum ents

CPNTR 0 integeramay of length \(M B+1\) such thatCPN TR (J)-CPN TR (1) +1 is the colum \(n\) index of the firstpoint colum \(n\) in the \(J\) th block colum n. CPN TR M B+1) is set to M +CPN TR (1). Thus, the num ber of pointcolum ns in the J-th block colum n is CPNTR ( \(\mathrm{J}+1\) )-CPNTR (J).

NO TE: For the current version CPN TR m ustequal RPN TR and a single aray can be passed forboth argum ents
BPN TRB 0 integer aray of length \(M B\) such thatBPN TRB (I)-BPNTRB (1)+1 points to location in B IND X of the firstblock entry of the I-th block row of A.

BPN TRE () integer array of length \(M B\) such thatBPN TRE (I) BPN TRB (1) points to location in B \(\mathbb{N} D \mathrm{X}\) of the last.block entry of the I-th block row of A.

B 0 rectangular aray w th first dim ension LD B.

LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC
W ORK 0 scratch array of length LW ORK. On exit, ifLW ORK = -1,W ORK (1) retums the optim um size of LW ORK.

LW ORK length of ORK array. LW ORK should be at least \(\mathrm{M}=\mathrm{RPNTR} \mathrm{M} \mathrm{B}+1)\) RPNTR (1).

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=\mathrm{M} * \mathrm{~N}\) _CPUS where N _CPUS is the \(m\) axim um num berof processors available to the program .

IfLW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK amray, and no enrorm essage related to LW ORK is issued

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U sers G uide available at:
http://m ath nist.gov/n csd/Staff/k Rem ington/Espblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such testsm ust.be perform ed before calling this routine.
2. If \(D E S C R A\) (3)=0, the low erorupper triangular part of each diagonalblock is used by the routine depending on DESCRA (2).
3. If \(D E S C R A(3)=1\), the unit diagonalblocksm ightorm ight not.be referenced in the VBR representation of a sparse \(m\) atrix. They are notused anyw ay .
4. If \(D E S C R A\) (3)=2, diagonalblocks are considered as dense \(m\) atrices and the LU factorization \(w\) ith partialpivoting is used by the routine. WORK (1)=0 on retum if the factorization for alldiagonalblocks has been com pleted successfully, otherw ise WORK (1) = -iw here is the block num ber forw hich the LU factorization could notbe com puted.
5.The routine can be applied for solving triangular system \(s\) w hen the upper or low er triangle of the general sparse m atrix A is used. H ow erverDESCRA (1) m ustbe equalto 3 in this case.
6. It is know \(n\) that there exists another representation of the variable block sparse row form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of six array instead of the seven used in the current im plem entation. Them ain difference is that only one array, IA , containing the pointers to the beginning ofeach block row in the array \(B \mathbb{N} D X\) is used instead of two arrays BPN TRB and BPN TRE.To use the routine w ith this kind of variable block sparse row

SUBROUTINEDVBRSM (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,
* \(\quad V A L, \mathbb{N} D X, B \mathbb{N} D X, R P N T R, C P N T R, \mathbb{A}, \mathbb{A}(2)\),
* B,LDB,BETA, C,LDC,WORK,LW ORK)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dw iener-perform W ienerdeconvolution oftw o signals

\section*{SYNOPSIS}

```

INTEGER N_PO INTS,ISW,\mathbb{ERR}
DOUBLE PRECISION ACOR (*),XCOR (*),FLTR (*),EROP (*)
SUBROUT\mathbb{NEDW ENER_64N_PO INTS,ACOR,XCOR,FLTR,EROP,ISW, ERR)}

```

```

DOUBLE PRECISION ACOR (*),XCOR (*),FLTR (*),EROP (*)

```

\section*{F95 INTERFACE}

SU BROUTINE W \(\left.\mathbb{E N E R} \mathbb{N} \_P O \mathbb{I N} T S, A C O R, X C O R, F L T R, E R O P, I S W, \mathbb{E R R}\right)\)
\(\mathbb{N} T E G E R:: N \_P O \mathbb{N} T S\), ISW , \(\mathbb{E R R}\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::ACOR,XCOR,FLTR,EROP

SU BROUTINEW \(\left.\mathbb{E N} E R \_64 \mathbb{N} \_P O \mathbb{I N} T S, A C O R, X C O R, F L T R, E R O P, I S W, \mathbb{E R R}\right)\)
\(\mathbb{N} T E G E R(8):: N \_P O \mathbb{N} T S, I S W, \mathbb{E R R}\)
REAL (8),D \(\mathbb{I}\) ENSION (:) ::ACOR,XCOR,FLTR,EROP

\section*{C INTERFACE}
\#include <sunperfh>
void dw iener(intn_points, double *acor, double *xcor, double *fter, double *erop, int *isw, int *ierr);
void dw iener_64 long n_points, double *acor, double *xcor,
double *fltr, double *erop, long *isw, long
*ienc);

\section*{PURPOSE}
dw ienerperform \(s W\) ienerdeconvolution of tw o signals.

\section*{ARGUMENTS}

N_POINTS (input)
O n entry, the num berofpoints in the inputconelations. U nchanged on exit.
ACOR (input)
O n entry, autocomelation coefficients. U nchanged on exit.

\section*{XCOR (input)}

On entry, cross-comelation coefficients.
U nchanged on exit.

FLTR (output)
O n exit, filter coefficients.

EROP (output)
O n exit, the prediction error.

ISW (input)
On entry, if ISW EQ. 0 then perform spiking
deconvolution, otherw ise perform generaldeconvolution. U nchanged on exit.

ERR (output)
O n exit, the deconvolution w as successfuliff \(\mathbb{E R R}\)
EQ.0, otherw ise there w as an error.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dzasum -Retum the sum of the absolute values of a vector x .

\section*{SYNOPSIS}
```

DOUBLE PRECISION FUNCTION D ZASUM (N,X,INCX)
DOUBLE COM PLEX X (*)
INTEGER N, INCX
DOUBLE PRECISION FUNCTION D ZASUM_64 N,X,INCX)
DOUBLE COM PLEX X (*)
INTEGER*8N,INCX
F95 INTERFACE
REAL (8) FUNCTION ASUM (N ],X,[\mathbb{NCX])}
COM PLEX (8),D IM ENSION (:) ::X
\mathbb{NTEGER ::N,\mathbb{NCX}}\mathbf{N}=\mp@code{N}
REAL (8)FUNCTION ASUM _64 (N ],X,[INCX ])
COM PLEX (8),D IM ENSION (:) ::X
INTEGER (8)::N,\mathbb{NCX}
C INTERFACE
\#include <sunperfh>
double dzasum(intn, doublecom plex *x, int incx);
double dzasum _64 (long n, doublecom plex *x, long incx);

```

\section*{PURPOSE}
dzasum Retum the sum of the absolute values of the elem ents of \(x\) where \(x\) is an \(n\)-vector. This is the sum of the absolute values of the real and com plex elem ents and not the sum of the squares of the realand com plex elem ents.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N m ustbe at leastone for the subroutine to have any visible effect. U nchanged on exit.
\(X\) (input)
( \(1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C X)\) ). On entry, the increm ented array \(X\) must contain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
dznm 2 -Retum the Euclidian norm of a vector.

\section*{SYNOPSIS}

DOUBLE PRECISION FUNCTION D ZNRM \(2(\mathbb{N}, \mathrm{X}, \mathbb{N} C X)\)

DOUBLE COM PLEX X (*)
\(\mathbb{N}\) TEGER \(N, \mathbb{I N C X}\)

DOUBLE PRECISION FUNCTION D ZNRM 2_64 \(\mathbb{N}, \mathrm{X}, \mathbb{N} C X)\)
DOUBLE COM PLEXX (*)
\(\mathbb{N} T E G E R * 8 N, \mathbb{N C X}\)

\section*{F95 INTERFACE}

REAL (8) FUNCTION NRM 2 ( \(\mathbb{N}], \mathrm{X},[\mathbb{N} C X]\) )
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::X
\(\mathbb{N} T E G E R:: N, \mathbb{N C X}\)
REAL (8) FUNCTION NRM 2_64 ( \(\mathbb{N}\) ], X, [ \(\mathbb{N} C X]\) )

COM PLEX (8),D \(\mathbb{I}\) ENSION (:) ::X
\(\mathbb{N} T E G E R(8):: N, \mathbb{N C X}\)

\section*{C INTERFACE}
\#include <sunperfh>
double dznim 2 (intn, doublecom plex *x, intincx);
double dznım 2_64 (long n, doublecom plex *x, long incx);

\section*{PURPOSE}
dznm 2 Retum the Euclidian norm of a vector \(x\) where \(x\) is an n -vector.

\section*{ARGUMENTS}

N (input)
O n entry, \(N\) specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.
X (input)
\((1+(n-1) * a b s(\mathbb{N} C X))\). On entry, the increm ented array \(X\) must contain the vectorx. U nchanged on exit.
\(\mathbb{N} C X\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X\) m ustbe positive. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
ezfftb - com putes a periodic sequence from its Fourier coefficients. EZFFTB is a sim plified butslow erversion of RFFTB.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE EZFFTB N,R,A ZERO,A,B,W SAVE)}
\mathbb{NTEGER N}
REALAZERO
REALR (*),A (*),B (*),W SAVE (*)
SUBROUTINE EZFFTB_64 N,R,AZERO,A,B,W SAVE)
INTEGER*8 N
REAL AZERO
REALR (*),A (*),B (*),W SAVE (*)
F95 INTERFACE
SUBROUT\mathbb{NE EZFFTB N,R,AZERO,A,B,W SAVE)}
\mathbb{NTEGER ::N}
REAL ::AZERO
REAL,D IM ENSION (:) ::R,A,B,W SAVE
SU BROUTINE EZFFTB_64 N,R,A ZERO,A,B,W SAVE)
\mathbb{NTEGER (8) ::N}
REAL ::AZERO
REAL,D IM ENSION (:) ::R,A,B,W SAVE
C INTERFACE
\#include <sunperfh>

```
void ezfflb (intn, float *r, float azero, float *a, float *b, float *W save);
void ezffltb_64 (long n, float *r, float azero, float *a, float*b, float *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be synthesized. The m ethod ism ostefficientw hen N is the product of smallprim es. \(N>=0\).

R (output)
On exit, the Fourier synthesis of the inputs.
AZERO (input)
On entry, the constant Fourier coefficient A 0 . U nchanged on exit.

A (input/output)
On entry, arrays that contain the rem aining Fourier coefficients. On exit, these arrays are unchanged.

B (input/output)
On entry, arrays that contain the rem aining Fourier coefficients. On exit, these amays are unchanged.

W SAVE (input)
On entry, an array \(w\) ith dim ension of at least (3 * \(\mathrm{N}+15\) ), in inialized by E ZFFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
ezffff-com putes the Fourier coefficients of a periodic sequence. EZFFTF is a sim plified butslow erversion of RFFTF.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE EZFFTF N,R,A ZERO,A,B,W SAVE)}
INTEGER N
REALAZERO
REALR (*),A (*),B (*),W SAVE (*)
SUBROUTINE EZFFTF_64 N,R,AZERO,A,B,W SAVE)
INTEGER*8 N
REAL AZERO
REALR (*),A (*),B (*),W SAVE (*)
F95 INTERFACE
SUBROUT\mathbb{NE EZFFTF N,R,A ZERO,A,B,W SAVE)}
\mathbb{NTEGER ::N}
REAL ::AZERO
REAL,D IM ENSION (:) ::R,A,B,W SAVE
SU BROUTINE EZFFTF_64 N,R,A ZERO,A,B,W SAVE)
\mathbb{NTEGER (8) ::N}
REAL ::AZERO
REAL,D IM ENSION (:) ::R,A,B,W SAVE
C INTERFACE
\#include <sunperfh>

```
void ezffff(intn, float *r, float azero, float *a, float
*b, float *w save);
void ezfflt 64 long \(n\), float *r, float azero, float *a, float*b, float *W save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. The m ethod is m ostefficientw hen N is the product of smallprim es. \(N>=0\).

R (output)
A realamay of length N containing the sequence to be transform ed. On exit, \(R\) is unchanged.

AZERO (output)
On exit, the sum from \(i=1\) to \(i=n\) ofr \((i) / n\).
A (input/output)
On entry, amays that contain the rem aining Fourier coefficients. On exit, these amays are unchanged.

B (input/output)
On entry, amays that contain the rem aining Fourier coefficients. On exit, these arrays are unchanged.

W SAVE (input)
O n entry, an array with dim ension of at least (3 * \(\mathrm{N}+15\) ), in inialized by E ZFFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
ezffti-initializes the array W SA VE, which is used in both EZFFTF andEZFFTB.

\section*{SYNOPSIS}

> SUBROUTINE EZFFTIN,W SAVE)
\(\mathbb{N}\) TEGER N
REALW SAVE (*)
SU BROUTINE EZFFTI_64 \(\mathbb{N}\),W SAVE)
\(\mathbb{N}\) TEGER*8N
REAL W SAVE (*)
F95 INTERFACE
SU BROUTINE EZFFTIN, W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL,D \(\mathbb{M}\) ENSION (:) ::W SAVE
SUBROUTINEEZFFTI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER (8) :: N
REAL,D \(\mathbb{M}\) ENSION (:) ::W SAVE

\section*{C INTERFACE}
\#include <sunperfh>
void ezfflti(intn, float *w save);
void ezffli_ 64 (long n, float *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. \(\mathrm{N}>=0\).

W SAVE (input)
O n entry, an array w ith a dim ension of at least (3
* \(\mathrm{N}+15\) ). The sam ew ork array can be used for both EZFFTF and EZFFTB as long as N rem ains unchanged. D ifferent W SAVE arrays are required fordifferent values of N . This initialization does not have to be repeated betw een calls to EZFFTF orEZFFTB as long as N and W SAVE rem ain unchanged, thus subsequent transform \(s\) can be obtained faster than the first.

\section*{Contents}
- NAME
- OVERVIEW
- MAPPING
- NOTES

\section*{NAME}
fft-FastFouriertransform subroutines

\section*{OVERVIEW}

The signal processing softw are in Sun Perform ance Library includes a set of routines based on public dom ain packages FFTPACK and VFFPACK that computes the Fast Fourier \(T\) ransform. These routines are now being replaced by a new interface (Perfib interface).

\section*{MAPPING}

Below is a mapping of routines from the FFTPA CK interface and the new Perflib interface. See individualm an pages for m ore detail.

FFTPACK interface Perflib interface
RFFTF (DFFTF) SFFTC (DFFTZ)

RFFTB (DFFTB) CFFTS (ZFFTD)
CFFTF (ZFFTF) CFFTC (ZFFTZ)
EZFFTF DEZFFTF) SFFTC (DFFTZ)
EZFFTB DEZFFTB) CFFTS (ZFFTD)
CFFTB (ZFFTB) CFFTC (ZFFTZ)
RFFT2F (DFFT2F) SFFT2C (DFFT2Z)
RFFT2B DFFT2B) CFFT2S (ZFFT2D)
CFFT2F (ZFFT2F) CFFT2C (ZFFT2Z)
CFFT2B (ZFFT2B) CFFT2C (ZFFT2Z)
RFFT3F (DFFT3F) SFFT3C (DFFT3Z)
RFFT3B (DFFT3B) CFFT3S (ZFFT3D)
CFFT3B (ZFFT3B) CFFT3C (ZFFT3Z)
CFFT3F (ZFFT3F) CFFT3C (ZFFT3Z)
\begin{tabular}{|c|c|}
\hline VCFFTF ( ZFFTF ) & CFFTCM (ZFFTZM) \\
\hline VCFFTB ( ZFFTB) & CFFTCM (ZFFTZM) \\
\hline VRFFTF ( \({ }^{\text {dFFTF) }}\) & SFFTCM (DFFTZM) \\
\hline VRFFTB (NDFFTB) & CFFTSM (ZFFTDM) \\
\hline RFFTI (DFFTI) & SFFTC (DFFTZ),CFFTS (ZFFTD) \\
\hline CFFTI (ZFFTI) & CFFTC (ZFFTZ) \\
\hline EZFFTI (DEZFFTI) & SFFTC ( \({ }^{\text {PFTZ }}\) ), CFFTS (ZFFTD) \\
\hline RFFT2I (DFFT2I) & SFFTC2 ( FFTZ2), CFFTS2 (ZFFTD 2) \\
\hline RFFT3I ( FFFT 31 ) & SFFTC 3 ( FFTZ3), CFFTS3 (ZFFTD 3) \\
\hline CFFT2I (ZFFT2I) & CFFTC2 (ZFFTZ2) \\
\hline CFFT3I (ZFFT3I) & CFFTC3 ( ZFFTZ 3 ) \\
\hline VCFFTI (VZFFTI) & CFFTCM (ZFFTZM) \\
\hline VRFFTI (NDFFTI) & SFFTCM ( FFTZM),CFFTSM (ZFFT \\
\hline
\end{tabular}

\section*{NOTES}

U nlike the FFTPA CK interface, the Perflib interface does not provide separate routines for initialization. C om putation and initialization can be selected by an argum ent in the calling sequence of each routine. Sim ilarto the FFTPA CK routines, the w eight and factor tables need to be initialized once for a particulartransform length. O nce these tables are initialized, they can be used repeatedly to com pute the forw ard and inverse tranform s fordifferent data sets until, of course, the transform length is changed. The appropriate transform routine is then called to initialize the tables for the new length.

The Perflib interface gives the user the option of com puting the FFT in-place (inputoverw ritten by transform results) or out-of-place (inputunchanged) in every routine. W hen an out-of-place transform is requested, the input and output arraysm ust not overlap in \(m\) em ory. In-place transform \(s\) require that there be perfectoverlay betw een the input and output amays. That is, the arraysm ustbegin at the sam e m em ory location. The routines assum e (and therefore do not check) that these conditions are satisfied. In som e cases, the dim ension (s) of the inputand output amays are related to each other. Below is a sum \(m\) ary of requirem ents of the array dim ensions. LD X 1 and LD X are leading dim ensions of the input arrays and LD Y 1 and LD Y are leading dim ensions of the outputarays. LD X 2 and LD Y 2 are the second dim ensions of the input and output anays, respectively. N 1 and N 2 are the first and second actualdim ensions of the problem .

Routine name in-place out-of-place

SFFTCM, DFFTZM LDX \(=2 \star\) LD Y LDX \(>=\) N 1
LD Y \(>=\mathrm{N} 1 / 2+1\) LD \(Y>=\mathrm{N} 1 / 2+1\)
```

CFFTSM,ZFFTDM LDX >=N 1/2+1 LDX >=N 1/2+1
LDY = 2*LDX LDY >=N 1
CFFTCM,ZFFTZM LDX >=N1 LDX >=N 1
LDY = LD X LDY >= N 1

```
SFFTC2,DFFTZ2 LDX = 2*LDY LDX >=N 1
    LD Y \(>=N 1 / 2+1\) LDY \(>=N 1 / 2+1\)
CFFTS2,ZFFTD 2 LDX \(>=N 1 / 2+1\) LDX \(>=N 1 / 2+1\)
    LD \(Y=2 * L D X \quad L D Y>=2 * L D X ; L D Y\) is even
CFFTC2,ZFFTZ2 LDX >=N1 LDX >=N1
    LD Y = LD X LD Y >= N 1
CFFTS3, ZFFTD 3 LDX1>=N1/2+1 LDX1 >=N \(1 / 2+1\)
    LDX2>=N2 LDX2>=N2
    LDY1 = 2*LDX1 LDY1 >= 2*LDX1; LDY1 is
even
    LD Y \(2=\operatorname{LD} X 2\) LD Y \(2>=\) N 2
CFFTC 3,2 ZFTZ3 LDX1>=N1 LDX1>=N1
    LDX2 >=N2 LDX2>=N2
    LDY1 = LDX1 LDY1>=N1
    LDY2 \(=\) LD 2 LDY2 \(2=N 2\)
```

SFFTC3,DFFTZ3 LDX1=2*LDY1 LDX1>=N1
LDX2>=N2 LDX2>=N2
LDY 1 >= N 1/2+1 LD Y 1 >= N 1/2+1
LDY2 = LDX2 LDY 2 >= N 2

```

In routines that com pute transform sbetw een com plex and real data type such as SFFTC 2 orCFFTS3 even though the transform length is N 1 , only \(\mathbb{N} 1 / 2+1\) ) com plex data points are referenced or computed. These data points \(m\) ake up the positive-frequency half of the spectrum of the D iscrete Fourier Transform . The rem aining \(\mathrm{N} 1-\mathbb{N} 1 / 2+1\) ) data points can be easily derived since they are com plex conjugates and therefore are not stored or referenced.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- NOTES

\section*{NAME}
icam ax-retum the index of the elem entw ith largest absolute value.

\section*{SYNOPSIS}
```

$\mathbb{N}$ TEGER FUNCTION ICAMAX $\mathbb{N}, \mathrm{X}, \mathbb{N} C X)$

```

COM PLEX X (*)
\(\mathbb{N} T E G E R N, \mathbb{N C X}\)
\(\mathbb{N}\) TEGER*8 FUNCTION ICAMAX_64 \(\mathbb{N}, \mathrm{X}, \mathbb{N} C X\) )
COM PLEX X \({ }^{(*)}\)
\(\mathbb{I N}\) TEGER*8 \(\mathrm{N}, \mathbb{N} C X\)

\section*{F95 INTERFACE}
\(\mathbb{N}\) TEGER FUNCTION \(\mathbb{I A M A X}(\mathbb{N}], X,[\mathbb{N} C X])\)
COMPLEX,D \(\mathbb{I M} \operatorname{ENSION(:)::X~}\)
\(\mathbb{N} T E G E R:: N, \mathbb{N C X}\)
\(\mathbb{N}\) TEGER (8) FUNCTION IAM AX_64 ( \(\mathbb{N}\) ], X , [ \(\mathbb{N} C X]\) )

COMPLEX,D \(\mathbb{I M}\) ENSION (:) ::X
\(\mathbb{I N} T E G E R(8):: N, \mathbb{N C X}\)

\section*{C INTERFACE}
\#include <sunperfh>
int icam ax (intn, com plex *x, intincx);

\section*{PURPOSE}
icam ax retum the index of the elem ent in \(x \mathrm{w}\) ith largest absolute value where \(x\) is an \(n\)-vector and absolute value is defined as the sum of the absolute value of the real part and the absolute value of the im aginary part.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.

X (input)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C X))\). On entry, the increm ented array \(X\) must contain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X\) m ustbe positive. U nchanged on exit.

\section*{NOTES}

If the vector contains all NaN s, the function retums 1. If the vector contains valid com plex num bers and one orm ore NaN s, the routine retums the index of the elem ent containing the largestabsolute value.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- NOTES

\section*{NAME}
idam ax-retum the index of the elem entw ith largest absolute value.

\section*{SYNOPSIS}
```

$\mathbb{N}$ TEGER FUNCTION $\mathbb{D} A M A X \mathbb{N}, X, \mathbb{N} C X)$

```
\(\mathbb{N}\) TEGER \(N, \mathbb{N C X}\)
DOUBLE PRECISIONX (*)
\(\mathbb{N} T E G E R * 8 F U N C T I O N \mathbb{D} A M A X \_64(\mathbb{N}, X, \mathbb{N} C X)\)
\(\mathbb{N} T E G E R * 8 N, \mathbb{N C X}\)
DOUBLE PRECISIONX (*)

\section*{F95 INTERFACE}
\(\mathbb{N}\) TEGER FUNCTION \(\mathbb{I A M A X}(\mathbb{N}], X,[\mathbb{N C X}])\)
\(\mathbb{N} T E G E R:: N, \mathbb{N} C X\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::X
\(\mathbb{N}\) TEGER (8) FUNCTION IAMAX_64 (N ], X, [ \(\mathbb{N} C X]\) )
\(\mathbb{N} T E G E R(8):: N, \mathbb{N C X}\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::X

\section*{C INTERFACE}
\#include <sunperfh>
int idam ax (intn, double *x, int incx);

\section*{PURPOSE}
idam ax retum the index of the elem ent in x w th largest absolute value \(w\) here \(x\) is an \(n\)-vector.

\section*{ARGUMENTS}

N (input)
On entry, N specifies the num ber of elem ents in the vector. N m ustbe at leastone for the subroutine to have any visible effect. U nchanged on exit.
X (input)
( \(1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C X)\) ). On entry, the increm ented array X m ust contain the vectorx. U nchanged on exit.
\(\mathbb{N} C X\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X\) m ustbe positive. U nchanged on exit.

\section*{NOTES}

If the vector contains all NaN s , the function retums 1 . If the vector contains valid floating pointnum bers and one or m ore NaN s, the routine retums the index of the lem ent containing the largest absolute value.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ilaenv - The nam e of the calling subroutine, in eitherupper case or low er case.

\section*{SYNOPSIS}
```

\mathbb{NTEGER FUNCTION LLAENV (ISPEC,NAME,OPTS,N1,N 2,N 3,N4)}
CHARACTER * 6NAMECHARACTER * 4OPTS
INTEGER ISPEC,N1,N2,N3,N4
INTEGER*8 FUNCTION LAENV_64(ISPEC,NAME,OPTS,N 1,N 2,N 3,N 4)
CHARACTER * 6NAMECHARACTER * 4OPTS
IN TEGER*8 ISPEC,N 1,N 2,N 3,N 4

```
F95 INTERFACE
    INTEGER FUNCTION HAENV (ISPEC,NAME,OPTS,N1,N2,N3,N4)
    CHARACTER (LEN=6) ::NAMECHARACTER (LEN=4) ::OPTS
    \(\mathbb{N}\) TEGER :: ISPEC,N1,N2,N3,N4
    \(\mathbb{N}\) TEGER (8) FUNCTION \(\mathbb{L} A E N V \_64(\) ISPEC,NAME,OPTS,N 1,N 2,N 3,N 4)
    CHARACTER (LEN=6) ::NAMECHARACTER (LEN=4) ::OPTS
    \(\mathbb{N}\) TEGER (8) :: ISPEC ,N1,N2,N3,N4
C INTERFACE
    \#include <sunperfh>
    int ilaenv (int ispec, char*nam e, char *opts, int n1, int
        n2, intn3, intn4);
long ilaenv_64 (long ispec, char *nam e, char *opts, long n1, long n2, long n3, long n4);

\section*{PURPOSE}
ilaenv is called from the LAPACK routines to choose problem-dependent param eters for the localenvironm ent. See ISPEC fora description of the param eters.

This version provides a setof param eters which should give good, butnotoptim al, penform ance on \(m\) any of the currently available com puters. U sers are encouraged to m odify this subroutine to set the tuning param eters for theirparticular \(m\) achine using the option and problem size inform ation in the argum ents.
This routine w illnot function correctly if it is converted to all low er case. C onverting it to allupper case is allow ed.

\section*{ARGUMENTS}

ISPEC (input)
Specifies the param eter to be retumed as the value of \(\amalg\) AENV . = 1 : the optim alblocksize; if this value is 1 , an unblocked algorithm will give the bestperform ance. \(=2\) : the \(m\) inim um block size forw hich the block routine should be used; if the usable block size is less than this value, an unblocked routine should be used. \(=3\) : the crossover point (in a block routine, for \(N\) less than this value, an unblocked routine should be used) = 4: the num ber of shifts, used in the nonsym \(m\) etric eigenvalue routines \(=5\) : the \(m\) inim um colum \(n\) dim ension for blocking to be used; rectangularblocks \(m\) usthave dim ension at least \(k\) by \(m\), where \(k\) is given by \(\mathbb{L A E N V}(2, \ldots)\) and \(m\) by \(\mathbb{L A E N V}(5, \ldots)=\) 6: the crossoverpoint for the SVD (w hen reducing an \(m\) by \(n m\) atrix to bidiagonal form, if \(m\) ax \((m, n)\) fr in \((m, n)\) exceeds this value, \(a Q R\) factorization is used first to reduce the \(m\) atrix to a triangular form .) \(=7\) : the num ber of processors = 8: the crossover point for the \(m\) ultishift \(Q R\) and \(Q Z \mathrm{~m}\) ethods fornonsym \(m\) etric eigenvalue problem \(s\). = 9:m axim um size of the subproblem sat the bottom of the com putation tree in the divide-and-conquer algorithm (used by xGELSD and xGESDD) =10: ieee N aN arithm etic can be trusted not to trap
\(=11\) : infinity arithm etic can be trusted not to trap

NAME (input)
The nam ef the calling subroutine, in either upper case or low er case.

OPTS (input)
The character options to the subroutine NAM E, concatenated into a single characterstring. For exam ple, \(\mathrm{UPLO}=\mathrm{U}\) ', TRANS \(=T\) ', and D \(\mathbb{A} G=\mathrm{N}^{\prime}\) for a triangular routine w ould be specified as OPTS = UTN '.

N 1 (input)
\(\mathbb{N}\) TEGER

N 2 (input)
\(\mathbb{N}\) TEGER
N 3 (input)
\(\mathbb{N}\) TEGER

N 4 (input)
\(\mathbb{N}\) TEGER
N 1, N 2, N 3, N 4 are problem dim ensions for the subroutine NAM E; these m ay not allbe required.
\(>=0\) : the value of the param eter specified by ISPEC
< 0: if ILAENV \(=-\mathrm{k}\), the k -th argum ent had an illegalvalue. < 0: if ILAENV \(=-k\), the \(k\)-th argum ent had an illegalvahue.

\section*{FURTHER DETAILS}

The follow ing conventions have been used w hen calling IIA EN V from the LAPACK routines:
1) OPTS is a concatenation of allof the character options to
subroutine NAM E, in the sam e order that they appear in the
argum ent list forN AM E, even if they are not used in determ ining
the value of the param eter specified by ISPEC.
2) The problem dim ensions \(1, N 2, N 3, N 4\) are specified in the order
that they appear in the argum ent list forNAME. N1 is used
first, N 2 second, and so on, and unused problem dim ensions are
passed a value of -1 .
3) The param etervalue retumed by ILAENV is checked for validity in
the calling subroutine. For exam ple, ILA EN V is used to retrieve
the optim alblocksize forSTRTR I as follow s:

NB = \(\amalg \operatorname{LAENV}(1, \operatorname{STRTRI}, \mathrm{UPLO} / / \mathrm{D} \operatorname{IA}, \mathrm{N},-1,-1,-1)\) \(\mathbb{F}(\mathrm{NB} L E 1) \mathrm{NB}=\mathrm{MAX}(1, \mathrm{~N})\)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- NOTES

\section*{NAME}
isam ax -retum the index of the elem entw ith largest absolute value.

\section*{SYNOPSIS}
\(\mathbb{N} T E G E R F U N C T I O N\) ISAMAX \(\mathbb{N}, X, \mathbb{N} C X)\)
\(\mathbb{N} T E G E R N, \mathbb{N} C X\)
REALX (*)
\(\mathbb{N} T E G E R * 8 F U N C T I O N\) ISAMAX_64(N,X, \(\mathbb{N} C X)\)
\(\mathbb{N} T E G E R * 8 N, \mathbb{N C X}\)
REALX (*)

\section*{F95 INTERFACE}
\(\mathbb{N}\) TEGER FUNCTION \(\mathbb{I A M A X}(\mathbb{N}], X,[\mathbb{N C X}])\)
\(\mathbb{N} T E G E R:: N, \mathbb{N} C X\)
REAL,D \(\mathbb{M}\) ENSION (:) ::X
\(\mathbb{N}\) TEGER (8) FUNCTION \(\mathbb{A}\) M AX_64 ( \(\mathbb{N}\) ],X , [ \(\mathbb{N} C X]\) )
\(\mathbb{N} T E G E R(8):: N, \mathbb{N C X}\)
REAL,D \(\mathbb{I}\) ENSION (:) ::X

\section*{C INTERFACE}
\#include <sunperfh>
int isam ax (intn, float * \(x\), int incx);

\section*{PURPOSE}
isam ax retum the index of the elem ent in x w ith largest absolute value \(w\) here \(x\) is an \(n\)-vector.

\section*{ARGUMENTS}
\(N\) (input)
On entry, N specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.
\(X\) (input)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C X))\). On entry, the increm ented array \(X\) ust contain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{I N C X ~ m ~ u s t b e ~ p o s i t i v e . ~ U ~ n c h a n g e d ~}\) on exit.

\section*{NOTES}

If the vector contains all NaN s , the function retums 1 . If the vectorcontains valid floating pointnum bers and one or m ore NaN s, the routine retums the index of the elem ent containing the largest absolute value.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- NOTES

\section*{NAME}
izam ax -retum the index of the elem entw ith largest absolute value.

\section*{SYNOPSIS}
```

$\mathbb{N} T E G E R F U N C T \mathbb{I} N \mathbb{Z} A M A X \mathbb{N}, X, \mathbb{N} C X)$

```

DOUBLE COM PLEX X (*)
\(\mathbb{I N}\) TEGER \(\mathrm{N}, \mathbb{I N C X}\)
\(\mathbb{N} T E G E R * 8 F U N C T I O N\) ZAM AX_64 \(\mathbb{N}, X, \mathbb{N C X})\)
DOUBLE COM PLEXX (*)
\(\mathbb{N} T E G E R * 8 N, \mathbb{N} C X\)
F95 INTERFACE
\(\mathbb{N}\) TEGER FUNCTION \(\mathbb{I A M A X}(\mathbb{N}], X,[\mathbb{N C X}])\)
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::X
\(\mathbb{I N T E G E R}:: N, \mathbb{N C X}\)
\(\mathbb{N}\) TEGER (8) FUNCTION \(\mathbb{A}\) M AX_64 ( \(\mathbb{N}\) ],X , [ \(\mathbb{N} C X]\) )

COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::X
\(\mathbb{I N} \operatorname{TEGER}\) (8) ::N, \(\mathbb{N C X}\)

\section*{C INTERFACE}
\#include <sunperfh>
int izam ax (intn, doublecom plex *x, int incx);

\section*{PURPOSE}
izam ax retum the index of the elem ent in x w ith largest absolute value where \(x\) is an \(n\)-vector and absolute value is defined as the sum of the absolute value of the real part and the absolute value of the im aginary part.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.

X (input)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C X))\). On entry, the increm ented array \(X\) must contain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X\) m ustbe positive. U nchanged on exit.

\section*{NOTES}

If the vector contains all NaN s , the function retums 1 . If the vector contains valid double com plex num bers and one or m ore NaN s , the routine retums the index of the elem ent containing the largest absolute value.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
lsame-retums TRUE.if CA is the same letter as CB regardless of case

\section*{SYNOPSIS}

> LOG ICALFUNCTION LSAME (CA, CB)

CHARACTER * 1 CA, CB
LOGICAL*8FUNCTION LSAME_64 (CA, CB)
CHARACTER * 1 CA, CB
F95 INTERFACE
LOG ICALFUNCTION LSAME (CA, CB)

CHARACTER (LEN=1): :CA,CB
LOG ICAL (8) FUNCTION LSAME_64 (CA,CB)

CHARACTER (LEN=1) ::CA,CB
C INTERFACE
\#include <sunperfh>
int lsam e (charca, chardb);
long lsam e_64 (charca, charcb);

\section*{PURPOSE}
lsam e retums .TRUE.ifCA is the same letter as CB regardless of case.

\section*{ARGUMENTS}

CA (input)
O n entry, CA is a single character to com pare w th CB.U nchanged on exit.

CB (input)
O n entry, CB is a single character to com pare w ith CA.U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
rfft2b -com pute a periodic sequence from its Fourier coefficients. The RFFT operations are unnorm alized, so a call ofRFFT2F followed by a callof RFFT2B w ill multiply the input sequence by \(\mathrm{M} * \mathrm{~N}\).

\section*{SYNOPSIS}
```

SUBROUTINE RFFT2B(PLACE,M,N,A,LDA,B,LDB,W ORK,LW ORK)
CHARACTER * 1 PLACE
INTEGER M,N,LDA,LDB,LW ORK
REALA (LDA,*),B (LDB,*),W ORK (*)
SUBROUTINE RFFT2B_64(PLACE,M ,N,A,LDA,B,LDB,W ORK,LW ORK)
CHARACTER * 1 PLACE
\mathbb{NTEGER*8M,N,LDA,LDB,LW ORK}
REALA (LDA,*),B (LDB,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE FFT2B (PLACE, \(\mathbb{M}], \mathbb{N}], A,[L D A], B,[L D B], W\) ORK,LW ORK)
CHARACTER (LEN=1) ::PLACE
\(\mathbb{N} T E G E R:: M, N, L D A, L D B, L W\) ORK
REAL,D \(\mathbb{M}\) ENSION (:) ::W ORK
REAL,D IM ENSION (: :: : ::A, B
SUBROUTINE FFT2B_64(PLACE, \(\mathbb{M}], \mathbb{N}], A,[L D A], B,[L D B], W\) ORK,LW ORK)
CHARACTER (LEN=1) ::PLACE
\(\mathbb{N}\) TEGER (8) ::M , N , LDA, LD B, LW ORK
REAL,D IM ENSION (:) ::W ORK
REAL,D IM ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void nff2b (charplace, intm, intn, float *a, int lda, float *b, int ldlb, float *w ork, int liw ork);
void rffi2b_64 (charplace, long m, long n, float *a, long lda, float *b, long ldb, float *w ork, long lw ork);

\section*{ARGUMENTS}

PLACE (input)
Character. IfPLA CE = 'I'or i' (for in-place), the input and outputdata are stored in array A. IfPLACE = 0 ' or \(b^{\prime}\) (for out-of-place), the input data is stored in anray \(B\) while the output is stored in A.
\(M\) (input) Integer specifying the number of row s to be transform ed. It is m ost efficientw hen M is a productofsm allprin es. \(\mathrm{M}>=0\); when \(\mathrm{M}=0\), the subroutine retums im mediately w ithout changing any data.

N (input) Integer specifying the num ber of colum ns to be transform ed. It is \(m\) ostm ostefficientw hen \(N\) is a productofsm allprim es. \(\mathrm{N}>=0\); when \(\mathrm{N}=0\), the subroutine retums im \(m\) ediately \(w\) ithout changing any data.

A (input/output)
Realarray ofdim ension (LDA N). On entry, the tw o-dim ensional array A (LD A , N) contains the input data to be transform ed if an in-place transform is requested. O therw ise, it is not referenced. U pon exit, results are stored in \(A(1 \mathbb{M}, 1 \mathbb{N})\).

LD A (input)
Integer specifying the leading dim ension ofA. If an out-of-place transform is desired LDA \(>=M\). Else if an in-place transform is desired LDA >= \(2 * M / 2+1\) ).

B (input/output)
Realarray ofdim ension ( \(2 *\) LD \(B, N\) ). On entry, if an out-of-place transform is requested \(B\) contains the input data. O therw ise, \(B\) is not referenced. \(B\) is unchanged upon exit.

LD B (input)
Integer. If an out-of-place transform is desired, \(2 *\) LD B is the leading dim ension of the amay B which contains the data to be transform ed and \(2 * \operatorname{LDB}>=2 *(2+1)\). O therw ise it is notreferenced.

W ORK (input/output)
O ne-dim ensional real array of length at least LW ORK. On input, W ORK m usthave been initialized by RFFT2I.

LW ORK (input)
Integer. LW ORK >= \(M+2 \star \mathrm{~N}+\mathrm{MAX} \mathrm{M}, 2 \star \mathrm{~N})+30\) )

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
rfft2f-com pute the Fourier coefficients of a periodic
sequence. The RFFT operations are unnorm alized, so a call ofRFFT2F followed by a callof RFFT2B will multiply the input sequence by M * N .

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE RFFT2F (PLACE,FULL,M ,N,A,LDA,B,LDB,W ORK,LW ORK)}
CHARACTER * 1 PLACE,FULL
INTEGERM,N,LDA,LDB,LW ORK
REALA (LDA,*),B (LDB,*),W ORK (*)
SUBROUTINE RFFT2F_64(PLACE,FULL,M,N,A,LDA,B,LDB,W ORK,LW ORK)
CHARACTER * 1 PLACE,FULL
INTEGER*8M,N,LDA,LDB,LW ORK
REALA (LDA,*),B (LDB,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE FFT2F (PLACE,FULL, \(\mathbb{M}], \mathbb{N}], A,[L D A], B,[L D B], W\) ORK, LW ORK)

CHARACTER (LEN=1) ::PLACE,FULL
\(\mathbb{N}\) TEGER ::M,N,LDA,LDB,LW ORK
REAL,D IM ENSION (:) ::W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A,B

SU BROUTINE FFT2F_64 (PLACE,FULL, M ], \(\mathbb{N}], A,[L D A], B,[L D B], W\) ORK, LW ORK)

CHARACTER (LEN=1) ::PLACE,FULL
\(\mathbb{N}\) TEGER (8) ::M , N,LDA, LD B, LW ORK

REAL,D \(\mathbb{I M} E N S I O N(:):: W O R K\)
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void rffi2f(charplace, char fill, intm, intn, float *a, int lda, float *b, int ldb, float*w ork, int lw ork);
void rfft2f_64 (charplace, char full, long m, long n, float
*a, long lda, float *b, long ldb, float *w ork, long lw ork);

\section*{ARGUMENTS}

PLACE (input)
Character. IfPLA CE = I'or i' (for in-place), the input and outputdata are stored in anray A. IfPLACE = \(0^{\prime}\) or \(b^{\prime}\) (for out-of-place), the input data is stored in aray B while the output is stored in A.

FULL (input)
Indicates \(w\) hether or not to generate the full result \(m\) atrix. \(F^{\prime}\) or 'f'w ill cause RFFT \(2 F\) to generate the filll resultm atrix. O therw ise only a partial \(m\) atrix that takes advantage of sym \(m\) etry w illbe generated.

M (input) Integer specifying the num ber of row \(s\) to be transform ed. It is m ost efficientw hen M is a productofsm allprim es. \(M>=0\); when \(M=0\), the subroutine retums im mediately w ithoutchanging any data.

N (input) Integerspecifying the num ber of colum ns to be transform ed. It ism ostm ostefficientw hen N is a productofsm allprim es. \(\mathrm{N}>=0\); when \(\mathrm{N}=0\), the subroutine retums im \(m\) ediately \(w\) thout changing any data.

A (input/output)
O n entry, a tw o-dim ensional aray A (LDA \(N\) ) that contains the data to be transform ed. U pon exit, A is unchanged if an out-of-place transform is done. If an in-place transform \(w\) ith partial result is requested, \(A(1:(M / 2+1) * 2,1 \mathbb{N})\) w ill contain the transform ed results. If an in-place transform
w th full result is requested, \(\mathrm{A}(1: 2 \star \mathrm{M}, 1 \mathrm{~N}) \mathrm{w}\) ill contain com plete transform ed results.

LD A (input)
Leading dim ension of the array containing the data to be transform ed. LDA must be even if the transform ed sequences are to be stored in A.
IfPLACE \(=\left(D^{\prime}\right.\) orb) LDA \(>=M\)
IfPLACE \(=\) (I'or li) LDA mustbe even. If
\(F U L L=\left(F^{\prime}\right.\) or \({ }^{\prime}\) ) \(), L D A>=2 * M\)
FULL is not ( \(\mathrm{F}^{\prime}\) or \(\mathrm{I}^{\prime}\) ), LDA \(>=(\mathrm{M} / 2+1\) ) 2
B (input/output)
U pon exit, a tw o-dim ensionalamay B ( \(2 *\) LD B , N ) that
contains the transform ed results if an out-of-
place transform is done. O therw ise, \(B\) is not used.
If an out-of-place transform is done and FULL is
not \(\mathrm{F}^{\prime}\) or \(\mathrm{'}^{\prime}\) ' \(\left.\mathrm{B}(1: M / 2+1) * 2,1 \mathbb{N}\right)\) w illcontain the partial transform ed results. IfFU LL \(=F\) 'or ' f ', \(\mathrm{B}(1: 2 * \mathrm{M}, 1 \mathbb{N})\) w ill contain the com plete transform ed results.

LD \(B\) (input)
\(2 * L D B\) is the leading dim ension of the array \(B\). If
an in-place transform is desired LD B is ignored.
IfPLACE is ( 0 'or \(b^{\prime}\) ) and
FULL is ( \(\mathrm{F}^{\prime}\) or \({ }^{\prime} \mathrm{F}\) ), LDB \(>=\mathrm{M}\)
FULL is not ( F ' or ' f ), LD B \(>=\mathrm{M} / 2+1\)
\(N\) ote thateven though LD \(B\) is used in the argum ent list, \(2 *\) LD B is the actual leading dim ension ofB .

W ORK (input/output)
O ne-dim ensional real anray of length at least
LW ORK. On input, W ORK m usthave been initialized by RFFT2I.

LW ORK (input)
Integer. LW ORK >= \(M+2 \star N+M A X M, 2 \star N)+30\) )

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
rfft2i-initialize the array W SAVE, which is used in both the forw ard and backw ard transform s.

\section*{SYNOPSIS}

SUBROUTINERFFT2IM,N,W ORK)
\(\mathbb{N}\) TEGER \(\mathrm{M}, \mathrm{N}\)
REALW ORK (*)
SU BROUTINERFFT2I_64 M ,N,W ORK)
\(\mathbb{N}\) TEGER*8 M , N
REALW ORK (*)

F95 INTERFACE
SU BROUTINE FFT2IM ,N,WORK)
\(\mathbb{N} T E G E R:: M, N\)
REAL,D IM ENSION (:) ::W ORK

SU BROUTINE FFT2I_64 M ,N,W ORK)
\(\mathbb{N} T E G E R(8):: M, N\)
REAL,D \(\mathbb{I M}\) ENSION (:) ::W ORK
C INTERFACE
\#include <sunperfh>
void rffli(intm, intn, float*W ork);
void rffl2i 64 (long m, long n, float *w ork);

\section*{ARGUMENTS}

M (input) N um ber of row s to be transform ed. \(\mathrm{M}>=0\).

N (input) N um ber of colum ns to be transform ed. \(\mathrm{N}>=0\).

W ORK (input/output)
On entry, an aray ofdim ension \(M+2 * N+M A X M\), \(2 * N\) ) +30 ) orgreater. RFFT2I needs to be called only once to initialize aray W O RK before calling RFFT2F and/or RFFT2B if M, N andW ORK rem ain unchanged betw een these calls. Thus, subsequent transform s or inverse transform s of sam e size can be obtained faster than the first since they do not require initialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
rffl3b - com pute a periodic sequence from its Fourier coefficients. The RFFT operations are unnorm alized, so a call ofRFFT3F follow ed by a call of RFFT 3B w ill multiply the input sequence by \(\mathrm{M} * \mathrm{~N} * \mathrm{~K}\).

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE RFFT3B (PLACE,M,N,K,A,LDA,B,LDB,W ORK,LW ORK)}
CHARACTER * 1 PLACE
INTEGERM,N,K,LDA,LDB,LW ORK
REALA (LDA,N,*),B (LDB,N,*),W ORK (*)
SU BROUT\mathbb{NE RFFT3B_64(PLACE,M ,N,K,A,LDA,B,LDB,W ORK,LW ORK)}

```
CHARACTER * 1 PLACE
\(\mathbb{N}\) TEGER*8M,N,K,LDA,LDB,LW ORK
REALA (LDA \(\left.\mathbb{N}^{*},{ }^{\star}\right), \mathrm{B}(\mathrm{LDB}, \mathbb{N}, \star), \mathrm{W} O R K(\star)\)

\section*{F95 INTERFACE}

SU BROUTINE FFT3B (PLACE, \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], B,[L D B], W\) ORK, LW ORK)

CHARACTER (LEN=1) ::PLACE
\(\mathbb{N}\) TEGER ::M , N, K, LD A, LD B, LW ORK
REAL,D IM ENSION (:) ::W ORK
REAL,D IM ENSION (:,:,:) ::A,B

SU BROUTINE FFT3B_64 (PLACE, \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], B,[L D B], W\) ORK, LW ORK)

CHARACTER (LEN=1) ::PLACE
\(\mathbb{N}\) TEGER (8) ::M ,N,K,LDA,LDB,LW ORK

REAL,D \(\mathbb{M} E N S I O N(:):: W O R K\)
REAL,D \(\mathbb{M}\) ENSION (:,:\%) ::A , B

\section*{C INTERFACE}
\#include < sunperfh>
void nffthb (charplace, intm , intn, intk, float *a, int lda, float *b, int ldb, float *w ork, int lw ork);
void rfft3b_64 (charplace, long m, long \(n\), long \(k\), float *a, long lda, float *b, long ldb, float *w ork, long lw ork);

\section*{ARGUMENTS}

PLACE (input)
Selectan in-place (I'or li) or out-of-place ( D 'or b ') transform.
\(M\) (input) Integer specifying the num ber of row s to be transform ed. It is \(m\) ost efficientw hen \(M\) is a productofsm allprim es. \(\mathrm{M} \mathrm{>=0;}\) when \(\mathrm{M}=0\), the subroutine retums im mediately w ithout changing any data.

N (input) Integer specifying the num ber of colum ns to be transform ed. It is m ost efficientw hen N is a productofsm allprim es. \(\mathrm{N}>=0\); when \(\mathrm{N}=0\), the subroutine retums im \(m\) ediately w ithout changing any data.
\(K\) (input) Integerspecifying the num ber of planes to be transform ed. It is \(m\) ost efficientw hen \(K\) is a productofsm allprim es. \(\mathrm{K}>=0\); when \(\mathrm{K}=0\), the subroutine retums im m ediately withoutchanging any data.

A (input/output)
O n entry, the three-dim ensional array A (LD A , N, K) contains the data to be transform ed if an in-place transform is requested. O therw ise, it is not referenced. Upon exit, results are stored in A \((\mathbb{M}, 1 \mathbb{N}, 1: K)\).

LD A (input)
Integer specifying the leading dim ension ofA. If an out-of-place transform is desired LDA \(>=M\). E lse if an in-place transform is desired LDA >= \(2 * M / 2+1)\).

B (input/output)
Realarray ofdim ension \(B(2 * L D B, N, K)\). On entry, if an out-of-place transform is requested
B ( \(\left.\left.1: 2^{\star} M / 2+1\right), 1 \mathbb{N}, 1: K\right)\) contains the input data. O therw ise, \(B\) is not referenced. \(B\) is unchanged upon exit.

LD B (input)
If an out-of-place transform is desired, \(2 *\) LD B is the leading dim ension of the array B which contains the data to be transform ed and \(2 *\) LD B \(>=\) \(2 * M / 2+1)\). O therw ise it is not referenced.

W ORK (input/output)
O ne-dim ensional real array of length at least
LW ORK. On input, W ORK m usthave been initialized by RFFT3I.

LW ORK (input)
Integer. LW ORK >= \(M+2 \star(N+K)+4 * K+45)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
rffl3f-com pute the Fourier coefficients of a realperiodic
sequence. The RFFT operations are unnorm alized, so a call ofRFFT3F followed by a callof RFFT3B w ill multiply the input sequence by \(\mathrm{M} * \mathrm{~N} * \mathrm{~K}\).

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE RFFT3F (PLACE,FULL,M ,N,K,A,LDA,B,LDB,W ORK,LW ORK)}
CHARACTER * 1 PLACE,FULL
INTEGERM,N,K,LDA,LDB,LW ORK
REALA (LDA N N ,*),B (LDB N ,*),W ORK (*)
SUBROUT\mathbb{NE RFFT3F_64(PLACE,FULL,M,N,K,A,LDA,B,LDB,W ORK,}
LW ORK)

```
CHARACTER * 1 PLACE,FULL
\(\mathbb{N}\) TEGER*8M,N,K,LDA,LDB,LW ORK
REALA (LDA \(N,{ }^{*}\) ), \(\mathrm{B}(\mathrm{LDB}, \mathrm{N}, \star), \mathrm{W} O R K(\star)\)

\section*{F95 INTERFACE}

SUBROUTINE FFT3F (PLACE,FULL, M ], \(\mathbb{N}],[K], A,[L D A], B,[L D B]\), W ORK,LW ORK)

CHARACTER (LEN=1) ::PLACE,FULL
\(\mathbb{N} T E G E R:: M, N, K, L D A, L D B, L W\) ORK
REAL,D IM ENSION (:) ::W ORK
REAL,D IM ENSION (:,:,:) ::A,B
SUBROUT \(\mathbb{N} E\) FFT3F_64 (PLACE,FULL, \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], B,[L D B]\), W ORK,LW ORK)

CHARACTER (LEN=1) ::PLACE,FULL
\(\mathbb{N} T E G E R(8):: M, N, K, L D A, L D B, L W\) ORK
REAL,D \(\mathbb{I M}\) ENSION (:) ::W ORK
REAL,D IM ENSION (:,:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void rffllf (charplace, char full, int \(m\), int \(n\), int \(k\), float *a, int lda, float *b, int ldlo, float *w ork, intlw ork);
void rffl3f_64 (charplace, char full, long m, long n, long k, float *a, long lda, float *b, long ldb, float *w ork, long lw ork);

\section*{ARGUMENTS}

PLACE (input)
Selectan in-place ('I'or li) or out-of-place ( D 'or b') transform .

FU LL (input)
Selecta full ( F ' or ' f ) or partial (' )
representation of the results. If the caller
selects full representation then an M xN xK real array will transform to produce an \(\mathrm{M} \times N \times \mathrm{K}\) com plex array. If the caller does not select full representation then an \(\mathrm{M} x \mathrm{NXK}\) real array will transform to a \(M / 2+1) x N x K\) complex aray that takes advantage of the sym \(m\) etry properties of a transform ed realsequence.
\(M\) (input) Integer specifying the num ber of row s to be transform ed. It is \(m\) ost efficientw hen \(M\) is a productofsm allprim es. \(\mathrm{M}>=0\); when \(\mathrm{M}=0\), the subroutine retums im mediately w ithout changing any data.

N (input) Integerspecifying the num ber of colum ns to be transform ed. It is m ost efficientw hen N is a product ofsm allprim es. \(\mathrm{N}>=0\); when \(\mathrm{N}=0\), the subroutine retums im mediately w ithout changing any data.
\(K\) (input) Integerspecifying the num ber of planes to be transform ed. It is \(m\) ost efficientw hen \(K\) is a productofsm allprim es. \(\mathrm{K}>=0\); when \(\mathrm{K}=0\), the subroutine retums im m ediately withoutchanging any data.

A (input/output)
O n entry, a three-dim ensional array A (LDA, N, K) that contains input data to be transform ed. On exit, if an in-place transform is done and FU LL is not \(\mathrm{F}^{\prime}\) or \(\left.\mathrm{I}^{\prime}, \mathrm{A}\left(1: 2^{*} \mathrm{M} / 2+1\right), 1 \mathrm{~N}, 1: \mathrm{K}\right) \mathrm{w}\) ill contain the partial transform ed results. If \(F U L L=\) F 'or \(\mathrm{I}^{\prime}, \mathrm{A}(1: 2 \star \mathrm{M}, 1: \mathbb{N}, 1: K)\) w illcontain the com plete transform ed results.

LD A (input)
Leading dim ension of the array containing the data to be transform ed. LD A must be even if the transform ed sequences are to be stored in A.
IfPLACE \(=(\mathrm{D}\) 'or b) LDA \(>=\mathrm{M}\)
IfPLACE \(=\) ( \(I\) 'or \({ }^{\prime \prime}\) ) LD A m ustbe even. If
FULL \(=\left(\mathrm{F}^{\prime}\right.\) or ' \({ }^{\text {f }}\) ), LD \(\mathrm{A}>=2 \star \mathrm{M}\)
FULL is not ( F 'or \({ }^{\prime}\) ' ), LD A \(>=2 * \mathrm{M} / 2+1\) )
B (input/output)
U pon exit, a three-dim ensional array B ( \(2 *\) LD B , \(N, K\) ) that contains the transform ed results if an out-of-place transform is done. O therw ise, \(B\) is not used.
If an out-of-place transform is done and FU LL is not \(\mathrm{F}^{\prime}\) or \(\left.\mathrm{f}^{\prime}, \mathrm{B}\left(1 \cdot 2^{*} \mathrm{M} / 2+1\right), 1 \mathbb{N}, 1 \mathrm{~K}\right) \mathrm{w}\) illcontain the partialtransform ed results. If FU LL \(=\) F 'or 'f', B ( \(1: 2 \star \mathrm{M}, 1 \mathrm{~N}, 1 \mathrm{~K})\) w ill contain the com plete transform ed results.

LD B (input)
\(2 *\) LD B is the leading dim ension of the array \(B\). If an in-place transform is desired LD \(B\) is ignored.
IfPLACE is ( 0 'or \(b^{\prime}\) ) and
FULL is ( F 'or \({ }^{\prime} \mathrm{F}\) ), then \(\mathrm{LDB}>=\mathrm{M}\)
FULL is not ( F ' or ' f '), then LD B \(>=\mathrm{M} / 2+1\)
\(N\) ote thateven though LD \(B\) is used in the argum ent list, 2*LD B is the actual leading dim ension ofB.

W ORK (input/output)
O ne-dim ensional real array of length at least
LW ORK. W ORK m usthave been initialized by RFFT 3I.

LW ORK (input)
Integer. LW ORK >= \(\left.\left.M+2^{\star} \mathbb{N}+K\right)+4 * K+45\right)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
rffliti-initialize the array W SAVE, which is used in both RFFT 3F and RFFT3B.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE RFFT3IM,N,K,W ORK)}
INTEGERM,N,K
REALW ORK (*)
SUBROUT\mathbb{NERFFT3I_64M,N,K,W ORK)}
INTEGER*8M,N,K
REALWORK (*)
F95 INTERFACE
SUBROUT\mathbb{NE FFT3IM ,N,K,W ORK)}
\mathbb{NTEGER ::M,N,K}
REAL,D IM ENSION (:) ::W ORK
SU BROUTINE FFT3I_64M ,N,K,W ORK)
INTEGER (8)::M ,N,K
REAL,DIM ENSION (:) ::W ORK
C INTERFACE
\#include <sunperfh>
void rffl3i(intm,intn, intk, float *W ork);
void rff3i_64 (long m , long n, long k, float *w ork);

```

\section*{ARGUMENTS}

M (input) N um ber of row s to be transform ed. \(\mathrm{M}>=0\).

N (input) N um ber of colum ns to be transform ed. \(\mathrm{N}>=0\).

K (input) N um ber of planes to be transform ed. \(\mathrm{K}>=0\).

W ORK (input/output)
O n entry, an aray ofdim ension \(M+2 *(\mathbb{N}+K)+\) \(4 * K+45\) ) orgreater. RFFT3Ineeds to be called only once to initialize array W O RK before calling RFFT3F and/or RFFT3B ifM , N, K andW ORK rem ain unchanged betw een these calls. Thus, subsequent transform s or inverse transform s of sam e size can be obtained fasterthan the first since they do not require initialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
rfftb -com pute a periodic sequence from its Fourier coefficients. The RFFT operations are unnorm alized, so a call of RFFTF follow ed by a callofRFFTB w ill multiply the input sequence by N .

\section*{SYNOPSIS}
```

    SUBROUTINE RFFTB N,X,W SAVE)
    INTEGER N
    REALX (*),W SAVE (*)
    SUBROUT\mathbb{NERFFTB_64 N,X,W SAVE)}
    INTEGER*8 N
    REALX (*),W SAVE (*)
    F95 INTERFACE
SUBROUT\mathbb{NE FFTB (N ],X,W SAVE)}
\mathbb{NTEGER ::N}
REAL,D IM ENSION (:) ::X,W SAVE
SUBROUT\mathbb{NEFFTB_64(N ],X,W SAVE)}
\mathbb{NTEGER (8) ::N}
REAL,D IM ENSION (:) ::X,W SAVE
C INTERFACE
\#include <sunperfh>
void rfflb (intn, float *x, float *w save);

```

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product ofsm allprim es. \(\mathrm{N}>=0\).
\(X\) (input) \(O n\) entry, an array of length \(N\) containing the sequence to be transform ed.

W SAVE (input) O n entry, W SAVE m ustibe an array ofdim ension (2 * \(\mathrm{N}+15\) ) orgreater and m usthave been initialized by RFFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
rffff-com pute the Fourier coefficients of a periodic
sequence. The FFT operations are unnorm alized, so a callof RFFTF follow ed by a callofRFFTB w ill m ultiply the input sequence by N .

\section*{SYNOPSIS}
```

    SUBROUTINE RFFTF N,X,W SAVE)
    INTEGER N
    REALX (*),W SAVE (*)
    SUBROUT\mathbb{NE RFFTF_64 N,X,W SAVE)}
    INTEGER*8 N
    REALX (*),W SAVE (*)
    F95 INTERFACE
SUBROUT\mathbb{NE FFTF (N ],X,W SAVE)}
\mathbb{NTEGER ::N}
REAL,D IM ENSION (:) ::X,W SAVE
SUBROUT\mathbb{NE FFTF_64(N ],X,W SAVE)}
\mathbb{NTEGER (8) ::N}
REAL,D IM ENSION (:) ::X,W SAVE
C INTERFACE
\#include <sunperfh>
void rfflf(intn, float *x, float *w save);

```

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product ofsm allprim es. \(\mathrm{N}>=0\).
\(X\) (input) \(O n\) entry, an array of length \(N\) containing the sequence to be transform ed.

W SAVE (input) O n entry, W SAVE m ustbe an array ofdim ension (2 * \(\mathrm{N}+15\) ) orgreater and m usthave been initialized by RFFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
rffti-initialize the array W SA VE, which is used in both RFFTF and RFFTB.

\section*{SYNOPSIS}

\section*{SU BROUTINERFFTIN,W SAVE)}
\(\mathbb{N}\) TEGER N
REALW SAVE (*)
SU BROUTINERFFTI_64 N, W SAVE)
\(\mathbb{N}\) TEGER*8 N
REALW SAVE (*)

\section*{F95 INTERFACE}

SUBROUTINE FFTIN,W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL,D \(\mathbb{M}\) ENSION (:) ::W SAVE

SU BROUTINE FFTI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER (8) :: N
REAL,D \(\mathbb{M}\) ENSION (:) ::W SAVE

\section*{C INTERFACE}
\#include <sunperfh>
void rffli(intn, float *w save);
void rffti_ 64 (long n, float *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. \(\mathrm{N}>=0\).

W SAVE (input)
O n entry, an array ofdim ension ( 2 * \(\mathrm{N}+15\) ) or greater. RFFTI needs to be called only once to initialize array \(W\) ORK before calling RFFTF and/or RFFTB if \(N\) and W SAVE rem ain unchanged betw een these calls. Thus, subsequent transform \(s\) or inverse transform s of sam e size can be obtained faster than the first since they do not require indialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE

\section*{NAME}
rfftopt-com pute the length of the closest fastFFT

\section*{SYNOPSIS}
\(\mathbb{N}\) TEGER FUNCTION RFFTOPT (LEN)
\(\mathbb{N}\) TEGER LEN
\(\mathbb{N}\) TEGER*8 FUNCTION RFFTOPT_64 (LEN)
\(\mathbb{N}\) TEGER*8 LEN

F95 INTERFACE
INTEGER FUNCTION RFFTOPT (LEN)
\(\mathbb{N}\) TEGER ::LEN
\(\mathbb{N}\) TEGER (8) FUNCTION RFFTOPT_64 (LEN)
\(\mathbb{N} T E G E R(8):: L E N\)
C INTERFACE
\#include <sunperfh>
int rfftopt(int len);
long rfftopt_64 (long len);

\section*{PURPOSE}

Fourier transform algorithm s , including those used in Perform ance L ibrary, w ork bestw ith vector lengths that are products of sm all prim es. Forexam ple, an FFT of length \(32=2 * * 5 \mathrm{w}\) ill run fasterthan an FFT of prime length 31 because 32 is a productofsm allprim es and 31 is not. If your application is such that you can taper or zero pad your vector to a larger length then this function \(m\) ay help you select a better length and run your FFT faster.

RFFTOPT w ill retum an integerno sm aller than the input argum ent N that is the closestnum ber that is the product of sm allprim es. RFFTO PT w ill retum 16 for an input of \(\mathrm{N}=16\) and retum 18=2*3*3 for an inputof \(\mathrm{N}=17\).

N ote that the length com puted here is not guaranteed to be optim al, only to be a
product of sm allprim es. A lso, the value retumed \(m\) ay change as the underlying
FFT sbecom e capable of handling larger prim es. For exam ple, passing in \(N=51\) to day \(w\) ill retum \(52=2 \star 2 \star 13\) rather than \(51=3 * 17\) because the FFT s in Perform ance Library do not have fast radix 17 code. In the future, radix 17 code m ay be added
and then \(\mathrm{N}=51 \mathrm{w}\) ill retum 51 .

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sasum -Retum the sum of the absolute vahues of a vectorx.

\section*{SYNOPSIS}
```

REALFUNCTION SASUM (N,X,INCX)
\mathbb{NTEGER N, INCX}
REALX (*)
REAL FUNCTION SASUM _64 N,X,\mathbb{NCX)}
INTEGER*8N,\mathbb{NCX}
REALX (*)
F95 INTERFACE
REALFUNCTION ASUM (N ],X, [NCX])
\mathbb{NTEGER ::N,\mathbb{NCX}}\mathbf{N}=\mp@code{N}
REAL,D IM ENSION (:) ::X
REAL FUNCTION ASUM _64 (N ],X,[\mathbb{NCX ])}
\mathbb{NTEGER (8) ::N,\mathbb{NCX}}\mathbf{N}=\mp@code{N}
REAL,D IM ENSION (:) ::X
C INTERFACE
\#include <sunperfh>
float sasum (intn, float *x, int incx);
float sasum _64 (long n, float *x, long incx);

```

\section*{PURPOSE}
sasum Retum the sum of the absolute values of \(x\) where \(x\) is an \(n\)-vector.

\section*{ARGUMENTS}

N (input)
O n entry, \(N\) specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.
X (input)
\((1+(n-1) * a b s(\mathbb{N} C X))\). On entry, the increm ented array X m ust contain the vectorx. U nchanged on exit.
\(\mathbb{N} C X\) (input)
O n entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{I N C X}\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

saxpy - com pute y := alpha * x + y

```

\section*{SYNOPSIS}
```

SUBROUTINE SAXPY N,ALPHA,X,\mathbb{NCX,Y,\mathbb{NCY)}}\mathbf{N},\textrm{N},\textrm{N}
INTEGERN,\mathbb{NCX,INCY}
REAL A LPHA
REALX (*),Y (*)
SUBROUT\mathbb{NE SAXPY_64 N,ALPHA,X,}\mathbb{NCX,Y,}\mathbb{N}CY)
INTEGER*8N,INCX,INCY
REAL ALPHA
REALX (*),Y (*)

```
F95 INTERFACE
SUBROUT \(\mathbb{N E E A X P Y}(\mathbb{N}], A L P H A, X,[\mathbb{N} C X], Y,[\mathbb{N} C Y])\)
\(\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N C Y}\)
REAL ::ALPHA
REAL,D IM ENSION (:) :: X,Y
SU BROUTINEAXPY_64 (N ],A LPHA, X, [ \(\mathbb{N} C X], Y,[\mathbb{N C Y}])\)
\(\mathbb{N} T E G E R(8):: N, \mathbb{I N C X}, \mathbb{N} C Y\)
REAL ::ALPHA
REAL,D IM ENSION (:) :: X,Y

\section*{C INTERFACE}
\#include <sunperfh>
void saxpy (intn, float alpha, float *x, intincx, float *y, int incy);
void saxpy_64 (long n, float alpha, float *x, long incx, float * y , long incy);

\section*{PURPOSE}
saxpy com pute \(\mathrm{y}:=\) alpha * \(\mathrm{x}+\mathrm{y}\) w here alpha is a scalar and \(x\) and \(y\) are \(n\)-vectors.

\section*{ARGUMENTS}

N (input)
On entry, N specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.

\section*{ALPHA (input)}

On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C X))\). Before entry, the increm ented array \(X\) must contain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X\) m ustnotbe zero. U nchanged on exit.

Y (input/output)
( \(1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C Y)\) ). On entry, the increm ented aray \(Y \mathrm{~m}\) ust contain the vectory. On exit, \(Y\) is overw ritten by the updated vectory.
\(\mathbb{N} C Y\) (input)
O n entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of Y. \(\mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

saxpyi-C om pute $y:=$ alpha * $x+y$

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SAXPYINZ,A,X,INDX,Y)}
REALA
REALX (*),Y (*)
INTEGER NZ
INTEGER INDX(*)
SUBROUTINE SAXPYI_64NZ,A,X,\mathbb{NDX,Y)}
REAL A
REALX (*),Y (*)
INTEGER*8NZ
INTEGER*8 \mathbb{NDX (*)}
F95 IN TERFACE
SUBROUT\mathbb{NE AXPYI(NZ],[A],X,NNDX,Y)}
REAL::A
REAL,D IM ENSION (:) ::X,Y
\mathbb{NTEGER ::NZ}
\mathbb{NTEGER,D IM ENSION (:) :: \mathbb{NDX}}\mathbf{N}=\mp@code{N}

```

```

REAL::A
REAL,D IM ENSION (:) ::X,Y
INTEGER (8)::N Z

```


\section*{PURPOSE}

SA XPY IC om pute \(\mathrm{y}:=\) alpha * \(\mathrm{x}+\mathrm{y}\) w here alpha is a scalar, x is a sparse vector, and \(y\) is a vector in full storage form
```

do i=1,n
y (indx (i)) = alpha * x (i) + y (indx (i))
enddo

```

\section*{ARGUMENTS}
\(\mathrm{N} Z\) (input) - \(\mathbb{N}\) TEGER
\(N\) um ber of elem ents in the com pressed form .
U nchanged on exit.

A (input)
On entry, A (LPH A) specifies the scaling value.
U nchanged on exit. A is defaulted to 1.0 E 0 forF 95
\(\mathbb{N}\) TERFACE.
X (input)
V ector containing the values of the com pressed form .
U nchanged on exit.
\(\mathbb{N} D \mathrm{X}\) (input) - \(\mathbb{N}\) TEGER
\(V\) ector containing the indices of the com pressed
form. It is assum ed that the elem ents in \(\mathbb{N} D \mathrm{X}\) are distinctand greater than zero. U nchanged on exit.

Y (output)
V ectoron inputw hich contains the vectorY in full storage form. On exit, only the elem ents
corresponding to the indices in \(\mathbb{N}\) D X have been
m odified.

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}

> sbcom m -block coordinate m atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SBCOMM(TRANSA,M B,N,KB,A LPHA,DESCRA,}

* VAL,B\mathbb{NDX,BJNDX,BNNZ,LB,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,MB,N,KB,DESCRA (5),BNNZ,LB,
* LDB,LDC,LW ORK
\mathbb{NTEGER BINDX (BNNZ),BUNDX (BNNZ)}
REAL ALPHA,BETA
REAL VAL (LB*LB*BNNZ),B (LDB,*),C (LDC **),W ORK (LW ORK)
SUBROUTINE SBCOMM_64(TRANSA,M B,N,KB,ALPHA,DESCRA,
* VAL,BINDX,BJNDX,BNNZ,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,MB,N,KB,DESCRA (5),BNNZ,LB,
* LDB,LDC,LW ORK
INTEGER*8 B}\mathbb{N}DX(BNNZ),BJNDX (BNNZ
REAL ALPHA,BETA
REAL VAL (LB*LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}

SUBROUTINE BCOMM (TRANSA, MB,N, \(K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D, B J N D\), * BNNZ,LB,B,[LDB],BETA,C,[LDC],[WORK],[LW ORK])
\(\mathbb{N} T E G E R\) TRANSA, MB,N,KB,BNNZ,LB
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \quad\) DESCRA, B \(\mathbb{N} D X, B J N D X\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M}\) ENSION (:) ::VAL
REAL,D IM ENSION (:, :) :: B,C
SUBROUTINE BCOMM_64 (TRANSA, MB, N, KB,ALPHA, DESCRA, VAL, B INDX,BUND,
* BNNZ,LB,B,[LDB],BETA,C,[LDC],[WORK],[LWORK])
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,N,KB,BNNZ,LB
\(\mathbb{N} T E G E R * 8, D \mathbb{I M} E N S \mathbb{O N}(:):: D E S C R A, B \mathbb{N} D X, B J N D X\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M}\) ENSION (:) ::VAL
REAL,D \(\mathbb{I M}\) ENSION (:, :) :: B,C

\section*{DESCRIPTION}
\[
C \text { <-alpha op (A) B + beta C }
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrioes, \(A\) is a m atrix represented in block coordinate form at and op (A) is one of
```

op(A)=A or op(A )= A' or op(A )= conjg(A').

```
( 'indicates m atrix transpose)

\section*{ARGUMENTS}
\begin{tabular}{|c|c|}
\hline \multirow[t]{4}{*}{TRANSA} & Indicates how to operate w th the sparse m atrix \\
\hline & 0 : operate w ith m atrix \\
\hline & 1 : operate w th transpose m atrix \\
\hline & 2 : operate \(w\) th the conjugate transpose ofm atrix. 2 is equivalent to 1 if the \(m\) atrix is real. \\
\hline M B & N um berofblock row s in m atrix A \\
\hline N & \(N\) um berof colum \(n s\) in \(m\) atrix \(C\) \\
\hline KB & \(N\) um ber ofblock colum ns in m atrix A \\
\hline A LPHA & Scalarparam eter \\
\hline \multirow[t]{13}{*}{DESCRA} & 0 D escriptor argum ent. Five elem ent integeranay \\
\hline & DESCRA (1) m atrix structure \\
\hline & 0 : general \\
\hline & 1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) ) \\
\hline &  \\
\hline & 3 :Triangular \\
\hline & 4 : Skew (A nti)-Symm etric ( \(A=-\mathrm{A}\) ) \\
\hline & 5 : D iagonal \\
\hline & \(6:\) Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJ}\) ( A\()\) ) \\
\hline & ESSCRA (2) upper/low er triangular indicator \\
\hline & 1: low er \\
\hline & 2 :upper \\
\hline & ESSCRA (3) m ain diagonaltype \\
\hline
\end{tabular}

0 :non-unit
1 : unit
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED )
0 :C C++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL ( scalar array of length LB *LB *BNN Z consisting of the non-zero block entries of \(A\), in any order. Each block is stored in standard colum n-m ajor form .
\(B \mathbb{N} D X(\) integer array of length \(B N N Z\) consisting of the block row indiaes of the block entries of .

B JND X 0 integer anray of length BNNZ consisting of the block colum \(n\) indiges of the block entries of \(A\).

BNNZ num berofblock entries

LB dim ension of dense blocks com posing A.
B 0 rectangular array with first dim ension LD B .
LD B leading dim ension ofB
BETA Scalarparam eter

C 0 rectangular array w ith firstdim ension LD C .

LD C leading dim ension ofC
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the currentversion.

LW ORK length ofW ORK array.LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov \(\mathrm{m}_{\mathrm{c}}\) csd/Staffk Rem ington/tspoblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

sbdim m -block diagonal form atm atrix m atrix m ultiply

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SBD IMM (TRANSA,M B,N,KB,ALPHA,DESCRA,}

* VAL,BLDA,\mathbb{BDIAG,NBDIAG,LB,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,MB,N,KB,DESCRA (5),BLDA,NBDIAG,LB,}
* LDB,LDC,LWORK
INTEGER \mathbb{BDIAG NBDIAG)}
REAL ALPHA,BETA
REAL VAL(LB*LB*BLDA*NBDIAG),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINESBD IMM_64(TRANSA,M B N,NB,ALPHA,DESCRA,
* VAL,BLDA,\mathbb{BDIAG,NBDIAG,LB,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
NNTEGER*8 TRANSA,MB,N,KB,DESCRA (5),BLDA,NBDIAG,LB,
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 IBD\mathbb{AG NBDIAG)}}\mathbf{N}=()
REAL ALPHA,BETA
REAL VAL (LB*LB*BLDA*NBDIAG),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}

SUBROUTINE BD \(\mathbb{I M} M\) (TRANSA, \(M B, \mathbb{N}], K B, A L P H A, D E S C R A, V A L, B L D A\), * \(\mathbb{B D} \mathbb{I} G, N B D \mathbb{I A G}, \mathrm{LB}, \mathrm{B},[\mathrm{LD} \mathrm{B}], \mathrm{BETA}, \mathrm{C},[\mathrm{LDC}],[\mathrm{W} O R K],[\mathrm{LW} O R K])\)
\(\mathbb{N} T E G E R\) TRANSA, MB,KB,BLDA,NBD \(\mathbb{I A} G, L B\)
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \quad\) DESCRA, \(\mathbb{B D} \mathbb{I} G\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M}\) ENSION (:) ::VAL
REAL,D IM ENSION (:, :) :: B,C
SUBROUTINE BD \(\mathbb{M} M \_64\) (TRANSA, MB, \(\left.\mathbb{N}\right], K B, A L P H A, D E S C R A, V A L, B L D A\),
* \(\mathbb{B D} \mathbb{I} G, N B D \mathbb{I} G, L B, B,[L D B], B E T A, C,[L D C],[W\) ORK], [LW ORK])
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,KB,BLDA,NBDIAG,LB
\(\mathbb{N} T E G E R * 8, D \mathbb{I} \operatorname{ENS} \mathbb{O} N(:):: \quad \mathrm{DESCRA}, \mathbb{B D} \mathbb{I A}\)
REAL ALPHA,BETA
REAL,D IM ENSION (:) ::VAL
REAL,D \(\mathbb{I M}\) ENSION (:, :) :: B,C

\section*{DESCRIPTION}
\[
C \text { <-alpha op (A) B + beta C }
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrioes, \(A\) is a m atrix represented in block diagonal form at and op(A) is one of \(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).
('indicatesm atrix transpose)

\section*{ARGUMENTS}
\begin{tabular}{|c|c|}
\hline \multirow[t]{4}{*}{TRANSA} & A Indicates how to operate w th the sparse \(m\) atrix \\
\hline & 0 : operate w th m atrix \\
\hline & 1 : operate w ith transpose m atrix \\
\hline & 2 : operate \(w\) th the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real. \\
\hline M B & N um berofblock row s in m atrix A \\
\hline N & \(N\) um berof colum \(n s\) in \(m\) atrix \(C\) \\
\hline K B & \(N\) um ber ofblock colum ns in m atrix A \\
\hline A LPH A & Scalar param eter \\
\hline \multirow[t]{13}{*}{DESCRA} & () D escriptor argum ent. Five elem ent integer amay \\
\hline & DESCRA (1) m atrix structure \\
\hline & 0 : general \\
\hline & 1 : symmetric ( \(\mathrm{A}=\mathrm{A}\) ) \\
\hline & 2: Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) ) \\
\hline & 3 :Triangular \\
\hline & 4 : Skew (A nti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) ) \\
\hline & 5 :D iagonal \\
\hline & 6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJG}(\mathrm{A})\) ) \\
\hline & D ESCRA (2) upper/low er triangular indicator \\
\hline & 1 : low er \\
\hline & 2 : upper \\
\hline & D ESCRA (3) m ain diagonaltype \\
\hline
\end{tabular}

0 : non-unit
1 : unit
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL \(0 \quad\) tw o-dim ensionalLB *LB *BLD A -by-N BD IA G scalar anay consisting of the NBD IA G nonzero block diagonal in any order. Each dense block is stored in standard colum n.m ajor form .

BLD A leading block dim ension ofV A L ( ).
IBD IA G 0 integer amay of length N BD IA G consisting of the corresponding diagonaloffsets of the non-zero block diagonals ofA in VA L. Low ertriangular block diagonals have negative offsets, the \(m\) ain block diagonal has offset 0, and uppertriangular block diagonals have positive offset.

NBD IA G the num berofnon-zero block diagonals in A.
LB dim ension of dense blocks com posing A.
B 0 rectangular array w ith first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w th first dim ension LD C.

LD C leading dim ension ofC
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib org/utk/papers/sparse .ps

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

sbdism - block diagonal form attriangular solve

```

\section*{SYNOPSIS}
```

SUBROUTINESBD ISM (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,

* VAL,BLDA, $\mathbb{B D} \mathbb{A} G, N B D \mathbb{I A}, L B$,
* B,LDB,BETA, C,LDC,W ORK,LW ORK )
$\mathbb{N} T E G E R$ TRANSA,MB,N,UNITD,DESCRA (5), BLDA,NBD $\mathbb{I A} G, L B$,
* LDB,LDC,LW ORK
$\mathbb{N} T E G E R \quad \mathbb{B D} \mathbb{I} G \mathbb{N} B D \mathbb{I} G)$
REAL ALPHA,BETA
REAL DVMB*LB*LB),VAL (LB*LB*BLDA,NBD IAG),B(LDB,*),C (LDC,*),
* WORK (LW ORK)
SUBROUTINE SBD ISM_64 (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,
* VAL,BLDA, $\mathbb{B D} \mathbb{I} G, N B D \mathbb{A} G, L B$,
* B,LDB,BETA, C,LDC,W ORK,LW ORK )
$\mathbb{N} T E G E R * 8$ TRANSA, M B,N,UN ITD,DESCRA (5), BLDA,NBD IAG,LB,
* LDB,LDC,LWORK
$\mathbb{N} T E G E R * 8 \mathbb{B D} \mathbb{A} G \mathbb{N} \operatorname{BD} \mathbb{A})$
REAL ALPHA,BETA
REAL DVMB*LB*LB),VAL(LB*LB*BLDA,NBDIAG),B(LDB,*),C(LDC,*),
* $\quad$ WORK (LWORK)

```

\section*{F95 INTERFACE}

SUBROUTINEBD ISM (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,VAL,BLDA, * \(\mathbb{B D} \mathbb{I} G, N B D \mathbb{I} G, L B, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\) \(\mathbb{N} T E G E R\) TRANSA, MB,N,UNITD,BLDA,NBDIAG,LB \(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \quad \mathrm{E} E C R A, \mathbb{B D} \mathbb{I} G\)
REAL ALPHA,BETA
REAL,DIM ENSION (:) ::VAL,DV
REAL,D \(\mathbb{M}\) ENSION (:, :) :: B,C

SUBROUT INE BD ISM _64 (TRANSA, MB,N,UNITD,DV, ALPHA,DESCRA,VAL, BLDA,
* \(\mathbb{B D} \mathbb{I A G}, N B D \mathbb{I A G}, \mathrm{LB}, \mathrm{B},[\mathrm{LDB}], B E T A, \mathrm{C},[\mathrm{LDC}],[W O R K],[L W O R K])\)
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,N,UNITD, BLDA,NBD IAG, LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O}(:):: \quad \mathrm{DESCRA}, \mathbb{B D} \mathbb{I} G\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M} E N S I O N(:):: V A L, D V\)
REAL,D \(\mathbb{I}\) ENSION (:, :) :: B , C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { op (A)B+BETA C } \\
& C<-A L P H A \text { OP (A)D B + BETA } C
\end{aligned}
\]
where ALPHA and BETA are scalar, \(C\) and \(B\) are m by \(n\) dense \(m\) atrices, \(D\) is ablock diagonalm atrix, \(A\) is a unit, ornon-unit, upperor low ertriangularm atrix represented in block diagonal form at and op (A) is one of \(\operatorname{op}(A)=\operatorname{inv}(A)\) or op \((A)=\operatorname{inv}(A) \operatorname{or} \operatorname{op}(A)=\operatorname{inv}\left(\operatorname{conjg}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix 0 : operate w ith m atrix 1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(s\) in \(m\) atrix \(A\)
\(N \quad N\) um berof colum \(n s\) in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity \(m\) atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)

DV () A rray of length M B *LB *LB containing the elem ents of the diagonalblocks of them atrix \(D\). The size of each square block is LB \(-b y-4 B\) and each block is stored in standard colum n-m ajor form .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay DESCRA (1) m atrix structure
            0 : general
            1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
            2 : Herm itian ( \(A=\operatorname{CONJG}(A))\)
            3 :Triangular
            4 : Skew (A nti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) )
            5 :D iagonal
            6 : Skew Herm itian ( \(A=-C O N J(A)\) )
                            N ote: For the routine, D ESCRA (1)=3 is only supported.
                            D ESCRA (2) upper/low er triangular indicator
            1 : low er
            2 :upper
DESCRA (3) m ain diagonaltype
            0 : non-identity blocks on the \(m\) ain diagonal
            1 : identity diagonalblocks
            2 : diagonalblocks are dense \(m\) atrices
            DESCRA (4) A may base \(\mathbb{N O T} \mathbb{I}\) PLEM ENTED)
            0 :C C ++ com patible
            1 :Fortran com patible
                    DESCRA (5) repeated indices? \(\mathbb{N O T} \mathbb{I}\) PLEM ENTED)
                    0 : unknown
                            1 : no repeated indices
VAL () Two-dim ensionalLB *LB *B LD A -by-N BD IA G scalaranay
consisting of the N BD IA G non-zero block diagonal.
Each dense block is stored in standard colum n-m ajor form .
B LD A Leading block dim ension ofV A L (). Should be greater
    than orequal to M B .
IBD IA G 0 integer amay of length NBD IA G consisting of the corresponding diagonal offsets of the non-zero block diagonals ofA in VA L. Low ertriangularblock diagonals have negative offsets, them ain block diagonalhas offset 0 , and upper triangularblock diagonals have positive offset. Elem ents of IBD IA G M UST be sorted in increasing order.
NBD IA G The num berofnon-zero block diagonals in A.
LB D im ension of dense blocks com posing A.
B 0 Rectangular aray with firstdim ension LD B .
LD B Leading dim ension of B .
BETA Scalarparam eter.
C 0 Rectangular array w ith first dim ension LD C .
LD C Leading dim ension of C .

W ORK 0 scratch array of length LW ORK. On exit, ifLW ORK=-1,W ORK (1) retums the optim um size ofLW ORK.

LW ORK length ofW ORK array. LW ORK should be at least M B *LB.

Forgood penform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=M B * L B * N \_C P U S\) where \(N\) _CPU \(S\) is the \(m\) axim um num berof processors available to the program .

IfLW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no enrorm essage related to LW ORK is issued by X ERBLA.

\section*{SEE ALSO}

N IST FORTRA N Sparse B las U sers G uide available at: http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/4tk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such tests \(m\) ustbe perform ed before calling this routine.
2. If \(D E S C R A(3)=0\), the low er or upper triangularpart of each diagonalblock is used by the routine depending on DESCRA (2).
3. If \(D E S C R A\) (3)=1, the unit diagonalblocksm ightorm ight notbe referenced in the BD I representation of a sparse \(m\) atrix. They are notused anyw ay .
4. If \(D E S C R A\) (3)=2, diagonalblocks are considered as dense \(m\) atrices and the LU factorization \(w\) th partial pivoting is used by the routine. WORK (1)=0 on retum if the factorization for alldiagonalblocks has been com pleted successfully, otherw ise WORK (1) = -iw here is the block
num ber forw hich the LU factorization could notbe com puted.
5. The routine can be applied for solving triangular system \(s\) w hen the upper or low er triangle of the general.sparse m atrix \(A\) is used. H ow erver \(D E S C R A\) (1) m ustbe equal to 3 in this case.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sbdsdc - com pute the singular value decom position (SV D ) of a
realN -by -N (upper or low er) bidiagonalm atrix B

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SBDSDC (UPLO,COM PQ,N,D,E,U,LDU,VT,LDVT,Q,IQ,}
W ORK,IN ORK,\mathbb{NFO)}
CHARACTER * 1 UPLO,COMPQ
\mathbb{NTEGER N,LDU,LDVT, INFO}
\mathbb{NTEGER IQ (*), IN ORK (*)}
REALD (*),E (*),U (LDU ,*),VT (LDVT,*),Q (*),W ORK (*)
SU BROUT\mathbb{NE SBD SDC_64 (UPLO,COM PQ,N,D,E,U,LDU ,VT,LDVT,Q,IQ,}
W ORK,\mathbb{IN ORK, INFO)}
CHARACTER * 1 UPLO,COMPQ
\mathbb{NTEGER*8N,LDU,LDVT,NNFO}
\mathbb{NTEGER*8 ID (*), IN ORK (*)}
REALD (*),E (*),U (LDU ,*),VT (LDVT **),Q (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE BDSDC (UPLO, COMPQ, \(\mathbb{N}], D, E, U,[L D U], V T,[L D V T], Q, \mathbb{Q}\), [W ORK], [IW ORK], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1): :UPLO, COMPQ
\(\mathbb{N} T E G E R:: N, L D U, L D V T, \mathbb{N F O}\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{Q}, \mathbb{I N} O R K\)
REAL,D \(\mathbb{M}\) ENSION (:) ::D , E, Q, W ORK
REAL,D IM ENSION (:,:) ::U,VT

SU BROUTINE BDSDC_64 (UPLO, COMPQ, \(\mathbb{N}], D, E, U,[L D U], V T,[L D V T], Q\), IQ , [W ORK ], [IW ORK ], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1): : UPLO, COMPQ
\(\mathbb{N}\) TEGER (8) ::N,LDU,LDVT, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{Z}, \mathbb{I W} O R K\)
REAL,D \(\mathbb{M}\) ENSION (:) ::D ,E,Q,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::U,VT

\section*{C INTERFACE}
\#include < sunperfh>
void sbdsdc (charuplo, char com pq, intn, float *d, float *e, float*u, int ldu, float *vt, int ldvt, float *q, int *iq, int *info);
void sodsdc_64 (charuplo, char com pq, long n, float *d, float *e, float *u, long ldu, float *vt, long ldvt, float *q, long *iq, long *info);

\section*{PURPOSE}
sbdsdc com putes the singular value decom position (SVD ) of a realN -by -N (upper or low er) bidiagonalm atrix \(\mathrm{B}: \mathrm{B}=\mathrm{U} * \mathrm{~S}\) * V T, using a divide and conquerm ethod, where \(S\) is a diagonalm atrix w ith non-negative diagonalelem ents the singular values ofB ), and U and V T are orthogonalm atrioes of left and right singularvectors, respectively. SBD SD C can be used to com pute all singular values, and optionally, singular vectors or singularvectors in com pact form .

This code \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. It willw ork on m achines w ith a guard digit in add/subtract, or on those binary machines w thout guard digits which subtract like the C ray \(\mathrm{X}-\mathrm{M} \mathrm{P}, \mathrm{C}\) ray Y M P, C ray \(\mathrm{C}-90\), or Cray-2. Itcould conœívably fail on hexadecim al or decin al machines w thout guard digits, butw e know of none. See SLA SD 3 fordetails.

The code currently callSLA SD Q if singularvalues only are desired. H ow ever, it can be slightly \(m\) odified to com pute singular values using the divide and conquerm ethod.

\section*{ARGUMENTS}
```

UPLO (input)
$=\mathrm{U}$ ': B is upperbidiagonal.
= L ': B is low erbidiagonal.

```

COMPQ (input)
Specifies w hether singularvectors are to be com puted as follow s:
= N ': C om pute singular values only;
\(=P^{\prime}:\) C om pute singularvalues and com pute singular vectors in compact form; = 'I': C om pute singular vahues and singular vectors.

N (input) The order of the m atrix \(\mathrm{B} . \mathrm{N}>=0\).

D (input/output)
O n entry, the n diagonalelem ents of the bidiagonal \(m\) atrix \(B\). On exit, if \(\mathbb{N} F O=0\), the singular values ofB.
E (input/output)
O n entry, the elem ents of \(E\) contain the offdiagonalelem ents of the bidiagonalm atrix whose SVD is desired. On exit, E has been destroyed.

U (output)
If \(C O M P Q=' I \prime\), then: \(O n\) exit, if \(\mathbb{N} F O=0, U\) contains the left singular vectors of the bidiagonalm atrix. Forothervalues of \(C O M P Q, U\) is not referenced.

LD U (input)
The leading dim ension of the array \(U\). LD U >= 1 . If singular vectors are desired, then LD U \(>=\max\) ( 1, N ).

VT (output)
If \(C O M P Q=I^{\prime}\), then: \(O n\) exit, if \(\mathbb{N} F O=0, V T '\) contains the right singularvectors of the bidiagonalm atrix. Forothervalues of COM PQ,VT is not referenced.

LDVT (input)
The leading dim ension of the array V T . LD V T >= 1 . If singularvectors are desired, then LDV T \(>=\mathrm{max}\) ( \(1, \mathrm{~N})\).
\(Q\) (input) If \(C O M P Q=P\) ', then: \(O n\) exit, if \(\mathbb{N} F O=0, Q\) and IQ contain the leftand rightsingularvectors in a com pact form, requiring \(O(N \log N\) ) space instead of \(2 * \mathrm{~N} * * 2\). In particular, \(Q\) contains all theREAL data in LDQ >= \(\mathrm{N}^{*}(11+2 * S M\) LSZ + \(8 * \mathbb{N}\) T (LO G_2 \(\mathbb{N} /(\) SM LSIZ+1)))) w ords ofm em ory, where SM LSIZ is retumed by ILAENV and is equal to the maxim um size of the subproblem sat the bottom of
the com putation tree (usually about 25). For othervahes of COM PQ, Q is notreferenced.

IQ (output)
If \(C O M P Q=P^{\prime}\), then: On exit, if \(\mathbb{N} F O=0, Q\) and IQ contain the leftand rightsingularvectors in a com pact form, requiring \(O \mathbb{N}\) log \(N\) ) space instead of \(2 * N * * 2\). In particular, IQ contains all \(\mathbb{N T E G E R}\) data in LD IQ \(>=N *(3+\) \(3 * \mathbb{N}\) T (LOG_2 \(\mathbb{N} /(\) SM LSIZ+1)))) w ords ofm em ory, where SM LSIZ is retumed by \(\mathbb{L A} E N V\) and is equal to the \(m\) axim um size of the subproblem s at the bottom of the com putation tree (usually about 25). For other values of COM PQ, \(\mathbb{Q}\) is notreferenced.

W ORK (w orkspace)
IfCOMPQ \(=N\) 'then LW ORK \(>=(2 * N)\). IfCOMPQ \(=\)
\(P^{\prime}\) then \(L W O R K>=(6 * N)\). IfCOMPQ \(=\) I'then
LW ORK >= ( \(3 * N * * 2+4 * N)\) 。

IV ORK (w orkspace)
dim ension ( \(8 * N\) )
\(\mathbb{I N F O}\) (output)
= 0: successfulexit.
\(<0:\) if \(\mathbb{N ~ F O ~}=-\) i, the ith argum enthad an illegalvalue.
\(>0\) : The algorithm failed to com pute an singular value. The update process ofdivide and conquer failed.

\section*{FURTHER DETAILS}

B ased on contributions by
\(M\) ing \(G u\) and \(H\) uan Ren, C om puterScience D ìvision, U niversity of

C alifomia at Berkeley, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sbdsqr - com pute the singular value decom position (SVD ) of a realN -by-N (upper or low er) bidiagonalm atrix B.

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SBDSQR (UPLO,N,NCVT,NRU,NCC,D,E,VT,LDVT,U,LDU,C,}
LDC,W ORK,\mathbb{NFO)}
CHARACTER * 1UPLO
INTEGERN,NCVT,NRU,NCC,LDVT,LDU,LDC,INFO
REALD (*),E (*),VT (LDVT,*),U (LDU ,*),C (LDC ,*),W ORK (*)
SUBROUTINE SBD SQR_64 (UPLO,N,NCVT,NRU,NCC,D,E,VT,LDVT,U,LDU,
C,LDC,W ORK,INFO)

```
CHARACTER * 1 UPLO
\(\mathbb{N}\) TEGER*8N,NCVT,NRU,NCC,LDVT,LDU,LDC, \(\mathbb{N} F O\)
REALD (*), E (*), VT (LDVT,*), U (LDU ,*), C (LDC ,*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE BDSQR (UPLO, \(\mathbb{N}], \mathbb{N} C V T], \mathbb{N} R U], \mathbb{N C C}], D, E, V T,[L D V T]\), \(\mathrm{U},[\operatorname{LDU}], \mathrm{C},[\operatorname{LDC}], \mathbb{W} O R K],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N} T E G E R:: N, N C V T, N R U, N C C, L D V T, L D U, L D C, \mathbb{N F O}\)
REAL,D IM ENSION (:) ::D,E,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::VT,U,C
SUBROUTINE BDSQR_64 (UPLO, \(\mathbb{N}], \mathbb{N C V T ]}, \mathbb{N} R U], \mathbb{N C C}], D, E, V T,[L D V T]\), U, [LDU ], C, [LDC], [W ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R(8):: N, N C V T, N R U, N C C, L D V T, L D U, L D C, \mathbb{N} F O\) REAL,D \(\mathbb{M}\) ENSION (:) ::D , E,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::VT,U,C

\section*{C INTERFACE}
\#include < sunperfh>
void sbdsqr(charuplo, intn, intncvt, int nnu, int ncc, float *d, float *e, float *vt, int ldvt, float *u, int ldu, float * c , int ldc, int*info);
void sbdsqr_64 (charuplo, long n, long ncvt, long nua, long ncc, float *d, float *e, float * vt, long ldvt, float*u, long ldu, float *C, long ldc, long *info);

\section*{PURPOSE}
sbdsqr com putes the singular value decom position (SVD ) of a realN -by-N (upper or low er) bidiagonalm atrix \(B: B=Q * S\) * \(P^{\prime}\) ( \(P\) 'denotes the transpose of ), w here \(S\) is a diagonal \(m\) atrix \(w\) ith non-negative diagonal elem ents the singular values of \(B\) ), and \(Q\) and \(P\) are orthogonalm atrices.

The routine com putes \(S\), and optionally com putes \(U\) * \(Q, P^{\prime}\) * \(V T\), orQ '* \(C\), for given real inputm atrices \(U, V T\), and \(C\).

See "C om puting Sm allSingularV ahues ofB idiagonalM atrioes W ith G uaranteed H igh Relative A ccuracy," by J.D em m eland W . K ahan, LA PA CK W orking N ote \#3 (orSIAM J.Sci. Statist. Com put.vol.11, no.5,pp.873-912,Sept1990) and "A ccurate singular values and differential qd algorithm s," by B. Parlett and V.Femando, TechnicalReportCPAM -554, \(M\) athem atics D epartm ent, U niversity of C alifomia at Berkeley, July 1992 for a detailed description of the algorithm .

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}: \mathrm{B}\) isupperbidiagonal;
= L ': B is low erbidiagonal.

N (input) The order of the matrix \(\mathrm{B} . \mathrm{N}>=0\).
NCVT (input)
The num ber of colum ns of the \(m\) atrix VT.NCVT \(>=0\).

NRU (input)
The num berof row s of the \(m\) atrix \(U . N R U>=0\).

NCC (input)
The num ber of colum ns of the m atrix C . \(\mathrm{NCC}>=0\).
D (input/output)
O n entry, the n diagonalelem ents of the bidiagonal matrix B. On exit, if \(\mathbb{N} F O=0\), the singular values ofB in decreasing order.

E (input/output)
O n entry, the elem ents ofE contain the offdiagonalelem ents of the bidiagonalm atrix whose SVD is desired. On norm al exit ( \(\mathbb{N F O}=0\) ), E is destroyed. If the algorithm does not converge ( \(\mathbb{N}\) FO > 0), D and E w illcontain the diagonal and superdiagonal elem ents of a bidiagonal \(m\) atrix orthogonally equivalent to the one given as input. \(E(\mathbb{N})\) is used forw orkspace.

VT (input/output)
On entry, an N-by-N CVT m atrix VT. On exit, VT is overw ritten by \(\mathrm{P}^{\prime} * \mathrm{VT} . \mathrm{VT}\) is not referenced if \(\mathrm{NCVT}=0\).

LDVT (input)
The leading dim ension of the array VT. LDVT >= \(\mathrm{max}(1, \mathrm{~N})\) if \(\mathrm{NCVT}>0\); LDVT \(>=1\) ifNCVT \(=0\) 。

U (input/output)
On entry, an NRU by N m atrix U. On exit, \(U\) is overw ritten by \(U * Q\). \(U\) is not referenced if \(N R U\) \(=0\).

LD U (input)
The leading dim ension of the anay \(U\). LD \(U\) >= max ( \(1, N R U\) ).

C (input/output)
On entry, an N-by -NCC matrix C. On exit, C is overw rilten by Q '* C . C is not referenced if N C C \(=0\).

LD C (input)
The leading dim ension of the amay C. LD C >= \(m\) ax \((1, N)\) if \(N C C>0 ; L D C>=1\) if \(N C C=0\).

W ORK (w orkspace)
dim ension ( \(4 * \mathrm{~N}\) )
\(\mathbb{N}\) FO (output)
= 0: successfulexit
<0: If \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue
> 0 : the algorithm did notconverge; D and E contain the elem ents of a bidiagonalm atrix w hich is orthogonally sim ilarto the input \(m\) atrix \(B\); if \(\mathbb{N F O}=i\) ielem ents of E have notconverged to zero.

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
sbelm m -block Ellpack form atm atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SBELMM(TRANSA,M B,N,KB,ALPHA,DESCRA,}

* VAL,B\mathbb{NDX,BLDA,MAXBNZ,LB,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,MB,N,KB,DESCRA (5),BLDA,MAXBNZ,LB,
* LDB,LDC,LW ORK
INTEGER BINDX (BLDA,MAXBNZ)
REAL ALPHA,BETA
REAL VAL (LB*LB*BLDA*MAXBNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NE SBELMM _64(TRANSA,M B ,N,KB,ALPHA,DESCRA,}
* VAL,B\mathbb{NDX,BLDA,MAXBNZ,LB,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,M B,N,KB,DESCRA (5),BLDA,MAXBNZ,LB,
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 B\mathbb{NDX (BLDA,MAXBNZ)}}\mathbf{M}=()
REAL ALPHA,BETA
REAL VAL (LB*LB*BLDA*M AXBNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}

SUBROUTINEBELMM (TRANSA, MB, \(\mathbb{N}], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D\), * BLDA,MAXBNZ,LB,B,[LDB],BETA,C,[LDC],[WORK], [LW ORK])
\(\mathbb{N} T E G E R\) TRANSA, MB,KB,BLDA,MAXBNZ,LB
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \quad D E S C R A, B \mathbb{N} D\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M}\) ENSION (:) ::VAL
REAL,D \(\mathbb{I}\) ENSION (: : : :: B,C
SUBROUTINEBELMM_64(TRANSA, MB, \(\mathbb{N}], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\),

BLDA, MAXBNZ,LB,B,[LDB],BETA,C,[LDC], [WORK],[LWORK])
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,KB,BLDA,MAXBNZ,LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: \quad \mathrm{DESCRA}, \mathrm{B} \mathbb{N} D \mathrm{X}\)
REAL ALPHA,BETA
REAL, D \(\mathbb{I M}\) ENSION (:) ::VAL
REAL,D IM ENSION (:, :) :: B , C

\section*{DESCRIPTION}
C <-ałha op (A ) B + beta C
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, \(A\) is a m atrix represented in block Elloack form at and \(o p(A)\) is one of
```

op(A) =A or op(A )=A' or op (A ) = conjg(A').

```
( 'indicates m atrix transpose)

\section*{ARGUMENTS}
\begin{tabular}{|c|c|}
\hline \multirow[t]{4}{*}{TRANSA} & A Indicates how to operate w ith the sparse m atrix \\
\hline & 0 : operate w ith m atrix \\
\hline & 1 : operate w ith transpose \(m\) atrix \\
\hline & 2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real. \\
\hline M B & N um berofblock row s in m atrix A \\
\hline N & \(N\) um berof colum ns in m atrix C \\
\hline KB & \(N\) um ber ofblock 00 lum ns in m atrix A \\
\hline A LPH A & Scalar param eter \\
\hline \multirow[t]{13}{*}{DESCRA} & () D escriptor argum ent. Five elem ent integer amay \\
\hline & DESCRA (1) m atrix structure \\
\hline & 0 : general \\
\hline & 1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) ) \\
\hline & \(2:\) Herm itian ( \(\mathrm{A}=\mathrm{CONJ}\) ( A ) ) \\
\hline & 3 :Triangular \\
\hline & 4 : Skew (Anti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) ) \\
\hline & 5 :D iagonal \\
\hline & 6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJ}\) ( A ) ) \\
\hline & D ESCRA (2) upper/low er triangular indicator \\
\hline & 1 : low er \\
\hline & 2 :upper \\
\hline & D ESCRA (3) m ain diagonal type \\
\hline
\end{tabular}

0 : non-unit
1 : unit
DESCRA (4) A ray base NOT IM PLEM ENTED )
0 :C C++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL 0 scalar array of length LB *LB *BLDA *M AXBN Z containing \(m\) atrix entries, stored 00 lum \(n-m\) ajorw thin each dense block.
\(B \mathbb{N} D X_{0} \quad\) tw o-dim ensional integerBLD A -by \(-M A X B N Z\) aray such B IND X (i,:) consists of the block colum \(n\) indices of the nonzero blocks in block row i, padded by the integer value i if the num ber of nonzero blocks is less than MAXBNZ.

BLDA leading dim ension of \(\operatorname{INDX(:,:).}\)

M A X BN Z max num berof nonzerosblocks per row .
LB row and colum \(n\) dim ension of the dense blocks com posing VAL.

B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension ofB
BETA Scalarparam eter

C 0 rectangular array w th first dim ension LD C .
LD C leading dim ension of \(C\)
W ORK () scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK aray. LW ORK is not referenced in the cumentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U sers G uide available at:
http://m ath nist.gov/n csd/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)

Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
sbelsm -block Ellpack form at triangular solve

\section*{SYNOPSIS}
```

SUBROUTINE SBELSM (TRANSA, M B,N,UNITD,DV,ALPHA,DESCRA,

* VAL,BINDX,BLDA,MAXBNZ,LB,
* B,LDB,BETA, C,LDC,W ORK,LW ORK )
$\mathbb{N} T E G E R$ TRANSA, MB,N,UNITD,DESCRA (5),BLDA,MAXBNZ,LB,
* LDB,LDC,LW ORK
$\mathbb{N}$ TEGER B $\mathbb{N} D \mathrm{X}($ BLDA, MAXBNZ)
REAL ALPHA,BETA
REAL $D V(M B * L B * L B), V A L(L B * L B * B L D A * M A X B N Z), B(L D B, *), C(L D C, *)$,
* $\quad$ WORK (LWORK)
SUBROUTINE SBELSM _64(TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,
* VAL,BINDX,BLDA,MAXBNZ,LB,
* B,LDB,BETA, C,LDC,W ORK,LW ORK )
$\mathbb{N} T E G E R * 8$ TRANSA,MB,N,UNITD,DESCRA (5),BLDA,MAXBNZ,LB,
* LDB,LDC,LWORK
$\mathbb{N} T E G E R * 8 B \mathbb{N} D X(B L D A, M A X B N Z)$
REAL ALPHA,BETA
REAL $D V M B * L B * L B), V A L(L B * L B * B L D A * M A X B N Z), B(L D B, *), C(L D C, *)$,
* $\quad$ WORK (LWORK)

```

\section*{F95 INTERFACE}

SUBROUTINE BELSM (TRANSA, MB, \(\mathbb{N}], \operatorname{UN} I T D, D V, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\), * BLDA,MAXBNZ,LB,B,[LDB],BETA,C,[LDC], [W ORK], [LW ORK])
\(\mathbb{N} T E G E R\) TRANSA,MB,UNITD, BLDA,MAXBNZ,LB
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \quad D E S C R A, B \mathbb{N} D X\)
REAL ALPHA,BETA
REAL,D IM ENSION (:) ::VAL,DV
REAL,D \(\mathbb{M}\) ENSION (:, :) :: B,C

SUBROUT \(\mathbb{N} E \operatorname{BELSM}\) _64 (TRANSA, MB, \(\mathbb{N}], U N T D, D V, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\),
* \(B L D A, M A X B N Z, L B, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,UNITD, BLDA, MAXBNZ,LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O}(:):: D E S C R A, B \mathbb{N} D X\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M} E N S I O N(:):: V A L, D V\)
REAL,D \(\mathbb{M}\) ENSION (: :) :: B , C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { op (A)B+BETA C } \\
& C<-A L P H A \text { OP (A)D B + BETA } C
\end{aligned}
\]
where ALPHA and BETA are scalar, \(C\) and \(B\) are m by \(n\) dense matrices, \(D\) is ablock diagonalm atrix, \(A\) is a unit, ornon-unit, upperor low ertriangularm atrix represented in block Elhoack form at and op (A ) is one of \(\operatorname{op}(A)=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}(A) \operatorname{or} \operatorname{op}(A)=\operatorname{inv}\left(\operatorname{conjg}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix 0 : operate \(w\) ith \(m\) atrix 1 : operate \(w\) th transpose \(m\) atrix 2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(s\) in \(m\) atrix \(A\)

N \(\quad N\) um berof colum ns in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity \(m\) atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)

DV () A may of the length M B *LB *LB consisting of the block entries ofblock diagonalm atrix \(D\) w here each block is stored in standard colum n-m ajor form .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general

1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2: Herm itian ( \(A=\operatorname{CONJ}(A)\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) )
5 :D iagonal
6 : Skew Herm titian ( \(A=-\operatorname{CON}\) J ( \(A\) ) )
N ote: For the routine, D ESCRA (1)=3 is only supported.
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 : upper
DESCRA (3) m ain diagonal type
0 : non-identity blocks on the \(m\) ain diagonal
1 : identity diagonalblocks
2 : diagonalblocks are dense \(m\) atrices
DESCRA (4) A ray base \(\mathbb{N O T} \mathbb{M}\) PLEM ENTED )
0 : C C C+ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? \(\mathbb{N} O T \mathbb{M}\) PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () scalar array of length LB *LB *BLD A *M A X BN Z containing \(m\) atrix entries, stored colum \(n-m\) ajorw thin each dense block.

B \(\mathbb{N}\) D X () tw o-dim ensionalintegerB LD A boy-M A X BN Z array such B IND X ( \(i\), : ) consists of the block colum \(n\) indices of the nonzero blocks in block row i, padded by the integer value iif the num ber ofnonzero blocks is less than M A X BN Z. The block colum \(n\) indioesM U ST be sorted in increasing order foreach block row.

BLDA leading dim ension ofB INDX (:,:).

M AXBNZ max num berofnonzerosblocks per row .
LB row and colum \(n\) dim ension of the dense blocks com posing A.

B 0 rectangular array w ith first dim ension LD B .

LD B leading dim ension of \(B\)

BETA Scalarparam eter

C 0 rectangular aray w ith first dim ension LD C .

LD C leading dim ension of C

W ORK () scratch array of length LW ORK.
On exit, ifLW ORK \(=-1, W\) ORK (1) retums the minim um
size ofLW ORK.

LW ORK length ofW ORK anay.LW ORK should be at least M B *LB.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=\mathrm{M} \mathrm{B} * \mathrm{LB} * \mathrm{~N}\) _CPU \(S\) where \(\mathrm{N} \_\)CPUS is the \(m\) axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W O RK array, and no errorm essage related to LW ORK is issued by XERBLA .

\section*{SEE ALSO}

\section*{N IST FO RTRA N Sparse B las U ser's G uide available at:} http:/m ath nist.gov/m cso/Staff/K Rem ington/Espblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib org/utk/papers/sparse _ps

\section*{NOTES /BUGS}
1.N o test for singularity ornear-singularity is included in this routine. Such tests m ust.be perform ed before calling this routine.
2. If \(D E S C R A(3)=0\), the low er or upper triangularpart of each diagonalblock is used by the routine depending on DESCRA (2) .
3. If \(D E S C R A(3)=1\), the unitdiagonalblocksm ightorm ight notbe referenced in the B EL representation of a sparse \(m\) atrix. They are notused anyw ay .
4. If \(D E S C R A(3)=2\), diagonalblocks are considered as dense m atrices and the LU factorization w ith partialpivoting is used by the routine. WORK (1)=0 on retum if the factorization for alldiagonalblocks has been com pleted successfully, otherw ise WORK (1) = -iw here i is the block num ber forw hich the LU factorization could notbe com puted.
5.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse
m atrix \(A\) is used. H ow erver DESCRA (1) m ust.be equalto 3 in this case.

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

sbscm m -block sparse colum n m atrix-m atrix m ultiply

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SBSCMM(TRANSA,M B,N,KB,ALPHA,DESCRA,}

* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,MB,N,KB,DESCRA (5),LB,}
* LDB,LDC,LW ORK
INTEGER B INDX (BNNZ),BPNTRB (KB),BPNTRE (KB)
REAL ALPHA,BETA
REAL VAL (LB*LB*BNNZ),B (LDB,*),C (LDC **),W ORK (LW ORK)
SUBROUTINE SBSCMM_64(TRANSA,M B,N,KB,ALPHA,DESCRA,
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,MB,N,KB,DESCRA (5),LB,
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 B INDX (BNNZ),BPNTRB (KB),BPNTRE (KB)}
REAL ALPHA,BETA
REAL VAL (LB*LB *BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
where: BNN Z = BPN TRE (K B )-BPNTRB (1)

```

\section*{F95 INTERFACE}

SUBROUTINE BSCMM (TRANSA, MB, \(\mathbb{N}], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\), * BPNTRB,BPNTRE, LB, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
\(\mathbb{N} T E G E R\) TRANSA, MB, KB,LB
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: D E S C R A, B \mathbb{N} D X, B P N T R B, B P N T R E\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M}\) ENSION (:) ::VAL
REAL, D \(\mathbb{I M}\) ENSION (: : : :: B, C

SUBROUT \(\mathbb{N} E \operatorname{BSCM} M \_64\) (TRANSA, MB, \(\left.\mathbb{N}\right], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\),
* BPNTRB,BPNTRE,LB,B, [LDB],BETA,C,[LDC], [WORK], [LWORK])
\(\mathbb{I N T E G E R * 8 ~ T R A N S A , ~ M B , K B , L B ~}\)
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: \quad \mathrm{DESCRA}, \mathrm{B} \mathbb{N} D X, B P N T R B, B P N T R E\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M}\) ENSION (:) ::VAL
REAL,D \(\mathbb{M}\) ENSION (: :) :: B, C

\section*{DESCRIPTION}
C <-aloha op (A ) B + beta C
where A LPHA andBETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, \(A\) is a m atrix represented in block sparse colum n form at and op (A) is one of \(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix.
2 is equivalentto 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row s in m atrix A

N \(\quad \mathrm{N}\) um berof colum ns in \(m\) atrix \(C\)

K B \(\quad\) Number ofblock colum ns in m atrix A

ALPHA Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(A=A\) )
2: Herm Itian ( \(A=C O N J(A)\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(A=-C O N J\) ( \(A\) ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 : upper

DESCRA (3) m ain diagonaltype
0 : non-unit
1 :unit
DESCRA (4) A ray base NOT \(\mathbb{M}\) PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 :unknown
1 : no repeated indices
VAL () scalar array of length \(\mathrm{LB} * \mathrm{LB} * \mathrm{BNN} Z\) consisting of the block entries stored collm n-m ajorw thin each dense block .
\(B \operatorname{IND}\) X (integer array of length BNNZ consisting of the block row indioes of the block entries ofA .

BPN TRB 0 integer aray of length \(K B\) such that BPN TRB (J) BPN TRB (1)+1 points to location in B IND X of the firstblock entry of the J-th block colum n of A.
BPNTRE ( integeramay of length \(K B\) such that BPN TRE (J) BPN TRB (1) points to location in B IN D X of the last.block entry of the J-th block colum n of A.

LB dim ension of dense blocks com posing A.
B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension of \(B\)
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension ofC
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the current version.

LW ORK length ofW ORK array. LW ORK is notreferenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U sers G uide available at:
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse .ps

\section*{NOTES/BUGS}

It is know \(n\) that there exists another representation of the block sparse colum \(n\) form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each block colum \(n\) in the arrays VAL and B INDX is used instead of tw o arrays BPN TRB and BPN TRE. To use the routine \(w\) ith this kind ofblock sparse colum \(n\) form at the follow ing calling sequence should be used

CALL SBSCMM (TRANSA, MB,N,KB,ALPHA,DESCRA, * \(\quad V A L, B \mathbb{N D}, \mathbb{A}, \mathbb{A}(2), L B\), * B,LDB,BETA, C,LDC,WORK,LWORK )

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
sbscsm -block sparse colum \(n\) form at triangular solve

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SBSCSM(TRANSA,M B ,N,UNTID,DV,A LPHA,DESCRA,}

* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,MB,N,UNITD,DESCRA (5),LB,
* LDB,LDC,LW ORK
INTEGER B\mathbb{NDX (BNNZ),BPNTRB MB),BPNTREMB)}
REAL ALPHA,BETA
REAL DV M B*LB*LB),VAL (LB*LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINE SBSCSM_64(TRANSA,M B,N,UNITD,DV,ALPHA,DESCRA,
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,MB,N,UNITD,DESCRA (5),LB,
* LDB,LDC,LW ORK
INTEGER*8 BINDX (BNNZ),BPNTRB MB),BPNTREMB)
REAL ALPHA,BETA
REAL DV M B *LB*LB),VAL (LB*LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
where: \(\operatorname{BNNZ}=\mathrm{BPNTRE} \mathrm{MB})-\mathrm{BPNTRB}\) (1)

\section*{F95 INTERFACE}

SUBROUTINE BSCSM (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,VAL,B \(\mathbb{N} D X\), * BPNTRB,BPNTRE, LB, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
\(\mathbb{N} T E G E R \quad\) TRANSA, MB,N,UNITD,LB
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: D E S C R A, B \mathbb{N} D X, B P N T R B, B P N T R E\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M}\) ENSION (:) ::VAL,DV
REAL,D IM ENSION (: :):: B,C

SUBROUTINE BSCSM_64 (TRANSA, MB,N, UNITD,DV,ALPHA,DESCRA,VAL,BINDX,
* BPNTRB,BPNTRE,LB,B, [LDB],BETA,C, [LDC], [W ORK], [LWORK])
\(\mathbb{I N T E G E R * 8}\) TRANSA, MB,N,UNITD, LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: \quad \mathrm{DESCRA}, \mathrm{B} \mathbb{N} D X, B P N T R B, B P N T R E\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M} E N S I O N(:):: V A L, D V\)
REAL,D \(\mathbb{M}\) ENSION (: :) :: B, C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { op (A)B+BETA C } \\
& C<-A L P H A \text { OP (A)D B + BETA } C
\end{aligned}
\]
where ALPHA and BETA are scalar, \(C\) and \(B\) are m by \(n\) dense \(m\) atrices, \(D\) is ablock diagonalm atrix, \(A\) is a unit, ornon-unit, upperor low er triangularm atrix represented in block sparse colum \(n\) form at and op (A) is one of op (A) \()=\operatorname{inv}(A)\) or op (A \()=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}\left(\operatorname{cong}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix 0 : operate w ith m atrix 1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(s\) in matrix \(A\)
\(N \quad N\) um berof colum \(n s\) in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity matrix (argum entD V [] is ignored)
2 : Scale on left (row block scaling)
3 : Scale on right (colum n block scaling)

DV () A rray of the length M B *LB *LB consisting of the block entries ofblock diagonalm atrix \(D\) w here each
block is stored in standard colum n-m ajor form .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem ent integer anay
DESCRA (1) m atrix structure
0 : general

1 : symmetric ( \(A=A\) )
2: Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\operatorname{CONJG}(\mathrm{A})\) )
N ote: For the routine, D ESCRA \((1)=3\) is only supported.
D ESCRA (2) upper/low er triangular indicator
1: low er
2 : upper
DESCRA (3) m ain diagonal type
0 : non-identity blocks on the \(m\) ain diagonal
1 : identily diagonalblocks
2 : diagonalblocks are densem atrices
DESCRA (4) A ray base NOT IM PLEM ENTED )
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () scalar aray of length LB *LB *BNN Z consisting of the block entries stored colum \(n\)-m ajorw thin each dense block.
\(B \mathbb{N}\) D X ( integer array of length BNN Z consisting of the block row indices of the block entries of \(A\).
The block row indicesM U ST be sorted
in increasing order foreach block colum \(n\).
BPN TRB () integer aray of length \(M B\) such that
BPN TRB (J) BPN TRB (1)+1 points to location in B IN D X of the first.block entry of the J-th block colum n of A.

BPN TRE 0 integer array of length \(M B\) such that BPN TRE (J) BPN TRB (1) points to location in B IND X of the lastblock entry of the \(J\) th block colum n of A.

LB dim ension of dense blocks com posing A.
B 0 rectangular array w th first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C.
LD C leading dim ension of \(C\)

W ORK 0 scratch array of length LW ORK. On exit, if LW ORK = \(-1, W\) ORK (1) retums the optim um size ofLW ORK.

LW ORK length ofW ORK aray. LW ORK should be at least M B*LB.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=\mathrm{M} \mathrm{B} * \mathrm{LB}\) *N_CPUS where N_CPUS is the \(m\) axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.

If LW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FORTRAN Sparse B las U sers G uide available at: http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
htyp://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. No test for singularity ornear-singularity is included in this routine. Such tests \(m\) ustbe perform ed before calling this routine.
2. If \(D E S C R A\) ( 3 )=0, the low erorupper triangularpart of each diagonalblock is used by the routine depending on DESCRA (2).
3. If \(\operatorname{DESCRA}(3)=1\), the unit diagonalblocksm ightorm ight notbe referenced in the BSC representation of a sparse \(m\) atrix. They are notused anyw ay .
4. If \(D E S C R A(3)=2\), diagonalblocks are considered as dense m atrices and the LU factorization w th partialpivoting is used by the routine. WORK (1)=0 on retum if the factorization for alldiagonalblocks has been com pleted
successfully, otherw ise wORK (1) = -iw here is the block
num ber forw hich the LU factorization could notbe com puted.
5. The routine can be applied for solving triangular system \(s\) w hen the upper or low er triangle of the general sparse m atrix A is used. H ow erver DESCRA (1) m ustbe equal to 3 in this case.
6. It is know \(n\) that there exists another representation of the block sparse colum n form at (see for exam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three anray instead of the fourused in the current im plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning ofeach block colum \(n\) in the arrays VAL and B \(\mathbb{N D}\) D is used instead of tw o arrays BPN TRB and BPN TRE. To use the routine \(w\) ith this kind ofblock sparse colum \(n\) form at the follow ing calling sequence should be used

CALL SBSCSM (TRANSA,MB,N,UNITD,DV,ALPHA,DESCRA,
* \(V A L, B \mathbb{N} D, \mathbb{A}, \mathbb{A}(2), L B\),
* B,LDB,BETA, C,LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

sbsmm m-block sparse row form atm atrix-m atrix m ultiply

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SBSRMM (TRANSA,M B,N,KB,ALPHA,DESCRA,}

* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,MB,N,KB,DESCRA (5),LB,}
* LDB,LDC,LW ORK
\mathbb{NTEGER B}\mathbb{N}DX(BNNZ),BPNTRB MB),BPNTRE MB)
REAL ALPHA,BETA
REAL VAL (LB*LB*BNNZ),B (LDB,*),C (LDC **),W ORK (LW ORK)
SUBROUTINE SBSRMM_64(TRANSA,M B,N,KB,ALPHA,DESCRA,
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,MB,N,KB,DESCRA (5),LB,
* LDB,LDC,LW ORK
INTEGER*8 BINDX (BNNZ),BPNTRB M B),BPNTREMB)
REAL ALPHA,BETA
REAL VAL (LB *LB *BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
where: BNN Z = BPNTRE M B )-BPNTRB (1)

```

\section*{F95 INTERFACE}

SUBROUTINE BSRMM (TRANSA, MB, \(\mathbb{N}], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\), * BPNTRB,BPNTRE, LB, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
\(\mathbb{N} T E G E R\) TRANSA, MB, KB,LB
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \quad \mathrm{DESCRA}, \mathrm{B} \mathbb{N} \mathrm{X}, \mathrm{BPN} T R B, B P N T R E\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M}\) ENSION (:) ::VAL
REAL, D \(\mathbb{I M}\) ENSION (: : : :: B, C

SUBROUT \(\mathbb{N} E \operatorname{BSRM} M \_64(T R A N S A, M B,[N], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\),
* BPNTRB,BPNTRE,LB,B, [LDB],BETA,C,[LDC], [WORK], [LWORK])
\(\mathbb{I N T E G E R * 8 ~ T R A N S A , ~ M B , K B , L B ~}\)
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: \quad \mathrm{DESCRA}, \mathrm{B} \mathbb{N} D X, B P N T R B, B P N T R E\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M}\) ENSION (:) ::VAL
REAL,D \(\mathbb{M}\) ENSION (: :) :: B, C

\section*{DESCRIPTION}
C <-aloha op (A ) B + beta C
where A LPHA andBETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, A is a m atrix represented in block sparse row form at and op (A ) is one of \(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRA N SA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix \(A\) is real.

M B \(\quad\) Num ber ofblock row s in m atrix A

N \(\quad \mathrm{N}\) um berof colum ns in \(m\) atrix \(C\)

K B \(\quad\) Number ofblock colum ns in m atrix A

ALPHA Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(A=A\) )
2: Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(A=-C O N J\) ( \(A\) ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper

DESCRA (3) main diagonal type
0 : non-unit
1 :unit
DESCRA (4) A ray base NOT \(\mathbb{M}\) PLEM ENTED)
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () scalar array of length LB *LB*BNNZ consisting of the block entries stored colum n-m ajorw thin each dense block .
\(B \operatorname{IND}\) X (integer array of length BNNZ consisting of the block colum n indices of the block entries of A.

BPN TRB () integeramay of length \(M B\) such that BPN TRB (J) BPN TRB (1)+1 points to location in B IND X of the first.block entry of the \(J\)-th block row of A.
BPN TRE () integer array of length \(M B\) such that BPN TRE (J)-BPN TRB (1) points to location in B IND X of the lastblock entry of the \(J\) th block row ofA.

LB dim ension of dense blocks com posing A.
B 0 rectangular array w th first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w th first dim ension LD C.
LD C leading dim ension of \(C\)
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the currentversion.

LW ORK length ofW ORK array. LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B lasU ser's G uide available at:
htep://m ath nist.gov/m csd/Staff/k Rem ington/Aspblas/
"D ocum ent forthe B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse дps

\section*{NOTES /BUGS}

It is know \(n\) that there exists another representation of the block sparse row form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem s",W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. The \(m\) ain difference is that only one array, \(\mathbb{A}\), containing the pointers to the beginning of each block row in the amays \(V A L\) and \(B \mathbb{N D X}\) is used instead of tw o arrays BPN TRB and BPN TRE. To use the routine w ith this kind ofblock sparse row form at the follow ing calling sequence should be used

CALL SBSRMM (TRANSA, MB,N,KB,ALPHA,DESCRA, * \(V A L, B \mathbb{N} D X, \mathbb{I}, \mathbb{I A}(2), L B\),
* \(\quad \mathrm{B}, \mathrm{LD} B, B E T A, C, L D C, W\) ORK,LWORK )

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
sbsrsm -block sparse row form at triangular solve

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SBSRSM (TRANSA,M B,N,UNITD,DV,ALPHA,DESCRA,}

* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,MB,N,UNITD,DESCRA (5),LB,
* LDB,LDC,LW ORK
INTEGER B\mathbb{NDX (BNNZ),BPNTRB MB),BPNTRE MB)}
REAL ALPHA,BETA
REAL DV M B*LB*LB),VAL (LB*LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NESBSRSM_64(TRANSA,M B,N,UNITD,DV,ALPHA,DESCRA,}
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,MB,N,UNITD,DESCRA (5),LB,
* LDB,LDC,LW ORK
INTEGER*8 BINDX (BNNZ),BPNTRB MB),BPNTREMB)
REAL ALPHA,BETA
REAL DV M B *LB*LB),VAL (LB*LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
where: \(\operatorname{BNN} \mathrm{Z}=\mathrm{BPN} \operatorname{TRE} \mathrm{M} \mathrm{B})\)-BPNTRB (1)

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE BSRSM (TRANSA,M B,N,UNITD,DV,ALPHA,DESCRA,VAL,BINDX,}

```
* BPNTRB,BPNTRE, LB, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
\(\mathbb{N} T E G E R \quad\) TRANSA, MB,N,UNITD,LB
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: D E S C R A, B \mathbb{N} D X, B P N T R B, B P N T R E\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M}\) ENSION (:) ::VAL,DV
REAL,D IM ENSION (: :):: B,C

SUBROUTINE BSRSM_64 (TRANSA, MB,N, UNITD,DV,ALPHA,DESCRA,VAL,BINDX,
* BPNTRB,BPNTRE,LB,B, [LDB],BETA,C, [LDC], [W ORK], [LWORK])
\(\mathbb{I N T E G E R * 8}\) TRANSA, MB,N,UNITD, LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: \quad \mathrm{DESCRA}, \mathrm{B} \mathbb{N} D X, B P N T R B, B P N T R E\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M} E N S I O N(:):: V A L, D V\)
REAL,D \(\mathbb{M}\) ENSION (: :) :: B, C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { op (A)B+BETA C } \\
& C<-A L P H A \text { OP (A)D B + BETA } C
\end{aligned}
\]
where ALPHA and BETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense \(m\) atrices, \(D\) is ablock diagonalm atrix, \(A\) is a unit, ornon-unit, upperor low ertriangularm atrix represented in block sparse row form at form atand op (A ) is one of \(\operatorname{op}(A)=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}(A)\) or op (A) \()=\operatorname{inv}\left(\operatorname{conjg}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix 0 : operate \(w\) ith \(m\) atrix 1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(s\) in \(m\) atrix \(A\)
\(N \quad N\) um berof colum \(n s\) in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 : Scale on left (row block scaling)
3 : Scale on right (colum n block scaling)

DV () A rray of the length M B *LB *LB consisting of the block entries ofblock diagonalm atrix \(D\) w here each
block is stored in standard colum n-m ajor form .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem ent integer anay
DESCRA (1) m atrix structure
0 : general

1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2: Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\operatorname{CONJG}(\mathrm{A})\) )
N ote: For the routine, D ESCRA \((1)=3\) is only supported.
D ESCRA (2) upper/low er triangular indicator
1: low er
2 : upper
DESCRA (3) m ain diagonal type
0 : non-identity blocks on the \(m\) ain diagonal
1 : identily diagonalblocks
2 : diagonalblocks are densem atrices
DESCRA (4) A ray base NOT IM PLEM ENTED )
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () scalar aray of length LB *LB *BNN Z consisting of the block entries stored colum \(n\)-m ajorw thin each dense block.
\(B \operatorname{IN}\) X ( integer array of length BNNZ consisting of the block colum n indices of the block entries of A. The block colum n indices M U ST be sorted in increasing order foreach block row .

BPN TRB () integer array of length \(M B\) such that BPN TRB (J) BPN TRB (1)+1 points to location in B IND X of the firstblock entry of the J-th block row of A.

BPN TRE ( integer array of length \(M B\) such that BPN TRE (J)-BPN TRB (1) points to location in B IND X of the last.block entry of the J-th block row of A.

LB dim ension of dense blocks com posing A.
B 0 rectangular array w th first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C.
LD C leading dim ension of \(C\)

W ORK 0 scratch array of length LW ORK. On exit, if LW ORK \(=-1, W\) ORK (1) retums the optim um size ofLW ORK.

LW ORK length ofW ORK array. LW ORK should be at least M B*LB.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=\mathrm{M} \mathrm{B} * \mathrm{LB} * \mathrm{~N}\) _CPU \(S\) where \(N\) _CPUS is the \(m\) axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FORTRAN Sparse B las U sers G uide available at: http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
htyp://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. No test for singularity ornear-singularity is included in this routine. Such tests \(m\) ustbe perform ed before calling this routine.
2. If \(D E S C R A\) ( 3 )=0, the low erorupper triangularpart of each diagonalblock is used by the routine depending on DESCRA (2).
3. If \(\operatorname{DESCRA}\) (3)=1, the unitdiagonalblocksm ightorm ight not.be referenced in the BSR representation of a sparse \(m\) atrix. They are notused anyw ay .
4. If \(D E S C R A(3)=2\), diagonalblocks are considered as dense m atrices and the LU factorization w th partialpivoting is used by the routine. WORK (1)=0 on retum if the factorization for alldiagonalblocks has been com pleted
successfully, otherw ise WORK (1) = -iw here i is the block
num ber forw hich the LU factorization could notbe com puted.
5. The routine can be applied for solving triangular system \(s\) w hen the upper or low er triangle of the general sparse m atrix A is used. H ow erver DESCRA (1) m ustbe equal to 3 in this case.
2. It is know \(n\) that there exists another representation of the block sparse row form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem S", W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each block row in the amays VAL and B \(\mathbb{N D}\) D is used instead of tw o arrays BPN TRB and BPN TRE. To use the routine w ith this kind ofblock sparse row form at the follow ing calling sequence should be used

CALL SBSRSM (TRANSA,MB,N,UNITD,DV,ALPHA,DESCRA,
* \(V A L, B \mathbb{N} D, \mathbb{A}, \mathbb{A}(2), L B\),
* B,LDB,BETA, C,LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
scasum -Retum the sum of the absolute values of a vector x .

\section*{SYNOPSIS}

REALFUNCTION SCASUM \(\mathbb{N}, \mathrm{X}, \mathbb{N} C X)\)
COM PLEX X \({ }^{(*)}\)
\(\mathbb{N} T E G E R \mathbb{N}, \mathbb{N} C X\)

REALFUNCTION SCASUM _64 \(\mathbb{N}, \mathrm{X}, \mathbb{N} C X)\)
COM PLEX X \({ }^{(*)}\)
\(\mathbb{N}\) TEGER*8 \(\mathrm{N}, \mathbb{N} C X\)

\section*{F95 INTERFACE}

REAL FUNCTION ASUM ( \(\mathbb{N}\) ],X, \([\mathbb{N} C X]\) )
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::X
\(\mathbb{N} T E G E R:: N, \mathbb{N C X}\)
REAL FUNCTION ASUM _64 ( \(\mathbb{N}\) ], \(X,[\mathbb{N} C X])\)

COM PLEX,D \(\mathbb{M}\) ENSION (:) ::X
\(\mathbb{N} T E G E R(8):: N, \mathbb{N} C X\)

\section*{C INTERFACE}
\#include < sunperfh>
float scasum (intn, com plex *x, int incx);
floatscasum _64 (long n, com plex *x, long incx);

\section*{PURPOSE}
scasum Retum the sum of the absolute values of the elem ents of \(x\) where \(x\) is an \(n\)-vector. This is the sum of the absolute values of the real and com plex elem ents and not the sum of the squares of the realand com plex elem ents.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N m ustbe at leastone for the subroutine to have any visible effect. U nchanged on exit.
\(X\) (input)
( \(1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C X)\) ). On entry, the increm ented array \(X\) must contain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
scnm 2 -Retum the Euclidian norm of a vector.

\section*{SYNOPSIS}
```

REALFUNCTION SCNRM 2N,X,\mathbb{NCX)}
COM PLEX X (*)
\mathbb{NTEGERN,INCX}
REALFUNCTION SCNRM 2_64 N,X,INCX)
COM PLEX X (*)
INTEGER *8 N, INCX

```
F95 INTERFACE
    REAL FUNCTION NRM 2 ( \(\mathbb{N}\) ], X, [ \(\mathbb{N C X}]\) )
    COMPLEX,D \(\mathbb{M}\) ENSION (:) ::X
    \(\mathbb{N} T E G E R:: N, \mathbb{N C X}\)
    REAL FUNCTION NRM 2_64 ( \(\mathbb{N}\) ], X, [ \(\mathbb{N} C X]\) )
    COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::X
    \(\mathbb{N}\) TEGER (8) ::N, \(\mathbb{N} C X\)
C INTERFACE
    \#include <sunperfh>
    float scnm 2 (intn, com plex *x, intincx);
    float scnim 2_64 (long n, com plex *x, long incx);

\section*{PURPOSE}
scnm 2 R etum the Euclidian norm of a vector \(x\) w here \(x\) is an n-vector.

\section*{ARGUMENTS}

N (input)
O \(n\) entry, \(N\) specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.
X (input)
(1+(n-1)*abs( \(\mathbb{N} C X)\) ). On entry, the increm ented array \(X\) m ust contain the vectorx. U nchanged on exit.
\(\mathbb{N} C X\) (input)
On entry, \(\mathbb{I N C X}\) specifies the increm ent for the elem ents ofX. \(\mathbb{N}\) CX m ustbe positive. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- NOTES

\section*{NAME}
scnvcor-com pute the convolution or comelation of real vectors

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SCNVCOR (CNVCOR,FOUR,NX,X,\mathbb{FX,}\mathbb{N}CX,NY,NPRE,M,Y,}
\mathbb{F}Y,\mathbb{NC}C1Y,\mathbb{N}C2Y,NZ,K,Z,\mathbb{FZ},\mathbb{NC}C1Z,\mathbb{NC}2Z,W ORK,LW ORK)
CHARACTER * 1 CNVCOR,FOUR

```

```

K,\mathbb{FZ,}\mathbb{NC1Z,INC2Z,LW ORK}
REALX (*),Y (*),Z (*),W ORK (*)
SUBROUT\mathbb{NE SCNVCOR_64 (CNVCOR,FOUR,NX,X,\mathbb{FX,INCX,NY,NPRE,M,Y,}}\mathbf{N},\textrm{N},

```

```

CHARACTER * 1 CNVCOR,FOUR
\mathbb{NTEGER*8NX,\mathbb{FX,}\mathbb{NCX,NY,NPRE,M, FYY, NNC1Y, \mathbb{NC}2Y,NZ,}}\mathbf{N}=,

```

```

REALX (*),Y (*),Z (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE CNVCOR (CNVCOR,FOUR, \(\mathbb{N} X], X, \mathbb{F X},[\mathbb{N C X}], N Y, N P R E, M, Y\), \(\mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N} C 2 Y, N Z, K, Z, \mathbb{F} Z, \mathbb{N C} 1 Z, \mathbb{N} C 2 Z, W\) ORK, (LW ORK ])

CHARACTER (LEN=1): :CNVCOR,FOUR
\(\mathbb{N} T E G E R:: N X, \mathbb{F} X, \mathbb{N} C X, N Y, N P R E, M, \mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N} C 2 Y\),
\(\mathrm{NZ}, \mathrm{K}, \mathbb{F} Z, \mathbb{N} C 1 Z, \mathbb{N} C 2 Z, L W\) ORK
REAL,D \(\mathbb{M}\) ENSION (:) :: X,Y,Z,W ORK

SU BROUTINE CNVCOR_64 (CNVCOR,FOUR, \(\mathbb{N} X], X, \mathbb{F X},[\mathbb{N} C X], N Y, N P R E, M\),

CHARACTER (LEN=1) ::CNVCOR,FOUR
\(\mathbb{N} T E G E R(8):: N X, \mathbb{F X}, \mathbb{N} C X, N Y, N P R E, M, \mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N C} 2 Y\), \(\mathrm{NZ}, \mathrm{K}, \mathbb{F} Z, \mathbb{N C} 1 \mathrm{Z}, \mathbb{N} \mathrm{C} 2 \mathrm{Z}, \mathrm{LW}\) ORK
REAL,D IM ENSION (:) :: X,Y,Z,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void scnvcor(char cnvcor, char four, intnx, float *x, int ifx, intincx, intny, intnpre, intm, float * \(y\), intify, int incly, int inc2y, int \(n z\), int \(k\), float \({ }^{2}\), int ify, intinc1z, intinc2z, float *w ork, intlw ork);
void scnvcor_64 (charcnvcor, char four, long nx, float *x, long ifx, long incx, long ny, long npre, long m, float *y, long ify, long inc1y, long inc2y, long nz, long k, float *z, long ifz, long inclz, long inc \(2 z\), float *w ork, long lw ork);

\section*{PURPOSE}
scnvcorcom putes the convolution or comelation of realvectors.

\section*{ARGUMENTS}

CNVCOR (input)
\(V\) 'or \(V^{\prime}\) if convolution is desired, \(R\) ' or ' \(r\) ' if comelation is desired.

FOUR (input)
T'or t'if the Fourier transform \(m\) ethod is to be used, D 'or d'ifthe com putation should be done directly from the definition. The Founier transform m ethod is generally faster, but itm ay introduce noticeable errors into certain results, notably w hen both the filter and data vectors consistentirely of integers or vectors where ele\(m\) ents of either the filtervector or a given data vectordiffer significantly in \(m\) agnitude from the 1 -norm of the vector.

NX (input)
Length of the filtervector. \(\mathrm{NX}>=0\). SCNVCOR w ill retum im m ediately if \(\mathrm{N}=0\).

X (input)
Filtervector.

IFX (input)
Index of the firstelem entofX. \(\mathrm{NX}>=\mathbb{F} \mathrm{X}>=1\).
\(\mathbb{N} C X\) (input)
Stride betw een elem ents of the filtervector in X . \(\mathbb{N} C X>0\).

NY (input)
Length of the inputvectors. NY >= 0. SCNVCOR w ill retum im m ediately if \(\mathrm{N} Y=0\).
NPRE (input)
The num ber of im plicit zeros prepended to the \(Y\) vectors. NPRE \(>=0\).

M (input)
N um berof inputvectors. \(\mathrm{M}>=0\). SCNVCOR will retum imm ediately if \(M=0\).

Y (input)
Inputvectors.

FY (input)
Index of the firstelem entof \(. \mathrm{NY}>=\mathbb{F} Y>=1\).
\(\mathbb{N} C 1 Y\) (input)
Stride betw een elem ents of the inputvectors in Y.
\(\mathbb{I N C} 1 \mathrm{Y}>0\).
\(\mathbb{N} C 2 Y\) (input)
Stride betw een the inputvectors in Y. \(\mathbb{N} C 2 Y>0\).

N Z (input)
Length of the outputvectors. NZ \(>=0\). SCNVCOR
\(w\) ill retum im m ediately if \(\mathrm{NZ}=0\). See the N otes section below for inform ation abouthow this argu\(m\) ent interacts \(w\) ith \(N X\) and \(N Y\) to control circular versus end-off shifting.

K (input)
N um berof Z vectors. \(\mathrm{K}>=0\). If \(\mathrm{K}=0\) then SCNVCOR will retum immediately. If \(K<M\) then only the firstK inputvectors w ill.be processed. If \(K>M\) then \(M\) inputvectors \(w\) ill.be processed.

Z (output)
Resultvectors.

FZ (input)
Index of the firstelem entofZ. NZ >= \(\mathbb{F Z}\) >=1.
\(\mathbb{N} C 1 Z\) (input)
Stride betw een elem ents of the output vectors in Z. \(\mathbb{N} C 1 Z>0\).
\(\mathbb{N} C 2 Z\) (input)
Stride betw een the outputvectors in Z. \(\mathbb{N N}\) C 2 Z > 0 .

W ORK (input/output)
Scratch space. Before the first call to SCNVCOR w ith particular values of the integer argum ents the firstelem entofW ORK m ustbe set to zero. If W ORK is w ritten between calls to SCNVCOR or if SCNVCOR is called w th different values of the integer argum ents then the firstelem entofW ORK m ustagain be set to zero before each call. If W ORK has notbeen w rilten and the sam e values of the integer argum ents are used then the firstele\(m\) entofW ORK to zero. This can avoid certain initializations that store their results into \(W\) ORK, and avoiding the intialization can \(m\) ake SCN V COR nun faster.

LW ORK (input)
Length ofW ORK. LW ORK >= 4*m ax NX,NPRE+NY,NZ)+15.

\section*{NOTES}

If any vector overlaps a w ritable vector, eitherbecause of argum ent aliasing or ill-chosen values of the various \(\mathbb{I N} C\) argum ents, the results are undefined and \(m\) ay vary from one nun to the next.

Them ost com \(m\) on form of the com putation, and the case that executes fastest, is applying a filter vectorX to a series of vectors stored in the colum ns of \(Y\) w ith the resultplaced into the colum nsof \(Z\). In that case, \(\mathbb{N} C X=1, \mathbb{N} C 1 Y=1\), \(\mathbb{N} C 2 Y>=N Y, \mathbb{N} C 1 Z=1, \mathbb{N} C 2 Z>=N Z\). A nothercomm on form is applying a filtervectorX to a series of vectors stored in the row sof \(Y\) and store the result in the row of \(Z\), in which case \(\mathbb{N} C X=1, \mathbb{N} C 1 Y>=N Y, \mathbb{N} C 2 Y=1, \mathbb{N} C 1 Z>=N Z\), and \(\mathbb{N} C 2 Z=1\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
scnvcor2 -com pute the convolution or comelation of real m atrices

\section*{SYNOPSIS}
```

SUBROUTINE SCNVCOR2 (CNVCOR,METHOD,TRANSX,SCRATCHX,TRANSY,
SCRATCHY,M X,NX,X,LDX,M Y,NY,MPRE,NPRE,Y,LDY,M Z,NZ,Z,
LD Z,W ORK NN,LW ORK)
CHARACTER * 1 CNVCOR,METHOD, TRANSX, SCRATCHX, TRANSY,
SCRATCHY
COM PLEX W ORK\mathbb{N (*)}
\mathbb{N TEGER M X,NX,LDX,M Y,NY,M PRE,NPRE,LDY,M Z, NZ, LD Z,}
LW ORK
REALX (LDX ,*),Y (LDY ,*),Z (LD Z,*)
SU BROUTINE SCNVCOR2_64 (CNVCOR,M ETHOD,TRANSX,SCRATCHX,TRANSY,
SCRATCHY,M X,NX,X,LDX,MY,NY,M PRE,NPRE,Y,LDY,M Z,NZ,Z,
LD Z,W ORK\mathbb{N,LW ORK)}
CHARACTER * 1 CNVCOR,METHOD,TRANSX, SCRATCHX, TRANSY,
SCRATCHY
COM PLEX W ORK\mathbb{N (*)}
\mathbb{NTEGER*8MX,NX,LDX,MY,NY,M PRE,NPRE,LDY,M Z N Z,LD Z,}
LW ORK
REAL X (LDX,*),Y (LDY,*),Z (LD Z ,*)

```

\section*{F95 INTERFACE}

SU BROUTINE CNVCOR2 (CNVCOR,METHOD,TRANSX,SCRATCHX,TRANSY, SCRATCHY, \(\mathbb{M} X], \mathbb{N} X], X,[L D X], \mathbb{M} Y], \mathbb{N} Y], M P R E, N P R E, Y,[L D Y]\), \(\mathbb{M} Z], \mathbb{N} Z], Z,[L D Z], W\) ORK \(\mathbb{N},[L W\) ORK])

CHARACTER (LEN=1) :: CNVCOR, METHOD, TRANSX, SCRATCHX,
TRANSY,SCRATCHY
COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::W ORK \(\mathbb{N}\)
\(\mathbb{N} T E G E R:: M X, N X, L D X, M Y, N Y, M P R E, N P R E, L D Y, M Z, N Z\),
LD Z,LW ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::X,Y,Z

SUBROUTINE CNVCOR2_64 CNVCOR,METHOD,TRANSX,SCRATCHX,TRANSY, SCRATCHY, MX], NX],X, [LDX], MY], \(\mathbb{N} Y\) ],MPRE,NPRE, Y, [LDY], \(\mathbb{M} Z], \mathbb{N} Z], Z,[L D Z], W\) ORK \(\mathbb{N},[L W O R K])\)

CHARACTER (LEN=1) :: CNVCOR, METHOD, TRANSX, SCRATCHX,
TRANSY,SCRATCHY
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::W ORK \(\mathbb{N}\)
\(\mathbb{N}\) TEGER (8) ::M X,NX,LDX,MY,NY,M PRE,NPRE,LDY,M Z, NZ,
LD Z,LW ORK
REAL,D \(\mathbb{M}\) ENSION (: : : : : X, Y,Z

\section*{C INTERFACE}
\#include <sunperfh>
void scnvcor2 (char cnvcor, charm ethod, char transx, char scratchx, char transy, char scratchy, intm \(x\), int \(n x\), float * \(x\), int ldx, intm \(y\), intny, int \(m\) pre, int npre, float * \(y\), int ldy, intm \(z\), intnz, float *z, int ldz, com plex *w orkin, int lw ork);
void scnvcor2_64 (char cnvcor, charm ethod, chartransx, char scratchx, char transy, char scratchy, long \(m \mathrm{x}\), long \(n x\), float *x, long ldx, long my, long ny, long \(m\) pre, long npre, float *y, long ldy, long m z, long nz, float *z, long ldz, com plex *w orkin, long lw ork);

\section*{PURPOSE}
scnvcor2 com putes the convolution or correlation of real \(m\) atrices.

\section*{ARGUMENTS}

CNVCOR (input)
V 'or f'to com pute convolution, R 'or \(\mathrm{I}^{\prime}\) to com pute comelation.

METHOD (input)
T'or t'if the Fourier transform \(m\) ethod is to be used, D 'or d'to com pute directly from the
definition.

TRANSX (input)
\(N\) 'or h'ifX is the filterm atrix, \(T\) ' or \(t^{\prime}\) if transpose \((X)\) is the filterm atrix.

SCRATCHX (input)
N 'or h 'ifX m ustbe preserved, S 'or S 'ifX can be used as scratch space. The contents ofX are undefined after retuming from a call in which \(X\) is allow ed to be used for scratch.

TRANSY (input)
N 'or h'ify is the inputm atrix, \(T\) 'or \(\mathrm{t}^{\prime}\) if transpose \((Y)\) is the inputm atrix.

\section*{SCRATCHY (input)}

N 'or h'ifY mustbe preserved, S'or S'ifY can be used as scratch space. The contents of \(Y\) are undefined after retuming from a call in which \(Y\) is allow ed to be used for scratch.

M X (input)
N um ber of row s in the filterm atrix. \(\mathrm{M} \mathrm{X}>=0\).

NX (input)
N um ber of Colum ns in the filterm atrix. \(\mathrm{NX}>=0\).

X (input) dim ension (LD X ,N X )
O \(n\) entry, the filterm atrix. Unchanged on exit if SCRATCHX is N' or h', undefined on exitif SCRATCHX is S'or \({ }^{\prime}\) '.

LD X (input)
Leading dim ension of the array that contains the filterm atrix.

M Y (input)
N um ber of row s in the inputm atrix. \(\mathrm{M} \mathrm{Y}>=0\).

NY (input)
\(N\) um ber of colum ns in the inputm atrix. \(N Y>=0\).

M PRE (input)
N um ber of im plicit zeros to prepend to each row of the inputm atrix. M PRE \(>=0\).

NPRE (input)
\(N\) um ber of im plicit zeros to prepend to each colum n of the inputm atrix. NPRE \(>=0\).

Y (input) dim ension (LD Y ,*)
Inputm atrix. U nchanged on exit if SCRATCHY is

or \(\mathrm{s}^{\prime}\).

LD Y (input)
Leading dim ension of the aray that contains the inputm atrix.
M Z (input)
N um ber of row s in the output m atrix. \(\mathrm{M} \mathrm{Z} \mathrm{>=0} 0\).
SCNVCOR2 w ill retum im m ediately ifM Z \(=0\).

N Z (input)
N um ber of colum ns in the outputm atrix. \(\mathrm{NZ}>=0\). SCN V COR 2 w ill retum im m ediately if \(\mathrm{Z} Z=0\).

Z (output)
dim ension (LD Z,*)
Resultm atrix.

LD Z (input)
Leading dim ension of the array that contains the resultm atrix. LD Z >= M AX (1, M Z).

W ORK \(\mathbb{N}\) (input/output)
(input/scratch) dim ension (LW ORK)
On entry for the first call to SCNVCOR2, REAL (WORK \(\mathbb{N}\) (1)) m ustcontain 0.0. A fter the first call, REAL (W ORK \(\mathbb{N}\) (1)) m ustbe set to 0.0 iff \(W\) ORK \(\mathbb{N}\) has been altered since the last call to this subroutine or if the sizes of the arays have changed.

LW ORK (input)
Length of the w ork vector. The upperbound of the w orkspace length requirem ent is 2 * \(M Y C+N Y C)+\) 15, where \(M Y C=M A X(M A X(M X, N X)\), \(M A X(M Y, N Y)+N P R E)\) and \(N Y C=M A X(M A X(M X, N X), M A X(M Y, N Y)+M P R E)\). If LW ORK indicates a w orkspace that is too sm all, the routine will allocate its ow n w ork.space. If the
FFT is notused, the value of LW ORK is unim portant.

\section*{Contents}
- NAME
- SYNOPSIS

\title{
- F95 INTERFACE
}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

scoomm -coordinatem atrix-m atrix m ultiply

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SCOOMM(TRANSA,M,N,K,ALPHA,DESCRA,}

* VAL,INDX,JNDX,NNZ,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,M,N,K,DESCRA (5),NNZ
* LDB,LDC,LW ORK
INTEGER INDX NNZ),JNDX NNZ)
REAL ALPHA,BETA
REAL VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINE SCOOMM_64(TRANSA,M,N,K,ALPHA,DESCRA,
* VAL,\mathbb{NDX,JNDX,NNZ,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,M,N,K,DESCRA (5),NNZ
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 INDX NNZ),NNDX NNZ)}
REAL ALPHA,BETA
REAL VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NECOOMM(TRANSA,M, N ],K,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,NNDX,NNZ,B,[LDB],BETA,C,[LDC],}
* [W ORK], [LW ORK])
INTEGER TRANSA,M,K,NNZ

```

```

REAL ALPHA,BETA
REAL,D IM ENSION (:) ::VAL
REAL,D IM ENSION (:, :):: B,C

```

SUBROUTINECOOMM_64(TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A\),
* \(V A L, \mathbb{N D} X, \operatorname{JNDX}, N N Z, B,[L D B], B E T A, C,[L D C]\), * [W ORK], [LW ORK])
\(\mathbb{I N T E G E R *}\) TRANSA, M, K, NNZ
\(\mathbb{N} T E G E R * 8, D \mathbb{I M} E N S \mathbb{I O N}(:):: D E S C R A, \mathbb{N} D X, J N D X\)
REAL ALPHA,BETA
REAL,D \(\mathbb{M}\) ENSION (:) ::VAL
REAL,D IM ENSION (: : : :: B,C

\section*{DESCRIPTION}
C <-alpha op (A) B + beta C
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrioes, \(A\) is a \(m\) atrix represented in coordinate form at and op(A) is one of \(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{con} \dot{g}\left(A^{\prime}\right)\).
( 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRA N SA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalentto 1 ifm atrix is real.

M \(\quad N\) um ber of row \(s\) in \(m\) atrix A

N \(\quad \mathrm{N}\) um berof colum ns in \(m\) atrix \(C\)

K \(\quad \mathrm{N}\) um berof colum \(n\) in in atrix \(A\)

ALPHA Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
\(2: \mathrm{Herm}\) itian \((\mathbb{A}=\mathrm{CONJG}(\mathbb{A}))\)
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) )
5 :D iagonal
6 : Skew Herm itian ( \(A=-C O N J(A)\) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper

DESCRA (3) main diagonal type
0 : non-unit
1 : unit
DESCRA (4) A ray base NOT IM PLEM ENTED)
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 :no repeated indices
VAL () scalar array of length NNZ consisting of the non-zero entries of \(A\), in any order.
\(\mathbb{I N D X}\) () integer array of length NNZ consisting of the comesponding row indices of the entries of A.

JND X () integer amray of length NNZ consisting of the corresponding colum \(n\) indioes of the entries of A.

NN Z number of non-zero elem ents in A.
B 0 rectangular array w th first dim ension LD B.
LD B leading din ension ofB

BETA Scalarparam eter
C 0 rectangular anray with firstdim ension LD C.

LD C leading dim ension of \(C\)
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is notreferenced in the current version.

\section*{SEE ALSO}

N IST FORTRA N Sparse B las U sers G uide available at:
http://m ath nistgov/m csd/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib.org/utk/papers/sparse.ps

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

scopy -C opy x to y

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SCOPY (N,X,\mathbb{NCX,Y,INCY)}}\mathbf{N},\mp@code{N}
\mathbb{NTEGER N, INCX,INCY}
REALX (*),Y (*)

```


```

REALX (*),Y (*)

```
F95 INTERFACE
    SU BROUTINE COPY ( \(\mathbb{N}], X,[\mathbb{N C X}], Y,[\mathbb{N} C Y])\)
    \(\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y\)
    REAL,D IM ENSION (:) :: X,Y
    SU BROUTINE COPY_64 (N ],X, [ \(\mathbb{N} C X], Y,[\mathbb{N} C Y])\)
    \(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{N} C X, \mathbb{N} C Y\)
    REAL,D \(\mathbb{I M}\) ENSION (:) :: X,Y
C INTERFACE
    \#include <sunperfh>
    void scopy (intn, float * \(x\), int incx, float * \(y\), int incy);
    void scopy_64 long \(n\), float *x, long incx, float *y, long
        incy);

\section*{PURPOSE}
scopy \(C\) opy \(x\) to \(y\) where \(x\) and \(y\) are \(n\)-vectors.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.
X (input)
\((1+(n-1) \star \operatorname{abs}(\mathbb{N} C X))\). Before entry, the increm ented array \(X\) m ust contain the vectorx. U nchanged on exit.
\(\mathbb{N} C X\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{I N C X ~ m ~ u s t n o t b e ~ z e r o . ~ U ~ n c h a n g e d ~}\) on exit.

Y (output)
( \(1+(m-1) * a b s(\mathbb{N} C Y)\) ). On entry, the increm ented array \(Y\) m ust contain the vectory. On exit, \(Y\) is overw rilten by the vectorx.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y\). \(\mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS

> - F95 INTERFACE
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}

\section*{scscm m -com pressed sparse colum n form atm atrix-m atrix} m ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SCSCMM (TRANSA,M,N,K,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,M,N,K,DESCRA (5),
* LDB,LDC,LWORK
\mathbb{NTEGER }\mathbb{N}DX (NNZ),PNTRB(K),PNTRE(K)
REAL ALPHA,BETA
REAL VAL NNZ),B(LDB,*),C (LDC,*),W ORK([WOORK)
SUBROUT\mathbb{NE SCSCM M _64(TRANSA,M ,N,K,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,M,N,K,DESCRA (5),}
* LDB,LDC,LWORK
\mathbb{NTEGER*8 \mathbb{NDX (NNZ),PNTRB(K),PNTRE(K)}}\mathbf{(K)}\mathrm{ (K)}
REAL ALPHA,BETA
REAL VAL NNZ),B (LDB,*),C (LDC ,*),W ORK (LW ORK)
w here N N Z = PN TRE (K )PN TRB (1)

```

\section*{F95 INTERFACE}

SUBROUTINECSCMM (TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N} D X\), * PNTRB,PNTRE, B, [LDB],BETA, C, [LDC], [WORK], [LWORK])
\(\mathbb{N}\) TEGER TRANSA, M, K
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \quad\) ESCRA, \(\mathbb{N} D X, P N T R B, P N T R E\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M}\) ENSION (:) ::VAL

REAL,D \(\mathbb{I M}\) ENSION (:, :) :: B , C

SUBROUTINECSCMM_64 (TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N D X}\),

\(\mathbb{N}\) TEGER*8 TRANSA, M, K
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{I} N(:):: \operatorname{DESCRA}, \mathbb{N} D X, P N T R B, P N T R E\)
REAL ALPHA,BETA
REAL,D \(\mathbb{M} E N S I O N(:):: V A L\)
REAL,D \(\operatorname{IM}\) ENSION (:, :) :: B , C

\section*{DESCRIPTION}
\[
C<- \text { alpha op (A) B + beta C }
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, \(A\) is a m atrix represented in com pressed sparse colum \(n\) form at and \(o p(A)\) is one of \(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix. 2 is equivalentto 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in matrix A

N \(\quad\) um berof colum ns in matrix C

K \(\quad \mathrm{N}\) um berof colum ns in \(m\) atrix \(A\)

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem ent integer anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm itian ( \(A=C O N J G(A)\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJ}\) ( A ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er

2 :upper
DESCRA (3) main diagonal type
0 :non-unit
1 : unit
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED)
0 :C C ++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM EN TED )
0 : unknown
1 : no repeated indices

VAL () scalar array of length NN Z consisting of nonzero entries ofA.

IND X \(0 \quad\) integer array of length NN Z consisting of the row indices of nonzero entries ofA .

PN TRB 0 integer amray of length \(K\) such thatPN TRB (J)-PN TRB (1)+1 points to location in VA L of the firstnonzero elem ent in colum n J .
PN TRE 0 integer array of length \(K\) such thatPN TRE (J)-PN TRB (1) points to location in V A L of the lastnonzero elem ent in colum n J .

B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension ofB

BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C.
LD C leading dim ension of \(C\)

W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the currentversion.

LW ORK length ofW ORK array.LW ORK is notreferenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov fin csd/Staffk Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee,
http://w w w netlib .org/utk/papers/sparse .ps

\section*{NOTES/BUGS}

It is know \(n\) that there exists another representation of the com pressed sparse colum n form at (see forexam ple Y Saad, "IterativeM ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each colum \(n\) in the arrays VA L and \(\mathbb{N} D \mathrm{X}\) is used instead oftw o arraysPN TRB and PN TRE.To use the routine \(w\) th this kind of sparse colum \(n\) form at the follow ing calling sequence should be used

SUBROUTINE SCSCMM (TRANSA, M,N,K,ALPHA,DESCRA,
* \(V A L, \mathbb{N} D X, \mathbb{A}, \mathbb{I}(2), B, L D B, B E T A\),
* C,LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

scscsm -com pressed sparse colum n form at triangular solve

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NESCSCSM(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,M,N,UNITD,DESCRA (5),
* LDB,LDC,LWORK
\mathbb{NTEGER INDX NNZ),PNTRB(M),PNTREM)}
REAL ALPHA,BETA
REAL DV M),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NE SCSCSM _64(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,M,N,UNITD,DESCRA (5),}
* LDB,LDC,LW ORK
INTEGER*8 \mathbb{NDX (NNZ),PNTRB M),PNTRE M)}
REAL ALPHA,BETA
REAL DV M),VAL (NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
where NN Z = PN TRE M )PNTRB (1)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE CSCSM (TRANSA,M, N ],UNITD,DV,ALPHA,DESCRA,VAL, INDX,}

```
* PNTRB,PNTRE, B, [LDB],BETA,C, [LDC],[WORK],[LWORK])
\(\mathbb{N}\) TEGER TRANSA, M, UNITD
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \operatorname{DESCRA}, \mathbb{N} D \mathrm{X}, \mathrm{PN} T R B, \operatorname{PNTRE}\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M}\) ENSION (:) ::VAL,DV
REAL,D IM ENSION (:, :):: B,C

SUBROUTINECSCSM_64 (TRANSA, M, \(\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X\),
* PNTRB,PNTRE, B, [LDB],BETA, C , [LDC], [W ORK], [LW ORK])
\(\mathbb{N}\) TEGER*8TRANSA, M , UN ITD
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):\) DESCRA, \(\mathbb{N} D X\), PNTRB, PNTRE
REAL ALPHA,BETA
REAL, D \(\mathbb{M}\) ENSION (:) ::VAL, DV
REAL,D \(\mathbb{M} E N S I O N(:,:): B, C\)

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { op }(A) B+B E T A C \\
& C<-A L P H A \text { OP }(A) D B+B E T A C
\end{aligned}
\]
where ALPHA and BETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense \(m\) atrices, \(D\) is a diagonalscaling \(m\) atrix, \(A\) is a unit, ornon-unit, upper or low er triangularm atrix represented in com pressed sparse colum n form atand op (A ) is one of op (A) \()=\operatorname{inv}(A)\) or op (A \()=\operatorname{inv}(A)\) or op (A) \()=\operatorname{inv}\left(\operatorname{con} \dot{g}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix 0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um ber of row \(s\) in \(m\) atrix \(A\)

N \(\quad\) umberof colum \(n s\) in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)
4 :A utom atic colum n scaling (see section N OTES for furtherdetails)

DV () A rray of length M containing the diagonalentries of the scaling diagonalm atrix D.

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay DESCRA (1) m atrix structure
\[
\begin{aligned}
& 0 \text { : general } \\
& 1 \text { : sym m etric ( } A=A \text { ) } \\
& 2: \text { H erm itian ( } A=\operatorname{CON} \text { JG (A }) \text { ) } \\
& 3 \text { : Triangular } \\
& 4 \text { : Skew (A nti)-Sym m etric ( } A=-A \text { ) } \\
& 5 \text { : D iagonal } \\
& 6 \text { : Skew Herm itian ( } A=-\operatorname{CON} \text { JG (A ) ) }
\end{aligned}
\]

N ote: For the routine, D ESCRA (1)=3 is only supported.

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonaltype
0 : non-unit
1 :unit
DESCRA (4) A may base \(\mathbb{N O T} \mathbb{M}\) PLEM ENTED)
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? \(N\) OT \(\mathbb{M}\) PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () scalar array of length N N Z consisting of nonzero entries ofA.
\(\mathbb{N} D \mathrm{X}\) () integer array of length N N Z consisting of the row indices of nonzero entries of . (R ow indigesM UST be sorted in increasing order for each colum n).

PNTRB () integer amay of length \(M\) such thatPN TRB (J) PN TRB (1)+1 points to location in VA L of the firstnonzero elem ent in colum \(n \mathrm{~J}\).

PN TRE () integer array of length \(M\) such thatPN TRE (J)-PN TRB (1) points to location in VA L of the lastnonzero elem ent in colum \(n \mathrm{~J}\).

B 0 rectangular array w ith first dim ension LD B .

LD B leading dim ension ofB

BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch array of length LW ORK.

On exit, if LW ORK \(=-1, W\) ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK array. LW ORK should be at leastM .
Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=M\) *N_CPUS where N_CPUS is the m axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
htep://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such tests \(m\) ust.be perform ed before calling this routine.
2. If UN ITD \(=4\), the routine scales the colum ns of A such that their 2 -norm s are one. The scaling \(m\) ay im prove the accuracy of the com puted solution. C orresponding entries of VA L are changed only in the particular case. On retum D V \(m\) atrix stored as a vector contains the diagonalm atrix by which the colum ns have been scaled. UN ITD = 3 should be used for the next calls to the routine \(w\) ith overw ritten VA L and DV.

WORK (1)=0 on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here i is the colum n num berw hich 2 -norm is exactly zero.
3. If \(D E S C R A(3)=1\) and \(U N\) ITD < 4, the unitdiagonalelem ents m ightorm ightnotbe referenced in the C SC representation
of a sparse \(m\) atrix. They are notused anyw ay in these cases. ButifU N ITD = 4, the unitdiagonalelem ents M U ST be referenced in the CSC representation.
4.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general.sparse \(m\) atrix A is used. H ow ever DESCRA (1) m ustbe equal to 3 in this case.
5. It is know \(n\) that there exists another representation of the com pressed sparse colum n form at (see for exam ple Y Saad, "Iterative M ethods forSparse LinearSystem \(s\) ", W PS, 1996). Its data structure consists of three anray instead of the fourused in the cumentim plem entation. Them ain difference is thatonly one amray, IA , containing the pointers to the beginning ofeach colum \(n\) in the arrays VA L and \(\mathbb{I N D X}\) is used instead of tw o amaysPNTRB and PN TRE.To use the routine \(w\) th this kind of sparse colum \(n\) form at the follow ing calling sequence should be used

SUBROUTINESCSCSM (TRANSA, M,N,UNITD,DV,ALPHA,DESCRA,
* \(V A L, \mathbb{N} D X, \mathbb{A}, \mathbb{A}(2), B, L D B, B E T A\),
* \(\quad\), LDC,\(W\) ORK,LW ORK )

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
scsmm -com pressed sparse row form atm atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SCSRMM(TRANSA,M,N,K,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,M,N,K,DESCRA (5),}
* LDB,LDC,LW ORK
\mathbb{NTEGER INDX NNZ),PNTRB (M),PNTREM)}
REAL ALPHA,BETA
REAL VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NE SCSRMM_64(TRANSA,M,N,K,A LPHA,DESCRA,}
* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,M,N,K,DESCRA (5),
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 INDX NNZ),PNTRB(M),PNTRE M)}
REAL ALPHA,BETA
REAL VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
where NN Z = PN TRE M )PNTRB (1)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NECSRMM (TRANSA,M, N ],K,ALPHA,DESCRA,VAL, INDX,}

```
* PNTRB,PNTRE, B, [LDB],BETA,C, [LDC], [WORK], [LW ORK])
\(\mathbb{N}\) TEGER TRANSA, M, K
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: D E S C R A, \mathbb{N} D X, P N T R B, P N T R E\)
REAL ALPHA,BETA
REAL,D IM ENSION (:) ::VAL
REAL, D \(\mathbb{I M}\) ENSION (: : : :: B, C

SUBROUTINE CSRMM_64 (TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N D X}\),
* PNTRB, PNTRE, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])

INTEGER*8TRANSA, M, K
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O}(:):: \operatorname{DESCRA}, \mathbb{N} D X, \operatorname{PNTRB}, \operatorname{PNTRE}\)
REAL ALPHA,BETA
REAL, D \(\mathbb{M}\) ENSION (:) ::VAL
REAL,D \(\mathbb{I}\) ENSION (:, :) :: B , C

\section*{DESCRIPTION}
C <-alpha op (A ) B + beta C
where A LPHA andBETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, \(A\) is a m atrix represented in com pressed sparse row form atand op (A) is one of \(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRA N SA Indicates how to operate \(w\) th the sparse \(m\) atrix
0 : operate w ith m atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um ber of row \(s\) in \(m\) atrix A

N \(\quad \mathrm{N}\) um berof colum ns in \(m\) atrix \(C\)

K \(\quad\) um berof colum ns in matrix A

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(A=A\) )
2: Herm Itian ( \(A=C O N J(A)\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(A=-C O N J\) ( \(A\) ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper

DESCRA (3) main diagonal type
0 : non-unit
1 : unit
DESCRA (4) A ray base NOT \(\mathbb{M}\) PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 :no repeated indices
VAL ( scalar array of length NN Z consisting of nonzero entries ofA.
\(\mathbb{I N D X} 0 \quad\) integer array of length NN Z consisting of the colum \(n\) indioes of nonzero entries of \(A\).

PN TRB () integer array of length \(M\) such thatPN TRB (J) PN TRB (1)+1
points to location in VA L of the firstnonzero elem ent in row J .
PNTRE ( integerarray of length \(M\) such thatPNTRE (J)-PNTRB (1) points to location in V A L of the lastnonzero elem ent in row J .

B 0 rectangular array w th first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C \(0 \quad\) rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC
W ORK () scratch array of length LW ORK.W ORK is not referenced in the curment version.

LW ORK length ofW ORK aray. LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/n cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:

\section*{NOTES/BUGS}

It is know \(n\) that there exists another representation of the com pressed sparse row form at (see forexam ple Y Saad, "Herative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three anray instead of the fourused in the currentim plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each row in the arrays VA L and \(\mathbb{N D} \mathrm{X}\) is used instead of tw o arrays PN TRB and PN TRE. To use the routine w th this kind of com pressed sparse row form at the follow ing calling sequence should be used

SUBROUTINESCSRMM (TRANSA, M, N, K,ALPHA,DESCRA,
* \(\quad V A L, \mathbb{N} D X, \mathbb{A}, \mathbb{I A}(2), B, L D B, B E T A\),
* C,LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
scsrsm -com pressed sparse row form at triangular solve

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SCSRSM(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,M,N,UNITD,DESCRA (5),}
* LDB,LDC,LW ORK
\mathbb{NTEGER INDX NNZ),PNTRB(M),PNTREM)}
REAL ALPHA,BETA
REAL DV M),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NE SCSRSM _64(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,M,N,UNITD,DESCRA (5),}
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 INDX NNZ),PNTRB(M),PNTRE M)}
REAL ALPHA,BETA
REAL DV M),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
where NN Z = PN TRE M )PNTRB (1)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE CSRSM (TRANSA,M, N ],UNITD,DV,ALPHA,DESCRA,VAL, INDX,}

```
* PNTRB,PNTRE, B, [LDB],BETA,C, [LDC], [W ORK], [LW ORK])
\(\mathbb{I N}\) TEGER TRANSA, M, UN ITD
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: D E S C R A, \mathbb{N} D X, P N T R B, P N T R E\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M}\) ENSION (:) ::VAL,DV
REAL,D IM ENSION (:, :):: B,C

SUBROUTINECSRSM_64 (TRANSA, M, \(\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X\),
* PNTRB, PNTRE, B, [LDB],BETA, C, [LDC], [W ORK], [LWORK])
\(\mathbb{I N T E G E R * 8 T R A N S A , M , U N I T D ~}\)
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):\) DESCRA, \(\mathbb{N} D X\), PNTRB, PNTRE
REAL ALPHA,BETA
REAL, D \(\mathbb{M}\) ENSION (:) ::VAL, DV
REAL,D \(\mathbb{I M} E N S I O N(:,:):\) B,C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \quad \text { OP (A) B + BETA } C \quad C<-A L P H A D \text { Op (A) B + BETA C } \\
& C<-A L P H A \text { OP (A)D B + BETA } C
\end{aligned}
\]
where ALPHA and BETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense \(m\) atrices, \(D\) is a diagonalscaling \(m\) atrix, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix represented in com pressed sparse row form atand op (A ) is one of op (A) \(=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}\left(\operatorname{conjg}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix 0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in \(m\) atrix A

N \(\quad\) umberof colum \(n s\) in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity \(m\) atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)
4 : A utom atic row scaling (see section N O TES for furtherdetails)

DV () A ray of length \(M\) containing the diagonalentries of the scaling diagonalm atrix D.

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem ent integer anay DESCRA (1) m atrix structure
\[
\begin{aligned}
& 0 \text { : general } \\
& 1 \text { : sym m etric ( } A=A \text { ) } \\
& 2: \text { H erm itian ( } A=\operatorname{CON} \text { JG (A }) \text { ) } \\
& 3 \text { : Triangular } \\
& 4 \text { : Skew (A nti)-Sym m etric ( } A=-A \text { ) } \\
& 5 \text { : D iagonal } \\
& 6 \text { : Skew Herm itian ( } A=-\operatorname{CON} \text { JG (A ) ) }
\end{aligned}
\]

N ote: For the routine, only DESCRA (1)=3 is supported.

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 : non-unit
1 :unit
DESCRA (4) A ray base \(\mathbb{N O T} \mathbb{M}\) PLEM ENTED )
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? \(\mathbb{N O T} \mathbb{I}\) PLEM ENTED)
0 : unknown
1 : no repeated indices
VAL () scalar array of length N N Z consisting of nonzero entries ofA.
\(\mathbb{I N D X ( ) \quad i n t e g e r a m a y ~ o f ~ l e n g t h ~ N ~ N ~ Z ~ c o n s i s t i n g ~ o f ~ t h e ~ c o l u m ~ n ~}\) indices of nonzero entries ofA (colum n indices M U ST be sorted in increasing order for each row )

PNTRB () integer amay of length \(M\) such thatPN TRB (J) PN TRB (1)+1 points to location in VA L of the firstnonzero elem ent in row J.

PN TRE () integer amay of length M such thatPNTRE (J)-PN TRB (1) points to location in VA L of the lastnonzero elem ent in row J.

B 0 rectangular array w ith first dim ension LD B .

LD B leading dim ension ofB

BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch array of length LW ORK.

On exit, ifLW ORK \(=-1, W\) ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK array. LW ORK should be at leastM .
Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=M\) *N_CPUS where N_CPUS is the m axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
htep://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such tests \(m\) ustbe perform ed before calling this routine.
2. IfUN ITD \(=4\), the routine scales the row s of \(A\) such that their 2 -nom s are one. The scaling \(m\) ay im prove the accuracy of the com puted solution. C orresponding entries ofV A L are changed only in the particular case. On retum D V m atrix stored as a vector contains the diagonalm atrix by which the row s have been scaled. U N ITD = 2 should be used for the next calls to the routine w ith overw ritten VAL and DV.

WORK (1)=0 on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here i is the row num berw hich 2 -norm is exactly zero.
3. If \(\operatorname{DESCRA}(3)=1\) and UN ITD < 4, the unitdiagonalelem ents \(m\) ightorm ightnotbe referenced in the CSR representation of a sparse m atrix. They are notused anyw ay in these cases.

ButifUN ITD \(=4\), the unit diagonalelem ents M U ST be referenced in the CSR representation.
4.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse m atrix \(A\) is used. H ow ever \(\operatorname{DESCRA}\) (1) m ustbe equal to 3 in this case.
5. It is know \(n\) that there exists another representation of the com pressed sparse row form at (see for exam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, IA , containing the pointers to the beginning ofeach row in the amaysVA L and \(\mathbb{N} D \mathrm{X}\) is used instead of tw o arrays PN TRB and PN TRE. To use the routine \(w\) ith this kind of com pressed sparse row form at the follow ing calling sequence should be used

SUBROUTINESCSRSM (TRANSA,M,N,UNTID,DV,ALPHA,DESCRA,
* \(\quad V A L, \mathbb{N} D X, \mathbb{A}, \mathbb{I A}(2), B, L D B, B E T A, C\),
* LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

sdiam m -diagonal form atm atrix-m atrix m ultiply

```

\section*{SYNOPSIS}
```

SUBROUTINE SDIAMM(TRANSA,M,N,K,ALPHA,DESCRA,

* VAL,LDA,\mathbb{D IAG,NDIAG,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,M,N,K,DESCRA (5),LDA,NDIAG,}
* LDB,LDC,LWORK
\mathbb{NTEGER IDIAG NDIAG)}
REAL ALPHA,BETA
REAL VAL(LDA,NDIAG),B (LD B,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINE SDIAMM_64(TRANSA,M,N,K,A LPHA,DESCRA,
* VAL,LDA,\mathbb{DIAG,NDIAG,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
NNTEGER*8 TRANSA,M,N,K,DESCRA (5),LDA,NDIAG,
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 \mathbb{D IAG NDIAG)}}\mathbf{N}=()
REAL ALPHA,BETA

```

\section*{F95 INTERFACE}

SUBROUTINEDIAMM (TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L,[L D A]\), * \(\mathbb{D} \mathbb{I} G, N D \mathbb{I} G, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\) \(\mathbb{N} T E G E R\) TRANSA, \(M, K, N D \mathbb{I} G\) \(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \quad D E S C R A, \mathbb{D} \mathbb{A} G\)
REAL ALPHA,BETA
REAL,D \(\mathbb{M}\) ENSION (:, :) :: VAL,B,C
SU BROU TINED IAMM_64(TRANSA, M, N ],K,ALPHA,DESCRA,VAL, [LDA],
* \(\mathbb{D} \mathbb{I A G}, N D \mathbb{I} G, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N} T E G E R * 8\) TRANSA, M, K, ND \(\mathbb{I A} G\)

\section*{DESCRIPTION}
C <-alpha op (A ) B + beta C
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, \(A\) is a \(m\) atrix represented in diagonal form atand op (A) is one of
\(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalentto 1 ifm atrix is real.

M \(N\) um ber of row \(s\) in matrix A
\(N \quad N\) um berof colum \(n s\) in \(m\) atrix \(C\)

K \(\quad \mathrm{N}\) um berof colum \(n s\) in \(m\) atrix \(A\)

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger array
0 : general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm titian ( \(A=-C O N\) J ( \(A\) ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonalype
0 : non-unit
1 :unit
DESCRA (4) A ray base \(\mathbb{N O T} \mathbb{I}\) PLEM ENTED)
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? \(\mathbb{N O T} \mathbb{I}\) PLEM ENTED)
0 :unknown
1 : no repeated indices

VAL ( ) tw o-dim ensionalLD A by ND IA G array such thatVA L (: II consists of non-zero elem ents on diagonal ID IA G (I) ofA. D iagonals in the low ertriangularpart of \(A\) are padded from the top, and those in the upper triangularpart are padded from the bottom.

LDA leading dim ension ofVA L, m ustibe GE.M \(\mathbb{N} \mathbb{M}, \mathrm{K}\) )

ID IA G () integer array of length ND IA G consisting of the comesponding diagonal offsets of the non-zero diagonals ofA in VA L. Low ertriangular diagonals have negative offsets, them ain diagonal has offset 0 , and upper triangular diagonals have positive offset.

ND IA G num berof non-zero diagonals in A.
B 0 rectangular array \(w\) ith first dim ension LD B .
LD B leading dim ension ofB
BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC
W ORK ( scratch array of length LW ORK.W ORK is not referenced in the cument version.

LW ORK length ofW ORK array. LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B lasU ser's G uide available at:
http://m ath nistgov/m csol/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

sdiasm -diagonal form at triangular solve

```

\section*{SYNOPSIS}
```

SUBROUTINE SDIA SM(TRANSA,M ,N,UNITD,DV,ALPHA,DESCRA,

* VAL,LDA,\mathbb{D IAG,NDIAG,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,M,N,UNITD,DESCRA (5),LDA,ND IAG,}
* LDB,LDC,LWORK
\mathbb{NTEGER IDIAG NDIAG)}
REAL ALPHA,BETA
REAL DV M),VAL (LDA,NDIAG),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINE SD IA SM _64(TRANSA,M,N,UNITD ,DV,ALPHA,DESCRA,
* VAL,LDA,\mathbb{DIAG,NDIAG,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,M,N,UNITD,DESCRA (5),LDA,NDIAG,}
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 \mathbb{D IAG NDIAG)}}\mathbf{N}\mathrm{ )}
REAL ALPHA,BETA
REAL DV M),VAL (LDA,NDIAG),B (LDB,*),C (LDC,\star),W ORK (LW ORK)

```

\section*{F95 INTERFACE}
```

SUBROUTINEDIASM (TRANSA,M, NN,UNITD,DV,ALPHA,DESCRA,VAL,

* [LDA],\mathbb{D IAG,NDIAG,B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])}
INTEGER TRANSA,M,ND IAG
\mathbb{NTEGER,D IM ENSION (:) :: DESCRA,\mathbb{D IAG}}\mathbf{N}=\mp@code{L}
REAL ALPHA,BETA
REAL,D\mathbb{M ENSION (:) :: DV}
REAL,D IM ENSION (:, :) :: VAL,B,C

```
SUBROUTINEDIASM_64 (TRANSA, M, N ],UNITD,DV,ALPHA,DESCRA,VAL,
* [LDA], \(\mathbb{D} \mathbb{I A G}, N D \mathbb{I A G}, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N} T E G E R * 8\) TRANSA, M,NDIAG
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{D} N(:):: \quad\) DESCRA, \(\mathbb{D} \mathbb{I} G\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I M} E N S I O N(:):: D V\)
REAL,D \(\mathbb{I}\) ENSION (:, :) :: VAL,B,C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { op(A)B + BETA } C \quad C<-A L P H A D \text { op(A)B+BETA C } \\
& C<-A L P H A \text { op(A)D B + BETA } C
\end{aligned}
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense \(m\) atrices, \(D\) is a diagonalscaling \(m\) atrix, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix represented in diagonal form at and op (A ) is one of
\(\mathrm{op}(\mathrm{A})=\operatorname{inv}(\mathrm{A})\) or op (A) \(=\operatorname{inv}(\mathrm{A})\) or op (A) \()=\operatorname{inv}\left(\right.\) (oonjg ( \(\left.\mathrm{A}^{\prime}\right)\) ) (inv denotesm atrix inverse, 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate w th transpose m atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in \(m\) atrix A

N \(\quad N\) um berof colum ns in matrix C
UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 :Scale on left (row scaling)
3 : Scale on right (colum \(n\) scaling)
4 :A utom atic row scaling (see section NOTES for furtherdetails)

DV 0 A ray of length M containing the diagonalentries of the scaling diagonalm atrix D .

A LPH A Scalarparam eter
DESCRA 0 D escriptor argum ent. Five elem ent integer amay
DESCRA (1) m atrix structure
0 : general

1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm itian ( \(A=C O N J(A)\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \((A=-A\) )
5 :D iagonal
6 : Skew Herm titian ( \(A=-\operatorname{CON}\) J ( \(A\) ) )
N ote: For the routine, only D ESCRA (1)=3 is supported.

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 :non-unit
1 :unit
DESCRA (4) A ray base \(\mathbb{N O T} \mathbb{M}\) PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? (NOT \(\mathbb{M}\) PLEM ENTED)
0 : unknown
1 : no repeated indices
VAL () tw o-dim ensionalLD A boy-ND IA G aray such thatVAL(:,I) consists of non-zero elem ents on diagonal ID IA G (I) of A. D iagonals in the low er triangularpart of A are padded from the top, and those in the upper triangularpart are padded from the bottom.

LDA leading dim ension ofVAL, m ustbe GE.M \(\mathbb{N}(M, K)\)

ID IA G () integer anay of length ND IA G consisting of the corresponding diagonaloffsets of the non-zero diagonals ofA in VAL. Low ertriangular diagonals have negative offsets, them ain diagonalhas offset 0 , and uppertriangular diagonals have positive offset. Elem ents of \(\mathbb{D}\) IA G ofM UST be sorted in increasing order.

ND IA G num berofnon-zero diagonals in A.

B 0 rectangular array w ith firstdim ension LD B .

LD B leading dim ension of \(B\)

BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch amay of length LW ORK. On exit, if LW ORK = -1,W ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK aray. LW ORK should be at leastM.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=M\) *N_CPU \(S\) where \(N\) _CPU \(S\) is the \(m\) axim um num berof processors available to the program .

IfLW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK anray, retums this value as the firstentry of theW ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov fn csd/Staff/k Rem ington/tspoblas/
"D ocum ent for the Basic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib.org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. No test for singularity ornear-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.
2. If U N ITD \(=4\), the routine scales the row sofA such that their 2 -norm sare one. The scaling \(m\) ay im prove the accuracy of the com puted solution. C orresponding entries ofV A L are changed only in the particular case. On retum DV m atrix stored as a vector contains the diagonalm atrix by w hich the row s have been scaled. U N ITD \(=2\) should be used for the next calls to the routine w ith overw ritten VA L and DV .

WORK ( 1 )=0 on retum if the scaling has been com pleted successfully, otherw ise \(W O R K(1)=-i w\) here \(i\) is the row num berw hich 2 -norm is exactly zero.
3. If \(D E S C R A(3)=1\) and \(U N\) ITD \(<4\), the unitdiagonalelem ents m ightorm ightnotbe referenced in the D IA representation of a sparse m atrix. They are notused anyw ay in these cases. ButifUN ITD = 4, the unit diagonalelem ents M U ST be referenced in the \(D \mathbb{I A}\) representation.
4.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse
m atrix \(A\) is used. H ow ever DESCRA (1) m ustbe equal to 3 in this case.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sdisna - com pute the reciprocal condition num bers for the eigenvectors of a real sym \(m\) etric or com plex \(H\) erm itian \(m\) atrix or for the leftor right singular vectors of a general mby \(-\mathrm{n} m\) atrix

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SD ISNA (JOB,M,N,D,SEP, INFO)}
CHARACTER * 1 JOB
\mathbb{NTEGER M,N,NNFO}
REALD (*),SEP (*)
SU BROUT\mathbb{NE SD ISNA_64 (JO B ,M ,N,D ,SEP, IN FO )}
CHARACTER * 1 JB
\mathbb{NTEGER*8M,N,INFO}
REALD (*),SEP (*)
F95 INTERFACE
SU BROUT\mathbb{NE D ISNA (JOB,M ,N,D ,SEP, [NNFO ])}
CHARACTER (LEN=1)::JOB
INTEGER ::M,N,\mathbb{NFO}
REAL,D IM ENSION (:) ::D ,SEP
SU BROUTINE D ISNA_64 (OD B,M ,N,D ,SEP,[INFO ])
CHARACTER (LEN=1) :: JOB
\mathbb{NTEGER (8) ::M ,N,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REAL,D IM ENSION (:) ::D ,SEP

```

\section*{C INTERFACE}
\#include <sunperfh>
void sdisna (char job, intm , intn, float *d, float *sep, int*info);
void sdisna_64 (char jंjb, long m, long n, float *d, float *sep, long *info);

\section*{PURPOSE}
sdisna com putes the reciprocal condition num bers for the eigenvectors of a realsym m etric or com plex H erm itian m atrix orforthe left or rightsingular vectors of a general m-by-n matrix. The reciprocalcondition num ber is the gap' betw een the corresponding eigenvalue or singular value and the nearest other one.

The bound on the error, \(m\) easured by angle in radians, in the I-th com puted vector is given by

SLAMCH (E')* (ANORM /SEP (I))
where ANORM \(=2\) norm \((A)=m a x(\operatorname{abs}(D(\mathcal{D}))\) ). SEP (I) is not allow ed to be sm allerthan SLAM CH (E')*ANORM in orderto lim it the size of the errorbound.

SD ISN A m ay also be used to com pute enrorbounds for eigenvectors of the generalized sym \(m\) etric definite eigenproblem .

\section*{ARGUMENTS}

JOB (input)
Specifies forw hich problem the reciprocal condition num bers should be com puted:
\(=\mathrm{E}\) ': the eigenvectors of a sym m etric/H erm titian m atrix;
= IL ': the leftsingular vectors of a general
\(m\) atrix;
\(=\mathrm{R}\) ': the rightsingularvectors of a general \(m\) atrix.
\(M\) (input) The num ber of row \(s\) of the \(m\) atrix. \(M>=0\).

N (input) If \(\mathrm{OB}=\mathrm{L}\) 'or R ', the num ber of colum ns of the \(m\) atrix, in which case \(N>=0\). Ignored if \(J O B=\) E'

D (input) dim ension ( \(m\) in \(M, N\) )) if \(30 B=L\) ' or \(R^{\prime}\) The eigenvalues (if \(\mathrm{JOB}=\mathrm{E}\) ) or singular values (if \(J O B=L\) ' or \(R\) ) of the matrix, in either increasing or decreasing order. If singular values, they \(m\) ustbe non-negative.

SEP (output)
dimension ( \(m\) in \((M, N)\) ) if \(J O B=L^{\prime}\) or \(R^{\prime}\) The reciprocal condition num bers of the vectors.
\(\mathbb{I N} F O\) (output)
= 0: successfulexit.
<0: if \(\mathbb{N F O}=-\) i, the \(i\)-th argum ent had an illegal value.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sdot-com pute the dotproduct of tw \(o\) vectors \(x\) and \(y\).

\section*{SYNOPSIS}
```

REAL FUNCTION SDOT N,X,\mathbb{NCX,Y,\mathbb{NCY)}}\mathbf{N}=()
\mathbb{NTEGER N, INCX,INCY}
REALX (*),Y (*)
REAL FUNCTION SDOT_64N,X,NNCX,Y,\mathbb{NCY)}
INTEGER*8N,\mathbb{NCX,INCY}
REALX (*),Y (*)
F95 INTERFACE

```

```

    \mathbb{NTEGER ::N,\mathbb{NCX,}\mathbb{NCY}}\mathbf{N}={
    REAL,DIM ENSION (:) ::X,Y
    REALFUNCTION DOT_64 (N ],X, [\mathbb{NCX ],Y, [\mathbb{NCY])}}\mathbf{~}\mathrm{ ( }
    \mathbb{NTEGER (8)::N,\mathbb{NCX,}\mathbb{NCY}}\mathbf{N}={
    REAL,D IM ENSION (:) ::X,Y
    C INTERFACE
\#include <sunperfh>
float solot(intn, float *x, int incx, float *y, int incy);
floatsdot_64 (long n, float *x, long incx, float *y, long
incy);

```

\section*{PURPOSE}
sdot com pute the dotproductof \(x\) and \(y w h e r e x\) and \(y\) are n-vectors.

\section*{ARGUMENTS}

N (input)
O n entry, \(N\) specifies the num ber of elem ents in the vector. IfN is notpositive then the function retums the value 0.0. U nchanged on exit. \(X\) (input)
\((1+(n-1) * a b s(\mathbb{N} C X))\). On entry, the increm ented array \(X\) must contain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X\) m ustnotbe zero. U nchanged on exit.
\(Y\) (input)
( \(1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C Y))\). On entry, the increm ented amay \(Y\) must contain the vectory.
U nchanged on exit.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of Y. \(\mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS

\section*{NAME}
sdoti-C om pute the indexed dotproduct.

\section*{SYNOPSIS}

REAL FUNCTION SDOTINZ,X, \(\mathbb{N} D X, Y)\)
REALX (*) \({ }^{( }{ }^{(*)}\)
\(\mathbb{N}\) TEGER NZ
\(\mathbb{N}\) TEGER \(\mathbb{I N D X ( * )}\)
REALFUNCTION SDOTI_64(NZ, X, \(\mathbb{N} D X, Y)\)
REALX (*), Y (*)
\(\mathbb{N}\) TEGER*8NZ
\(\mathbb{I N} T E G E R * 8 \mathbb{N} D X(*)\)
F95 \(\mathbb{I N}\) TERFACE
REAL FUNCTION DOTI(NZ],X, \(\mathbb{N} D \mathrm{X}, \mathrm{Y})\)
REAL,D IM ENSION (:) :: X,Y
\(\mathbb{I N T E G E R}:: \mathrm{NZ}\)
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{N} D \mathrm{X}\)
REALFUNCTION DOTI_64 ( \(\mathbb{N} Z], X, \mathbb{N} D X, Y)\)
REAL,D IM ENSION (:) :: X,Y
\(\mathbb{N} T E G E R(8):: N Z\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{N D X}\)

\section*{PURPOSE}
fullstorage form.
```

dot $=0$
do $i=1, n$
$\operatorname{dot}=\operatorname{dot}+x(i)$ * $y($ ind $x(i))$
enddo

```

\section*{ARGUMENTS}

N Z (input)
N um ber of elem ents in the com pressed form .
U nchanged on exit.
\(X\) (input)
V ector in com pressed form . U nchanged on exit.
\(\mathbb{N} D X\) (input)
\(V\) ector containing the indiaes of the com pressed
form. It is assum ed that the elem ents in \(\mathbb{N} D X\) are distinctand greaterthan zero. U nchanged on exit.

Y (input)
V ector in fullstorage form. O nly the elem ents corresponding to the indices in \(\mathbb{N}\) D X w illbe accessed.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sdsdot - com pute a constantplus the double precision dot product oftw \(o\) single precision vectors \(x\) and \(y\)

\section*{SYNOPSIS}

REAL FUNCTION SDSDOT \(N, S B, S X, \mathbb{N C X}, S Y, \mathbb{N C Y})\)
\(\mathbb{N} T E G E R N, \mathbb{N} C X, \mathbb{N} C Y\)
REAL SB
REALSX (*), SY (*)
REAL FUNCTION SDSDOT_64 \(\mathbb{N}, S B, S X, \mathbb{N} C X, S Y, \mathbb{N} C Y\) )
\(\mathbb{N} T E G E R * 8 N, \mathbb{N} C X, \mathbb{N} C Y\)
REALSB
REALSX (*), SY (*)

\section*{F95 INTERFACE}

REAL FUNCTION SDSDOT \(\mathbb{N}, S B, S X, \mathbb{N C X}, S Y, \mathbb{N C Y})\)
\(\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y\)
REAL :: SB
REAL,D IM ENSION (:) ::SX ,SY
REAL FUNCTION SDSDOT_64 \(\mathbb{N}, S B, S X, \mathbb{N C X}, S Y, \mathbb{N C Y})\)
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{N} C X, \mathbb{N} C Y\)
REAL :: SB
REAL,D IM ENSION (:) :: SX , SY
C INTERFACE
\#include <sunperfh>
float sdsdot(intn, floatsb, float *sx, int incx, float *sy, int incy);
float sdsdot 64 (long n , float sb, float *sx, long incx, float *sy, long incy);

\section*{PURPOSE}
sdsdotCom putes a constantplus the double precision dot product of \(x\) and \(y\) where \(x\) and \(y\) are single precision \(n-\) vectors.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. If N is notpositive then the function retums the value 0.0 . U nchanged on exit.

SB (input)
O n entry, the constant that is added to the dot product before the result is retumed. U nchanged on exit.

SX (input)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C X))\). On entry, the increm ented array \(S X\) mustcontain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of SX. \(\mathbb{N} C X\) m ustnotbe zero. U nchanged on exit.

SY (input)
(1+(n-1)*abs( \(\mathbb{N} C Y)\) ). On entry, the increm ented array \(S Y\) mustcontain the vectory. U nchanged on exit.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of SY. \(\mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE

\section*{NAME}
second -retum the user tim e for a process in seconds

\section*{SYNOPSIS}

REALFUNCTION SECOND 0

REAL FUNCTION SECOND_640

F95 INTERFACE
REAL FUNCTION SECOND ()

REAL FUNCTION SECOND_640

C INTERFACE
\#include < sunperfh>
float second ();
floatsecond_64 ();

\section*{PURPOSE}
second retums the user tim e for a process in seconds. This version gets the tim e from the system function ET IM E.

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}

> selm m -Ellpack form atm atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SELLMM(TRANSA,M,N,K,ALPHA,DESCRA,}

* VAL,INDX,LDA,MAXNZ,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,M,N,K,DESCRA (5),LDA,MAXNZ,}
* LDB,LDC,LWORK
INTEGER INDX (LDA,MAXNZ)
REAL ALPHA,BETA
REAL VAL (LDA,MAXNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
SUBROUTINE SELLMM_64(TRANSA, M,N,K,ALPHA,DESCRA,
* VAL, \(\mathbb{N} D \mathrm{X}, \mathrm{LDA}, \mathrm{MAXNZ}\),
* B,LDB,BETA, C,LDC,WORK,LWORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, M, N, K, DESCRA (5), LDA, MAXNZ,
* LDB,LDC,LW ORK
\(\mathbb{N T E G E R * 8} \mathbb{N} D X(\mathbb{L} A, M A X N Z)\)
REAL ALPHA,BETA
REAL VAL (LDA, MAXNZ),B( LDB , *), C (LDC,*),WORK (LW ORK)

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE ELLMM (TRANSA,M, N ],K,ALPHA,DESCRA,VAL, INDX,}

* [LDA],MAXNZ,B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])
\mathbb{NTEGER TRANSA,M,K,MAXNZ}
INTEGER,D\mathbb{M ENSION (:) :: DESCRA}
INTEGER,D\mathbb{M ENSION (:,:):: \mathbb{NDX}}\mathbf{N}=\mp@code{N}
REAL ALPHA,BETA
REAL,D IM ENSION (:,:) :: VAL,B,C

```
SUBROUTINE ELLMM _64(TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N} D X\),

\section*{DESCRIPTION}
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, \(A\) is a \(m\) atrix represented in Ellpack form at form at and op (A) is one of
\(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conj}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix
0 : operate w th m atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalentto 1 ifm atrix is real.

M \(\quad\) Numberofrow sin matrix A
\(N \quad N\) um berof colum ns in \(m\) atrix \(C\)

K \(\quad \mathrm{N}\) um berof colum ns in \(m\) atrix \(A\)

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(A=A\) )
2 : Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) )
3 :Triangular
4 : Skew (Anti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CON}\) G ( A ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonaltype
0 : non-unit
1 :unit

DESCRA (4) A ray base NOT IM PLEM ENTED)
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? (NOT IM PLEM ENTED)
0 : unknown
1 : no repeated indices
VAL ( 0 tw o-dim ensionallD A -by M A XNZ array such thatVA L ( \(\mathrm{I}, \mathrm{:}\) ) consists of non-zero elem ents in row IofA, padded by zero values if the row contains less than M AXN Z .
\(\mathbb{I N D X} 0 \quad\) tw o-dim ensional integer LD A by -M A XN Z aray such \(\mathbb{N} D \mathrm{X}\) ( \(I\), :) consists of the colum n indices of the nonzero elem ents in row \(I\), padded by the integer value I if the num berof nonzeros is less than M AXNZ.

LD A leading dim ension ofVAL and \(\mathbb{N D} X\).

MAXNZ max num berofnonzeros elem ents per row .
B 0 rectangular array w th first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w th first dim ension LD C .

LD C leading dim ension of \(C\)
W ORK ( scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK anay. LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}

\section*{sellsm -Ellpack form at triangular solve}

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SELLSM(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,LDA,MAXNZ,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,M,N,UNITD,DESCRA (5),LDA,MAXNZ,}
* LDB,LDC,LWORK
INTEGER INDX (LDA,MAXNZ)
REAL ALPHA,BETA
REAL DVM),VAL(LDA,MAXNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINE SELLSM _64(TRANSA,M,N,UNITD,DV,A LPHA,DESCRA,
* VAL, \mathbb{NDX,LDA,MAXNZ,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,M,N,UNITD,DESCRA (5),LDA,MAXNZ,
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 INDX (LDA,MAXNZ)}
REAL ALPHA,BETA
REAL DV M),VAL (LDA,MAXNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}

SUBROUTINE ELLSM (TRANSA, M, \(\mathbb{N}], U N \mathbb{I T D , D V , A L P H A , D E S C R A , V A L , ~}\) * \(\mathbb{N} D X,[L D A], M A X N Z, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{I N T E G E R}\) TRANSA, M, MAXNZ
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: DESCRA \(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:, :) :: \(\mathbb{N D X}\)
REAL ALPHA,BETA
REAL,D \(\mathbb{M}\) ENSION (:) :: DV
REAL,D \(\mathbb{I M}\) ENSION (:, :) :: VAL,B,C

SUBROUTINE ELLSM _64 (TRANSA, M, \(\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L\), * \(\mathbb{N} D X,[L D A], M A X N Z, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N} T E G E R * 8\) TRANSA, M, MAXNZ
\(\mathbb{N} T E G E R * 8, D \mathbb{I M}\) ENSION (:) :: DESCRA
\(\mathbb{N} T E G E R * 8, D \mathbb{M}\) ENSION (: : : : : \(\mathbb{N} D X\)
REAL ALPHA,BETA
REAL,D \(\mathbb{M}\) ENSION (:) :: DV
REAL,D \(\mathbb{M}\) ENSION (:, :) :: VAL,B,C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { op(A)B + BETA C } \quad C<-A L P H A D \text { op (A) B + BETA C } \\
& C<-A L P H A \text { op(A)D B + BETA C }
\end{aligned}
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense \(m\) atrices, \(D\) is a diagonalscaling \(m\) atrix, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix represented in Ellpack form at and op (A) is one of \(\mathrm{op}(\mathrm{A})=\operatorname{inv}(\mathrm{A})\) or op (A) \(=\operatorname{inv}(\mathrm{A})\) or op(A) \(=\operatorname{inv}\left(\operatorname{conjg}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix 0 : operate with m atrix 1 : operate \(w\) th transpose m atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M \(\quad\) um berof row \(s\) in \(m\) atrix A
N \(\quad\) Uum berof colum ns in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)
4 :A utom atic row scaling (see section N O TES for furtherdetails)

DV 0 A ray of length M containing the diagonalentries of the scaling diagonalm atrix D .

A LPH A Scalarparam eter
DESCRA () D escriptor argum ent. Five elem ent integer aray DESCRA (1) m atrix structure
```

    0 :general
    1 : symm etric ( }A=A\mathrm{ ) )
    2:H erm Itian (A = CON JG (A ))
    3:Triangular
    4 :Skew (A nti)-Symm etric (A=-A )
    5 :D iagonal
    6:Skew Herm titian (A= CON JG (A ) )
    N ote:For the routine, only D ESCRA (1)=3 is supported.
    D ESCRA (2) upper/low er triangular indicator
        1 : low er
        2 :upper
    DESCRA (3) m ain diagonaltype
        0:non-unit
        1 :unit
    DESCRA (4) A ray base NOT IM PLEM ENTED )
        0 : C C ++ com patible
        1 :Fortran com patible
    DESCRA (5) repeated indices? (NOT IM PLEM ENTED)
        0 :unknow n
        1: no repeated indices
    VAL () tw o-dim ensionalLD A foy M A X N Z array such thatV A L (I,:)
consists of non-zero elem ents in row IofA, padded by
zero values if the row contains less than M AXN Z .
INDX () tw o-dim ensionalintegerLD A boy-M A XN Z array such
\mathbb{N D X (I,:) consists of the colum n indiges of the}
nonzero elem ents in row I, padded by the integer
value I if the num berofnonzeros is less than M A XN Z .
The colum n indices M U ST be sorted in increasing order
foreach row .
LDA leading dim ension ofV A L and \mathbb{ND X .}
M A X N Z m ax num ber ofnonzeros elem ents per row .
B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension ofC

```
    W ORK () scratch amay of length LW ORK.
        On exit, ifLW ORK \(=-1, W\) ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK array. LW ORK should be at leastM .
Forgood penform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=\mathrm{M}\) *N_CPUS where N_CPUS is the m axim um num berof processors available to the program .

IfLW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of theW ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
htep://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such tests \(m\) ustbe perform ed before calling this routine.
2. IfUN ITD \(=4\), the routine scales the row s of \(A\) such that their 2 -nom s are one. The scaling \(m\) ay im prove the accuracy of the com puted solution. C orresponding entries ofV A L are changed only in the particular case. On retum DV m atrix stored as a vector contains the diagonalm atrix by w hich the row s have been scaled. U N ITD = 2 should be used for the next calls to the routine w ith overw ritten VAL and DV.

WORK ( 1 )=0 on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here i is the row num berw hich 2 -norm is exactly zero.
3. If \(\operatorname{DESCRA}(3)=1\) and U N ITD < 4, the unitdiagonalelem ents \(m\) ightorm ightnotbe referenced in the ELL representation of a sparse m atrix. They are notused anyw ay in these cases.

ButifUN ITD \(=4\), the unit diagonalelem ents M U ST be referenced in the ELL representation.
4.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse m atrix A is used. H ow ever DESCRA (1) m ustbe equal to 3 in this case.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO

\section*{NAME}
sfftc - initialize the trigonom etric weight and factor tables or com pute the forw ard FastFourier T ransform of a realsequence.

\section*{SYNOPSIS}

\(\mathbb{N}\) TEGER IOPT,N, \(\mathbb{F} A C(*)\),LW ORK, \(\mathbb{E R R}\)
COM PLEX Y (*)
REALX (*), SCALE, TRIGS (*), W ORK (*)
SU BROUTINE SFFTC_64 (TOPT,N,SCALE,X,Y,TRIGS, IFAC,WORK,LWORK,ERR)
\(\mathbb{N} T E G E R * 8 \mathbb{I} P T, N, \mathbb{F A C}(*), L W O R K, \mathbb{E R R}\)
REALX (*), SCALE, TRIGS (*), W ORK (*)
COM PLEX Y (*)

\section*{F95 INTERFACE}

SU BROUTINE FFT (IOPT,N,SCALE,X,Y,TRIGS, IFAC,W ORK, [LW ORK ], ERR)
\(\mathbb{N}\) TEGER*4, \(\mathbb{N}\) TENT \((\mathbb{N})::\) IOPT
\(\mathbb{N}\) TEGER*4, \(\mathbb{N}\) TENT \((\mathbb{N})\),OPTIONAL ::N,LW ORK
REAL, \(\mathbb{N} T E N T(\mathbb{N})\), OPTIONAL::SCALE
REAL, \(\mathbb{N} T E N T(\mathbb{N})\),D \(\mathbb{M}\) ENSION (:) ::X
COMPLEX, \(\mathbb{N}\) TENT (OUT),D \(\mathbb{I M}\) ENSION (:) ::Y
REAL, \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{I M}\) ENSION (:) ::TRIGS
\(\mathbb{N}\) TEGER*4, \(\mathbb{N}\) TENT ( \(\mathbb{N} O U T\) ), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{F A C}\)
REAL, \(\mathbb{I N T E N T}\) (OUT),D \(\mathbb{M}\) ENSION (:) ::W ORK
\(\mathbb{N} T E G E R * 4, \mathbb{N} T E N T(O U T):: \mathbb{E R R}\)

SU BROUTINE FFT_64 (IOPT, \(\mathbb{N}],[S C A L E], X, Y, T R I G S, \mathbb{F A C}, W\) ORK, [LW ORK ], \(\mathbb{E R R}\) )
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N}):: \operatorname{IOPT}\)
\(\mathbb{N} T E G E R(8), \mathbb{N} T E N T(\mathbb{N})\), OPTIONAL :: N , LW ORK
REAL, \(\mathbb{N}\) TENT ( \(\mathbb{N}\) ), OPTIONAL :: SCALE
REAL, \(\mathbb{N} \operatorname{TENT}(\mathbb{N}), D \mathbb{M} E N S I O N(:):: X\)
COMPLEX, \(\mathbb{N}\) TENT (OUT), D \(\mathbb{M}\) ENSION (:) ::Y
REAL, \(\mathbb{N}\) TENT ( \(\mathbb{N} O U T\) ), D \(\mathbb{M}\) ENSION (:) ::TRIGS
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N O U T}), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{F A C}\)
REAL, \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M} E N S I O N(:):: W O R K\)
\(\mathbb{N}\) TEGER (8), \(\mathbb{N}\) TENT (OUT) :: \(\mathbb{E R R}\)

\section*{C INTERFACE}
\#include <sunperfh>
void sfftc_ (int*iopt, int *n, float *scale, float * x , com plex *y, float *trigs, int *ifac, float *w ork, int *lw ork, int *ient);
void sfftc_64_ long *iopt, long *n, float *scale, float *x, com plex *y, float *trigs, long *ifac, float *w ork, long *lw ork, long *ienc);

\section*{PURPOSE}
sfftc initializes the trigonom etric w eight and factor tables or com putes the forw ard FastFourier Transform of a real sequence as follow \(s\) :

N-1
\(Y(k)=\) scale * SUM W *X (i)
\(=0\)
where
\(k\) ranges from 0 to \(N-1\)
\(i=\operatorname{sqrt}(-1)\)
isign \(=-1\) for forw ard transform
\(W=\exp \left(i s i g n \star i^{\star} j^{\star} k \star 2 * p i N N\right)\)
In real-to-com plex transform of length \(N\), the \((\mathbb{N} / 2+1)\) com plex output data points stored are the positive-frequency half of the spectrum of the D iscrete Fourier T ransform. The other half can be obtained through com plex conjugation and therefore is notstored.

\section*{ARGUMENTS}

IO PT (input)
Integer specifying the operation to be perform ed:

IO PT \(=0\) com putes the trigonom etric \(w\) eight table and factor table
IO PT = -1 com putes forw ard FFT

N (input)
Integer specifying length of the input sequence X . N is \(m\) ostefficientw hen it is a product of sm all prim es. \(\mathrm{N}>=0\). U nchanged on exit.

SCALE (input)
Real scalarby w hich transform results are scaled. U nchanged on exit. SCA LE is defaulted to 1.0 for F95 \(\mathbb{I N}\) TERFACE.

X (input) On entry, X is a realaray whose first N elem ents contain the sequence to be transform ed.

Y (output)
C om plex array w hose first \(\mathbb{N} / 2+1\) ) elem ents contain the transform results. \(X\) and \(Y m\) ay be the sam \(e\) array starting at the sam e mem ory location, in which case the dimension of X ust.be at least \(2^{*}(\mathbb{N} / 2+1)\). O therw ise, it is assum ed that there is no overlap betw een \(X\) and \(Y\) in \(m\) em ory.

TR IG S (input/output)
Realarray of length \(2 * \mathrm{~N}\) that contains the trigonom etric w eights. The w eights are com puted w hen the routine is called with IOPT = 0 and they are used in subsequent calls when IOPT \(=-1\). U nchanged on exit.

FAC (input/output)
Integer array of dim ension at least 128 that contains the factors of \(N\). The factors are com puted when the routine is called w ith IO PT = 0 and they are used in subsequent calls where \(10 P T=-1\). U nchanged on exit.

W ORK (w orkspace)
Realarray of dim ension at leastN. The user can also choose to have the routine allocate its ow n w orkspace (see LW ORK).

LW ORK (input)
Integer specifying w orkspace size. If LW ORK \(=0\), the routine w illallocate its ow n w orkspace.
\(\mathbb{E R R}\) (output)
On exit, integer \(\mathbb{E} R R\) has one of the follow ing
values:
0 = norm alretum
\(-1=10 P T\) is not 0 or -1
\(-2=\mathrm{N}<0\)
\(-3=(L W O R K\) is not 0) and (LW ORK is less than N)
\(-4=m\) em ory allocation forw orkspace failed

\section*{SEE ALSO}
ffl

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO
- CAUTIONS

\section*{NAME}
sfftc2 -initialize the trigonom etric weight and factor
tables or com pute the tw o-dim ensional forw ard FastFourier
\(T\) ransform of a tw o-dim ensional real array.

\section*{SYNOPSIS}

SU BROUTINE SFFTC2 (IOPT,N1,N2,SCALE, X,LDX,Y,LDY,TRIGS, FAC,W ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER IOPT,N1,N2,LDX,LDY, \(\mathbb{F A C}\) (*) \(^{(*}\) LW ORK, \(\mathbb{E R R}\) COM PLEX Y (LDY,*)
REALX (LDX,*),SCALE,TRIGS (*),WORK (*)
SU BROUTINE SFFTC2_64 (IOPT,N1,N2,SCALE,X,LDX,Y,LDY,TRIGS, FFAC,W ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N} T E G E R * 8 \mathbb{I} P T, N 1, N 2, L D X, L D Y, \mathbb{F} A C(*), L W O R K, \mathbb{E R R}\)
REALX (LDX,*), SCALE,TRIGS (*), W ORK (*)
COM PLEX Y (LDY,*)

\section*{F95 INTERFACE}

SU BROUTINE FFT2 (IOPT, \(\mathbb{N} 1], \mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S\), \& \(\quad \mathbb{F A C}, \mathrm{W} O R K\), [LW ORK], \(\mathbb{E R R})\)
\(\mathbb{N}\) TEGER*4, \(\mathbb{N}\) TENT \((\mathbb{N})::\) IOPT
\(\mathbb{N}\) TEGER*4, \(\mathbb{N}\) TENT \((\mathbb{N})\), OPTIONAL ::N \(1, N 2, L D X, L D Y, L W\) ORK
REAL, \(\mathbb{N}\) TENT \((\mathbb{N})\), OPTIONAL :: SCALE
REAL, \(\mathbb{N} T E N T(\mathbb{N})\), D \(\mathbb{I M}\) ENSION \((:,:):\) :
COM PLEX, \(\mathbb{N}\) TENT (OUT), D \(\mathbb{M}\) ENSION (:,:) ::Y
REAL, \(\mathbb{N}\) TENT ( \(\mathbb{N O U T}\) ), D \(\mathbb{M}\) ENSION (:) ::TRIGS
\(\mathbb{N} T E G E R * 4, \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M}\) ENSION (:) :: \(\mathbb{F A C}\)

REAL, \(\mathbb{I N T E N T}(\mathrm{OUT}), \mathrm{D} \mathbb{I}\) ENSION (:) ::W ORK
\(\mathbb{N}\) TEGER*4, \(\mathbb{N}\) TENT (OUT) :: \(\mathbb{E R R}\)
SU BROUTINE FFT2_64 (DPPT, \(\mathbb{N} 1], \mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S\), تAC,W ORK, [LW ORK], ERR)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N}):: \mathbb{I O P T}\)
\(\mathbb{N}\) TEGER (8), \(\mathbb{I N}\) TENT \((\mathbb{N})\), OPT IO NAL ::N 1,N2, LDX,LDY,LW ORK
REAL, \(\mathbb{N} T E N T(\mathbb{N})\),OPTIONAL::SCALE
REAL, \(\mathbb{N} T E N T(\mathbb{N}), D \mathbb{M}\) ENSION (:,:)::X
COM PLEX, \(\mathbb{I N T E N T}\) (OUT),D \(\mathbb{M}\) ENSION (: : : : :: Y
REAL, \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{I M}\) ENSION (:) ::TRIGS
\(\mathbb{N} T E G E R(8), \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M}\) ENSION (:) :: \(\mathbb{F} A C\)
REAL, \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:) ::W ORK
\(\mathbb{N}\) TEGER (8), \(\mathbb{N}\) TENT (OUT) :: \(\mathbb{E R R}\)

\section*{C INTERFACE}
\#include <sunperfh>
void sfftc2_ (int*iopt, int*n1, int *n2, float *scale, float *x, int *ldx, com plex *y, int *ldy, float
*trigs, int*ifac, float*w ork, int *lw ork, int
*ienc);
void sfftce2_64_ (long *iopt, long *n1, long *n2, float
*scale, float *x, long *ldx, com plex *y, long
*ldy, float *trigs, long *ifac, float *w ork, long
*lw ork, long *ienr);

\section*{PURPOSE}
sfftc2 initializes the trigonom etric weight and factor tables orcom putes the tw o-dim ensional forw ard FastFourier \(T\) ransform of a tw o-dim ensional real array. In com puting the tw o-dim ensionalFFT, one-dim ensionalFFTs are com puted along the colum ns of the inputarray. O ne-dim ensional FFTs are then com puted along the row s of the interm ediate results.

> N2-1 N1-1
\(Y(k 1, k 2)=\) scale \(*\) SUM SUM W \(2 * W 1 * X(1,2)\)
\[
\mathfrak{p}=0 \quad \mathfrak{j}=0
\]
where
k 1 ranges from 0 to \(\mathrm{N} 1-1\) and \(k 2\) ranges from 0 to \(\mathrm{N} 2-1\)
\(i=\operatorname{sqrt}(-1)\)
isign \(=-1\) for forw ard transform
W \(1=\exp \left(i s i g n \star i^{\star} j 1 * k 1 * 2 * p i N 1\right)\)
W \(2=\exp \left(\right.\) isign \(\left.* i^{*} \sum_{2} * k 2 * 2 * p i / N 2\right)\)
In real-to-com plex transform of length N 1 , the \(\mathbb{N} 1 / 2+1\) ) com -
plex output data points stored are the positive-frequency
half of the spectrum of the \(D\) iscrete Fourier T ransform. The other half can be obtained through com plex conjugation and therefore is not stored.

\section*{ARGUMENTS}

IO PT (input)
Integer specifying the operation to be perform ed:
IO PT \(=0\) com putes the trigonom etric weight table and factor table
IO PT = -1 com putes forw ard FFT
N 1 (input)
Integer specifying length of the transform in the first dim ension. N 1 is m ostefficientw hen it is a productofsm allprim es. N \(1>=0\). U nchanged on exit.

N 2 (input)
Integer specifying length of the transform in the second dim ension. N 2 is m ostefficientw hen it is a productofsm allprim es \(\mathrm{N} 2>=0\). U nchanged on exit.

SCA LE (input)
Real scalarby which transform results are scaled. U nchanged on exit. SCA LE is defaulted to 1.0 for F95 \(\mathbb{I N}\) TERFACE.

X (input) X is a com plex array ofdim ensions (LD \(\mathrm{X}, \mathrm{N} 2)\) that contains input data to be transform ed. X and Y can be the sam e array.

LD X (input)
Leading dim ension of X . LD X >=N1 if X is not the sam e aray as Y.Else, LD X = 2*LD Y. U nchanged on exit.

Y (output)
Y is a com plex array ofdim ensions (LD Y,N2) that contains the transform results. \(X\) and \(Y\) can be the sam e array starting at the sam \(e m e m\) ory location, in which case the inputdata are overw ritten by their transform results. O therw ise, it is assum ed that there is no overlap betw een \(X\) and \(Y\) in \(m\) em ory.

LD Y (input)

Leading dim ension of \(Y\). LD \(Y>=N 1 / 2+1\) U nchanged on exit.

TRIGS (input/output)
Real array of length \(2 *(N 1+N 2)\) that contains the trigonom etric weights. The weights are com puted when the routine is called w ith IO PT = 0 and they are used in subsequent calls when IOPT \(=-1\).
U nchanged on exit.
IFAC (input/output)
Integerarray ofdim ension at least \(2 * 128\) that contains the factors of 1 and N2. The factors are com puted when the routine is called w ith IO PT
= 0 and they are used in subsequent calls w hen
IO PT \(=-1\). U nchanged on exit.
W ORK (w orkspace)
Real array of dimension at least MAX N1, \(2 *\) N 2 ) \({ }^{\text {N }}\) CPU S, where NCPU \(S\) is the num berof threads used to execute the routine. The user can also choose to have the routine allocate its own w orkspace (see LW ORK).

LW ORK (input)
Integer specifying workspace size. If LW ORK = 0, the routine w illallocate its ow n w orkspace.
\(\mathbb{E R R}\) (output)
On exit, integer \(\mathbb{E R R}\) has one of the follow ing
values:
0 = norm alretum
\(-1=\mathbb{I O P T}\) is not0 or -1
\(-2=\mathrm{N} 1<0\)
\(-3=N 2<0\)
\(-4=(L D X<N 1)\) or (LD X notequal2*LD \(Y\) when \(X\) and
\(Y\) are sam e array)
\(-5=(\mathbb{L D Y}<\mathrm{N} 1 / 2+1)\)
\(-6=(L W O R K\) not equal 0) and (LWORK <
MAX (N1,2*N2)*NCPUS)
\(-7=m\) em ory allocation failed

\section*{SEE ALSO}
fft

\section*{CAUTIONS}

Y \(\mathbb{N} 1 / 2+1: L D Y,:)\) is used as scratch space. U pon retuming,
the original contents of \(Y \mathbb{N} 1 / 2+1: L D Y,:)\) w ill be lost, \(w\) hereas \(Y(1: \mathbb{N} 1 / 2+1,1 \mathbb{N} 2)\) contains the transform results.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO
- CAUTIONS

\section*{NAME}
sfftc3-initialize the trigonom etric weight and factor
tables or com pute the three-dim ensional forw ard FastF ourier
T ransform of a three-dim ensionalcom plex aray.

\section*{SYNOPSIS}

SU BROUTINE SFFTC3 (IO PT,N1,N2,N3,SCA LE, X,LDX 1, LD X 2, Y, LD Y 1, LD Y 2, TRIGS, \(\mathbb{F} A C, W\) ORK,LW ORK, \(\mathbb{E R R}\) )

LW ORK, ERR
COM PLEX Y (LDY 1,LDY \(2, *\) )
REALX (LDX1,LDX2,*),SCALE,TRIGS (*),W ORK (*)
SU BROUTINE SFFTC3_64 (TOPT,N1,N2,N 3,SCALE, X,LDX 1,LD X 2, Y,LDY 1,LD Y 2, TRIGS, \(\mathbb{F A C}, W\) ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER*8 IO PT,N1,N2,N3,LDX 1, LD X 2, LD Y 1, LD Y 2, FAA (*), LW ORK, \(\mathbb{E R R}\)
COM PLEX Y (LD Y 1,LDY 2,*)
REALX (LDX1,LDX2,*),SCALE,TRIGS (*),W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE FFT3 (IO PT, \(\mathbb{N} 1], \mathbb{N} 2], \mathbb{N} 3],[S C A L E], X,[L D X 1], L D X 2, Y,[L D Y 1]\), LD Y 2, TRIGS, \(\mathbb{F A C}, \mathrm{W}\) ORK, (LW ORK], \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER*4, \(\mathbb{N}\) TEN T ( \(\mathbb{N}\) ) :: IO PT, LD X 2, LD Y 2
\(\mathbb{N} T E G E R * 4, \mathbb{N} T E N T(\mathbb{N}), O P T I O N A L:: N 1, N 2, N 3, L D X 1, L D Y 1\),
LW ORK
REAL, \(\mathbb{N} T E N T(\mathbb{N})\),OPTIONAL ::SCALE
REAL, \(\mathbb{N}\) TENT ( \(\mathbb{N}\) ), \(\mathrm{D} \mathbb{I}\) ENSION \((:,:\) : : X

COM PLEX, \(\mathbb{N}\) TENT (OUT), D \(\mathbb{M} E N S I O N(:,:):: Y\)
REAL, \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} E N S I O N(:):: T R I G S\)
\(\mathbb{N} T E G E R * 4, \mathbb{N}\) TENT ( \(\mathbb{N} O U T\) ) , D \(\mathbb{M} \operatorname{ENSION(:):~:\mathbb {FAC}}\)
REAL, \(\mathbb{N}\) TENT (OUT), D \(\mathbb{M}\) ENSION (:) ::W ORK
\(\mathbb{N}\) TEGER * \(4, \mathbb{N}\) TENT (OUT) :: \(\mathbb{E R R}\)

SU BROU T \(\mathbb{N} E\) FFT3_64 ( \(\mathbb{O}\) PT, \(\mathbb{N} 1], \mathbb{N} 2], \mathbb{N} 3],[S C A L E], X,[L D X 1], L D X 2, Y\), [LD Y 1], LD Y 2, TR IG \(\operatorname{s}, \mathbb{F} A C, W\) ORK, [LW ORK ], \(\mathbb{E R R}\) )
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N}\) TENT ( \(\mathbb{N}\) ) :: IOPT,LDX 2,LDY2
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N})\), OPTIDNAL\(:: N 1, N 2, N 3, L D X 1, L D Y 1\),
LW ORK
REAL, \(\mathbb{N} T E N T(\mathbb{N})\), OPTIONAL ::SCALE
REAL, \(\mathbb{N} T E N T(\mathbb{N})\), D \(\mathbb{M} E N S \mathbb{O}(:,:):\) X
COM PLEX , \(\mathbb{N}\) TENT (OUT), D \(\mathbb{M}\) ENSION (:,:) ::Y
REAL, \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} E N S I O N(:):: T R I G S\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N} O U T), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{F A C}\)
REAL, \(\mathbb{N}\) TENT (OUT), D \(\mathbb{M}\) ENSION (:) ::W ORK
\(\mathbb{N}\) TEGER (8), \(\mathbb{N}\) TENT(OUT) :: \(\mathbb{F R R}\)

\section*{C INTERFACE}
\#include <sunperfh>
void sfftce3_ (int*iopt, int*n1, int*n2, int *n3, float *scale, float *x, int*ldx1, int*ldx2, com plex *y, int *ldy1, int *ldy2, float *trigs, int *ifac, float *W ork, int *lw ork, int *ienr);
void sfftce3_64_ (long *iopt, long *n1, long *n2, long *n3, float *scale, float *x, long *ldx1, long *ldx2, com plex *y, long *ldy1, long *ldy2, float *trigs, long *ifac, float *w ork, long *lw ork, long *ienc);

\section*{PURPOSE}
sfftc3 initializes the trigonom etric w eight and factor tables or computes the three-dim ensional forw ard Fast Fourier T ransform of a three-dim ensional com plex amay.

N 3-1 N 2-1 N 1-1
Y (k1,k2,k3) = scale * SUM SUM SUM W 3*W 2*W 1*X ( \(1, \mathfrak{2}, \mathfrak{\jmath})\)
\[
\mathfrak{j}=0 \quad \mathfrak{2}=0 \quad \mathfrak{j}=0
\]
where
k 1 ranges from 0 to \(\mathrm{N} 1-1 ; k 2\) ranges from 0 to \(\mathrm{N} 2-1\) and \(k 3\)
ranges from 0 to \(\mathrm{N} 3-1\)
\(i=\operatorname{sqnt}(-1)\)
isign \(=-1\) for forw ard transform
W \(1=\exp (i s i g n * i * j * k 1 * 2 * p i / N 1)\)
W \(2=\exp \left(i \operatorname{sign} * i^{\star} 2 * k 2 * 2 * p i N 2\right)\)

W \(3=\exp \left(\right.\) isign*i* \(\left.{ }^{2} * k 3 * 2 * p i N 3\right)\)

\section*{ARGUMENTS}

IO PT (input)
Integer specifying the operation to be perform ed:
IO PT \(=0\) com putes the trigonom etric weight table and factor table
IO \(P T=-1\) com putes forw ard FFT

N 1 (input)
Integer specifying length of the transform in the first dim ension. N 1 is m ostefficientw hen it is a productofsm allprim es. N \(1>=0\). U nchanged on exit.

N 2 (input)
Integer specifying length of the transform in the second dim ension. N 2 is m ostefficientw hen it is a product ofsm allprim es. N \(2>=0\). U nchanged on exit.
N 3 (input)
Integer specifying length of the transform in the third dim ension. N 3 ism ostefficientw hen it is a productofsm allprim es. N \(3>=0\). U nchanged on exit.

SCALE (input)
Realscalarby which transform results are scaled. U nchanged on exit. SCA LE is defaulted to 1.0 for F95 \(\mathbb{N}\) TERFACE.

X (input) X is a real array ofdim ensions (LD X 1, LD X 2, N 3)
that contains inputdata to be transform ed. \(X\) can be sam e array as \(Y\).

LD X 1 (input)
first dim ension ofX. If \(X\) is notsame array as
Y, LDX1 >= N1 Else, LDX1 = 2*LD Y 1 Unchanged on exit.

LD X 2 (input)
second dim ension of X. LD X 2 >= N 2 U nchanged on exit.

Y (output)
Y is a com plex aray of dim ensions (LD Y 1, LD Y 2,
N 3 ) that contains the transform results. \(X\) and \(Y\)
can be the sam e array starting at the sam e \(m\) em ory location, in which case the input data are overw rilten by their transform results. O therw ise, it is assum ed that there is no overlap betw een \(X\) and \(Y\) in \(m\) em ory.

LD Y 1 (input)
firstdin ension ofY. LD Y \(1>=\mathrm{N} 1 / 2+1\) U nchanged on
exit.
LD Y 2 (input)
second dim ension of \(Y\). If \(X\) and \(Y\) are the sam \(e\) aray, LD Y 2 = LD X 2 Else LD Y 2 >= N 2 U nchanged on exit.

TR IG S (input/output)
Realarray of length \(2 *(\mathbb{N} 1+\mathrm{N} 2+\mathrm{N} 3\) ) that contains the trigonom etric w eights. The weights are com puted w hen the routine is called w ith \(\mathbb{I D}\) PT \(=0\) and they are used in subsequent calls when IO PT \(=-1\). U nchanged on exit.

IFAC (input/output)
Integeramay ofdim ension at least \(3 * 128\) that contains the factors of N1, N2 and N3. The factors are com puted w hen the routine is called w ith IOPT \(=0\) and they are used in subsequent calls when \(\mathbb{I O P T}=-1\). U nchanged on exit.

W ORK (w orkspace)
Realarray of dim ension at least MAX \(\mathbb{N}, 2 * N 2,2 * N 3\) )
\(+16 * N 3)\) * NCPUS where NCPUS is the num berof
threads used to execute the routine. The user can
also choose to have the routine allocate its ow \(n\)
w orkspace (see LW ORK).
LW ORK (input)
Integer specifying w orkspace size. If LW ORK = 0, the routine w illallocate its ow n w orkspace.
\(\mathbb{E R R}\) (output)
On exit, integer \(\mathbb{E R R}\) has one of the follow ing
values:
0 = norm alretum
\(-1=10\) PT is not 0 or -1
\(-2=\mathrm{N} 1<0\)
\(-3=\) N \(2<0\)
\(-4=\) N \(3<0\)
\(-5=(\mathbb{L D X} 1<\mathrm{N} 1)\) or ( \(\mathbb{L D} \mathrm{X}\) notequal \(2 *\) LD Y when X
and \(Y\) are sam e array)
\(-6=(\mathrm{LDX} 2<\mathrm{N} 2)\)
\(-7=(\mathbb{L D} Y 1<N 1 / 2+1)\)
\(-8=(L D Y 2<N 2)\) or (LDY 2 notequal LD X 2 when \(X\) and \(Y\) are sam e anray)
\(-9=(L W O R K\) not equal 0) and (LW ORK <
(MAX (N,2*N2,2*N3) + 16*N 3) *N CPUS)
\(-10=m\) em ory allocation failed

\section*{SEE ALSO}
ff

\section*{CAUTIONS}

This routine uses \(Y(\mathbb{N} 1 / 2+1)+1: \operatorname{LD} Y 1,:,:\) ) as scratch space. Therefore, the original contents of this subaray w illbe
lost upon retuming from routine while subaray Y \((1: N 1 / 2+1,1 \mathbb{N} 2,1: N 3)\) contains the transform results.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO

\section*{NAME}
sfftom -initialize the trigonom etric weight and factor tables or com pute the one-dim ensional forw ard FastF ourier \(T\) ransform of a set of realdata sequences stored in a twodim ensionalaray.

\section*{SYNOPSIS}

SUBROUTINE SFFTCM (IOPT,N1,N2,SCALE,X,LDX,Y,LDY,TRIGS, \(\mathbb{F} A C, W\) ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER IOPT,N1,N2,LDX,LDY, \(\mathbb{F A C}(*), L W O R K, \mathbb{E R R}\)
REALX (LDX,*), SCALE,TRIGS (*),W ORK (*)
COM PLEX Y (LDY,*)
SU BROUTINE SFFTCM_64 (IOPT,N1,N2,SCALE,X,LDX,Y,LDY,TRIGS, FFAC,W ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N} T E G E R * 8 \mathbb{I} P T, N 1, N 2, L D X, L D Y, \mathbb{F A C}(*), L W O R K, \mathbb{E R R}\)
REALX (LDX,*),SCALE,TRIGS (*),W ORK (*)
COM PLEX Y (LDY,*)

\section*{F95 INTERFACE}

SU BROUTINE FFTM (IOPT, \(\mathbb{N} 1], \mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S\), تAC,W ORK, [LW ORK], \(\mathbb{E R R}\) )
\(\mathbb{N} \operatorname{TEGER} * 4, \mathbb{N}\) TENT( \(\mathbb{N}\) ) :: \(\mathbb{I O P T}\)
\(\mathbb{N}\) TEGER*4, \(\mathbb{N}\) TENT \((\mathbb{N})\),OPTIONAL ::N 1,N2,LDX,LDY,LW ORK
REAL, \(\mathbb{N} T E N T(\mathbb{N})\), OPTIONAL ::SCALE
REAL, \(\mathbb{N}\) TENT ( \(\mathbb{N}\) ), \(\mathrm{D} \mathbb{I}\) ENSION (:,:) : : X
COMPLEX, \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:,:) ::Y
REAL, \(\mathbb{N}\) TENT ( \(\mathbb{N O U T}\) ), D \(\mathbb{I M}\) ENSION (:) ::TRIGS
\(\mathbb{N}\) TEGER*4, \(\mathbb{N}\) TENT ( \(\mathbb{N} O U T\) ), D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{F A C}\)

SU BROUTINE FFTM _64 (IOPT, \(\mathbb{N} 1], \mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S\), تAC, W ORK, [LW ORK], ERR)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N}):: \mathbb{I D P T}\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} T E N T(\mathbb{N}), O P T I O N A L:: N 1, N 2, L D X, L D Y, L W\) ORK
REAL, \(\mathbb{N} T E N T(\mathbb{N})\),OPTIONAL ::SCALE
REAL, \(\mathbb{N} T E N T(\mathbb{N}), D \mathbb{I} \operatorname{ENSION}(:,:):\) X
COM PLEX, \(\mathbb{I N T E N T}\) (OUT),D \(\mathbb{M}\) ENSTON (:,:) ::Y
REAL, \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{I M}\) ENSION (:) ::TRIGS
\(\mathbb{N}\) TEGER (8), \(\mathbb{N}\) TENT ( \(\mathbb{N O U T}\) ), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{F A C}\)
REAL, \(\mathbb{I N T E N T}(O U T), D \mathbb{M} E N S I O N(:):: W\) ORK
\(\mathbb{N} T E G E R(8), \mathbb{N} T E N T\) (OUT) :: \(\mathbb{E R R}\)

\section*{C INTERFACE}
\#include <sunperfh>
void sfftem _ (int*iopt, int*n1, int *n2, float *scale,
float *x, int *ldx, com plex *y, int *ldy, float
*trigs, int *ifac, float *w ork, int *lw ork, int
*ient);
void sfffom_64_ (long *iopt, long *n1, long *n2, float
*scale, float *x, long *ldx, com plex *y, long
*ldy, float *trigs, long *ifac, float *w ork, long
*ly ork, long *ienr);

\section*{PURPOSE}
sfftom initializes the trigonom etric weight and factor tables orcom putes the one-dim ensional forw ard FastFourier Transform of a set of real data sequences stored in a tw o-dim ensionalarray:

N 1-1
\(Y(k, l)=\) scale * SUM W *X (jl)
\[
\dot{j} 0
\]
where
\(k\) ranges from 0 to N 1-1 and lranges from 0 to N 2-1
\(i=\operatorname{sqrt}(-1)\)
isign \(=-1\) for forw ard transform
\(W=\exp \left(i s i g n * i^{\star}{ }^{j} \star k * 2 \star\right.\) piN 1)
In real-to-com plex transform of length \(N 1\), the \((N 1 / 2+1)\) com -
plex output data points stored are the positive-frequency half of the spectrum of the discrete Fourier transform. The other half can be obtained through com plex conjugation and therefore is not stored.

\section*{ARGUMENTS}

IO PT (input)
Integer specifying the operation to be perform ed:
IO PT \(=0\) com putes the trigonom etric weight table and factor table
IO \(P T=-1\) com putes forw ard FFT

N 1 (input)
Integer specifying length of the input sequences. N 1 is m ostefficientw hen it is a productofsm all prim es. N \(1>=0\). U nchanged on exit.

N 2 (input)
Integerspecifying num ber of input sequences. N 2 \(>=0\). U nchanged on exit.

SCALE (input)
Real scalarby which transform results are scaled. U nchanged on exit. SCA LE is defaulted to 1.0 for F95 \(\mathbb{I N}\) TERFACE.

X (input) X is a realarray of dim ensions (LD \(\mathrm{X}, \mathrm{N} 2\) ) that contains the sequences to be transform ed stored in its colum ns.

LD X (input)
Leading dim ension of \(X\). If \(X\) and \(Y\) are the same aray, LDX \(=2 *\) LD E Else LD \(\mathrm{X}>=\mathrm{N} 1\) U nchanged on exit.

Y (output)
\(Y\) is a com plex array ofdim ensions (LD Y ,N 2) that contains the transform results of the input sequences. \(X\) and \(Y\) can be the sam e array starting at the sam e \(m\) em ory location, in which case the input sequences are overw rilten by their transform results. O therw ise, it is assum ed that there is no overlap betw een \(X\) and \(Y\) in \(m\) em ory.

LD \(Y\) (input)
Leading dim ension of \(\mathrm{Y} . \operatorname{LD} \mathrm{Y}>=\mathrm{N} 1 / 2+1\) U nchanged on exit.

TRIG S (input/output)
Realarray of length \(2 * \mathrm{~N} 1\) that contains the trigonom etric w eights. The w eights are com puted w hen
the routine is called \(w\) ith IO PT \(=0\) and they are used in subsequent calls when IOPT \(=-1\). U nchanged on exit.

FAC (input/output)
Integer array of dim ension at least 128 that contains the factors of 1 . The factors are com puted when the routine is called w ith IO PT = 0 and they are used in subsequent calls when IOPT \(=-1\). U nchanged on exit.

W ORK (w orkspace)
Real array ofdim ension at least 1 1. The user can also choose to have the routine allocate its ow \(n\) w orkspace (see LW ORK).

LW ORK (input)
Integer specifying w orkspace size. If LW ORK = 0,
the routine w illallocate its ow n w orkspace.
ERR (output)
On exit, integer \(\mathbb{E R R}\) has one of the follow ing
values:
\(0=\) norm alretum
\(-1=\mathbb{1 O P T}\) is not 0 or -1
\(-2=\mathrm{N} 1<0\)
\(-3=N 2<0\)
\(-4=(\mathbb{L D X}<\mathrm{N} 1)\) or (LDX notequal2*LD \(Y\) when \(X\) and
Y are sam e array)
\(-4=(\mathbb{L D Y}<\mathrm{N} 1 / 2+1)\)
\(-6=(\mathrm{LW} O R K\) notequal0) and (LWORK <N1)
\(-7=m\) em ory allocation failed

\section*{SEE ALSO}
ff

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgbbrd - reduce a realgeneralm -by-n band \(m\) atrix \(A\) to upper bidiagonal form B by an orthogonal transform ation

\section*{SYNOPSIS}
```

SUBROUTINE SGBBRD NECT,M,N,NCC,KL,KU,AB,LDAB,D,E,Q,LDQ,
PT,LDPT,C,LDC,W ORK,INFO)
CHARACTER * 1 VECT
\mathbb{NTEGERM,N,NCC,KL,KU,LDAB,LDQ,LDPT,LDC,INFO}
REALAB (LDAB ,*),D (*),E (*),Q (LD Q ,*),PT (LDPT,*),C (LD C ,*),
W ORK (*)
SUBROUT\mathbb{NE SGBBRD_64NECT,M,N,NCC,KL,KU,AB,LDAB,D,E,Q,LDQ,}
PT,LDPT,C,LDC,W ORK,INFO)
CHARACTER * 1VECT
INTEGER*8M,N,NCC,KL,KU,LDAB,LDQ,LDPT,LDC, INFO
REALAB (LDAB,*),D (*),E (*),Q (LDQ ,*),PT (LDPT,*),C (LDC,*),
W ORK (*)

```
F95 INTERFACE
    SU BROUTINE GBBRD \(N E C T, M, \mathbb{N}], \mathbb{N C C}], K L, K U, A B,[L D A B], D, E, Q\),
        [LD Q ], PT, [LDPT ], C , [LD C ], [W ORK ], [ \(\mathbb{N} F \mathrm{~F}]\) )
    CHARACTER (LEN=1) ::VECT
    \(\mathbb{N} T E G E R:: M, N, N C C, K L, K U, L D A B, L D Q, L D P T, L D C, \mathbb{N F O}\)
    REAL,D \(\mathbb{M}\) ENSION (:) ::D, E,W ORK
    REAL,D \(\mathbb{I M}\) ENSION (:,:) ::AB, \(\mathrm{Q}, \mathrm{PT}, \mathrm{C}\)
    SU BROUTINE GBBRD_64 \(N E C T, M, \mathbb{N}], \mathbb{N C C}], K L, K U, A B,[L D A B], D, E\),
        \(Q,[L D Q], P T,[L D P T], C,[L D C],[W O R K],[\mathbb{N F O}])\)

CHARACTER ( \(L E N=1\) ) : : VECT
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{NCC}, \mathrm{KL}, \mathrm{KU}, \mathrm{LD} A B, L D Q, L D P T, L D C, \mathbb{N} F O\)
REAL,D \(\mathbb{I M}\) ENSION (:) ::D , E,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::AB,Q,PT,C

\section*{C INTERFACE}
\#include <sunperfh>
void sgbbord (charvect, intm, intn, intncc, int kl, int ku, float *ab, int ldab, float *d, float *e, float * \(q\), int ldq, float *pt, int ldpt, float * C , int ldc, int*info);
void sgbbird_64 (charvect, long m, long n, long ncc, long kl, long ku, float *ab, long ldab, float *d, float *e, float*q, long ldq, float*pt, long ldpt, float * C, long ldc, long *info);

\section*{PURPOSE}
sgbbrd reduces a realgeneralm \(-b y-n\) band \(m\) atrix \(A\) to upper bidiagonal form B by an orthogonaltransform ation: Q '* A * \(P=B\).

The routine com putes B, and optionally form s Q or P', or com putes \(Q\) *C for given \(m\) atrix \(C\).

\section*{ARGUMENTS}

\section*{VECT (input)}

Specifies w hether ornot the \(m\) atrices \(Q\) and \(P\) 'are
to be form ed. = N ': do not form Q orP ';
\(=Q\) ': form Q only;
= P ': form P 'only;
= B ': form both.

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of Colm ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NCC (input)
The num berof colum ns of the m atrix C. NCC \(>=0\).

KL (input)
The num ber of subdiagonals of them atrix A . K L >= 0.

KU (input)
The num ber of superdiagonals of the m atrix A. K U \(>=0\).

AB (input/output)
REA L array, dim ension (LD A B N ) On entry, the m -by-n band \(m\) atrix \(A\), stored in row \(s 1\) to \(K L+K U+1\). The \(j\) th column ofA is stored in the \(j\) th column of the array AB as follow s: AB \((k u+1+i-j)=A(i, j)\) form ax \((1, j k u)<=i<=m\) in \((m, j+k l)\). On exit, \(A\) is overw ritten by values generated during the reduction.

LDAB (input)
The leading dim ension of the array A. LD AB >= K L+KU+1.

D (output)
REAL array, dim ension ( \(m\) in \(M, N\) )) The diagonal ele\(m\) ents of the bidiagonalm atrix B .

E (output)
REAL array, dim ension ( \(m\) in \(M, N\) )-1) The superdiagonalelem ents of the bidiagonalm atrix \(B\).

Q (output)
REAL array, dim ension (LD Q M ) IfVECT = Q 'or B', the \(m\) by \(m\) orthogonalm atrix \(Q\). IfVECT \(=N\) 'or P ', the array Q is not referenced.

LDQ (input)
The leading dim ension of the array \(Q\). LDQ >= \(m a x(1, M)\) ifVECT \(=Q\) 'or \(B\) '; LD \(Q>=1\) otherw ise.

PT (output)
REAL array, dim ension (LDPT,N) If VECT = P' or B ', the n -by-n orthogonalm atrix \(\mathrm{P}^{\prime}\). IfVECT \(=\) \(N\) 'or \(Q\) ', the anay PT is not referenced.

LDPT (input)
The leading dim ension of the array PT. LD PT >= \(\max (1, N)\) if VECT = P'or B ; LDPT >= 1 otherw ise.

C (input/output)
REAL array, dim ension (LD C NCC) On entry, an m-byncc \(m\) atrix \(C\). On exit, \(C\) is overw ritten by \(Q\) *C .
\(C\) is not referenced if \(\mathrm{NCC}=0\).
LD C (input)

The leading dim ension of the array \(C\). LD \(C>=\) \(\max (1, M)\) if \(N C C>0 ; L D C>=1\) if \(N C C=0\).
W ORK (w orkspace)
REAL array, dim ension ( \(2 *\) M AX M \(N\) ))
\(\mathbb{I N} F O\) (output)
= 0: successfulexit.
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgbcon -estim ate the reciprocal of the condition num ber of a real general band \(m\) atrix \(A\), in either the 1-norm or the infinity-norm,

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGBCON NORM,N,NSUB,NSUPER,A,LDA,\mathbb{PIVOT,ANORM,}}\mathbf{N},\mp@code{N},
RCOND,W ORK,W ORK2,INFO)
CHARACTER * 1 NORM
\mathbb{NTEGER N,NSUB,NSUPER,LDA, INFO}
INTEGER \mathbb{PIVOT (*),W ORK2 (*)}
REALANORM,RCOND
REAL A (LDA,*),W ORK (*)
SUBROUT\mathbb{NE SGBCON_64 NORM,N,NSUB,NSUPER,A,LDA, IPIVOT,ANORM,}
RCOND,W ORK,W ORK2,INFO)
CHARACTER * 1 NORM
\mathbb{NTEGER*8 N,NSUB,N SUPER,LDA, NNFO}
\mathbb{NTEGER*8 P\mathbb{IVOT (*),W ORK2 (*)}}\mathbf{(})
REALANORM,RCOND
REALA (LDA,*),W ORK (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE GBCON $\mathbb{N} O R M, \mathbb{N}], N S U B, N S U P E R, A,[L D A], \mathbb{P} \mathbb{I} O T, A N O R M$, RCOND, [W ORK], [W ORK 2], [ $\mathbb{N F O}$ ])
CHARACTER (LEN=1) ::NORM
$\mathbb{N} T E G E R:: N, N S U B, N S U P E R, L D A, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}, \mathrm{W}$ ORK 2
REAL ::ANORM,RCOND

```

REAL,D \(\mathbb{I M}\) ENSION (:) ::W ORK
REAL,D IM ENSION (:,:) ::A

SU BROUTINE GBCON_64 \(\mathbb{N} O R M, \mathbb{N}], N S U B, N S U P E R, A,[L D A], \mathbb{P} \mathbb{I} O T, A N O R M\), RCOND, [WORK], [WORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::NORM
\(\mathbb{N}\) TEGER (8) :: N , N SUB , N SUPER, LD A , \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T\), W ORK 2
REAL ::ANORM,RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::W ORK
REAL,D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void sgbcon (charnorm, intn, intnsub, int nsuper, float
*a, int lda, int *ipivot, float anorm, float
*rcond, int*info);
void sgbcon_64 (charnorm , long n, long nsub, long nsuper, float *a, long lda, long *ipivot, floatanorm, float *roond, long *info);

\section*{PURPOSE}
sgbcon estim ates the reciprocal of the condition num ber of a real general band m atrix A, in either the 1 -nom orthe infinity-norm, using the LU factorization computed by SGBTRF .

A \(n\) estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as
\(\operatorname{RCOND}=1 /(\operatorname{norm}(A) * \operatorname{norm}(\operatorname{inv}(A)))\).

\section*{ARGUMENTS}

NORM (input)
Specifies w hether the 1-norm condition num ber or the infinity-norm condition num ber is required:
= I'or \(\mathrm{D}^{\prime}\) : 1-norm;
= \(\mathrm{I}^{\prime}: \quad\) Infinity-norm .

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

N SUB (input)
The num ber of subdiagonals \(w\) thin the band of \(A\).
N SUB \(>=0\) 。

\section*{N SU PER (input)}

The num ber of superdiagonals w ithin the band of A. N SU PER \(>=0\).

A (input) D etails of the LU factorization of the band \(m\) atrix A, as com puted by SGBTRF. U is stored as an upper triangularband \(m\) atrix \(w\) ith N SU B \(+N\) SU PER superdiagonals in row s 1 to NSUB+NSUPER+1, and them ultipliers used during the factorization are stored in row sN SU B +N SU PER + 2 to \(2 *\) N SU B + N SU PER +1 .

LD A (input)
The leading dim ension of the anay A. LDA >= \(2 *\) N SU B + N SU PER +1 .
\(\mathbb{P I V O T}\) (input)
The pivot indices; for \(1<=i<=N\), row \(i\) of the \(m\) atrix \(w\) as interchanged w ith row \(\mathbb{P}\) IV OT (i).

ANORM (input)
IfNORM = ' 1 'or 0 ', the 1 -norm of the original \(m\) atrix \(A\). IfNORM = \(I\) ', the infinity-norm of the originalm atrix A.

RCOND (output)
The reciprocal of the condition number of the \(m\) atrix \(A\), com puted as RCOND \(=1 /(\) norm (A) * nom (inv (A))).

W ORK (w orkspace)
dim ension ( \(3 * \mathrm{~N}\) )

W ORK 2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgbequ - com pute row and colum n scalings intended to equilibrate an M -by-N band matrix A and reduce its condition num ber

\section*{SYNOPSIS}
```

SUBROUTINE SGBEQU M,N,KL,KU,A,LDA,R,C,ROW CN,
COLCN,AMAX,INFO)
\mathbb{N TEGER M,N,KL,KU,LDA, NNFO}
REAL ROW CN,COLCN,AMAX
REALA (LDA,*),R (*),C (*)
SUBROUT\mathbb{NE SGBEQU_64M,N,KL,KU,A,LDA,R,C,ROW CN,}
COLCN,AMAX,\mathbb{NFO)}
\mathbb{NTEGER*8M,N,KL,KU,LDA,}\mathbb{N}FO
REAL ROW CN,COLCN,AMAX
REAL A (LDA,*),R (*),C (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GBEQU ( \(\mathbb{M}], \mathbb{N}], K L, K U, A,[L D A], R, C\),
ROW CN, COLCN,AMAX, \(\mathbb{N} F O]\) )
\(\mathbb{N}\) TEGER :: \(\mathrm{M}, \mathrm{N}, \mathrm{KL}, \mathrm{KU}, \mathrm{LD} A, \mathbb{N} F \mathrm{O}\)
REAL ::ROW CN, COLCN,AMAX
REAL,D IM ENSION (:) ::R,C
REAL,D \(\mathbb{M}\) ENSION (: : : : : A

SU BROUTINE GBEQU_64 (M) ROW CN, COLCN,AMAX, [ \(\mathbb{N} F O]\) )
\(\mathbb{N}\) TEGER (8) :: M , N , KL, KU, LDA, \(\mathbb{N} F O\)
REAL ::ROW CN, COLCN, AMAX
REAL,D \(\mathbb{I M} E N S I O N\) (:) ::R,C
REAL,D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void sgbequ (intm , intn, intkl, int ku, float *a, int lda, float *r, float *C, float * row cn, float * colen, float *am ax, int*info);
void sgbequ_64 (long m, long n, long kl, long ku, float *a, long lda, float * \(r\), float * c, float *row cn, float *colen, float *am ax, long *info);

\section*{PURPOSE}
sgbequ com putes row and colum \(n\) scalings intended to equilibrate an M boy-N band m atrix A and reduce its condition num ber. \(R\) retums the row scale factors and \(C\) the colum \(n\) scale factors, chosen to try to \(m\) ake the largestelem ent in each row and colum \(n\) of the \(m\) atrix \(B \quad w\) ith elem ents \(B(i, j)=R(i) \star A(i, j) * C(i)\) have absolute value 1.

R (i) and C (i) are restricted to be betw een SM LN UM = sm allest safe num ber and B IG N UM = largestsafe num ber. U se of these scaling factors is notguaranteed to reduce the condition num berofA butw orks w ellin practice.

\section*{ARGUMENTS}

M (input) The num ber of row s of the matrix \(A . M>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

K L (input)
The num ber of subdiagonals \(w\) thin the band of \(A\). \(\mathrm{KL}>=0\) 。

KU (input)
The num ber of superdiagonals \(w\) thin the band of \(A\). \(K U>=0\) 。

A (input) The band \(m\) atrix \(A\), stored in row 1 to \(K L+K U+1\). The \(j\) th colum \(n\) of \(A\) is stored in the \(j\) th column of the array \(A\) as follow s: A \((k u+1+i-j, j)=A(i, j)\)
form ax \((1, j \mathrm{j} k)<=i<=m\) in \((m, j+k l)\).

LD A (input)
The leading dim ension of the aray A. LDA >= \(K L+K U+1\).

R (output)
If \(\mathbb{N} F O=0\), or \(\mathbb{N} F O>M, R\) contains the row scale
factors forA.
C (output)
If \(\mathbb{N} F O=0, C\) contains the colum \(n\) scale factors forA.
ROW CN (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O>M, R O W C N\) contains the ratio
of the sm allest \(R\) (i) to the largest \(R\) (i). If
ROW CN >= 0.1 and AM AX is neither too large nor too
sm all, it is notw orth scaling by \(R\).
COLCN (output)
If \(\mathbb{N} F O=0, C O L C N\) contains the ratio of the sm allest C (i) to the largestC (i). IfC OLCN >=0.1, it is notw orth scaling by \(C\).

\section*{AM AX (output)}

A bsolute value of largestm atrix elem ent. If A M A X is very close to overflow orvery close to underflow , the m atrix should be scaled.
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
<0: if \(\mathbb{I N}\) FO = -i, the i-th argum ent had an illegalvalue
>0: if \(\mathbb{N} F O=i\), and \(i\) is
\(<=\mathrm{M}\) : the i-th row ofA is exactly zero
> M : the ( \(j-\mathrm{M}\) ) -th collum n ofA is exactly zero

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgbom v -perform one of the m atrix-vectoroperations \(\mathrm{y}:=\) alpha*A *x + beta* \(y\) ory \(:=\) alpha*A *x + beta* \(y\)

\section*{SYNOPSIS}
```

SU BROUTINE SGBMV (TRANSA,M,N,NSUB,NSUPER,A LPHA,A,LDA,X, INCX,
BETA,Y,\mathbb{NCY)}
CHARACTER * 1 TRANSA
\mathbb{NTEGERM,N,NSUB,NSUPER,LDA, INCX,INCY}
REAL ALPHA,BETA
REALA (LDA,*),X (*),Y(*)
SU BROUT\mathbb{NE SGBM V_64 (TRANSA,M,N ,N SUB,N SUPER,ALPHA,A,LDA,X,}
INCX,BETA,Y,\mathbb{NCY)}
CHARACTER * 1 TRANSA
\mathbb{NTEGER*8M,N,NSUB,NSUPER,LDA,INCX,INCY}
REALALPHA,BETA
REAL A (LDA,*),X (*),Y (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GBMV ([TRANSA], \(\mathbb{M}], \mathbb{N}], N S U B, N S U P E R, A L P H A, A,[L D A], X\), \([\mathbb{N} C X], B E T A, Y,[\mathbb{N C Y}])\)

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N} T E G E R:: M, N, N S U B, N S U P E R, L D A, \mathbb{N C X}, \mathbb{N} C Y\)
REAL ::ALPHA,BETA
REAL,D IM ENSION (:) :: X,Y
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A
SU BROUTINE GBM V_64 ([TRANSA], M ], \(\mathbb{N}], N \operatorname{SUB}, N \operatorname{SUPER}, A L P H A, A,[L D A]\),
\(X,[\mathbb{N C X}], B E T A, Y,[\mathbb{N C Y}])\)

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N}\) TEGER (8) :: M , N , N SU B , N SUPER , LD A , \(\mathbb{N} C X, \mathbb{N} C Y\)
REAL ::ALPHA,BETA
REAL,D \(\mathbb{I M}\) ENSION (:) :: X,Y
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void sgbm v (chartransa, intm, intn, intnsub, int nsuper, float alpha, float *a, int lda, float *x, int incx, floatbeta, float *y, int incy);
void sgbm v_64 (chartransa, long m, long n, long nsub, long nsuper, float alpha, float *a, long lda, float *x, long incx, floatbeta, float *y, long incy);

\section*{PURPOSE}
sgbom v perform s one of the \(m\) atrix-vector operations \(y:=\) alpha*A *x + beta*y ory := alpha*A *x + beta*y, w here alpha and beta are scalars, \(x\) and \(y\) are vectors and \(A\) is an \(m\) by \(n\) band m atrix, w ith nsub sub-diagonals and nsupersuperdiagonals.

\section*{ARGUMENTS}

TRANSA (input)
On entry, TRAN SA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} y:=\) alpha*A * \(\mathrm{x}+\) beta* \(^{\mathrm{y}}\).

TRANSA \(=\) T'or \(t^{\prime} \mathrm{y}:=\) alpha*A \({ }^{*} \mathrm{x}+\) beta* y .

TRANSA \(=C^{\prime}\) or \(C^{\prime} y:=\) alpha*A *x + beta* \(y\).

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

M (input)
O n entry, M specifies the num ber of row s of the \(m\) atrix \(A . M>=0\). U nchanged on exit.

N (input)

O n entry, \(N\) specifies the num ber of colum ns of the \(m\) atrix \(A . N>=0\). U nchanged on exit.

NSUB (input)
On entry, NSUB specifies the number of subdiagonals of them atrix A.NSUB \(>=0\). U nchanged on exit.

N SU PER (input)
On entry, N SU PER specifies the num ber of superdiagonals of the m atrix A. N SU PER \(>=0\). U nchanged on exit.
ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
Before entry, the leading (nsub + nsuper+1) by \(n\) part of the array A m ust contain the matrix of coefficients, supplied colum \(n\) by colum \(n\), with the leading diagonal of the \(m\) atrix in row (nsuper+ 1 ) of the array, the first super-diagonal starting at position 2 in row nsuper, the firstsubdiagonal starting atposition 1 in row (nsuper + 2 ), and so on. Elem ents in the array A that do not comespond to elem ents in the band matrix (such as the top leftnsuperby nsupertriangle) are not referenced. The follow ing program segm ent w ill transfer a band \(m\) atrix from conventional full \(m\) atrix storage to band storage:
\[
\text { DO } 20, \mathrm{~J}=1, \mathrm{~N}
\]
\[
K=N S U P E R+1-J
\]
\[
\text { DO } 10, I=\text { M AX }(1, J-N \text { SU PER }), M \mathbb{N}(M, J+
\]
N SUB )
\[
A(K+I, J)=m \operatorname{atrix}(I, J)
\]

10 CONTINUE
20 CONTINUE
U nchanged on exit.
LDA (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A \(>=\) (
nsub + nsuper+ 1 ). U nchanged on exit.
X (input)
\((1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C X))\) when TRANSA \(=\mathrm{N}\) 'or
\(h^{\prime}\) and at least \((1+(m-1) * a b s(\mathbb{N} C X))\)
otherw ise. Before entry, the increm ented amay \(X\)
m ustcontain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X<>0\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then \(Y\) need notbe set on input. U nchanged on exit.

Y (input/output)
\((1+(m-1) \star \operatorname{abs}(\mathbb{N} C Y))\) when TRANSA \(=\mathrm{N}\) 'or \(h^{\prime}\) and at least \((1+(n-1) * a b s(\mathbb{N} C Y))\)
otherw ise. Before entry, the increm ented array \(Y\) m ust contain the vectory. On exit, Y is overw ritten by the updated vectory.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents ofY. \(\mathbb{N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgbrifs - im prove the com puted solution to a system of linear equations when the coefficientm atrix is banded, and provides errorbounds and backw ard error estim ates for the solution

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGBRFS (TRANSA,N,KL,KU,NRHS,A,LDA,AF,LDAF,}
\mathbb{PIVOT,B,LDB,X,LDX,FERR,BERR,W ORK,W ORK 2, INFO)}
CHARACTER * 1 TRANSA
INTEGERN,KL,KU,NRHS,LDA,LDAF,LDB,LDX, INFO
INTEGER \mathbb{PIVOT (*),W ORK 2 (*)}
REAL A (LDA,*), AF (LDAF,*), B (LDB ,*), X (LDX ,*), FERR (*),
BERR (*),W ORK (*)
SUBROUT\mathbb{NE SGBRFS_64(TRANSA,N,KL,KU,NRHS,A,LDA,AF,LDAF,}
\mathbb{PIVOT,B,LDB,X,LDX,FERR,BERR,W ORK,W ORK2,\mathbb{NFO)}}\mathbf{(})=
CHARACTER * 1 TRANSA
\mathbb{N}TEGER*8N,KL,KU,NRHS,LDA,LDAF,LDB,LDX,INFO
INTEGER*8 \mathbb{PIVOT (*),W ORK2 (*)}
REAL A (LDA,*), AF (LDAF,*), B (LDB ,*), X (LDX,*), FERR (*),
BERR (*),W ORK (*)

```

\section*{F95 INTERFACE}
```

SUBROUTINE GBRFS ([TRANSA], $\mathbb{N}], K L, K U, \mathbb{N} R H S], A,[L D A], A F$, [LDAF], $\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}], \mathrm{X},[\mathrm{LD} \mathrm{X}], \mathrm{FERR}, \mathrm{BERR},[\mathrm{W}$ ORK ], [W ORK 2], [ $\mathbb{N} F O$ ])

```

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N}\) TEGER ::N,KL,KU,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{I} O T, W\) ORK 2 REAL,D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK REAL,D \(\mathbb{I}\) ENSION (:,:) ::A,AF,B,X

SU BROUTINE GBRFS_64 ([TRANSA], \(\mathbb{N}], K L, K U, \mathbb{N} R H S], A,[L D A]\), \(A F,[L D A F], \mathbb{P} \mathbb{I} O T, B,[L D B], X,[L D X], F E R R, B E R R,[W O R K]\), [W ORK2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N} T E G E R(8):: N, K L, K U, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N F O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I} O T, W\) ORK2
REAL,D \(\mathbb{I}\) ENSION (:) ::FERR,BERR,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A,AF,B,X

\section*{C INTERFACE}
\#include <sunperfh>
void sgbrfs (char transa, intn, intkl, int ku, int nrhs, float *a, int lda, float *af, int ldaf, int *ípivot, float*b, intldb, float *x, int ldx, float * ferr, float *berr, int *info);
void sgbrfs_64 (chartransa, long n, long kl, long ku, long nrhs, float *a, long lda, float*af, long ldaf, long *ipivot, float*b, long ldb, float *x, long ldx, float * ferr, float *berr, long *info);

\section*{PURPOSE}
sgbris im proves the com puted solution to a system of linear equations when the coefficientm atrix is banded, and provides errorbounds and backw ard error estim ates for the solution.

\section*{ARGUMENTS}

TRANSA (input)
Specifies the form of the system ofequations:
\(=\mathrm{N}: \mathrm{A} * \mathrm{X}=\mathrm{B} \quad(\mathrm{N}\) o transpose)
\(=T{ }^{\prime}: A * * T X=B \quad\) (Transpose)
\(=C^{\prime}: A * * H * X=B \quad\) (C onjugate transpose \(=T\) ran -
spose)

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).

K L (input)

The num berof subdiagonals w ithin the band of A. \(K L>=0\).

KU (input)
The num ber of superdiagonals within the band of A. \(K U>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the \(m\) atrioes \(B\) and \(X\). NRH \(S>=0\).

A (input) The originalband m atrix A, stored in row s 1 to \(K L+K U+1\). The \(j\) th column ofA is stored in the \(j\) th colum n of the array A as follow s: A (ku+1+i\(j, j)=A(i, j)\) form ax \((1, j \mathrm{jku})<=i<=m\) in \((n, j+k l)\).

LD A (input)
The leading dim ension of the array A. LDA >= K L+KU+1.

AF (input)
D etails of the LU factorization of the band \(m\) atrix A , as com puted by SG BTRF. U is stored as an upper triangularband \(m\) atrix \(w\) th \(K L+K U\) superdiagonals in row \(s 1\) to \(K L+K U+1\), and the \(m\) ultipliers used during the factorization are stored in row S \(K L+K U+2\) to \(2 * K L+K U+1\).

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(2 * K L * K U+1\).
\(\mathbb{P I V O T}\) (input)
The pivotindices from SGBTRF; for \(1<=i<=N\), row \(i\) of the \(m\) atrix \(w\) as interchanged w ith row \(\mathbb{P}\) IV OT (i).
\(B\) (input) The righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, N)\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by SGBTRS. On exit, the im proved solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the array X . LDX >= \(\max (1, N)\).

\section*{FERR (output)}

The estim ated forw ard errorbound for each solution vector \(X(\mathcal{)}\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{O}), \operatorname{FERR}(\underset{)}{(1)}\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in \((X(\mathcal{J})-X\) TRU \(E\) ) divided by the \(m\) agnitude of the largestelem ent in \(\mathrm{X}(\mathcal{j})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard emror of each solution vector \(X\) (i) (i.e., the sm allest relative change in any elem entofA orB thatm akes X (i) an exactsolution).

W ORK (w orkspace)
dim ension ( \(3 \star \mathrm{~N}\) )

W ORK 2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgbsv -com pute the solution to a real system of linear equations \(A * X=B\), where \(A\) is a band \(m\) atrix of order \(N\) \(w\) th \(K L\) subdiagonals and \(K U\) superdiagonals, and \(X\) and \(B\) are
N -by-N R H S m atrices

\section*{SYNOPSIS}

```

\mathbb{NTEGERN,KL,KU,NRHS,LDA,LDB,INFO}
INTEGER \mathbb{PIVOT (*)}
REALA (LDA,*),B (LDB,*)

```

```

        \mathbb{NFO)}
    NNTEGER*8 N,KL,KU,NRHS,LDA,LDB,NNFO
INTEGER*8 \mathbb{PIVOT (*)}
REALA (LDA,*),B (LDB,*)

```

\section*{F95 INTERFACE}

SU BROUTINE GBSV ( \(\mathbb{N}], K L, K U, \mathbb{N} R S], A,[L D A], \mathbb{P} \mathbb{I V} \operatorname{T}, \mathrm{B},[\mathrm{LDB}]\), [ \(\mathbb{N}\) FO ])
\(\mathbb{N}\) TEGER ::N,KL,KU,NRHS,LDA,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
REAL,D \(\mathbb{M}\) ENSION (: : : : :: A, B
SU BROUTINEGBSV_64 (N) \(\mathbb{N}, \mathrm{KL}, \mathrm{KU}, \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I} O T, B\), [LDB], [ \(\mathbb{N} F O]\) )
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{KL}, \mathrm{KU}, \mathrm{NRH}, \mathrm{LDA}, \mathrm{LD} B, \mathbb{N} F \mathrm{O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V O T}\)
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A, B

\section*{C INTERFACE}
\#include <sunperfh>
void sgbsv (intn, intkl, intku, intnihs, float *a, int lda, int *ipívot, float *b, int ldb, int *info);
void sgbsv_64 (long n, long kl, long ku, long nrhs, float *a, long lda, long *ipivot, float *b, long ldb, long *info);

\section*{PURPOSE}
sgbsv com putes the solution to a real system of linearequations A * \(X=B\), where A is a band \(m\) atrix of orderN w th K L subdiagonals and \(K U\) superdiagonals, and \(X\) and \(B\) are \(N\)-byN RH S m atriges.

The LU decom position \(w\) ith partialpivoting and row interchanges is used to factorA asA \(=\mathrm{L} * \mathrm{U}\), where L is a product of perm utation and unit low er triangularm atrices \(w\) ith \(K L\) subdiagonals, and \(U\) is uppertriangularw ith K L+K U superdiagonals. The factored form of \(A\) is then used to solve the system of equations \(A * X=B\).

\section*{ARGUMENTS}

N (input) The num ber of linear equations, i.e., the order of them atrix \(A . N>=0\).

KL (input)
The num ber of subdiagonals w ithin the band of A. \(K L>=0\).

KU (input)
The num ber of superdiagonals \(w\) ithin the band of \(A\). \(K U>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS >=0.

A (input/output)
On entry, the m atrix A in band storage, in row s
\(K L+1\) to \(2 * K L+K U+1\); row s 1 to \(K L\) of the anay need notbe set. The \(j\) th column of A is stored in the \(j\) th collumn of the array A as follows: \(A(K L+K U+1+i-j, j)=A(i, j)\) for \(\max (1, j\) \(K U)<=i<=m\) in ( \(N, j+K L\) ) On exit, details of the factorization: \(U\) is stored as an upper triangular band \(m\) atrix \(w\) ith \(K L+K U\) superdiagonals in row \(s 1\) to \(K L+K U+1\), and the \(m\) ultipliers used during the factorization are stored in rows \(K L+K U+2\) to \(2 \star K L+K U+1\). See below for further details.

LD A (input)
The leading dim ension of the array A. LDA >= \(2 * K L+K U+1\).
\(\mathbb{P I V O T}\) (output)
The pivot indices that define the perm utation \(m\) atrix \(P\); row i of the \(m\) atrix \(w\) as interchanged w ith row \(\mathbb{P I V O T}\) (i).

B (input/output)
On entry, the \(N-b y-N R H S\) righthand sidem atrix \(B\). On exit, if \(\mathbb{N} F O=0\), the \(N\) by \(-N R H S\) solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i, U(i, i)\) is exactly zero. The factorization has been com pleted, but the factor U is exactly singular, and the solution has notbeen com puted.

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, when \(M=N=6, K L=2, K U=1\) :

On entry: Onexit:
u36
* * + + + + * * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55 u66
a21 a32 a43 a54 a65 * m 21 m 32 m 43 m 54 m 65 * a31 a42 a53 a64 * * m31 m 42 m 53 m 64 * *

A rray elem entsm arked * are notused by the routine; ele\(m\) entsm arked + need notbe seton entry, but are required by the routine to store elem ents of \(U\) because of fill-in resulting from the row interchanges.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgbsvx - use the LU factorization to com pute the solution to a real system of linearequations \(A * X=B, A * * T * X=B\), orA ** \(_{\mathrm{H}}\) * \(\mathrm{X}=\mathrm{B}\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGBSVX (FACT,TRANSA,N,KL,KU,NRHS,A,LDA,AF,}
LDAF,\mathbb{PIVOT,EQUED,R,C,B,LDB,X,LDX,RCOND,FERR,}
BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1FACT,TRANSA,EQUED
INTEGERN,KL,KU,NRHS,LDA,LDAF,LDB,LDX,INFO
INTEGER \mathbb{PIVOT (*),W ORK2 (*)}
REAL RCOND
REALA (LDA,*),AF (LDAF,*),R (*),C (*), B (LDB,*), X (LDX,*),
FERR (*),BERR (*),WORK (*)
SUBROUT\mathbb{NE SGBSVX_64(FACT,TRANSA,N,KL,KU,NRHS,A,LDA,AF,}
LDAF,\mathbb{PIVOT,EQUED,R,C,B,LDB,X,LDX,RCOND,FERR,}
BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1FACT,TRANSA,EQUED
\mathbb{N}TEGER*8N,KL,KU,NRHS,LDA,LDAF,LDB,LDX,INFO
\mathbb{NTEGER*8 P\mathbb{IVOT (*),W ORK2 (*)}}\mathbf{(})
REALRCOND
REAL A (LDA,*),AF (LDAF,*),R(*),C (*), B (LDB,*), X (LDX,*),
FERR (*),BERR (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GBSVX \(\mathbb{E A C T},[T R A N S A], \mathbb{N}], K L, K U, \mathbb{N R H S}], A,[L D A]\), AF, [LDAF], \(\mathbb{P} \mathbb{I V O T}, E Q U E D, R, C, B,[L D B], X,[L D X]\), RCOND,FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::FACT,TRANSA, EQUED
\(\mathbb{N}\) TEGER :: N, KL, KU, NRHS,LDA, LDAF, LDB, LD X , \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T, W O R K 2\)
REAL ::RCOND
REAL,D IM ENSION (:) ::R, C,FERR,BERR,W ORK
REAL,D \(\mathbb{I M} \operatorname{ENSION}(:,:):: A, A F, B, X\)

SU BROUT \(\left.\mathbb{N} E \operatorname{GBSVX\_ 64(FACT,[TRANSA],~} \mathbb{N}\right], K L, K U,[N R H S], A\), \([L D A], A F,[L D A F], \mathbb{P} \mathbb{V} O T, E Q U E D, R, C, B,[L D B], X,[L D X]\), RCOND, FERR, BERR, \(\mathbb{W}\) ORK], [W ORK 2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::FACT,TRANSA, EQUED
\(\mathbb{N}\) TEGER (8) :: N, KL, KU,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T, W\) ORK 2
REAL ::RCOND
REAL,D IM ENSION (:) ::R, C,FERR,BERR,W ORK
REAL,D \(\mathbb{I M} E N S I O N\) (:,:) ::A,AF,B,X

\section*{C INTERFACE}
\#include <sunperfh>
void sgbsvx (char fact, chartransa, intn, intkl, int ku, int nrhs, float *a, int lda, float * af, int ldaf, int *ipivot, charequed, float * \(r\), float * C , float *b, int ldb, float *x, intldx, float*roond, float *ferr, float *berr, int *info);
void sgbsvx_64 (char fact, chartransa, long n, long kl, long
ku, long nıhs, float *a, long lda, float *af, long ldaf, long *ipivot, char equed, float *r, float *C, float*b, long ldb, float *x, long ldx, float *rcond, float *ferr, float *berr, long *info);

\section*{PURPOSE}
sgbsvx uses the LU factorization to com pute the solution to a real system of linear equations \(A * X=B, A * * T * X=B\), or \(A * * H * X=B, w h e r e A\) is aband \(m\) atrix of orderN w ith \(K L\) subdiagonals and KU superdiagonals, and X and B are N -byN R H S m atrioes.

Errorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed by this subroutine:
1. IfFACT \(=\) E', real scaling factors are com puted to equilibrate
the system :
TRANS \(=N^{\prime}: \operatorname{diag}(\mathbb{R}) * A * \operatorname{diag}(C) \quad * \operatorname{inv}(\operatorname{diag}(C)) \star X=\) \(\operatorname{diag}(R) * B\)
\(\operatorname{TRANS}=T::(\operatorname{diag}(\mathbb{R}) \star A * \operatorname{diag}(C)) * * T * \operatorname{inv}(\operatorname{diag}(\mathbb{R})) * X=\) diag (C)*B

TRANS \(=C\) ': \((\operatorname{diag}(\mathbb{R}) * A * \operatorname{diag}(C)) * * H * \operatorname{inv}(\operatorname{diag}(\mathbb{R})) \star X=\) diag (C)*B

W hether or not the system w illbe equilibrated depends on the scaling of the m atrix A , but ifequilibration is used, A is overw ritten by diag \((\mathbb{R}) \star A\) *diag \((C)\) and \(B\) by diag \((R) * B\) (if TRANS = N ) ordiag (C)*B (if TRANS = T'or C).
2. IfFACT \(=N\) 'or \(E\) ', the LU decomposition is used to factor the
\(m\) atrix A (afterequilibration ifFACT = E ) as
\[
A=L \star U
\]
where \(L\) is a productof perm utation and unit low er triangular
m atrioes w ith \(\mathrm{K} L\) subdiagonals, and U is upper triangular w ith
\(\mathrm{K} \mathrm{L}+\mathrm{K} \mathrm{U}\) superdiagonals.
3. If som e \(U(i, i)=0\), so that \(U\) is exactly singular, then the routine
retums w ith \(\mathbb{N F O}=i .0\) therw ise, the factored form of \(A\) is used
to estim ate the condition num ber of the \(m\) atrix A. If the reciprocal of the condition num ber is less than \(m\) achine precision,
\(\mathbb{N}\) FO \(=N+1\) is retumed as a w aming, but the routine stillgoes on
to solve for \(X\) and com pute error bounds as described below.
4. The system ofequations is solved forX using the factored form
of A.
5. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for 五.
6. If equilibration w as used, the \(m\) atrix \(X\) is prem ultiplied by
\(\operatorname{diag}(C)\) (iftRANS = N ) ordiag \((\mathbb{R})\) (ifTRANS = T' or
C) so
that it solves the original system before equilibration.

\section*{ARGUMENTS}

FACT (input)
Specifies w hether or not the factored form of the \(m\) atrix \(A\) is supplied on entry, and ifnot, whether them atrix A should be equilibrated before it is factored. = \(\mathrm{F}^{\prime}:\) On entry, AF and \(\mathbb{P} \mathbb{I V O T}\) contain the factored form of . IfEQUED is not \(N\) ', the \(m\) atrix A has been equilibrated w ith scaling factors given by R and \(\mathrm{C} . \mathrm{A}, \mathrm{AF}\), and \(\mathbb{P}\) IV OT are not m odified. \(=\mathrm{N}\) ': Them atrix A w illbe copied to A F and factored.
\(=\mathrm{E}\) : The matrix A w ill be equilibrated if necessary, then copied to A F and factored.
TRANSA (input)
Specifies the form of the system ofequations. =
\(\mathrm{N}^{\prime}: A * X=\mathrm{B} \quad\) (Notranspose)
\(=T ': A * * T * X=B \quad\) ( ranspose)
\(=C\) ': \(A * * H * X=B \quad\) (Transpose)

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

N (input) The num ber of linearequations, ie., the order of them atrix A. N >=0.

K L (input)
The num ber of subdiagonals \(w\) thin the band of \(A\). \(K L>=0\) 。

KU (input)
The num ber of superdiagonals \(w\) thin the band of \(A\).
\(K U>=0\) 。

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atriges B and X . NRH S \(>=0\).

A (input/output)
O n entry, the \(m\) atrix A in band storage, in row s 1 to \(K L+K U+1\). The \(j\) th colum \(n\) ofA is stored in the \(j\) th colum \(n\) of the array A as follow s: A (KU+1+i\(j, j)=A(i, j)\) form ax \((1, j K U)<=i<=m\) in \((\mathbb{N}, j+k l)\)

IfFACT = F'and EQUED is not \(N\) ', then \(A\) must have been equilibrated by the scaling factors in \(R\) and/orC. A is notm odified if \(\mathrm{FACT}=\mathrm{F}^{\prime}\) or \(\mathrm{N}^{\prime}\),
orifFACT = E'andEQUED = N'on exit.

Onexit, ifEQUED ne. \(N\) ', A is scaled as follow s: \(E Q U E D=R 1: A:=\operatorname{diag}(R) * A\) EQUED = C': A :=A * diag (C)
EQUED \(=B: A=\operatorname{diag}(R) * A * \operatorname{diag}(C)\).

LD A (input)
The leading dim ension of the array A. LDA >= \(K L+K U+1\).

AF (input/output)
If \(F A C T=F\) ', then \(A F\) is an inputargum entand on entry contains details of the LU factorization of the band \(m\) atrix A, as com puted by SGBTRF. U is stored as an upper triangularband \(m\) atrix \(w\) ith \(K L+K U\) superdiagonals in row \(s 1\) to \(K L+K U+1\), and the m ultipliers used during the factorization are stored in row \(s K L+K U+2\) to \(2 * K L+K U+1\). If EQUED ne. \(N\) ', then AF is the factored form of the equilibrated \(m\) atrix A.

IfFACT = N ', then \(A F\) is an output argum ent and on exit retums details of the LU factorization of A.

If \(F A C T=E\) ', then \(A F\) is an output argum ent and on exit retums details of the LU factorization of the equilibrated \(m\) atrix \(A\) (see the description of \(A\) for the form of the equilibrated \(m\) atrix).

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(2 * K L+K U+1\).

PIVOT (input)
IfFACT = \(\mathrm{F}^{\prime}\), then \(\mathbb{P} \mathbb{I V O T}\) is an input argum ent and on entry contains the pivot indioes from the factorization \(\mathrm{A}=\mathrm{L} * \mathrm{U}\) as com puted by SGBTRF; row i of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P}\) IV OT (i).

IfFACT \(=\mathrm{N}\) ', then \(\mathbb{P} \mathbb{I V O T}\) is an output argum ent and on exit contains the pivot indioes from the factorization \(\mathrm{A}=\mathrm{L} * \mathrm{U}\) of the originalm atrix A .

IfFACT = E ', then \(\mathbb{P I V O T}\) is an output argum ent and on exit contains the pivot indioes from the factorization \(A=L * U\) of the equilibrated \(m\) atrix A.

EQUED (input)
Specifies the form of equilibration thatw as done.
= \(\mathrm{N}^{\prime}\) : N o equilibration (alw ays true ifFACT =
N).
\(=R\) ': Row equilibration, ie., A has been
prem ultiplied by diag (R) = C ': C olum n equilibration, ie., A has been postm ultiplied by
diag (C). = B ': B oth row and colum \(n\) equilibra-
tion, ie., A has been replaced by diag \((\mathbb{R})\) * A * diag (C). EQUED is an inputargum entifFACT = F '; otherw ise, it is an output argum ent.
\(R\) (input/output)
The row scale factors forA. IfEQUED = R' or \(B\) ', \(A\) is multiplied on the left by diag \((R)\); if \(E Q U E D=N\) 'or \(C\) ', \(R\) is notaccessed. \(R\) is an input argum ent ifFACT = \(F\) '; otherw ise, \(R\) is an outputargum ent. IfFACT = \(\mathrm{F}^{\prime}\) and EQUED \(=\mathrm{R}\) 'or \(B\) ', each elem entofR m ustbe positive.

C (input/output)
The colum n scale factors for \(A\). IfEQ U ED = C 'or
B', A is multiplied on the rightby diag (C ) ; if EQUED = N 'or R', C is notaccessed. \(C\) is an input argum ent ifFACT=F'; otherw ise, \(C\) is an outputargum ent. IfFACT = F'and EQUED = C'or \(B\) ',each elem entof \(C\) ustbe positive.

B (input/output)
On entry, the righthand side m atrix B. On exit, ifEQUED \(=N^{\prime}\) ', \(B\) is notm odified; ifTRANSA \(=N^{\prime}\) and \(E Q U E D=R^{\prime}\) or \(B^{\prime}, B\) is overw ritten by \(\operatorname{diag}(R) * B\); ifTRANSA = T'or \(C^{\prime}\) and EQUED = \(C^{\prime}\) or \(B\) ', \(B\) is overw ritten by diag ( \(C\) )*B.

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, \mathbb{N})\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=N+1\), the \(\mathrm{N}-\) by -NRH S solution
\(m\) atrix \(X\) to the original system of equations.
\(N\) ote that \(A\) and \(B\) arem odified on exit if EQUED
ne. \(\mathrm{N}^{\prime}\), and the solution to the equilibrated
system is inv (diag (C ) ) \({ }^{2} \mathrm{X}\) if TRANSA \(=\mathrm{N}\) 'andEQUED
\(=C\) 'or \(B^{\prime}\),orinv \((\operatorname{diag}(R)) * X\) ifTRANSA \(=T\) 'or
\(C^{\prime}\) and \(E Q U E D=R\) 'or \(B '\).

LD X (input)
The leading dim ension of the array \(\mathrm{X} . \mathrm{LD} \mathrm{X}\) >= \(\max (1, N)\).

\section*{RCOND (output)}

The estim ate of the reciprocal condition num ber of the \(m\) atrix \(A\) after equilibration (ifdone). If RCOND is less than the \(m\) achine precision (in particular, ifRCOND \(=0\) ), the \(m\) atrix is singular to w orking precision. This condition is indicated by a retum code of \(\mathbb{N} F O>0\).

\section*{FERR (output)}

The estim ated forw ard errorbound for each solution vector \(X(\mathcal{)}\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution conesponding to \(X(\mathcal{O})\), FERR ( \()\) ) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(\mathrm{X}(\mathcal{J})\)-X TRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{)}\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each
solution vector \(X\) ( \()\) ) (i.e., the \(s m\) allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension ( \(3 *\) N ) On exit, W ORK (1) contains the reciprocal pivot grow th factornorm (A)/nom (U). The "m ax absolute element" norm is used. If W ORK (1) ism uch less than 1 , then the stability of the LU factorization of the (equilibrated) m atrix A could be poor. This also \(m\) eans that the solution \(X\), condition estim atorRCOND, and forw ard error bound FERR could be unreliable. If factorization fails w ith \(0<\mathbb{N} F O<=N\), then \(W\) ORK (1) contains the reciprocal pivot grow th factor for the leading INFO colum nsofA.

W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{N N}\) FO = -i, the i-th argum ent had an illegalvahue
\(>0\) : if \(\mathbb{N} F O=i\), and \(i\) is
\(<=\mathrm{N}: \mathrm{U}(i, i)\) is exactly zero. The factorization has been completed, but the factorU is exactly singular, so the solution and error bounds could
not be com puted. R COND \(=0\) is retumed. \(=\mathrm{N}+1: \mathrm{U}\) is nonsingular, butRCOND is less than machine precision, \(m\) eaning that the \(m\) atrix is singular to w orking precision. N evertheless, the solution and error bounds are com puted because there are a num ber of situations w here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgbtf2 - com pute an LU factorization of a real \(m\)-by-n band \(m\) atrix A using partialpivoting \(w\) th row interchanges

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGBTF2M,N,KL,KU,AB,LDAB,\mathbb{P}V,\mathbb{NFO)}}\mathbf{M}\mathrm{ (N,N}
INTEGERM,N,KL,KU,LDAB,INFO
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(})
REALAB (LDAB,*)
SUBROUT\mathbb{NE SGBTF2_64M ,N,KL,KU,AB,LDAB,}\mathbb{P}\mathbb{IV},\mathbb{N}FO)
INTEGER*8M,N,KL,KU,LDAB,INFO
\mathbb{NTEGER*8 \mathbb{PIV (*)}}\mathbf{*}\mathrm{ ( }
REALAB (LDAB,*)
F95 INTERFACE

```

```

\mathbb{NTEGER::M,N,KL,KU,LDAB,}\mathbb{N}FO
INTEGER,D IM ENSION (:) :: \mathbb{PIV}
REAL,DIM ENSION (:,:)::AB

```

```

\mathbb{NTEGER (8)::M,N,KL,KU,LDAB,INFO}
INTEGER (8),D IM ENSION (:) :: \mathbb{PIV}
REAL,D IM ENSION (:,:) ::AB

```
\#include < sunperfh>
void sgbtf2 (intm , intn, int kl, int ku, float *ab, int ldab, int *ịív, int *info);
void sgbtf2_64 (long m , long n, long kl, long ku, float *ab, long ldab, long *ịìiv, long *info);

\section*{PURPOSE}
sgbtf2 com putes an LU factorization of a real \(m\)-by-n band m atrix A using partialpivoting w ith row interchanges.

This is the unblocked version of the algorithm, calling Level2 BLAS.

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
KL (input)
The num ber of subdiagonals w ithin the band of A.
\(\mathrm{KL}>=0\).

KU (input)
The num ber of superdiagonals w ithin the band of A. \(K U>=0\).

AB (input/output)
O n entry, them atrix \(A\) in band storage, in row \(s\) \(K L+1\) to \(2 * K L+K U+1\); row s 1 to \(K L\) of the anay need not.be set. The \(j\) th column ofA is stored in the \(j\) th column of the array AB as follows: \(A B(k l+k u+1+i-j, j)=A(i, j)\) for \(m a x(1, j\) \(\mathrm{ku})<=i<=m\) in \((\mathrm{m}, \mathrm{j}+\mathrm{kl})\)

On exit, details of the factorization: U is stored as an upper triangular band \(m\) atrix \(w\) ith \(K L+K U\) superdiagonals in row s 1 to \(K L+K U+1\), and the \(m u l-\) tipliers used during the factorization are stored in row s \(K L+K U+2\) to \(2 * K L+K U+1\). See below for furtherdetails.

The leading dim ension of the array A B . LD AB >=
\(2 * K L+K U+1\).

IPIV (output)
The pivot indices; for \(1<=i<=m\) in \(M, N\) ), row i of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P} \mathbb{V}\) (i).
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0\) : if \(\mathbb{N N F O}=-\) i, the \(i\)-th argum ent had an illegalvalue \(>0\) : if \(\mathbb{N} F O=+i, U(i, i)\) is exactly zero. The factorization has been com pleted, but the factor \(U\) is exactly singular, and division by zero will occur if it is used to solve a system ofequations.

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, \(w\) hen \(M=N=6, K L=2, K U=1\) :

On entry: On exit:
```

    * * * + + + * * * u14 u25
    u36
* * + + + + * * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55
u66
a21 a32 a43 a54 a65 * m 21 m 32 m 43 m 54 m 65
*
a31 a42 a53 a64 * * m 31 m 42 m 53 m 64 *

```
*

A ray elem entsm arked * are notused by the routine; ele\(m\) entsm arked + need notbe seton entry, but are required by the routine to store elem ents of \(U\), because of fill-in resulting from the row interchanges.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgbtrf-com pute an LU factorization of a real \(m\)-by-n band \(m\) atrix A using partialpivoting \(w\) th row interchanges

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGBTRFM,N,KL,KU,AB,LDAB,\mathbb{PIVOT,INFO)}}\mathbf{M}\mathrm{ (N,N}
INTEGERM,N,KL,KU,LDAB,INFO
\mathbb{NTEGER PPIVOTM IN M N))}
REALAB (LDAB,N)

```

```

INTEGER*8M,N,KL,KU,LDAB,INFO
\mathbb{NTEGER*8 \mathbb{PIVOTM MN MN))}}\mathbf{M}\mathrm{ (N)}
REALAB (LDAB,N)

```

\section*{F95 INTERFACE}
```

SU BROUTINE GBTRF M, $\mathbb{N}], K L, K U, A B,[L D A B], \mathbb{P} \mathbb{I} O T,[\mathbb{N F O}])$
$\mathbb{N} T E G E R:: M, N, K L, K U, L D A B, \mathbb{N} F O$
$\mathbb{N}$ TEGER,D $\mathbb{I M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$
REAL,D IM ENSION (:,:) ::AB
SU BROUTINE GBTRF_64M, N ],KL,KU,AB, [LDAB], $\mathbb{P} \mathbb{I V O T},[\mathbb{N} F O])$
$\mathbb{N}$ TEGER ( 8 ) :: M, N, KL, KU, LDAB, $\mathbb{N}$ FO
$\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T$
REAL,D $\mathbb{I M}$ ENSION (:,:) ::AB

```
void sgbtrf(intm, intn, intkl, int ku, float *ab, int ldab, int *ipivot, int*info);
void sgbtrf_ 64 long m , long n, long kl, long ku, float *ab, long ldab, long *ipivot, long *info);

\section*{PURPOSE}
sgbtrf com putes an LU factorization of a real \(m\)-by-n band m atrix A using partialpivoting w ith row interchanges.

This is the blocked version of the algorithm, calling Level 3 BLAS.

\section*{ARGUMENTS}

M (input) Integer
The num ber of row s of the m atrix A. M \(>=0\).
N (input) Integer
The num berof 00 lum ns of the \(m\) atrix \(A . N>=0\).

K L (input) Integer
The num ber of subdiagonals w ithin the band of A. \(K L>=0\).

KU (input) Integer
The num ber of superdiagonals \(w\) ith in the band of A.
\(K U>=0\).
AB (input/output) Realarray ofdim ension (LD AB,N).
On entry, the \(m\) atrix \(A\) in band storage, in row \(s\) \(K L+1\) to \(2 * K L+K U+1\); row s 1 to \(K L\) of the array need notbe set. The \(j\) th colum n of \(A\) is stored in the \(j\) th column of the array AB as follows: AB \((\mathbb{K} L+K U+1+I J, J)=A(I, N)\) for MAX \((1, J\) \(K U)<=\mathbb{K}=M \mathbb{N}(M, N+K L)\)

O n exit, details of the factorization: U is stored as an upper triangular band \(m\) atrix \(w\) ith \(K L+K U\) superdiagonals in row s 1 to \(K L+K U+1\), and the \(m u l-\) tipliers used during the factorization are stored in row \(S K+K U+2\) to \(2 * K L+K U+1\). See below for furtherdetails.

LD AB (input) Integer
The leading dim ension of the array AB. LDAB >= \(2 * K L+K U+1\).
\(\mathbb{P} \mathbb{I V O T}\) (output) Integerarray ofdim ension \(M \mathbb{I N} M, N\) )
The pivotindioes; for \(1<=I<=M \mathbb{N} M, N\) ), row \(I\) of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P I V O T}\) (I).
\(\mathbb{N}\) FO (output) Integer
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) I, the I-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=+\mathbb{I}, U(I, I)\) is exactly zero. The factorization has been com pleted, but the factorU is exactly singular, and division by zero will occur if it is used to solve a system ofequations.

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, when \(M=N=6, K L=2, K U=1\) :

On entry: On exit:
* * * + + + * * * u14 u25
u36
* * + + + + * * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55
u66
a21 a32 a43 a54 a65 * m 21 m 32 m 43 m 54 m 65
*
a31 a42 a53 a64 * * m31 m 42 m 53 m 64 *
*

A ray elem entsm arked * are notused by the routine; ele\(m\) entsm arked + need notbe seton entry, but are required by the routine to store elem ents of \(U\) because of fill-in resulting from the row interchanges.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

sgbtrs - solve a system of linearequations A * X = B orA '

* X = B w th a generalbandm atrix A using the LU factoriza-
tion com puted by SGBTRF

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGBTRS (TRANSA,N,NSUB,NSUPER,NRHS,A,LDA, \mathbb{PIVOT,B,}}\mathbf{N},\textrm{N},\textrm{N}
LDB,\mathbb{NFO)}
CHARACTER * 1 TRANSA
INTEGER N,NSUB,NSUPER,NRHS,LDA,LDB, INFO
INTEGER \mathbb{PIVOT (*)}
REAL A (LDA,*),B (LDB,*)
SUBROUTINE SGBTRS_64 (IRANSA,N,NSUB,NSUPER,NRHS,A,LDA, \mathbb{PIVOT,}
B,LDB,INFO)
CHARACTER * 1 TRANSA
INTEGER*8N,NSUB,NSUPER,NRHS,LDA,LDB,INFO
NTEEGER*8 \mathbb{PIVOT (*)}
REALA (LDA,*),B (LDB,*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE GBTRS ([TRANSA], $\mathbb{N}], N S U B, N S U P E R, ~ N R H S], A,[L D A]$, $\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}],[\mathbb{N F O}])$
CHARACTER (LEN=1) ::TRANSA
$\mathbb{N}$ TEGER ::N,N SUB,N SUPER,NRHS,LDA,LDB, $\mathbb{N}$ FO
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: $\mathbb{P} \mathbb{V} O T$
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A,B
SU BROUTINE GBTRS_64 ([TRANSA], $\mathbb{N}], N S U B, N S U P E R, ~ \mathbb{N} R H S], A,[L D A]$,

```
```

\mathbb{PNOT,B,[LDB],[\mathbb{NFO])}}\mathbf{|}=()

```

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N}\) TEGER (8) :: N , N SUB , N SUPER, NRHS,LDA, LD B, \(\mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL, D \(\mathbb{M}\) ENSION (:,:) ::A , B

\section*{C INTERFACE}
\#include <sunperfh>
void sgbtrs (chartransa, intn, intnsub, int nsuper, int nrhs, float *a, intlda, int*ipivot, float*b, int ldlo, int *info);
void sgbtrs_64 (chartransa, long n, long nsub, long nsuper, long nrhs, float *a, long lda, long *ípivot, float *b, long ldb, long *info);

\section*{PURPOSE}
sgbtrs solves a system of linear equations
\(A * X=B\) or \(A^{\prime} \star X=B\) w th a general band \(m\) atrix \(A\) using the LU factorization com puted by SG B TRF .

\section*{ARGUMENTS}

TRANSA (input)
Specifies the form of the system ofequations. =
\(\mathrm{N}^{\prime}: A * X=B \quad\) N o transpose)
\(=T\) ': A * \(\mathrm{X}=\mathrm{B} \quad\) (Transpose)
\(=C: A\) * \(X=B\) (C onjugate transpose \(=\) Transpose)

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NSUB (input)
The num ber of subdiagonals w ithin the band of A. N SUB \(>=0\) 。

\section*{N SU PER (input)}

The num ber of superdiagonals \(w\) thin the band of A. N SU PER > \(=0\) 。

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of the m atrix B. NRH S \(>=0\).

A (input) D etails of the LU factorization of the band \(m\) atrix
A , as com puted by SG B TRF. U is stored as an upper triangularband m atrix w ith N SU B + N SU PER superdiagonals in row s 1 to N SU B + N SUPER +1 , and them ultipliers used during the factorization are stored in row s N SU B +N SU PER +2 to \(2 \star\) N SU B + N SU PER +1 .

LD A (input)
The leading dim ension of the anay A. LDA >= \(2 * N\) SU B \(+N\) SU PER +1 .
IPIVOT (input)
The pivotindices; for \(1<=i<=N\), row i of the m atrix \(w\) as interchanged w ith row IPIVOT (i).

B (input/output)
On entry, the right hand sidem atrix B. On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

IN FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgebak - form the rightor lefteigenvectors of a real gen-
eral matrix by backw ard transform ation on the com puted
eigenvectors of the balanced \(m\) atrix outputby SG EBA L

\section*{SYNOPSIS}

```

CHARACTER * 1 JOB,SIDE
\mathbb{NTEGERN,}\mathbb{NO},\mathbb{H}\textrm{I},\textrm{M},LDV,\mathbb{NFO}
REAL SCALE (*),V (LDV,*)

```

```

CHARACTER * 1 JOB,SIDE

```

```

REAL SCALE (*),V (LDV ,*)

```

\section*{F95 INTERFACE}
 [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) :: JOB,SDE
\(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathbb{Z} \mathrm{O}, \mathbb{H} \mathrm{I}, \mathrm{M}, \mathrm{LDV}, \mathbb{N} F O\)
REAL,D \(\mathbb{I}\) ENSION (:) ::SCALE
REAL,D \(\mathbb{I}\) ENSION (: : : : ::V
SU BROUTINE GEBAK_64 (JOB,SDE, \(\mathbb{N}], \mathbb{I} O, \mathbb{H} I, S C A L E, \mathbb{M}], V,[L D V]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) :: JOB,SDE
\(\mathbb{N} T E G E R(8):: N, \mathbb{L O}, \mathbb{H} \mathrm{I}, \mathrm{M}, \mathrm{LDV}, \mathbb{N} F O\)

REAL,D \(\mathbb{M}\) ENSION (:) :: SCALE
REAL,D IM ENSION (:,:) ::V

\section*{C INTERFACE}
\#include <sunperfh>
void sgebak (char jंb, char side, intn, int ilo, int ihi, float *scale, int \(m\), float *V, int ldv, int *info);
void sgebak_64 (char job, charside, long n, long ion, long ihi, float *scale, long m,float*v, long ldv, long *info);

\section*{PURPOSE}
sgebak form s the right or lefteigenvectors of a real general \(m\) atrix by backw ard transform ation on the com puted eigenvectors of the balanced m atrix outputby SG EBA L .

\section*{ARGUMENTS}

\section*{JOB (input)}

Specifies the type of backw ard transform ation
required: \(=\mathrm{N}\) ', do nothing, retum im m ediately;
\(=P\) ', do backw ard transform ation for perm utation only; = S', do backw ard transform ation forscaling only; = B', do backw ard transform ations for both perm utation and scaling. JO B m ustbe the sam e as the argum ent JO B supplied to SG EBA L .

SIDE (input)
\(=\mathrm{R}: \mathrm{V}\) contains righteigenvectors;
\(=\mathbb{L}\) ': V contains lefteigenvectors.

N (input) The num ber of row s of the m atrix \(\mathrm{V} . \mathrm{N}>=0\).

IIO (input)
The integers \(\mathbb{I L O}\) and \(\mathbb{H}\) Ideterm ined by SGEBAL. 1 \(<=\mathbb{H}<=\mathbb{H} I<=N\), if \(N>0\); \(\Pi O=1\) and \(\mathbb{H} I=0\), if \(\mathrm{N}=0\) 。

IH I (input)
See the description for \(\Pi\).

SCALE (input)
D etails of the perm utation and scaling factors, as
retumed by SGEBAL.

M (input) The num ber of colum \(n s\) of the \(m\) atrix \(V . M>=0\).

V (input/output)
O n entry, the \(m\) atrix of right or lefteigenvectors to be transform ed, as retumed by SHSEIN or STREVC. On exit, \(V\) is overw rilten by the transform ed eigenvectors.

LDV (input)
The leading dim ension of the anray V.LDV >= \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-i\), the i-th argum enthad an illegalvalue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgebal-balance a general realm atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGEBAL(JOB,N,A,LDA,\mathbb{LO,}\mathbb{H}I,SCA LE,INFO)}
CHARACTER * 1 JOB

```

```

REALA (LDA,*),SCALE (*)

```

```

CHARACTER * 1 J B
\mathbb{N TEGER*8N,LDA,}\mathbb{NO},\mathbb{H}I,\mathbb{N}FO
REAL A (LDA,*),SCALE (*)
F95 INTERFACE

```

```

    CHARACTER (LEN=1) :: JOB
    ```

```

    REAL,DIM ENSION (:) ::SCALE
    REAL,DIM ENSION (:,:)::A
    ```

```

    CHARACTER (LEN=1) :: ODB
    \mathbb{N TEGER (8) ::N,LDA,}\mathbb{NO},\mathbb{H}I,\mathbb{NNFO}
    REAL,DIM ENSION (:) ::SCALE
    REAL,D IM ENSION (:,:) ::A
    ```

\section*{C INTERFACE}
\#include <sunperfh>
void sgebal(char jjb, intn, float *a, int lda, int *io, int *ihi, float *scale, int *info);
void sgebal 64 (char j̣b, long n, float*a, long lda, long *ilo, long *ihi, float *scale, long *info);

\section*{PURPOSE}
sgebalbalances a general real m atrix A. This involves, first, perm uting A by a sim ilarity transform ation to isolate eigenvalues in the first1 to \(\mathbb{H O}-1\) and last IH I+1 to N ele\(m\) ents on the diagonal; and second, applying a diagonalsim ilarity transform ation to row s and colum ns \(\mathbb{H} O\) to \(\mathbb{H}\) Ito \(m\) ake the row s and colum ns as close in norm aspossible. Both steps are optional.
\(B\) alancing \(m\) ay reduce the 1 -norm of the \(m\) atrix, and im prove the accuracy of the com puted eigenvalues and/or eigenvectors.

\section*{ARGUMENTS}

JOB (input)
Specifies the operations to be perform ed on A:
\(=\mathrm{N}^{\prime}\) : none: simply set \(\Pi \mathrm{O}=1, \mathrm{H} I=\mathrm{N}\), SCALE (I) \(=1.0\) for \(i=1, \ldots, N ;=P ':\) perm ute only;
\(=\mathrm{S}\) ': scale only;
\(=B\) ': both perm ute and scale.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the inputm atrix A. On exit, A is overw rilten by the balanced \(m\) atrix. If \(\mathrm{OB}=\mathrm{N}^{\prime}\), A is not referenced. See FurtherD etails.

LD A (input)
The leading dim ension of the aray A. LD A >= \(\max (1, N)\).

ILO (output)
IHO and \(\mathbb{H}\) I are set to integers such thaton exit
\(A(i, j)=0\) if \(i>j a n d j=1, \ldots, I L O-1\) or \(I=\)
\(\mathbb{H} \mathrm{I}+1, \ldots, \mathrm{~N}\). If \(\mathrm{OB}=\mathrm{N}\) 'or \(\mathrm{S}^{\prime}, \mathbb{I} \mathrm{HO}=1\) and \(\mathbb{H} \mathrm{I}\)
\(=\mathrm{N}\).
IH I (output)
See the description for IIO .
SCALE (output)
D etails of the perm utations and scaling factors applied to \(A\). IfP ( \(j\) ) is the index of the row and colum \(n\) interchanged \(w\) th row and colum \(n\) jand \(D(1)\) is the scaling factorapplied to row and column \(j\) then SCALE \((j)=P(j) \quad\) for \(j=1, \ldots, I L O-1=D(j)\) for \(j=\mathbb{L O}, \ldots, \mathbb{H} I=P(j) \quad\) for \(j=\mathbb{H} I+1, \ldots, N\). The order in which the interchanges are m ade is N to \(\mathbb{H} \mathrm{I}+1\), then 1 to \(\mathbb{L O} \mathrm{O}\).
\(\mathbb{I N} F O\) (output)
= 0: successfulexit.
<0: if \(\mathbb{N N}\) FO = -i, the i-th argum enthad an illegalvahue.

\section*{FURTHER DETAILS}

The perm utations consist of row and column interchanges which put the \(m\) atrix in the form
\[
\left.\begin{array}{c}
(\mathrm{T} 1 \mathrm{X} \\
\mathrm{PAP}
\end{array}\right)=\left(\begin{array}{ll}
0 & \mathrm{~B} \\
\mathrm{P}
\end{array}\right) .
\]
where T1 and T2 are uppertriangularm atrices whose eigenvalues lie along the diagonal. The colum \(n\) indices \(\Pi \mathrm{HO}\) and IH Im ark the starting and ending colum ns of the subm atrix B. Balancing consists of applying a diagonal sim ilarity transform ation inv \((D) * B * D\) to \(m\) ake the 1 -norm \(s\) of each row of \(B\) and its comesponding colum n nearly equal. The outputm atrix is
\(\left(\begin{array}{lll}(11 & X * D & Y\end{array}\right)\)
\(\left(\begin{array}{lll}0 & \operatorname{inv}(D) * B * D & \operatorname{inv}(D) * Z\end{array}\right)\).
\(\left(\begin{array}{lll}0 & 0 & T 2\end{array}\right)\)

Inform ation about the perm utations P and the diagonalm atrix \(D\) is retumed in the vectorSCA LE.

This subroutine is based on the E ISPACK routine BA LANC.
M odified by Tzu-Y iChen, C om puterScience D ívision, U niversity of
C alifomia atB erkeley, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgebrd -reduce a general realM -by-N m atrix A to upper or low erbidiagonal form B by an orthogonal transform ation

\section*{SYNOPSIS}

\(\mathbb{N}\) TEGERM,N,LDA,LW ORK, \(\mathbb{N} F O\)
REALA (LDA , *), D (*), E (*),TAUQ (*), TAUP (*), W ORK (*)
SU BROUTINE SGEBRD_64M,N,A,LDA,D,E,TAUQ,TAUP,W ORK,LW ORK, \(\mathbb{N} F O\) )
\(\mathbb{N}\) TEGER* \(8 \mathrm{M}, \mathrm{N}\), LD A, LW ORK, \(\mathbb{N} F \mathrm{O}\)
REALA (LDA , *), D (*), E (*),TAUQ (*), TAUP (*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GEBRD (M ], \(\mathbb{N}], A,[L D A], D, E, T A U Q, T A U P,[W O R K],[L W ~ O R K]\), [ \(\mathbb{N}\) FO ])
\(\mathbb{N}\) TEGER ::M,N,LDA,LW ORK, \(\mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::D ,E,TAUQ,TAUP,W ORK
REAL,D \(\mathbb{M}\) ENSION (: : : : : A

SU BROUTINE GEBRD_64 (M ], \(\mathbb{N}], A,[L D A], D, E, T A U Q, T A U P,[W O R K]\), [LW ORK], \(\mathbb{N} F O\) ])
\(\mathbb{N}\) TEGER (8) ::M,N,LDA,LW ORK, \(\mathbb{N} F=\)
REAL,D \(\mathbb{M}\) ENSION (:) ::D ,E,TAUQ,TAUP,W ORK
REAL,D \(\mathbb{M}\) ENSION (: : : : : A

\section*{C INTERFACE}
\#include <sunperfh>
void sgebord (intm, intn, float *a, int lda, float *d, float
*e, float *tauq, float *taup, int *info);
void sgebrd_64 (long m, long n, float *a, long lda, float *d, float *e, float *tauq, float *taup, long *info);

\section*{PURPOSE}
sgebrd reduces a general realM -by \(-\mathrm{N} m\) atrix A to upper or low er bidiagonal form \(B\) by an orthogonaltransform ation: \(\mathrm{Q} * * \mathrm{~T} * \mathrm{~A} * \mathrm{P}=\mathrm{B}\).

Ifm \(>=n, B\) is upperbidiagonal; if \(m<n, B\) is low er bidiagonal.

\section*{ARGUMENTS}

M (input) The num ber of row s in the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of colum ns in them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the \(M-b y-N\) generalm atrix to be reduced. On exit, if \(m>=n\), the diagonal and the first superdiagonal are overw rilten w ith the upperbidiagonal m atrix B ; the elem ents below the diagonal, w th the array TAUQ , represent the orthogonal \(m\) atrix \(Q\) as a productofelem entary reflectors, and the elem ents above the first superdiagonal, w ith the amay TAUP, represent the orthogonal \(m\) atrix \(P\) as a productofelem entary reflectors; if \(\mathrm{m}<\mathrm{n}\), the diagonal and the first subdiagonalare overw ritten \(w\) ith the low er bidiagonal \(m\) atrix B; the elem ents below the first subdiagonal, w ith the array TAUQ, represent the orthogonalm atrix \(Q\) as a product ofelem entary reflectors, and the elem ents above the diagonal, with the array TA U \(P\), represent the orthogonalm atrix \(P\) as a product of elem entary reflectors. See FurtherD etails.

LD A (input)
The leading dim ension of the array A. LDA >= \(m a x(1, M)\).

D (output)
The diagonalelem ents of the bidiagonalm atrix B:
\(D(i)=A(i, i)\).

E (output)
The off-diagonalelem ents of the bidiagonalm atrix
\(B:\) ifm \(>=n, E(i)=A(i, i+1)\) for \(i=1,2, \ldots, n-\)
\(1 ;\) ifm \(<n, E(i)=A(i+1, i)\) for \(i=1,2, \ldots m-1\).

TAUQ (output)
The scalar factors of the elem entary reflectors which represent the orthogonal matrix Q. See FurtherD etails.
TAUP (output)
The scalar factors of the elem entary reflectors which represent the orthogonal \(m\) atrix P. See FurtherD etails.

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The length of the aray \(W\) ORK. LW ORK \(>=\) \(\max (1, M, N)\). For optim um perform ance LW ORK >= \((M+N) \star N B, w h e r e N B\) is the optim alblocksize.

If LW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK anay, retums this value as the first entry of the W ORK array, and no emorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an illegalvalue.

\section*{FURTHER DETAILS}

Them atrices \(Q\) and \(P\) are represented as products of elem entary reflectors:

Ifm \(>=n\),
\[
Q=H(1) H(2) \ldots H(n) \text { and } P=G(1) G(2) \ldots G(n-1)
\]

Each H (i) and G (i) has the form:
\(H(i)=I-\operatorname{tanq} * v^{*} v^{\prime}\) and \(G(i)=I-\operatorname{taup} * u^{*} u^{\prime}\)
w here tauq and taup are real scalars, and \(v\) and \(u\) are real vectors; \(\mathrm{v}(1: i-1)=0, \mathrm{v}(\mathrm{i})=1\), and \(\mathrm{v}(\mathrm{i}+1 \mathrm{~m})\) is stored on exitin \(A(i+1 \mathrm{~m}, \mathrm{i}) ; \mathrm{u}(1: i)=0, u(i+1)=1\), and \(u(i+2 \mathrm{~m})\) is stored on exit in A ( \(\mathbf{i}, \mathrm{i}+2 \mathrm{n}\) ); tauq is stored in TA U Q (i) and taup in TA UP (i).

Ifm \(<\mathrm{n}\),
\(Q=H(1) H(2) \ldots H(m-1)\) and \(P=G(1) G(2) \ldots G(m)\)

Each H (i) and G (i) has the form :
\(H(i)=I-\operatorname{tanq} * v^{*} v^{\prime}\) and \(G(i)=I-\operatorname{tanp} * u^{*} u^{\prime}\)
\(w\) here tauq and taup are realscalars, and \(v\) and \(u\) are real vectors; \(\mathrm{v}(1: i)=0, \mathrm{v}(\mathrm{i}+1)=1\), and \(\mathrm{v}(\mathrm{i}+2 \mathrm{~m})\) is stored on exitin \(A(i+2 m, i) ; u(1: i-1)=0, u(i)=1\), and \(u(i+1 m)\) is stored on exitin A ( \(\mathbf{i}, \mathrm{i}+1 \mathrm{n}\) ); tauq is stored in TA U Q (i) and taup in TA UP (i).

The contents ofA on exitare illustrated by the follow ing exam ples:
```

m = 6 and n=5 (m>n): m=5 and n=6 (m<n):
(d e ul ul u1) (d u1 u1 u1 u1
u1 )
( v1 d e u2 u2 ) ( e d u2 u2 u2
u2 )
( v1 v2 d e u3 ) ( v1 e d u3 u3
u3 )
( v1 v2 v3 d e ) ( v1 v2 e d u4
u4 )
( v1 v2 v3 v4 d ) ( v1 v2 v3 e d
u5 )
( v1 v2 v3 v4 v5 )

```
w here d and e denote diagonal and off-diagonal elem ents of \(B\), videnotes an elem ent of the vectordefining \(H\) (i), and ui an elem ent of the vectordefining G (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgecon - estim ate the reciprocal of the condition num ber of a general real \(m\) atrix \(A\), in either the 1 -nom or the infinity-norm, using the LU factorization com puted by SGETRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NESGECON NORM,N,A,LDA,ANORM,RCOND,WORK,W ORK2,\mathbb{NFO)}}\mathbf{N}\mathrm{ (N,A}
CHARACTER * 1NORM
NNTEGER N,LDA,}\mathbb{N}F
INTEGER W ORK2 (*)
REALANORM,RCOND
REAL A (LDA,*),W ORK (*)
SU BROUT\mathbb{NE SGECON_64 NORM,N,A,LDA,ANORM ,RCOND,W ORK,W ORK2,}
\mathbb{NFO)}
CHARACTER * 1 NORM
\mathbb{NTEGER*8N,LDA, INFO}
INTEGER*8W ORK2 (*)
REAL ANORM,RCOND
REALA (LDA,*),W ORK (*)

```
F95 INTERFACE
    SU BROUTINE GECON \(\mathbb{N} O R M, \mathbb{N}], A,[L D A], A N O R M, R C O N D,[W O R K],[W O R K 2]\),
        [ \(\mathbb{N} F O\) ])
    CHARACTER (LEN=1) ::NORM
    \(\mathbb{N} T E G E R:: N, L D A, \mathbb{N} F O\)
    \(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) ::W ORK2
    REAL ::ANORM,RCOND
    REAL,D IM ENSION (:) ::W ORK

SU BROUTINE GECON_64 NORM, \(\mathbb{N}], A,[L D A], A N O R M, R C O N D,[W O R K],[W O R K 2]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::NORM
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{LD} A, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) ::W ORK2
REAL ::ANORM,RCOND
REAL,D IM ENSION (:) ::W ORK
REAL,D \(\mathbb{M}\) ENSION (: : : : : A

\section*{C INTERFACE}
\#include < sunperfh>
void sgecon (charnorm, int n, float *a, int lda, float anorm, float*rcond, int*info);
void sgecon_64 (charnorm , long n, float *a, long lda, float anorm, float *rcond, long *info);

\section*{PURPOSE}
sgecon estim ates the reciprocal of the condition num berof a general real \(m\) atrix \(A\), in either the 1 -nom or the infinity-norm, using the LU factorization com puted by SGETRF.

A \(n\) estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as
\(\operatorname{RCOND}=1 /(\operatorname{norm}(A) *\) norm (inv (A))).

\section*{ARGUMENTS}

NORM (input)
Specifies w hether the 1-nom condition num ber or the infinity-norm condition num ber is required:
= 1 'or \(0^{\prime}\) : 1 -nom ;
= I': Infinity-norm .

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
A (input) The factors \(L\) and \(U\) from the factorization \(A=\) \(P * L * U\) as com puted by SGETRF.

LD A (input)
The leading dim ension of the amay A. LDA >= \(\max (1, \mathbb{N})\).

\section*{ANORM (input)}

IfNORM = 1 'or 0 ', the 1 -nom of the original
\(m\) atrix \(A\). IfNORM = \(I\) ', the infinity-norm of the originalm atrix A.

RCOND (output)
The reciprocal of the condition num ber of the
\(m\) atrix \(A\), computed as RCOND \(=1 /(\) norm (A) * norm (inv (A))).

W ORK (w orkspace)
dim ension ( \(4 * N\) )

W ORK 2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgeequ - com pute row and colum n scalings intended to equilibrate an \(M\)-by \(-\mathrm{N} m\) atrix \(A\) and reduce its condition num ber

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGEEQU M,N,A,LDA,R,C,ROW CN,COLCN,AMAX,}
\mathbb{NFO)}
\mathbb{NTEGERM,N,LDA,}\mathbb{NFO}
REAL ROW CN,COLCN,AMAX
REAL A (LDA,*),R (*),C (*)
SU BROUTINE SGEEQU_64 M,N,A,LDA,R,C,ROW CN,COLCN,AMAX,
INFO )
\mathbb{NTEGER*8M,N,LDA,INFO}
REAL ROW CN,COLCN,AMAX
REALA (LDA,*),R (*),C (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GEEQU (M ], \(\mathbb{N}\) ],A, [LDA],R,C,ROW CN,COLCN, AMAX, \([\mathbb{N F O}])\)
\(\mathbb{N}\) TEGER ::M,N,LDA, \(\mathbb{N} F O\)
REAL ::ROW CN, COLCN,AMAX
REAL,D IM ENSION (:) ::R,C
REAL,D \(\mathbb{I}\) ENSION (:,:) ::A
SU BROUTINE GEEQU_64 (M ], \(\mathbb{N}], A,[L D A], R, C, R O W C N, C O L C N\), AMAX, [ \(\mathbb{N F O}\) ])
\(\mathbb{N} T E G E R(8):: M, N, L D A, \mathbb{N} F O\)

REAL ::ROW CN, COLCN, AMAX
REAL,D \(\mathbb{I M}\) ENSION (:) ::R,C
REAL,D \(\mathbb{I}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void sgeequ (intm , intn, float *a, int lda, float * \(r\), float
*c, float *row cn, float *colen, float *am ax, int *info);
void sgeequ_64 (long m, long n, float*a, long lda, float * \(r_{\text {, }}\) float * c, float *row cn, float *colen, float *am ax, long *info);

\section*{PURPOSE}
sgeequ com putes row and colum \(n\) scalings intended to equilibrate an M -by-N m atrix A and reduce its condition num ber. R retums the row scale factors and \(C\) the column scale factors, chosen to try to \(m\) ake the largestelem ent in each row and column of the \(m\) atrix \(B \quad w\) ith elements \(B(i, j)=R(i) \star A(i, j) \star C(i)\) have absolute value 1.

R (i) and C (i) are restricted to be betw een SM LN UM = sm allest safe num ber and B IG N UM = largest,safe num ber. U se of these scaling factors is notguaranteed to reduce the condition num berofA butw orks w ellin practice.

\section*{ARGUMENTS}

M (input) The num ber of row s of the matrix \(A . M>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input) The \(M\) boy -N m atrix w hose equilibration factors are to be com puted.

LD A (input)
The leading dim ension of the amay A. LDA >= \(\max (1, M)\).

R (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O>M, R\) contains the row scale
factors forA.

C (output)

If \(\mathbb{N} F O=0, C\) contains the colum \(n\) scale factors forA.

ROW CN (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O>M, R O W C N\) contains the ratio of the sm allest \(R\) (i) to the largest \(R\) (i). If ROW CN >= 0.1 and AM AX is nether too large nortoo sm all, it is notw orth scaling by R .

COLCN (output)
If \(\mathbb{N} F O=0, C O L C N\) contains the ratio of the sm allest C (i) to the largestC (i). IfC O LCN >=0.1, it is notw orth scaling by \(C\).
AMAX (output)
A bsolute value of largestm atrix elem ent. IfA M A X
is very close to overflow orvery close to underflow , the m atrix should be scaled.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvahue
\(>0\) : if \(\mathbb{N F O}=i\), and \(i\) is
\(<=\mathrm{M}\) : the i-th row ofA is exactly zero
> M : the ( \((-\mathrm{M})\) )-th collum n ofA is exactly zero

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgees - com pute foran \(N\) boy -N real nonsym \(m\) etric \(m\) atrix \(A\), the eigenvalues, the realSchur form \(T\), and, optionally, the m atrix of Schurvectors Z

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SGEES (JOBZ,SORTEV,SELECT,N,A,LDA,NOUT,W R,W I, Z,}
LD Z,W ORK,LDW ORK,W ORK 3,\mathbb{NFO)}
CHARACTER * 1 JOBZ,SORTEV
\mathbb{NTEGERN,LDA,NOUT,LDZ,LDW ORK,INFO}
LOGICAL SELECT
LOG ICALW ORK 3 (*)
REALA (LDA,*),W R (*),W I(*),Z (LD Z,*),W ORK (*)
SUBROUTINE SGEES_64(JOBZ,SORTEV,SELECT,N,A,LDA ,NOUT,W R,W I, Z,
LD Z,W ORK,LDW ORK,W ORK 3,\mathbb{NFO)}
CHARACTER * 1 JOBZ,SORTEV
\mathbb{NTEGER*8N,LDA,NOUT,LD Z,LDW ORK,INFO}
LOG ICAL*8 SELECT
LOG ICAL*8W ORK 3 (*)
REALA (LDA,*),W R (*),W I(*),Z (LD Z ,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N E}\) GEES (JOBZ, SORTEV ,SELECT, \(\mathbb{N}], A,[L D A], N O U T, W R, W I, Z\), [LD Z ], [W ORK ], [LDW ORK ], [W ORK 3], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) :: JOBZ,SORTEV
\(\mathbb{N}\) TEGER ::N,LDA,NOUT,LDZ,LDW ORK, \(\mathbb{N} F O\)
LOG ICAL :: SELECT
LOG ICAL,D IM ENSION (:) ::W ORK 3

REAL,D \(\mathbb{M}\) ENSION (:) ::WR,W I,W ORK
REAL, D \(\mathbb{M}\) ENSION (: : : : : : A , Z

SU BROUTINE GEES_64 (OBB , SORTEV, SELECT, \(\mathbb{N}\) ], A, [LDA ],NOUT,W R,W I, Z, [LDZ], [W ORK], [LDW ORK ], [W ORK 3], [ \(\mathbb{N} F O]\) )

CHARACTER ( \(L E N=1\) ) : : JOBZ, SORTEV
\(\mathbb{N}\) TEGER (8) :: N , LDA , NOUT, LD Z , LDW ORK , \(\mathbb{N} F O\)
LOGICAL (8) :: SELECT
LOGICAL (8), D \(\mathbb{M}\) ENSION (:) ::WORK 3
REAL,D \(\mathbb{I M} E N S I O N(:):: W R, W I, W O R K\)
REAL,D IM ENSION (:,:) ::A , Z

\section*{C INTERFACE}
\#include <sunperfh>
void sgees (char jobz, char sortev, int (*select) (float,float), int n, float *a, int lda, int *nout, float *W r, float *W i, float *z, int ldz, int*info);
void sgees_64 (char jobz, char sortev, long (*select) (float,float), long n, float *a, long lda, long *nout, float*W r, float *w i, float *z, long ldz, long *info);

\section*{PURPOSE}
sgees com putes for an \(N\) boy- N realnonsym \(m\) etric \(m\) atrix \(A\), the eigenvalues, the real Schur form \(T\), and, optionally, the \(m\) atrix of Schurvectors \(Z\). This gives the Schur factorization \(A=Z * T *(Z * * T)\).

Optionally, italso orders the eigenvalues on the diagonal of the realSchur form so that selected eigenvalues are at the top left. The leading colum ns of \(Z\) then form an orthonorm albasis for the invariant subspace corresponding to the selected eigenvalues.

A matrix is in real Schur form if it is upper quasitriangularw th 1 -by-1 and 2 -by-2 blocks. 2 -by-2 blocks w ill be standardized in the form
[ a b ]
[ \(\left.\begin{array}{ll}\mathrm{c} & \mathrm{a}\end{array}\right]\)
where \(\mathrm{b}^{*} \mathrm{c}<0\). The eigenvalues of such a block are a +squt(bc).

\section*{ARGUMENTS}

JO BZ (input)
\(=\mathrm{N}\) ': Schurvectors are notcom puted;
\(=\mathrm{V}\) : Schurvectors are com puted.

SORTEV (input)
Specifies w hether or not to order the eigenvalues
on the diagonalof the Schur form . = N ': Eigenvalues are notordered;
\(=S\) ': Eigenvalues are ordered (see SELECT).

SELECT (input)
SELECT m ustbe declared EXTERNAL in the calling subroutine. If SORTEV \(=S^{\prime}\), SELECT is used to selecteigenvalues to sort to the top leftof the Schur form. If SORTEV = \(N^{\prime}\), SELECT is not referenced. A n eigenvalue W R ( 1\()+\operatorname{sqnt}(-1) * W I(1)\) is selected ifSELECT (W R ( 7 ) , \(\mathrm{N} I(\mathcal{j})\) ) is tue; ie., if either one of a com plex conjugate pair of eigenvalues is selected, then both com plex eigenvalues are selected. N ote that a selected com plex eigenvalue m ay no longer satisfy SELECT \((\mathbb{W} R(\mathcal{J}) \mathbb{W} I(\mathcal{J})=\) .TRUE.afterordering, since ordering may change the value of com plex eigenvalues (especially if the eigenvalue is ill-conditioned); in this case \(\mathbb{N}\) FO is set to \(N+2\) (see \(\mathbb{N}\) FO below).

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).

A (input/output)
On entry, the \(N\) boy -N m atrix A. On exit, A has been overw ritten by its realSchur form \(T\).

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

NOUT (output)
If SORTEV = N',NOUT = 0. IfSORTEV = S', NOUT
= num ber of eigenvalues (aftersorting) forw hich
SELECT is true. (C om plex conjugate pairs forw hich
SELECT is true foreithereigenvalue countas 2.)

\section*{W R (output)}

W R and W I contain the real and im aginary parts, respectively, of the com puted eigenvahues in the sam e order that they appear on the diagonal of the output Schur form T. C om plex conjugate pairs of eigenvalueswill appear consecutively w ith the
eigenvalue having the positive in aginary part first.

W I (output)
See the description forW \(R\).
Z (output)
If \(\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{Z}\) contains the orthogonalm atrix Z of Schurvectors. If \(J 0 \mathrm{BZ}=\mathrm{N}^{\prime}, \mathrm{Z}\) is notreferenced.
LD Z (input)
The leading dim ension of the array Z . LD \(\mathrm{Z}>=1\); if \(\mathrm{JO} \mathrm{BZ}=\mathrm{V}\) ', LD \(\mathrm{Z}>=\mathrm{N}\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) contains the optim al LDW ORK.

LDW ORK (input)
The dimension of the amay \(W\) ORK. LDW ORK >= \(m a x(1,3 \star N)\). For good perform ance, LDW ORK must generally be larger.

If LD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 3 (w orkspace)
dim ension ( N\() \mathrm{N}\) ot referenced if \(S O\) RTEV \(=\mathrm{N}^{\prime}\).
\(\mathbb{N F O}\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue.
\(>0:\) if \(\mathbb{N F O}=i\), and \(i\) is
\(<=N\) : the Q R algorithm failed to com pute all the eigenvalues; elem ents \(1: \mathbb{H}-1\) and i+ \(1 \mathbb{N}\) ofW R and W I contain those eigenvalues which have converged; if \(J O B Z=V^{\prime}, Z\) contains the \(m\) atrix which reduces \(A\) to its partially converged Schur form . \(=\mathrm{N}+1\) : the eigenvalues could not be reordered because som e eigenvalues w ere too close to separate (the problem is very ill-conditioned); \(=\mathrm{N}+2\) : after reordering, roundoff changed values of som e com plex eigenvalues so that leading eigenvalues in the Schur form no longersatisfy SELECT=TRUE. This could also be caused by underflow due to scaling.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

sgeesx - com pute foran N -by-N realnonsymm etric m atrix A,
the eigenvalues, the realSchur form T, and, optionally, the
m atrix of Schurvectors Z

```

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SGEESX (JOBZ,SORTEV,SELECT,SENSE,N,A,LDA,NOUT,W R,}
W I, Z,LD Z,SRCONE,RCONV,W ORK,LDW ORK,IN ORK 2,LDW RK 2,BW ORK 3,
\mathbb{NFO)}

```
CHARACTER * 1 JOBZ,SORTEV,SEN SE
\(\mathbb{N}\) TEGERN,LDA,NOUT,LDZ,LDW ORK,LDW RK2, \(\mathbb{N} F O\)
\(\mathbb{I N T E G E R} \mathbb{I N}\) ORK2(*)
LOG ICAL SELECT
LOG ICALBW ORK 3 (*)
REAL SRCONE,RCONV
REALA (LDA , *), W R (*), W I (*), Z (LD Z , \(\left.{ }^{\star}\right), \mathrm{W} O R K\left({ }^{*}\right)\)
SU BROUTINE SGEESX_64 (JOBZ, SORTEV, SELECT, SENSE,N,A,LDA,NOUT,
    W R,W I, Z,LD Z, SRCONE,RCONV,W ORK,LDW ORK, IV ORK2,LDW RK2,
    BW ORK \(3, \mathbb{N} F O\) )
CHARACTER * 1 JOBZ, SORTEV, SEN SE
\(\mathbb{N}\) TEGER*8N,LDA,NOUT,LD Z,LDW ORK,LDW RK 2, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R * 8 \mathbb{I N}\) ORK 2 (*)
LOG ICAL*8 SELECT
LOG ICAL*8BW ORK 3 (*)
REAL SRCONE,RCONV
REALA (LDA, \(\left.{ }^{\star}\right), \mathrm{W} R\left({ }^{\star}\right), \mathrm{W} I(*), \mathrm{Z}(\mathrm{LD} Z, \star), \mathrm{W} O R K\left({ }^{*}\right)\)

\section*{F95 INTERFACE}

SU BROUTINE GEESX (JOBZ, SORTEV, SELECT, SENSE, \(\mathbb{N}\) ],A, [LDA ],NOUT,

W R,W I, Z, [LDZ], SRCONE,RCONV, [W ORK], [LDW ORK], [IW ORK 2], [LDW RK2], [BW ORK 3], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) :: JOBZ, SORTEV , SEN SE
\(\mathbb{N}\) TEGER : : N, LDA, NOUT, LD Z, LDW ORK , LDW RK2, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{I W}\) ORK2
LOGICAL :: SELECT
LOGICAL,D \(\mathbb{M}\) ENSION (:) ::BW ORK 3
REAL :: SRCONE,RCONV
REAL,D \(\mathbb{I M} E N S I O N(:):: W R, W I, W O R K\)
REAL,D \(\mathbb{M}\) ENSION (: : : : : A , Z
```

SU BROUT INE GEESX_64 (OBZ, SORTEV, SELECT, SENSE, $\mathbb{N}$ ], A, [LDA ],NOUT,
W R, W I, Z, [LDZ], SRCONE,RCONV, [W ORK], [LDW ORK], [ $\mathbb{W}$ ORK2],
[LDW RK 2], [BW ORK 3], [ $\mathbb{N F O}$ ])

```

CHARACTER (LEN=1) :: JOBZ, SORTEV, SEN SE
\(\mathbb{N} T E G E R(8):: N\), LDA, NOUT,LDZ,LDW ORK, LDW RK2, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{I N} O R K 2\)
LOGICAL (8) :: SELECT
LOGICAL (8), D \(\mathbb{I M} E N S I O N(:):: B W O R K 3\)
REAL :: SRCONE,RCONV
REAL,D \(\mathbb{I M} E N S I O N\) (:) ::W R,W I,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A, Z

\section*{C INTERFACE}
\#include <sunperfh>
void sgeesx (char j̄bz, char sortev, int(*select) (float,float), char sense, int \(n\), float*a, int lda, int *nout, float *w r, float \({ }^{*}{ }_{W}\) i, float * \(z\), int ldz, float *srcone, float *rconv, int*info);
void sgeesx_64 (char jंbz, char sortev, long (*select) (float,float), char sense, long n, float *a, long lda, long *nout, float *w r, float *W i, float *z, long ldz, float*srcone, float *rconv, long *info);

\section*{PURPOSE}
sgeesx com putes for an \(N\) boy \(-N\) real nonsym m etric \(m\) atrix \(A\), the eigenvahues, the realSchur form T, and, optionally, the \(m\) atrix of Schurvectors \(Z\). This gives the Schur factorization \(A=Z * T *(Z * * T)\).

O ptionally, italso orders the eigenvalues on the diagonal of the realSchur form so thatselected eigenvalues are at
the top left; com putes a reciprocalcondition num ber for the average of the selected eigenvalues (RCONDE); and com putes a reciprocal condition num ber for the right invariant subspace comesponding to the selected eigenvalues (RCONDV). The leading colum ns of \(Z\) form an orthonorm al basis for this invariant subspace.

For further explanation of the reciprocal condition num bers RCONDE and RCONDV, see Section 4.10 of the LAPACK U sers' G uide (w here these quantities are called s and sep respectively).

A realm atrix is in realSchur form if it is upper quasitriangularw ith 1-by-1 and 2 -by- 2 blocks. 2 -by- 2 blocksw ill be standardized in the form
[ a b ]
[ \(\mathrm{c} a \mathrm{a}\) ]
\(w\) here \(b^{*} c<0\). The eigenvalues of such a block are a +sqrt(bc).

\section*{ARGUMENTS}

JOBZ (input)
= N ':Schurvectors are not com puted;
= V ':Schurvectors are com puted.

SORTEV (input)
Specifies w hether ornot to order the eigenvalues on the diagonal of the Schur form . = N ': Eigenvalues are not ordered;
= S':E igenvahues are ordered (see SELECT) .
SELECT (input)
SELECT mustbe declaredEXTERNAL in the calling subroutine. If SORTEV = S', SELECT is used to selecteigenvalues to sort to the top left of the Schurform. If SORTEV = N', SELECT is notreferenced. An eigenvalue \(W\) R \((j)+\operatorname{sqnt}(-1) * W I(j)\) is selected if SELECT \((\mathbb{N} R(\mathcal{J}), N I(\mathcal{J})\) is true; ie., if either one of a com plex conjugate pair of eigenvalues is selected, then both are. N ote that a selected com plex eigenvalue \(m\) ay no longer satisfy \(\operatorname{SELECT} \mathbb{N} R(\mathcal{J}, \mathbb{N} I(\mathcal{j})=\) TRUE.after ordering, since ordering \(m\) ay change the value of com plex eigenvalues (especially if the eigenvalue is illconditioned); in this case \(\mathbb{I N F O} \mathrm{m}\) ay be set to \(\mathrm{N}+3\) (see \(\mathbb{I N} F O\) below ).

D eterm ines which reciprocal condition num bers are com puted. = N ': N one are com puted;
\(=\mathrm{E}\) ': C om puted for average of selected eigenvalues only;
= V ': C om puted for selected right invariant subspace only;
\(=\mathrm{B}\) ': C om puted forboth. If SEN \(S E=\mathrm{E}^{\prime}, \mathrm{V}^{\prime}\) or B',SORTEV mustequal \(S^{\prime}\).

N (input) The order of the m atrix A \(. \mathrm{N}>=0\).

A (input/output)
On entry, the \(N\) boy \(-N m\) atrix A. On exit, A is overw ritten by its realSchur form T.
LDA (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

NOUT (output)
If SORTEV \(=\mathrm{N}\) ', NOUT \(=0\). If SORTEV \(=\mathrm{S}\) ', NOUT
= num ber of eigenvalues (aftersorting) forw hich
SELECT is true. (C om plex conjugate pairs forw hich
SELECT is true fore thereigenvalue countas 2 .)
W R (output)
W R and W Icontain the real and im aginary parts, respectively, of the com puted eigenvalues, in the sam e order that they appearon the diagonal of the output Schur form T. C om plex conjugate pairs of eigenvalues appear consecutively \(w\) ith the eigenvalue having the positive im aginary part first.

W I (output)
See the description forW R.
Z (output)
If \(\mathrm{OBBZ}=\mathrm{V}\) ', Z contains the orthogonalm atrix Z of Schurvectors. If JO BZ \(=N^{\prime}\), Z is notreferenced.

LD \(Z\) (input)
The leading dim ension of the array \(Z\). LD \(Z>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD \(\mathrm{Z}>=\mathrm{N}\).

SRCONE (output)
If SENSE = E' or B', SRCONE contains the reciprocal condition num ber for the average of the selected eigenvalues. N ot referenced if SEN SE = N 'or V '.

RCONV (output)
If SEN SE = V 'or B ', RCONV contains the reciprocal condition num ber for the selected right invariant subspace. N ot referenced if \(\operatorname{SEN} S E=\mathrm{N}^{\prime}\) or E'.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the array W ORK. LDW ORK >= \(\max (1,3 \star \mathrm{~N})\). A lso, if SENSE = E'or V 'or B',
LDW ORK >=N+2*NOUT* \((\mathbb{N}-N O U T)\), where NOUT is the
num ber of selected eigenvalues com puted by this routine. N ote that \(\left.\mathrm{N}+2{ }^{*} \mathrm{NOUT} \mathrm{N}^{*} \mathrm{~N}+\mathrm{NOUT}\right)<=\mathrm{N}+\mathrm{N} * \mathrm{~N} / 2\).
For good perform ance, LDW ORK mustgenerally be larger.

IW ORK 2 (w orkspace/output)
N ot referenced if SEN \(S E=\mathrm{N}\) 'or \(\mathrm{E}^{\prime}\). . On exit, if \(\mathbb{N} F O=0, \mathbb{I}\) ORK 2 (1) retums the optim alld W RK 2.

LD W RK 2 (input)
The dim ension of the array \(\mathbb{I W}\) ORK 2. LDW RK 2 >= 1;
if SENSE \(=V\) 'or \(B^{\prime}\),LDW RK2 \(>=\) NOUT* \((N-N O U T)\).

BW ORK 3 (w orkspace)
dim ension ( \(\mathbb{N}\) ) N ot referenced if \(S O R T E V=\mathrm{N}^{\prime}\).
\(\mathbb{I N F O}\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum ent had an illegalvalue.
\(>0:\) if \(\mathbb{I N F O}=i\), and \(i\) is
\(<=N\) : the \(Q R\) algorithm failed to com pute all the eigenvalues; elem ents \(1: \mathbb{I} \mathrm{O}-1\) and i+ 1 N ofW R and W I contain those eigenvalues which have converged; if \(\mathrm{JOBZ}=\mathrm{V}\) ', Z contains the transform ation w hich reduces A to its partially converged Schur form. \(=\mathrm{N}+1\) : the eigenvalues could not be reordered because some eigenvalues were too close to separate (the problem is very ill-conditioned); = \(\mathrm{N}+2\) : after reordering, roundoff changed values of som e com plex eigenvalues so that leading eigenvalues in the Schur form no longer satisfy SELEC T= TRU E. This could also be caused by underflow due to scaling.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgeev -com pute foran \(N\)-by -N real nonsym \(m\) etric \(m\) atrix \(A\), the eigenvalues and, optionally, the left and/or right eigenvectors

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGEEV (OOBVL,JOBVR,N,A,LDA,W R,W I,VL,LDVL,VR,LDVR,} W ORK,LDW ORK, $\mathbb{N} F O$ )

```

CHARACTER * 1 JobvL, JOBVR
\(\mathbb{N}\) TEGER N,LDA,LDVL,LDVR,LDW ORK, \(\mathbb{N} F O\)
REALA (LDA, \(\left.{ }^{\star}\right), \mathrm{W} R\left({ }^{\star}\right), \mathrm{W} I(*), \mathrm{VL}\left(\mathrm{LDVL},{ }^{*}\right), \mathrm{VR}(\mathbb{L} D V R, \star), \mathrm{W} O R K(*)\)

SU BROUTINE SGEEV_64 (JO BVL, JOBVR,N,A,LDA,W R,W I,VL,LDVL,VR, LDVR,W ORK,LDW ORK, \(\mathbb{N} F O\) )

CHARACTER * 1 JobvL, JobvR
\(\mathbb{N} T E G E R * 8 N, L D A, L D V L, L D V R, L D W O R K, \mathbb{N} F O\)


\section*{F95 INTERFACE}

SU BROUTINE GEEV (JOBVL, JOBVR, \(\mathbb{N}], A,[L D A], W R, W I, V L,[L D V L], V R\), [LDVR], [W ORK], [LDW ORK], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1) :: JOBVL, JOBVR
\(\mathbb{N} T E G E R:: N, L D A, L D V L, L D V R, L D W O R K, \mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::W R,W I,W ORK
REAL,D IM ENSION (:,:) ::A,VL,VR

SU BROUTINE GEEV_64 (JOBVL, JOBVR, \(\mathbb{N}], A,[L D A], W R, W I, V L,[L D V L]\), VR, [LDVR], [W ORK], [LDW ORK], [NFO])

CHARACTER (LEN=1) :: JOBVL, JOBVR
\(\mathbb{N}\) TEGER (8) ::N,LDA,LDVL,LDVR,LDW ORK, \(\mathbb{N}\) FO
REAL,D \(\mathbb{I M} E N S I O N(:):: W R, W I, W\) ORK
REAL,D IM ENSION (:,:) ::A,VL,VR

\section*{C INTERFACE}
\#include <sunperfh>
void sgeev (char jobvl, char jobvr, intn, float *a, int lda,
float *w r, float *w i, float*vl, int ldvl, float
*vr, int ldvr, int*info);
void sgeev_64 (char jobvl, char jंbver, long n, float *a, long
lda, float *W \(r\), float * w i, float * v , long ldvl, float *vr, long ldvr, long *info);

\section*{PURPOSE}
sgeev com putes for an \(N\)-by -N realnonsym \(m\) etric \(m\) atrix \(A\), the eigenvalues and, optionally, the left and/or righteigenvectors.

The righteigenvectorv \((\mathcal{I})\) of A satisfies
\[
\text { A * v ( } \mathcal{j})=\operatorname{lam} \operatorname{bda}(\mathcal{I}) * v(\mathcal{j})
\]
where lam bda ( \()\) ) is its eigenvalue.
The lefteigenvectoru ( \(\mathcal{F}\) ) of A satisfies
\(u()^{* *} \mathrm{H} * \mathrm{~A}=\operatorname{lam} \operatorname{bda}(\mathrm{\jmath}){ }^{*} \mathrm{u}()^{* * *} \mathrm{H}\)
where \(u(\mathcal{j}){ }^{* *}\) H denotes the conjugate transpose of \(u(j)\).
The com puted eigenvectors are norm alized to have Euclidean norm equal to 1 and largestcom ponent real.

\section*{ARGUMENTS}
\(J 0 \mathrm{BVL}\) (input)
\(=N\) : lefteigenvectors of A are not com puted;
= V': lefteigenvectors of A are com puted.
JOBVR (input)
= N ': righteigenvectors of A are not com puted;
= V ': righteigenvectors of A are com puted.
N (input) The order of the matrix A. \(\mathrm{N}>=0\).
A (input/output)
On entry, the \(\mathrm{N}-\) by -N m atrix A. On exit, A has been overw ritten.

LD A（input）
The leading dim ension of the array A．LDA＞＝ \(\max (1, N)\) ．

\section*{W R（output）}

W R and W Icontain the real and im aginary parts， respectively，of the com puted eigenvalues．Com－ plex conjugate pairs of eigenvalues appear con－ secutively w ith the eigenvalue having the positive im aginary part first．

W I（output）
See the description forW R ．
VL（output）
If \(\mathrm{OOBVL}=\mathrm{V}\)＇，the left eigenvectors \(u(1)\) are stored one afteranother in the colum ns of VL，in the sam e order as theireigenvahues．If \(\mathrm{OBVL}=\) \(N^{\prime}, \mathrm{VL}\) is notreferenced．If the \(j\) th eigenvalue is real，then \(u(J)=V L(:, ~ J)\) ，the \(j\) th column of \(V L\) ．If the \(j\) th and（ \(j+1\) ）－steigenvalues form \(a\) com plex conjugate pair，then \(u(i)=V L(:, i)+\) i＊VL（：, \(\mathfrak{j}+1\) ）and
\(u(j+1)=V L(: っ)-i^{*} V L(:, j+1)\) ．

LD V L（input）
The leading dim ension of the aray \(\mathrm{V} \mathrm{L} . \mathrm{LDVL}>=1\) ；
if \(J O B V L=V ', L D V L>=N\) 。

VR（input）
If \(\mathrm{OBVR}=\mathrm{V}\)＇，the right eigenvectors \(\mathrm{V}(\mathcal{1})\) are stored one after another in the colum ns of VR，in the sam e order as theireigenvahues．If \(\mathrm{OBVR}=\) \(N^{\prime}, V R\) is notreferenced．If the \(j\) th eigenvalue is real，then \(v(\mathcal{I})=V R(:, \mathcal{I})\) ，the \(j\) th colum \(n\) of VR．If the \(j\) th and（ \(j+1\) ）－steigenvalues form a complex conjugate pair，then \(\mathrm{V}(\mathcal{j})=\operatorname{VR}(:, 7)+\) i＊VR（：, j＋1）and \(\mathrm{V}(j+1)=\mathrm{VR}(:, \boldsymbol{j})-\mathrm{i}^{\star} \mathrm{VR}(:, j+1)\).

LDVR（input）
The leading dim ension of the array \(V\) R．LD VR＞＝1； if \(\mathrm{JOBVR}=\mathrm{V}^{\prime}, \mathrm{LDVR}>=\mathrm{N}\) 。

W ORK（w orkspace）
On exiv，if \(\mathbb{N F O}=0, W\) ORK（ 1 ）retums the optim al LDWORK．

LDW ORK（input）
The dim ension of the amay W ORK．LDW ORK＞＝
\(\max (1,3 * N)\), and if \(\mathrm{OBVL}=\mathrm{V}^{\prime}\) or \(\mathrm{OBCR}=\mathrm{V}^{\prime}\),
LDW ORK \(>=4 * N\). Forgood penform ance, LDW ORK must generally be larger.

If LD W ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LD W ORK is issued by XERBLA.
INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue.
\(>0:\) if \(\mathbb{N F O}=\) i, the \(Q \mathrm{R}\) algorithm failed to com pute all the eigenvalues, and no eigenvectors have been com puted; elem ents i+ 1 N ofW R and \(W\) Icontain eigenvalues w hich have converged.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgeevx - com pute for \(a n \mathrm{~N}\)-by N realnonsym \(m\) etric \(m\) atrix \(A\), the eigenvalues and, optionally, the left and/orright eigenvectors

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SGEEVX BALANC,JOBVL,JOBVR,SENSE,N,A,LDA,W R,W I,VL,}
LDVL,VR,LDVR,\PiO,IHI,SCALE,ABNRM,RCONE,RCONV,W ORK,
LDW ORK,INORK2,\mathbb{NFO)}

```
CHARACTER * 1 BALANC, JOBVL, JOBVR, SEN SE
\(\mathbb{N}\) TEGERN,LDA,LDVL,LDVR, \(\mathbb{L} O, \mathbb{H} I, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{I}\) ORK2(*)
REAL ABNRM

\(\operatorname{SCALE}(*), \operatorname{RCONE}(*), \operatorname{RCONV}(\star), \mathrm{WORK}\left(^{*}\right)\)
SU BROUTINE SGEEVX_64 BALANC, JOBVL, JOBVR,SENSE,N,A,LDA,WR,W I,
    VL,LDVL,VR,LDVR, \(\mathbb{L} O, \mathbb{H} I, S C A L E, A B N R M, R C O N E, R C O N V, W O R K\),
    LDW ORK, \(\mathbb{I N}\) ORK2, \(\mathbb{N} F O\) )
CHARACTER * 1 BALANC, JOBVL, JOBVR, SENSE
\(\mathbb{N}\) TEGER*8 \(\mathrm{N}, \mathrm{LDA}, L D V L, L D V R, \mathbb{L O}, \mathbb{H} \mathrm{I}, \mathrm{LD} W\) ORK, \(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER*8 \(\mathbb{I N}\) ORK 2 (*)
REALABNRM

SCALE (*), RCONE (*),RCONV (*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GEEVX BALANC, JOBVL, JO BVR,SENSE, \(\mathbb{N}], A,[L D A], W R, W I\), VL, [LDVL],VR, [LDVR], \(\mathbb{L O}, \mathbb{H} I, S C A L E, A B N R M, R C O N E, R C O N V\), [W ORK ], [LDW ORK], [IW ORK 2], [ \(\mathbb{N} F O\) ])

CHARACTER ( \(\llcorner E N=1\) ) : : BALANC, JOBVL, JOBVR, SEN SE
\(\mathbb{N}\) TEGER :: N, LDA ,LDVL,LDVR, \(\mathbb{L} O, \mathbb{H} I, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{I W} O R K 2\)
REAL ::ABNRM
REAL,D \(\mathbb{M}\) ENSION (:) ::WR,W I, SCALE,RCONE,RCONV,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A,VL,VR
 W I, VL, [LDVL], VR, [LDVR], ILO, IHI, SCALE, ABNRM,RCONE,RCONV, [W ORK], [LDW ORK], [ \(\mathbb{W}\) ORK 2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::BALANC, JOBVL, JOBVR, SEN SE
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{LD} A, L D V L, L D V R, \mathbb{L}, \mathbb{H} \mathrm{I}, \mathrm{LDW} O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I W}\) ORK2
REAL ::ABNRM
REAL,D \(\mathbb{M}\) ENSION (:) ::WR,W I, SCALE,RCONE,RCONV,W ORK
REAL, D \(\mathbb{M}\) ENSION (:,:) ::A, VL, VR

\section*{C INTERFACE}
\#include <sunperfh>
void sgeevx (charbalanc, char jobvl, char jobvr, char sense, int \(n\), float *a, int lda, float * W r, float * W i, float*vl, intldvl, float *vr, int ldvr, int *ilo, int*ihi, float*scale, float*abnm, float *rcone, float *rconv, int*info);
void sgeevx_64 (charbalanc, char jobvl, char jobvr, char sense, long \(n\), float *a, long lda, float * w r, float *W i, float* *l, long ldvl, float *vr, long ldvr, long *ilo, long *ihi, float*scale, float *abnm, float *rcone, float *rconv, long *info);

\section*{PURPOSE}
sgeevx com putes for an \(N\) boy- \(N\) real nonsym \(m\) etric \(m\) atrix \(A\), the eigenvalues and, optionally, the left and/or right eigenvectors.

Optionally also, it com putes a balancing transform ation to im prove the conditioning of the eigenvalues and eigenvectors ( \(\mathbb{L} O, \mathbb{H}\) I, SCA LE , and A BN RM ), reciprocal condition num bers for the eigenvalues (RCONDE), and reciprocal condition num bers for the right eigenvectors (RCONDV).

The righteigenvectorv (i) ofA satisfies
\[
A * v(\mathcal{J})=\operatorname{lam} \operatorname{bda}(\mathcal{J}) * v(\mathcal{I})
\]
w here lam bda ( 7 ) is its eigenvalue.
The lefteigenvectoru ( \()\) ) of A satisfies
\[
\mathrm{u}()^{*}{ }^{*} \mathrm{H} * \mathrm{~A}=\operatorname{lam} \operatorname{bda}(\mathrm{\jmath}) * \mathrm{u}()^{* *} \mathrm{H}
\]
where \(u(\mathcal{j}){ }^{* *} \mathrm{H}\) denotes the conjugate transpose of \(u(\mathcal{j})\).
The com puted eigenvectors are norm alized to have Euclidean norm equal to 1 and largest com ponent real.

B alancing a \(m\) atrix \(m\) eans perm uting the row sand colum ns to m ake itm ore nearly upper triangular, and applying a diagonalsim ilarity transform ation \(D\) * \(A\) * \(D\) ** \((-1)\), where \(D\) is a diagonalm atrix, to \(m\) ake its row \(s\) and colum ns closer in norm and the condition num bers of its eigenvalues and eigenvectors sm aller. The com puted reciprocalcondition num bers comespond to the balanced \(m\) atrix. Perm uting row \(s\) and colum ns will not change the condition num bers (in exact arithm etic) butdiagonalscaling w ill. For further explanation of balancing, see section 4.102 of the LA PACK U sers' G uide.

\section*{ARGUMENTS}

BALANC (input)
Indicates how the inputm atrix should be diagonally scaled and/orperm uted to im prove the conditioning of its eigenvalues. \(=\mathrm{N}\) ': D o not diagonally scale orperm ute;
\(=\mathrm{P}\) ':Perform perm utations to m ake the m atrix m ore nearly upper triangular. D o notdiagonally scale; = S ': D iagonally scale the m atrix, ie. replace \(A\) by \(D\) *A *D ** \((-1)\), where \(D\) is a diagonal \(m\) atrix chosen to \(m\) ake the row \(s\) and colum ns of \(A\) m ore equal in norm .D o notperm ute; \(=\mathrm{B}\) ':Both diagonally scale and perm ute A.

C om puted reciprocalcondition num bers w illbe for the \(m\) atrix afterbalancing and/orperm uting. Per\(m\) uting does not change condition num bers (in exact arithm etic), butbalancing does.

JO BVL (input)
\(=\mathrm{N}\) ': lefteigenvectors of A are not com puted;
\(=\mathrm{V}\) ': lefteigenvectors of A are com puted. If
SEN SE = E 'or B', JO BV L m ust= V'.

JO BVR (input)
\(=\mathrm{N}\) ': righteigenvectors of A are not com puted;
\(=\mathrm{V}\) ': righteigenvectors of A are com puted. If

SENSE \(=\) E'or B', JOBVR m ust \(=\) V'.

SEN SE (input)
D eterm ines which reciprocal condition num bers are com puted. = N ': N one are com puted;
\(=\mathrm{E}\) : C om puted foreigenvalues only;
\(=\mathrm{V}\) : C om puted for righteigenvectors only;
= B : \(: \mathrm{C}\) om puted foreigenvalues and right eigenvectors.

If SEN \(S E=E\) 'or \(B\) ', both left and right eigenvectors must also be computed ( \(\mathrm{JOBVL}=\mathrm{V}\) 'and JOBVR = V ).

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input/output)
On entry, the \(N-b y-N\) m atrix A. On exit, A has
been overw ritten. If JOBVL = V'orJOBVR=V', A contains the realSchur form of the balanced version of the inputm atrix \(A\).

LDA (input)
The leading din ension of the array A. LD A >= \(\max (1, N)\).

\section*{W R (output)}

W R and W Icontain the real and im aginary parts, respectively, of the com puted eigenvalues. Com plex conjugate pairs of eigenvalues w ill appear consecutively \(w\) ith the eigenvalue having the positive im aginary part first.

W I (output)
See the description forW R.

VL (output)
If \(\mathrm{JOBVL}=\mathrm{V}\) ', the left eigenvectors \(u(i)\) are stored one afteranother in the colum ns of V L, in the sam e order as theireigenvalues. If \(00 \mathrm{BVL}=\) N ', VL is not referenced. If the \(j\) th eigenvalue is real, then \(u(\eta)=V L(: r)\), the \(j\) th colum \(n\) of \(V L\). If the \(j\) th and ( \(j+1\) )-steigenvalues form a com plex conjugate pair, then \(u(\mathcal{J})=V L(:, 7)+\) \(i^{\star} V L(: j+1)\) and \(u(j+1)=V L(:, ~>)-i \star V L(:, j+1)\).

LD V L (input)
The leading dim ension of the array V L. LD V L >=1; if JO BVL \(=\mathrm{V}\) ', LD VL >= N .

\section*{VR (output)}

If JO BVR \(=\mathrm{V}\) ', the right eigenvectors \(\mathrm{v}(\mathrm{J})\) are stored one after another in the colum ns of VR, in the sam e order as theireigenvalues. If \(J O B V R=\) \(N^{\prime}, \mathrm{VR}\) is not referenced. If the \(j\) th eigenvalue is real, then \(v(j)=V R(: 1)\), the \(j\) th column of \(V R\). If the \(j\) th and ( \(j+1\) )-steigenvalues form a com plex conjugate pair, then \(\operatorname{V}(\mathcal{j})=\operatorname{VR}(:, 7)+\) i*VR (:, \(\mathrm{j}+1\) ) and \(\mathrm{V}(j+1)=\operatorname{VR}(:, \boldsymbol{\jmath})-\mathrm{i} \star \mathrm{VR}(:, \mathfrak{j}+1)\).
LDVR (input)
The leading dim ension of the array V R . LD V R >=1, and if \(J O B V R=V ', L D V R>=N\).

ㅍㅇ (output)
IO and \(\mathbb{H}\) I are integervalues determ ined when \(A\) w as balanced. The balanced \(\mathrm{A}(\mathrm{i}, \mathrm{J})=0\) if I> J and \(J=1, \ldots, \mathbb{I L} O-1\) or \(I=\mathbb{H} \mathrm{I}+1, \ldots, \mathrm{~N}\).

IH I (output)
See the description of ILO .

SCALE (output)
D etails of the perm utations and scaling factors applied w hen balancing A. IfP ( \()\) ) is the index of the row and column interchanged \(w\) ith row and colum \(n j\) and \(D(j)\) is the scaling factorapplied to row and colum \(n j\) then SCA LE \((J)=P(J)\), for \(J=1, \ldots\), IUO \(-1=\mathrm{D}(J)\), for \(J=\mathbb{H O}, \ldots, \mathbb{H} I=\) \(P\) (J) for \(J=\mathbb{H} I+1, \ldots, N\). The order in which the interchanges are \(m\) ade is \(N\) to \(\mathbb{H} \mathrm{I}+1\), then 1 to [10-1.

ABNRM (output)
The one-nom of the balanced \(m\) atrix (the \(m\) axim um of the sum of absolute values of elem ents of any colum n).

RCONE (output)
RCONE ( \()\) ) is the reciprocalcondition num ber of the \(j\) th eigenvalue.

RCONV (output)
RCONV ( 7 ) is the reciprocalcondition num ber of the jth righteigenvector.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dim ension of the anay \(W\) ORK. IfSENSE \(=N^{\prime}\)
or E', LDW ORK >= max ( \(1,2 \star \mathrm{~N}\) ), and if \(\mathrm{JOBVL}=\mathrm{V}^{\prime}\) or \(\operatorname{JOBVR}=\mathrm{V}^{\prime}\), LD W ORK \(>=3{ }^{\star} \mathrm{N}\). IfSENSE \(=\mathrm{V}^{\prime}\) or B', LD W ORK >= \(N *(N+6)\). Forgood perform ance, LDW ORK m ust generally be larger.

IfLDW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of theW ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

IW ORK 2 (w orkspace)
dim ension ( \(2 * \mathrm{~N}-2\) ) IfSEN SE \(=\mathrm{N}\) 'or E ', notreferenced.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvahue.
> 0 : if \(\mathbb{N} F O=i\), the \(Q R\) algorithm failed to com pute all the eigenvalues, and no eigenvectors or condition num bers have been com puted; elem ents \(1: I H O-1\) and i+ \(1 \mathbb{N}\) ofW R and W Icontain eigenvalues which have converged.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgegs - routine is deprecated and has been replaced by routine SG G ES

\section*{SYNOPSIS}
```

SU BROUTINE SGEGS (JO BV SL, JO BV SR,N,A,LDA,B,LD B,A LPHAR,A LPHA I,
BETA,VSL,LDVSL,VSR,LDVSR,W ORK,LDW ORK,INFO)
CHARACTER * 1 JOBVSL,JOBVSR
\mathbb{NTEGERN,LDA,LDB,LDVSL,LDVSR,LDW ORK,INFO}
REAL A (LDA,*), B (LDB,*), ALPHAR (*), ALPHAI(*), BETA (*),
VSL (LDVSL,*),VSR (LDVSR,*),W ORK (*)
SU BROUTINE SGEGS_64 (JO BV SL,NO BV SR,N,A ,LD A ,B,LD B,ALPHAR,
ALPHAI,BETA,VSL,LDVSL,VSR,LDVSR,WORK,LDW ORK,INFO)
CHARACTER * 1 JOBVSL, JO BV SR
$\mathbb{N} T E G E R * 8 N, L D A, L D B, L D V S L, L D V S R, L D W O R K, \mathbb{N F O}$

```

```

VSL (LDVSL,*),VSR (LDVSR, $\left.{ }^{\star}\right), \mathrm{W}$ ORK ( $\left.{ }^{\star}\right)$

```

\section*{F95 INTERFACE}

SU BROUTINE GEGS (ODBVSL, JO BVSR, \(\mathbb{N}], A,[L D A], B,[L D B], A L P H A R\), A LPHA I, BETA, V SL, [LDVSL],VSR, [LDVSR], [W ORK], [LDW ORK ], [NFO ])

CHARACTER (LEN=1)::JOBVSL, JO BV SR
\(\mathbb{N} T E G E R:: N, L D A, L D B, L D V S L, L D V S R, L D W O R K, \mathbb{N} F O\)
REAL,D \(\mathbb{I M} E N S I O N(:):: A L P H A R, A L P H A I, B E T A, W\) ORK
REAL,D IM ENSION (:,:) ::A,B,VSL,VSR
SU BROUTINE GEGS_64 (JOBVSL, JOBVSR, \(\mathbb{N}]\), A, [LDA ], B, [LD B ],A LPHAR, A LPHAI, BETA, VSL, [LDVSL],VSR, [LDVSR], [W ORK], [LDW ORK], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1) :: JO BV SL , JO BV SR
\(\mathbb{N}\) TEGER (8) ::N,LDA,LDB,LDVSL,LDVSR,LDW ORK, \(\mathbb{N} F O\)
REAL,D \(\mathbb{I M} E N S I O N(:):: A L P H A R, A L P H A I, B E T A, W\) ORK
REAL,D \(\mathbb{M}\) ENSION (: : : : : A, B,VSL,VSR

\section*{C INTERFACE}
\#include < sunperfh>
void sgegs (char jobvsl, char jobvssr, intn, float *a, int
lda, float *b, int ldb, float *alphar, float
*alphai, float *beta, float *vsl, int ldvsl, float
*vsr, int ldvsr, int*info);
void sgegs_64 (char jobvsll, char jobvssr, long n, float *a, long lda, float *b, long ldb, float *alphar, float *alphai, float *beta, float *vsl, long ldvsl, float *vsr, long ldvss, long *info);

\section*{PURPOSE}
sgegs routine is deprecated and has been replaced by routine SG GES.

SGEGS com putes for a pair of \(N\) by -N real nonsym \(m\) etric \(m\) atrices A, B: the generalized eigenvalues (alphar+/alphai*i, beta), the realSchur form (A, B), and optionally left and/or right Schurvectors ( V SL and V SR).
(Ifonly the generalized eigenvalues are needed, use the driverSGEGV instead.)

A generalized eigenvalue for a pairof \(m\) atrioes ( \(A, B\) ) is, roughly speaking, a scalar w ora ratio alpha/beta \(=\mathrm{w}\), such that A -w *B is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation forbeta \(=0\), and even forboth being zero. A good beginning reference is the book, "M atrix C om putations", by G.Golub \& C .van Loan (Johns H opkins U .Press)

The (generalized) Schur form of a pair of \(m\) atrioes is the result of \(m\) ultiplying both \(m\) atrices on the leftby one orthogonalm atrix and both on the rightby another orthogonalm atrix, these tw o orthogonalm atrices being chosen so as to bring the pairofm atrices into (real) Schur form .

A pair ofm atrices \(A, B\) is in generalized realSchur form if \(B\) is upper triangularw th non-negative diagonaland \(A\) is block uppertriangularw ith 1-by-1 and 2 -by-2 blocks. 1-by-1 blocks comespond to real generalized eigenvalues,
while 2-by-2 blocks ofA willbe "standardized" by making the corresponding elem ents ofB have the form :
[ a 0 ]
[ 0 b ]
and the pair of corresponding 2 -by -2 blocks in \(A\) and \(B\) w ill have a com plex conjugate pair of generalized eigenvalues.

The left and rightSchurvectors are the colum ns ofV SL and VSR, respectively, where VSL and VSR are the orthogonal \(m\) atrices \(w\) hich reduce \(A\) and \(B\) to Schur form :

Schurform of \((A, B)=((N S L) * * T A(N S R),(N S L) * * T B(N S R))\)

\section*{ARGUMENTS}

JO BV SL (input)
= N ': do notcom pute the leftSchurvectors;
\(=\mathrm{V}\) : com pute the leftSchurvectors.
\(J O B V S R\) (input)
\(=N\) ': do notcom pute the rightSchurvectors;
\(=\mathrm{V}\) : com pute the rightSchurvectors.
N (input) The order of the m atrices A , B , V SL, and V SR. N \(>=0\).

A (input/output)
O \(n\) entry, the first of the pair ofm atrioes whose generalized eigenvalues and (optionally) Schur vectors are to be com puted. On exit, the general ized Schur form of A. N ote: to avoid overflow, the Frobenius norm of the \(m\) atrix A should be less than the overflow threshold.

LD A (input)
The leading dim ension ofA . LD A \(>=\max (1, \mathbb{N})\).
B (input/output)
O \(n\) entry, the second of the pair ofm atrices w hose generalized eigenvalues and (optionally) Schur vectors are to be com puted. On exit, the generalized Schur form of B. N ote: to avoid overflow, the Frobenius norm of the matrix B should be less than the overflow threshold.

LD B (input)
The leading dim ension ofB . LD B \(>=m\) ax \((1, N)\).

ALPHAR (output)
On exit, (ALPHAR ( ) + ALPHAI ( 1 *i) BETA ( \(\boldsymbol{\lambda}\) ) ,于1,...,N, will be the generalized eigenvalues.
ALPHAR ( \() ~+~ A L P H A I(\beth * i, ~ j 1, \ldots, N\) and BETA ( 7\(), 1, \ldots, N\) are the diagonals of the com plex Schur form ( \(\mathrm{A}, \mathrm{B}\) ) thatw ould result if the 2-by-2 diagonal blocks of the realSchur form of ( \(A, B\) ) w ere further reduced to triangular form using 2 łoy -2 com plex unitary transform ations. If A LPHAI( \(\mathcal{j})\) is zero, then the \(j\) th eigenvalue is real; if positive, then the jth and (j+1)-st eigenvahues are a com plex conjugate pair, w ith A LPH A I(j+1) negative.

N ote: the quotients \(A \operatorname{LPHAR}\) ( 1 ) BETA ( \()\) ) and A LPHAI ( ) , BETA ( \()\) m ay easily over-orunderflow, and BETA (ㄱ) m ay even be zero. Thus, the user should avoid naively computing the ratio alpha/beta. H ow ever, A LPH AR and A LPH A I w ill be alw ays less than and usually com parable w ith norm (A) in magnitude, and BETA alw ays less than and usually com parable \(w\) ith norm (B).

\section*{A LPH A I (output)}

See the description forA LPH A R .

\section*{BETA (output)}

See the description forA LPH A R .

VSL (input)
If JOBVSL = V',VSL willcontain the left Schur vectors. (See "Punpose", above.) N ot referenced if \(J O B V S L=N^{\prime}\).

LDVSL (input)
The leading dim ension of the \(m\) atrix VSL. LDVSL \(>=1\), and if \(\mathrm{OBVSL}=\mathrm{V}^{\prime}, \mathrm{LDVSL}>=\mathrm{N}\).

VSR (input)
If OB BVSR \(=V\) ',VSR willcontain the right Schur
vectors. (See "Puppose", above.) N ot referenced if \(J O B V S R=N^{\prime}\) 。

LDVSR (input)
The leading dim ension of the \(m\) atrix \(V\) SR.LD VSR \(>=\) 1 , and if OB B \(S R=V^{\prime}\) ', LD \(V S R>=N\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al

LDW ORK.

LDW ORK (input)
The dim ension of the array \(W\) ORK. LDW ORK \(>=\) \(\max (1,4 \star N)\). For good perform ance, LDW ORK must generally be larger. To com pute the optim alvalue of LDW ORK, call ILAENV to getblocksizes (for SGEQRF, SORM QR, and SORGQR .) Then com pute: NB-
M AX of the blocksizes for SGEQRF, SORM QR, and SORGQR The optim alLDW ORK is \(2 * N+N * \mathbb{N B + 1 )}\).

IfLD W ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK aray, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LD W ORK is issued by XERBLA.
\(\mathbb{I N} F O\) (output)
= 0: successfiulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the i-th argum enthad an ille-
galvahue.
\(=1, \ldots, N\) : The \(\mathrm{Q} Z\) teration failed. ( \(\mathrm{A}, \mathrm{B}\) ) are
not in Schurform, butALPHAR ( \()\), ALPHAI( \()\), and
BETA ( 1 ) should be comect for \(\ddagger \mathbb{N} \mathrm{FO}+1, \ldots, N\). >
N : enors that usually indicate LA PA C K problem \(s\) :
\(=\mathrm{N}+1\) : error retum from SG GBAL
\(=\mathrm{N}+2\) : error retum from \(S G E Q R F\)
\(=\mathrm{N}+3\) : error retum from \(S O R M Q R\)
\(=\mathrm{N}+4\) : error retum from \(S O R G Q R\)
\(=\mathrm{N}+5\) : error retum from SG G H RD
\(=\mathrm{N}+6\) : emor retum from SHGEQZ (other than failed
iteration) \(=\mathrm{N}+7\) : enor retum from SG GBAK (com put-
ing V SL)
\(=\mathrm{N}+8\) : emor retum from SGGBAK (com puting \(V \mathrm{SR}\) )
\(=\mathrm{N}+9\) : error retum from SLA SCL (various places)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgegv - routine is deprecated and has been replaced by routine SG G EV

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGEGV (JOBVL,JOBVR,N,A,LDA,B,LDB,ALPHAR,ALPHAI,}
BETA,VL,LDVL,VR,LDVR,W ORK,LDW ORK,INFO)
CHARACTER * 1 JOBVL,JOBVR
\mathbb{NTEGERN,LDA,LDB,LDVL,LDVR,LDW ORK,INFO}
REAL A (LDA,*), B (LDB,*), ALPHAR (*), ALPHAI(*), BETA (*),
VL (LDVL,*),VR (LDVR,*),W ORK (*)
SU BROUTINE SGEGV_64(JOBVL,JOBVR,N,A,LDA,B,LD B,ALPHAR,A LPHA I,
BETA,VL,LDVL,VR,LDVR,W ORK,LDW ORK,\mathbb{NFO)}
CHARACTER * 1 JOBVL,JOBVR
\mathbb{NTEGER*8 N,LDA,LDB,LDVL,LDVR,LDW ORK, INFO}
REAL A (LDA,*), B (LDB,*), ALPHAR (*), ALPHAI(*), BETA (*),
VL (LDVL,*),VR (LDVR,*),WORK (*)

```

\section*{F95 INTERFACE}

SUBROUTINE GEGV (JOBVL, JOBVR, \(\mathbb{N}], A,[L D A], B,[L D B], A L P H A R\), ALPHAI, BETA, VL, [LDVL],VR, [LDVR], [W ORK], [LDW ORK], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1)::JOBVL, JOBVR
\(\mathbb{N} T E G E R:: N, L D A, L D B, L D V L, L D V R, L D W O R K, \mathbb{N} F O\)
REAL,D \(\mathbb{I M} E N S I O N(:):: A L P H A R, A L P H A I, B E T A, W\) ORK
REAL,D IM ENSION (:,:) ::A,B,VL,VR

SU BROUTINE GEGV_64 (JOBVL, JO BVR, \(\mathbb{N}], A,[L D A], B,[L D B], A L P H A R\),

CHARACTER ( \(L E N=1\) ) :: JOBVL, OBVR
\(\mathbb{N} \operatorname{TEGER}(8):: N, L D A, L D B, L D V L, L D V R, L D W O R K, \mathbb{N} F O\) REAL,D \(\mathbb{I}\) ENSION (:) ::ALPHAR,ALPHAI,BETA,WORK REAL,D IM ENSION (:,:) ::A,B,VL,VR

\section*{C INTERFACE}
\#include <sunperfh>
void sgegv (char jobvl, char j.jbvr, intn, float *a, int lda, float *b, int lalo, float *alphar, float*alphai, float *beta, float *vl, intldvl, float *vr, int ldvr, int*info);
void sgegv_64 (char j̣bvl, char jobvr, long n, float *a, long lda, float *b, long ldb, float*aliphar, float *alphai, float *beta, float*vl, long ldvl, float *vr, long ldvr, long *info);

\section*{PURPOSE}
sgegv routine is deprecated and has been replaced by routine SG GEV .

SG EGV com putes for a pair of \(n-b y-n\) real nonsymm etric \(m\) atrices A and B, the generalized eigenvalues (alphar +/alphai*i, beta), and optionally, the left and/or right generalized eigenvectors ( L and VR).

A generalized eigenvalue for a pair of \(m\) atrices \((A, B)\) is, roughly speaking, a scalar w ora ratio alpha/beta \(=\mathrm{w}\), such that A -w *B is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation forbeta \(=0\), and even forboth being zero. A good beginning reference is the book, "M atrix C om putations", by G .G olub \& C.van Loan (Johns H opkins U .Press)

A right generalized eigenvector corresponding to a generalized eigenvalue w fora pair ofm atrices \((A, B)\) is a vector \(r\) such that ( \(A-w B\) ) \(r=0\). A left generalized eigenvector is a vectorlsuch that \(1 * * H\) * \((A-w B)=0\), where \({ }^{1 * *} \mathrm{H}\) is the conjugate-transpose of l.

N ote: this routine perform s "fullbalancing" on A and B see "FurtherD etails", below .

\section*{ARGUMENTS}

JO BVL (input)
\(=\mathrm{N}\) ': do notcom pute the leftgeneralized eigenvectors;
\(=\mathrm{V}\) ': com pute the leftgeneralized eigenvectors.

JO BVR (input)
\(=\mathrm{N}\) : : do not com pute the right generalized eigenvectors;
\(=\mathrm{V}\) : com pute the right generalized eigenvectors.

N (input) The order of the m atriges \(\mathrm{A}, \mathrm{B}, \mathrm{VL}\), and \(\mathrm{VR} . \mathrm{N}>=\) 0.

A (input/output)
O n entry, the first of the pair ofm atrices w hose generalized eigenvalues and (optionally) generalized eigenvectors are to be com puted. On exit, the contents w ill have been destroyed. Fora description of the contents of A on exit, see "FurtherD etails", below .)

LD A (input)
The leading dim ension ofA. LD A \(>=\max (1, N)\).

B (input/output)
O n entry, the second of the pair ofm atrices w hose generalized eigenvalues and (optionally) generalized eigenvectors are to be com puted. On exit, the contents will have been destroyed. Fora description of the contents of B on exit, see "FurtherD etails", below .)

LD B (input)
The leading dim ension ofB. LD \(B>=m a x(1, N)\).

\section*{ALPHAR (output)}

On exit, (ALPHAR ( ) + ALPHAI ( 1 *i) BETA ( 1 ), \(j 1, \ldots, N\), w ill be the generalized eigenvalues. IfA LPH A I( \(\mathfrak{j}\) ) is zero, then the \(j\) th eigenvalue is real; if positive, then the \(j\) th and ( \(j+1\) )-st eigenvahues are a com plex conjugate pair, w ith A LPH A I (j+1) negative.

Note: the quotients ALPHAR ( 7 ) BETA ( \()\) and A LPHAI( ) BETA ( \()\) m ay easily over-orunderflow, and BETA ( 7 ) m ay even be zero. Thus, the user should avoid naively computing the ratio alpha/beta. H ow ever, A LPH AR and A LPHAI w ill be
alw ays less than and usually com parable w ith norm (A) in magnitude, and BETA alw ays less than and usually com parable w ith norm (B).

\section*{A LPH A I (output)}

See the description of A LPH AR .

BETA (output)
See the description of A LPH AR .
VL (output)
If \(30 \mathrm{BVL}=\mathrm{V}\) ', the left generalized eigenvectors.
(See "Purpose", above.) Realeigenvectors take one colum n, com plex take tw o colum ns, the first for the realpart and the second for the im aginary part. Complex eigenvectors correspond to an eigenvalue w ith posilive im aginary part. Each eigenvectorw illbe scaled so the largest com ponent w ill have abs (realpart) + abs(im ag. part) \(=1\), *except* that for eigenvalues \(w\) ith alpha=beta=0, a zero vectorw illbe retumed as the comesponding eigenvector. N ot referenced if JOBVL = N '.

LDVL (input)
The leading dim ension of the \(m\) atrix \(\mathrm{V} \mathrm{L} . \mathrm{LD} \mathrm{VL}>=1\), and if \(\mathrm{OOBVL}=\mathrm{V}^{\prime}, \mathrm{LDVL}>=\mathrm{N}\) 。

\section*{VR (output)}

If \(\mathrm{OB} \mathrm{BR}=\mathrm{V}\) ', the right generalized eigenvectors. (See "Purpose", above.) Realeigenvectors take one colum n, com plex take two colum ns, the first for the real partand the second for the im aginary part. C om plex eigenvectors correspond to an eigenvalue \(w\) ith positive im aginary part. Each eigenvectorw illbe scaled so the largest com ponent w ill have abs(real part) + abs(im ag. part) \(=1\), *except* that for eigenvalues \(w\) ith alpha=beta=0, a zero vectorw illbe retumed as the corresponding eigenvector. N otreferenced if JOBVR = N'.

LDVR (input)
The leading dim ension of the \(m\) atrix \(V R\).LD \(V R>=1\), and if \(\mathrm{OBVR}=\mathrm{V}^{\prime}, \mathrm{LDVR}>=\mathrm{N}\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)

The dim ension of the amay W ORK. LDW ORK >= \(m\) ax \((1,8 \star N)\). For good perform ance, LDW ORK m ust generally be larger. To com pute the optim alvalue of LDW ORK, call HAENV to getblocksizes (for SGEQRF, SORMQR, and SORGQR .) Then com pute: NB -
\(M A X\) of the blocksizes for \(S G E Q R F\), \(S O R M Q R\), and SORGQR; The optim alLDW ORK is: \(2{ }^{*} \mathrm{~N}+\mathrm{MAX}\left(6^{*} \mathrm{~N}\right.\), \(N\) * (NB+1)).

IfLDW ORK \(=-1\), then a w ork.space query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LD W ORK is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the \(i-\) th argum ent had an illegalvalue.
\(=1, \ldots, N\) : The Q Z iteration failed. N o eigenvec-
tors have been calculated, but ALPHAR (1),
A LPHAI( \()\), and BETA ( 1 ) should be correct for
\(\dot{F} \mathbb{N}\) FO \(+1, \ldots, N .>N\) : errors thatusually indi-
cate LA PA CK problem s:
\(=\mathrm{N}+1\) : error retum from SGGBAL
\(=\mathrm{N}+2\) : error retum from SGEQRF
\(=N+3\) : error retum from \(S O R M Q R\)
\(=\mathrm{N}+4\) : error retum from \(S O R G Q R\)
\(=\mathrm{N}+5\) : error retum from SGGHRD
\(=\mathrm{N}+6\) : error retum from \(\mathrm{SH} G E Q Z\) (other than failed
iteration) \(=\mathrm{N}+7\) : error retum from STGEVC
\(=\mathrm{N}+8\) : enor retum from SGGBAK (com puting VL )
\(=\mathrm{N}+9\) : error retum from SGGBAK (com puting \(V R\) )
\(=\mathrm{N}+10\) : enor retum from SLA SC L (various calls)

\section*{FURTHER DETAILS}

Balancing

This driver calls SG G B A L to both perm ute and scale row s and colum ns of \(A\) and \(B\). The perm utations \(P L\) and \(P R\) are chosen so that \(\mathrm{PL}{ }^{*} \mathrm{~A} * P R\) and \(P L * B * R\) w illbe upper triangular except for the diagonal blocksA (i:ji:j) and B (i:ji:j) w ith i and jas close together as possible. The diagonal scaling \(m\) atrices DL and DR are chosen so that the pair \(D L * P L * A * P R * D R, D L * P L * B * P R * D R\) have elem ents close to one (except for the elem ents that start out zero.)

A fter the eigenvalues and eigenvectors of the balanced
\(m\) atrices have been com puted, SG G BA K transform s the eigenvectors back to what they w ould have been (in perfect arithm etic) ifthey had notbeen balanced.

Contents of \(A\) and \(B\) on Exit

If any eigenvectors are com puted (either \(J O B V L=V\) ' or JO BVR=V' or both), then on exit the arrays \(A\) and \(B\) w ill contain the realSchur form [ \({ }^{\star}\) ] of the "balanced" versions of A and B. If no eigenvectors are com puted, then only the diagonalblocks w illbe comect.
[*] See SH GEQ Z, SGEGS, or read the book "M atrix Com putations",
by G olub \& van Loan, pub. by Johns H opkins U .Press.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgehrd -reduce a realgeneralm atrix A to upper H essenberg form H by an orthogonal sim ilarity transform ation

\section*{SYNOPSIS}


```

REALA (LDA,*),TAU (*),W ORK IN (*)

```


```

REALA (LDA,*),TAU (*),WORK IN (*)

```

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E \operatorname{GEHRD}(\mathbb{N}], \mathbb{L O}, \mathbb{H} I, A,[L D A], T A U,[W O R K \mathbb{N}],[L W O R K \mathbb{N}]\), [ \(\mathbb{N}\) FO ])
\(\mathbb{N} T E G E R:: N, \mathbb{L} O, \mathbb{H} I, L D A, L W O R K \mathbb{N}, \mathbb{N} F O\)
REAL,D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK \(\mathbb{N}\)
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A
SU BROUTINE GEHRD_64 ( \(\mathbb{N}], \mathbb{L} O, \mathbb{H} I, A,[L D A], T A U,[W O R K \mathbb{N}],[L W O R K \mathbb{N}]\), [ \(\mathbb{N} F O\) ])
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{Z} \mathrm{O}, \mathbb{H} \mathrm{I}, \mathrm{LD} A, L W O R K \mathbb{N}, \mathbb{N} F O\)
REAL,D \(\mathbb{I M} E N S \mathbb{I O N}(:):: T A U, W\) ORK \(\mathbb{N}\)
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A
\#include < sunperfh>
void sgehrd (intn, intile, int ini, float *a, int lda, float *tau, int*info);
void sgehrd_64 (long n, long ilo, long ihi, float *a, long lda, float *tau, long *info);

\section*{PURPOSE}
sgehrd reduces a realgeneralm atrix A to upper H essenberg
form \(H\) by an orthogonalsim ilarity transform ation: Q '* A * \(Q=H\).

\section*{ARGUMENTS}

N (input) The order of the matrix \(A . N>=0\).
ㅍO (input)
It is assum ed that A is already upper triangular in row s and colum ns \(1: \mathbb{I L O - 1}\) and \(\mathbb{H} \mathrm{I}+1 \mathbb{N} . \mathbb{W}\) and \(\mathbb{H}\) I are norm ally setby a previous call to SG EBA L; otherw ise they should be set to 1 and \(N\) respectively. See FurtherD etails.

IH I (input)
See the description of IIO .
A (input/output)
O \(n\) entry, the N -by -N generalm atrix to be reduced.
On exit, the upper triangle and the first subdiagonalofA are overw rilten w ith the upper H essenberg \(m\) atrix \(H\), and the elem ents below the first subdiagonal, w ith the array TAU, represent the orthogonal \(m\) atrix \(Q\) as a productof elem entary reflectors. See FurtherD etails.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, \mathbb{N})\).

TAU (output)
The scalar factors of the elem entary reflectors (see Further Details). Elem ents 1:THO-1 and IH IN -1 of TA U are setto zero.

W ORK \(\mathbb{N}\) (w orkspace)

On exit, if \(\mathbb{N F O}=0, W O R K \mathbb{N}\) (1) retums the optim allW ORK \(\mathbb{N}\).

LW ORK \(\mathbb{N}\) (input)
The length of the array \(W O R K \mathbb{N}\). LW ORK \(\mathbb{N}>=\) \(\max (1, \mathbb{N})\). For optim um perform ance LW ORK \(\mathbb{N}>=\) \(\mathrm{N} * \mathrm{NB}\), where NB is the optim alblocksize.

If LW ORK \(\mathbb{N}=-1\), then a workspace query is assum ed; the routine only calculates the optim al size of the \(W\) ORK \(\mathbb{N}\) array, retums this value as the firstentry of the \(W\) ORK \(\mathbb{N}\) array, and no error \(m\) essage related to LW ORK \(\mathbb{N}\) is issued by XERBLA.
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the i-th argum enthad an illegalvahue.

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a productof (hi-ib) ele\(m\) entary reflectors
\[
Q=H \text { ( } 0 \text { ) H (ib+1) . . H (ihi-1) . }
\]

Each H (i) has the form
\[
H(i)=I-\tan * V^{*} V^{\prime}
\]
where tau is a real scalar, and \(v\) is a realvectorw ith \(\mathrm{v}(1: i)=0, v(i+1)=1\) and \(v\) (init 1 n\()=0\); \(\mathrm{v}(\mathrm{i}+2\) : ihi) is stored on exitin A (i+2:ihi,i), and tau in TA U (i).

The contents of A are illustrated by the follow ing exam ple, w ith \(\mathrm{n}=7\), il = \(=2\) and ihi= 6:
```

on entry, on exit,

```
(a a a a a a a) (a a h h h h a) ( a a a a a a) ( a h h h ha) ( a a a a a a) ( \(h\) h h h h h ) ( a a a a a a) ( v2 h \(h \mathrm{~h} h \mathrm{~h})(\mathrm{a}\) a a a a a) ( v2 v3 h h h h ) ( a a a a a a) ( v2 v3 v4 h h h) (
a) (

> a)
\(w\) here a denotes an elem ent of the original \(m\) atrix \(A, h\) denotes a \(m\) odified elem ent of the upper \(H\) essenberg \(m\) atrix \(H\), and videnotes an elem ent of the vector defining \(H\) (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgelqf-com pute an LQ factorization of a realM -by-N m atrix A

\section*{SYNOPSIS}

```

INTEGER M,N,LDA,LDW ORK,NNFO
REALA (LDA,*),TAU (*),W ORK (*)
SUBROUT\mathbb{NE SGELQF_64M,N,A,LDA,TAU,W ORK,LDW ORK, INFO)}
\mathbb{NTEGER*8M,N,LDA,LDW ORK,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REALA (LDA,*),TAU (*),W ORK (*)
F95 INTERFACE

```


```

    REAL,DIM ENSION (:) ::TAU,W ORK
    REAL,D IM ENSION (:,:) ::A
    ```


```

    REAL,DIM ENSION (:) ::TAU,W ORK
    REAL,D IM ENSION (:::) ::A
    C INTERFACE
\#include <sunperfh>

```
void sgelqf(intm , intn, float *a, int lda, float *tau, int
*info);
void sgelqf_64 (long m, long n, float *a, long lda, float *tau, long *info);

\section*{PURPOSE}
sgelqf com putes an LQ factorization of a realM -by-N matrix \(A: A=L * Q\).

\section*{ARGUMENTS}

M (input) The num ber of row s of the matrix A. M >=0.
N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the M foy-N m atrix A. On exit, the ele\(m\) ents on and below the diagonal of the amay contain the \(m\) boy-m in ( \(m, n\) ) low er trapezoidalm atrix \(L\) ( \(L\) is low er triangularifm <= n); the elem ents above the diagonal, w ith the aray TA U , represent the orthogonalm atrix \(Q\) as a productofelem entary reflectors (see FurtherD etails).

LD A (input)
The leading dim ension of the aray A. LDA >= max (1, M ) .

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails) .

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK ( 1 ) retums the optim al LD W ORK.

LDW ORK (input)
The dim ension of the array \(W\) ORK. LDW ORK \(>=\) \(\mathrm{max}(1, \mathrm{M})\). Foroptim um perform ance LDW ORK >= M *NB, w here N B is the optim alblocksize.

If LD W ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK aray, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LD W ORK is issued by XERBLA .
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\(Q=H(k) \ldots H(2) H(1)\), where \(k=m\) in \((m, n)\).
Each \(H\) (i) has the form
H (i) \(=I-\tan * V^{*} V^{\prime}\)
w here tau is a real scalar, and v is a realvectorw ith \(v(1: i-1)=0\) and \(v(i)=1\); \(v(i+1 n)\) is stored on exit in A ( \(i, i+1 \mathrm{n})\), and tau in TAU (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgels -solve overdeterm ined or underdeterm ined real linear system \(s\) involving an \(M\) by -N matrix \(A\), or its transpose, using a \(Q R\) orLQ factorization ofA

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SGELS (TRANSA,M ,N,NRHS,A,LDA,B,LDB,W ORK,LDW ORK,}
\mathbb{NFO)}
CHARACTER * 1 TRANSA
\mathbb{NTEGERM,N,NRHS,LDA,LDB,LDW ORK,INFO}
REALA (LDA,*),B (LDB,*),W ORK (*)
SU BROUT\mathbb{NE SGELS_64 (TRANSA,M ,N ,NRHS,A,LDA,B,LDB,W ORK,LDW ORK,}
INFO)
CHARACTER * 1 TRANSA
\mathbb{NTEGER*8M,N,NRHS,LDA,LDB,LDW ORK, INFO}
REALA (LDA,*),B (LDB,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GELS ([TRANSA], \(\mathbb{M}], \mathbb{N}], \mathbb{N R H S}], A,[L D A], B,[L D B],[W O R K]\), LDW ORK, [ \(\mathbb{N} F \mathrm{~F}\) ])

CHARACTER ( \(4 E N=1\) ) ::TRANSA
\(\mathbb{N}\) TEGER ::M,N,NRHS,LDA,LDB,LDW ORK, \(\mathbb{N}\) FO
REAL,D IM ENSION (:) ::W ORK
REAL,D IM ENSION (:,:) ::A, B

SU BROUTINE GELS_64 ([TRANSA], \(\mathbb{M}], \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B]\), [ \(\mathbb{W}\) ORK ],LDW ORK, [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N}\) TEGER (8) ::M , N , NRHS,LDA,LDB,LDW ORK, \(\mathbb{N}\) FO
REAL,D \(\mathbb{I M}\) ENSION (:) ::W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void sgels (char, int, int, int, float*, int, float*, int, int*);
void sgels_64 (char, long, long, long, float*, long, float*, long, long*);

\section*{PURPOSE}
sgels solves overdeterm ined or underdeterm ined real linear system \(s\) involving an \(M\)-by -N m atrix \(A\), or its transpose, using a QR orLQ factorization ofA. It is assum ed that A has full rank.

The follow ing options are provided:
1. IfTRANS \(=N\) 'and \(m>=n\) : find the leastsquares solution of
an overdeterm ined system , i.e., solve the least squares problem
\[
\mathrm{m} \text { inim ize }\|\mathrm{B}-\mathrm{A} * \mathrm{X}\| .
\]
2. IfTRAN \(S=N\) 'and \(m<n\) : find the \(m\) inim um norm solution of an underdeterm ined system \(A * X=B\).
3. IfTRANS \(=T\) 'and \(m>=n\) : find the \(m\) in \(m\) um norm solution of
an undeterm ined system \(A * * T * X=B\).
4. IfTRANS = T'and \(\mathrm{m}<\mathrm{n}\) : find the least squares solution of
an overdeterm ined system , ie., solve the least squares problem
\[
\mathrm{m} \text { inim ize }\|\mathrm{B}-\mathrm{A} * * \mathrm{~T} * \mathrm{X}\| .
\]

Several righthand side vectors \(b\) and solution vectors x can be handled in a single call; they are stored as the colum ns of the M by-NRHS righthand side m atrix B and the N -by-NRHS solution \(m\) atrix \(X\).

\section*{ARGUMENTS}

TRANSA (input)
\(=N^{\prime}\) : the linearsystem involves A;
\(=T\) ': the linear system involves \(A * * T\).

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

M (input) The num ber of row s of the matrix A. M >=0.

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the m atrioes B and X .NRH S \(>=0\).
A (input/output)
On entry, the M -by -N m atrix A. On exit, if \(\mathrm{M}>=\) \(\mathrm{N}, \mathrm{A}\) is overw ritten by details of its Q R factorization as retumed by SGEQRF; if \(M<N, A\) is overw rilten by details of its LQ factorization as retumed by SGELQF.

LDA (input)
The leading dim ension of the aray A. LDA >= \(\max (1, M)\).

B (input/output)
O n entry, the \(m\) atrix \(B\) of righthand side vectors,
stored colum nw ise; B is M byy-NRHS ifTRANSA = N',
orN boy-NRHS ifTRANSA = T'. On exit, B is
overw rilten by the solution vectors, stored colum nw ise: ifTRAN SA \(=\mathrm{N}\) 'and \(\mathrm{m}>=\mathrm{n}\), row s 1 to \(n\) ofB contain the least squares solution vectors; the residual.sum ofsquares for the solution in each colum \(n\) is given by the sum of squares of ele\(m\) ents \(N+1\) to \(M\) in thatcolumn; ifTRANSA \(=N\) 'and \(m<n\), row \(s 1\) to \(N\) ofB contain them inim um norm solution vectors; ifTRANSA \(=T\) 'and \(m>=n\), row \(s\) 1 to M ofB contain them inim um norm solution vectors; ifTRANSA \(=T\) 'and \(m<n\), row 1 to \(M\) of \(B\) contain the least squares solution vectors; the residual.sum of squares forthe solution in each colum \(n\) is given by the sum of squares of elem ents \(M+1\) to \(N\) in that colum \(n\).

LD B (input)
The leading dim ension of the aray \(\mathrm{B} . \mathrm{LD} \mathrm{B}>=\) M AX ( \(1, \mathrm{M}, N\) ) 。

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (output)
The dim ension of the amay W ORK. LDW ORK >= \(\max (\)
1, \(\mathrm{M} N+\max (\mathrm{M} N, \mathrm{NRHS})\) ). Foroptim alperfor
\(m\) ance, LDW ORK \(>=\max (1, M N+\max (M N, N R H S) * N B\)
). where \(M N=m\) in \(M N\) ) and \(N B\) is the optim um
block size.

IfLD W ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
theW ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LD W ORK is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an ille-
galvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgelsd - com pute them inim um -norm solution to a real linear
least squares problem

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SGELSD M ,N,NRHS,A,LDA,B,LDB,S,RCOND,RANK,W ORK,}
LW ORK,\mathbb{IN ORK,INFO)}
\mathbb{NTEGER M,N,NRHS,LDA,LDB,RANK,LW ORK,INFO}
INTEGER IN ORK (*)
REALRCOND
REALA (LDA,*),B (LDB,*),S (*),W ORK (*)
SUBROUT\mathbb{NESGELSD_64M,N,NRHS,A,LDA,B,LDB,S,RCOND,RANK,}
W ORK,LW ORK,\mathbb{IN ORK,INFO)}
INTEGER*8M,N,NRHS,LDA,LDB,RANK,LW ORK,\mathbb{NFO}
INTEGER*8 IN ORK (*)
REALRCOND
REALA (LDA,*),B (LDB,*),S (*),W ORK (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE GELSD ( $\mathbb{M}], \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], S, R C O N D$, RANK, [W ORK], [LW ORK ], [IW ORK ], [ $\mathbb{N} F O$ ])
$\mathbb{N} T E G E R:: M, N, N R H S, L D A, L D B, R A N K, L W O R K, \mathbb{N} F O$ $\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: $\mathbb{I N}$ ORK
REAL ::RCOND
REAL,D IM ENSION (:) ::S,W ORK
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A,B

```

SUBROUTINE GELSD_64 ( \(\mathbb{M}], \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], S, R C O N D\), RANK, [W ORK ], [LW ORK ], [ \(\mathbb{W}\) ORK ], [ \(\mathbb{N F O}\) ])
\(\mathbb{N}\) TEGER ( 8 ) :: M, N,NRHS,LDA,LDB,RANK,LW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{I V}\) ORK
REAL ::RCOND
REAL,DIM ENSION (:) ::S,W ORK
REAL,D IM ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void sgelsd (intm, intn, intnrhs, float *a, int lda, float
*b, int ldb, float *s, float roond, int * rank, int *info);
void sgelsd_64 (long m, long n, long nihs, float *a, long lda, float *b, long ldb, float *s, float roond, long *rank, long *info);

\section*{PURPOSE}
sgelsd com putes the \(m\) inim um -norm solution to a real linear least squares problem :
\(m\) inim ize 2 -norm (|b-A *x )
using the singularvalue decom position (SVD) ofA.A is an M -by-N m atrix which \(m\) ay be rank-deficient.

Several righthand side vectors b and solution vectors \(x\) can be handled in a single call; they are stored as the colum ns of the M -by-NRHS righthand sidem atrix \(B\) and the \(N\)-by-NRH S solution \(m\) atrix \(X\).

The problem is solved in three steps:
(1) Reduce the coefficientm atrix A to bidiagonal form \(w\) ith H ouseholder transform ations, reducing the originalproblem
into a "bidiagonal least squares problem " (BLS)
(2) Solve the BLS using a divide and conquer approach.
(3) A pply back all the \(H\) ouseholder tranform ations to solve the original least squares problem .

The effective rank of A is determ ined by treating as zero those singular values which are less than RCO ND tim es the largest singularvalue.

The divide and conqueralgorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on \(m\) achines w ith a guard digit in add/subtract, or on those binary m achines w ithout guard digits w hich subtract like the C ray

X - M P , C ray Y M P , C ray C -90, orC ray-2. It could conceivably fail on hexadecim al or decim al machines \(w\) thout guard digits, butw e know of none.

\section*{ARGUMENTS}

M (input) The num ber of row sofA. \(\mathrm{M}>=0\).
N (input) The num ber of colum nsofA . \(\mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the \(m\) atrices \(B\) and X.NRHS \(>=0\).
A (input/output)
On entry, the M -by -N m atrixA. On exit, A has been destroyed.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, M)\).

B (input/output)
On entry, the M -by \(-N\) RH \(S\) righthand side \(m\) atrix \(B\).
On exit, B is overw ritten by the \(N\) by-NRHS solution \(m\) atrix \(X\). Ifm \(>=n\) and RANK \(=n\), the residual sum-of-squares for the solution in the \(i\)-th colum \(n\) is given by the sum of squares of elem ents \(\mathrm{n}+1 \mathrm{~m}\) in thatcolum n .

LD B (input)
The leading dim ension of the array \(\mathrm{B} . \mathrm{LD} \mathrm{B}>=\) \(\max (1, m a x(M, N))\).

S (output)
The singularvalues of A in decreasing order. The condition number of \(A\) in the 2 -norm \(=\) \(S(1) / S(m\) in \((m, n))\).

RCOND (input)
RCOND is used to determ ine the effective rank of A. Singularvalues \(S\) (i) <= RCOND *S (1) are treated as zero. IfRCOND \(<0, \mathrm{~m}\) achine precision is used instead.

RANK (output)
The effective rank of A, i.e., the num ber of singular values w hich are greater than RCO N D *S (1).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK.LW ORK \(>=1\). The exact \(m\) inim um am ount of w orkspace needed depends on M , N and NRHS. As long as LW ORK is at least \(12 * \mathrm{~N}+2 * \mathrm{~N} * \mathrm{SM} \mathrm{LSIZ}+8 * \mathrm{~N} * \mathrm{NLVL}+\mathrm{N} * \mathrm{NRHS}\) * (SMLSLZ +1 ) \({ }^{* *} 2\), ifM is greater than orequal to \(N\) or \(12 * \mathrm{M}+2 * \mathrm{M} * \mathrm{SMLSIZ}+8 * \mathrm{M} * \mathrm{NLVL}+\mathrm{M}\) *NRHS + (SM LSIZ + 1) **2, ifM is less than N , the code will execute correctly. SM LSIZ is retumed by ILA ENV and is equal to the \(m\) axim um size of the subproblems at the bottom of the com putation tree (usually about 25) , and \(\mathrm{NLVL}=\mathbb{N} T\left(\mathrm{LOG} \_2\right.\) ( \(\mathrm{M} \mathbb{N}(\mathrm{M}, \mathbb{N}\) )/(SM LSIZ+1) ) ) + 1 Forgood penform ance, LW ORK should generally be larger.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IW ORK (w orkspace)

\(M \mathbb{N} M N=M \mathbb{N}(M, N)\) 。

IN FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an illegalvahue.
\(>0\) : the algorithm forcom puting the SVD failed to converge; if \(\mathbb{N} F O=\) i, ioff-diagonalelem ents of an interm ediate bidiagonalform did not converge to zero.

\section*{FURTHER DETAILS}

B ased on contributions by
M ing Gu and Ren-C ang Li, Computer Science D ivision, U niversity of C alifomia atB erkeley, U SA

O sniM arques, LBNLNERSC , U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgelss - com pute the minim um norm solution to a real linear least squares problem

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGELSS M,N,NRHS,A,LDA,B,LDB,SING,RCOND, RANK,}
W ORK,LDW ORK, INFO)
INTEGERM,N,NRHS,LDA,LDB, \mathbb{RANK,LDW ORK, INFO}
REALRCOND
REAL A (LDA,*),B (LDB,*),SING (*),W ORK (*)
SUBROUT\mathbb{NE SGELSS_64M,N,NRHS,A,LDA,B,LDB,S\mathbb{NG,RCOND,\mathbb{RANK,}}\mathbf{N},}\mathbf{N},
W ORK,LDW ORK,INFO)

```
\(\mathbb{N} T E G E R * 8 M, N, N R H S, L D A, L D B, \mathbb{R A N K}, L D W O R K, \mathbb{N F O}\)
REALRCOND
REALA (LDA , *), B (LDB,*), SING (*), WORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GELSS (M) \(\mathbb{M}], \mathbb{N} R H S], A,[L D A], B,[L D B], S \mathbb{N} G, R C O N D\), \(\mathbb{R} A N K, \mathbb{W}\) ORK], [LDW ORK], [ \(\mathbb{N F O}])\)
\(\mathbb{N} T E G E R:: M, N, N R H S, L D A, L D B, \mathbb{R A N K}, L D W O R K, \mathbb{N} F O\)
REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::SING,W ORK
REAL,D IM ENSION (:,:)::A,B
SUBROUTINE GELSS_64 (M) \(\mathbb{M}, \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], S \mathbb{N} G\), RCOND, \(\mathbb{R A N K},\left[\begin{array}{l}\text { W ORK }],[L D W O R K],[\mathbb{N} F O])\end{array}\right.\)
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{NRHS}, \mathrm{LD} A, L D B, \mathbb{R} A N K, L D W O R K, \mathbb{N} F O\)

REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::SING,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include < sunperfh>
void sgelss (intm, intn, intnihs, float *a, int lda, float
*b, int ldb, float * sing, float rcond, int *irank, int*info);
void sgelss_64 (long m, long n, long nihs, float *a, long lda, float *b, long ldb, float *sing, float roond, long *irank, long *info);

\section*{PURPOSE}
sgelss com putes the \(m\) inim um norm solution to a real linear least squares problem :
\(M\) inim ize 2 -norm (|b-A *x \()\).
using the singularvalue decom position (SVD) ofA.A is an M -by-N m atrix which m ay be rank-deficient.

Several righthand side vectors \(b\) and solution vectors x can be handled in a single call; they are stored as the colum ns of the M - by-NRH S righthand side m atrix B and the N -by-NRH S solution \(m\) atrix \(X\).

The effective rank of \(A\) is determ ined by treating as zero those singular values which are less than RCOND tim es the largest singularvalue.

\section*{ARGUMENTS}

M (input) The num ber of row sof the \(m\) atrix \(A . M>=0\).
N (input) The num ber of collm ns of the \(m\) atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the \(m\) atrices \(B\) and \(X . N R H S>=0\).

A (input/output)
On entry, the \(M\) by -N m atrix A. On exit, the first
\(m\) in \((m, n)\) row \(s\) of A are overw ritten \(w\) ith its right
singularvectors, stored row w ise.
LD A (input)
The leading dim ension of the array A. LDA >= \(\mathrm{max}(1, \mathrm{M})\).

B (input/output)
On entry, the M -by-NRHS righthand sidem atrix B . On exit, B is overw rilten by the \(N\)-by NRHS solution \(m\) atrix \(X\). Ifm \(>=n\) and \(\mathbb{R} A N K=n\), the residual sum-of-squares for the solution in the \(i\)-th colum \(n\) is given by the sum of squares of elem ents \(\mathrm{n}+1 \mathrm{~m}\) in that colum n .
LD B (input)
The leading dim ension of the array \(B . L D B>=\) \(\max (1, m a x M, N)\) ).

SING (output)
The singularvalues of A in decreasing order. The condition number of \(A\) in the 2 -norm \(=\) \(S \mathbb{N} G(1) / S \mathbb{N} G(m\) in \((m, n))\).

RCOND (input)
RCOND is used to determ ine the effective rank of
A. Singular values \(S \mathbb{N} G\) (i) \(<=\) RCOND*SNI (1) are treated as zero. If RCOND \(<0, \mathrm{~m}\) achine precision is used instead.

RANK (output)
The effective rank of A, i.e., the num ber of singular values which are greater than RCOND*SING (1).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dim ension of the array \(W\) ORK.LDW ORK >=1, and also: LDW ORK \(>=3 \star_{m}\) in \(\left.M, N\right)+m a x(2 \star m\) in \(M, N)\), max (M,N),NRHS ) For good perform ance, LDW ORK should generally be larger.

IfLDW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
= 0: successfulexit
< 0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum enthad an illegalvalue.
>0: the algorithm forcom puting the SVD failed to converge; if \(\mathbb{N} F O=\) i, ioff-diagonalelem ents of an interm ediate bidiagonal form did not converge to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgelsx - routine is deprecated and has been replaced by routine SG ELSY

\section*{SYNOPSIS}

```

    W ORK,INFO)
    \mathbb{NTEGERM,N,NRHS,LDA,LDB, \mathbb{RANK,INFO}}\mathbf{N},\mp@code{N}
INTEGER JPIVOT (*)
REAL RCOND
REALA (LDA,*),B (LDB,*),W ORK (*)
SUBROUT\mathbb{NE SGELSX_64M,N,NRHS,A,LDA,B,LDB,JPIVOT,RCOND,}
\mathbb{RANK,WORK,\mathbb{NFO)}}\mathbf{N}=(
\mathbb{NTEGER*8M,N,NRHS,LDA,LDB,\mathbb{RANK,INFO}}\mathbf{N},\mp@code{N}
\mathbb{NTEGER*8 JPIVOT (*)}
REALRCOND
REALA (LDA,*),B (LDB,*),W ORK (*)

```

\section*{F95 INTERFACE}
```

SU BROUT $\mathbb{N} E \operatorname{GELSX}(\mathbb{M}], \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], J P I V O T, R C O N D$, $\mathbb{R} A N K,[\mathbb{W} O R K],[\mathbb{N} F O])$

```
\(\mathbb{N}\) TEGER :: M , N,NRHS,LDA,LDB, \(\mathbb{R} A N K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: JPIVOT
REAL ::RCOND
REAL,D IM ENSION (:) ::W ORK
REAL,D \(\mathbb{I M}\) ENSION \((:,:):: A, B\)
SU BROUTINE GELSX_64 (M) \(\mathbb{M} \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], \mathbb{P} I V O T\),
\(\mathbb{N}\) TEGER (8) ::M , N , NRHS,LDA, LD B, \(\mathbb{R} A N K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} I V O T\)
REAL ::RCOND
REAL,D IM ENSION (:) ::W ORK
REAL,D IM ENSION (:,:) ::A, B

\section*{C INTERFACE}
\#include <sunperfh>
void sgelsx (intm, intn, intnrhs, float *a, int lda, float
*b, int ldlb, int * j íivot, float rcond, int *irank, int*info);
void sgelsx_64 (long m, long n, long nrhs, float *a, long
lda, float *b, long ldb, long * jpivot, float rcond, long *irank, long *info);

\section*{PURPOSE}
sgelsx routine is deprecated and has been replaced by routine SGELSY .

SG ELSX com putes the minim um -norm solution to a real linear
least squares problem :
m inim ize \(\| \mathrm{A}\) * \(\mathrm{X}-\mathrm{B} \|\)
using a com plete orthogonal factorization of A. A is an Mby \(-\mathrm{N} m\) atrix \(w\) hich \(m\) ay be rank-deficient.

Several righthand side vectors \(b\) and solution vectors x can be handled in a single call; they are stored as the colum ns of the M -by-N RH S righthand side m atrix B and the N -by-NRHS solution \(m\) atrix \(X\).

The routine firstcom putes \(a Q R\) factorization \(w\) th \(c o l u m n\) pivoting:
\[
A * P=Q *[R 11 R 12]
\]
[ 0 R22]
w ith R 11 defined as the largest leading subm atrix whose estim ated condition num ber is less than \(1 \not R C O N D\). The order ofR11,RAN \(K\), is the effective rank ofA.

Then, R 22 is considered to be negligible, and R 12 is annihilated by orthogonal transform ations from the right, ariving at the com plete orthogonal factorization:
\[
A * P=Q *[T 110] * Z
\]
[ 0 0 \(]\)
Them inim um norm solution is then
\(\mathrm{X}=\mathrm{P} * \mathrm{Z}^{\prime}[\operatorname{inv}(\mathrm{T} 11) * \mathrm{Q} 1\) * B\(]\)
where Q 1 consists of the firstRA K colum ns of Q .

\section*{ARGUMENTS}

M (input) The num ber of row s of the matrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of collm ns of the m atrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns ofm atrices \(B\) and \(X\).NRH \(S>=0\). A (input/output)

On entry, the M -by -N m atrix A. On exit, A has been overw rilten by details of its complete orthogonal factorization.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, M)\).

B (input/output)
On entry, the M by -N RH S righthand side \(m\) atrix \(B\).
On exit, the N -by-N RH S solution \(m\) atrix X . Ifm >= \(n\) and \(\mathbb{R A N K}=\mathrm{n}\), the residual sum -of-squares for the solution in the \(i\)-th colum \(n\) is given by the sum of squares of elem ents \(N+1 \mathrm{M}\) in that colum \(n\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, M, N)\).

JPIVOT (input/output)
On entry, if \(\mathbb{P} \mathbb{I V O T}\) (i) ne.0, the i-th colum n of \(A\) is an initial colum \(n\), otherw ise it is a free colum \(n\). Before the \(Q R\) factorization of \(A\), all initial colum ns are perm uted to the leading positions; only the rem aining free colum ns are m oved as a result of colum \(n\) pivoting during the factorization. On exit, if JPIV OT \((i)=k\), then the \(i\)-th colum \(n\) of A *P was the \(k\)-th collm \(n\) of A.

RCOND (input)
RCOND is used to determ ine the effective rank of A, which is defined as the order of the largest leading triangular subm atrix R 11 in the \(Q R\) factorization w ith pivoting ofA , whose estim ated condition num ber \(<1\) RCOND.

IRANK (output)
The effective rank ofA, ie., the order of the subm atrix R11. This is the sam e as the order of the subm atrix T11 in the com plete orthogonal factorization of A.

W ORK (w orkspace)
\((m\) ax \((m\) in \(M, N)+3 * N, 2 * m\) in \((M, N)+N R H S))\), INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgelsy - com pute them inim um norm solution to a real linear
least squares problem

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGELSY M,N,NRHS,A,LDA,B,LDB,JPVT,RCOND,RANK,}
W ORK,LW ORK,INFO)
\mathbb{NTEGERM,N,NRHS,LDA,LDB,RANK,LW ORK, INFO}
\mathbb{NTEGER JPVT (*)}
REALRCOND
REALA (LDA,*),B (LDB,*),W ORK (*)
SUBROUT\mathbb{NE SGELSY_64M,N,NRHS,A,LDA,B,LDB,JPVT,RCOND,RANK,}
W ORK,LW ORK,INFO)
\mathbb{NTEGER*8M,N,NRHS,LDA,LDB,RANK,LW ORK, INFO}
INTEGER*8 JPVT (*)
REALRCOND
REALA (LDA,*),B (LDB,*),W ORK (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE GELSY (M ], $\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], J P V T, R C O N D$, RANK, [W ORK], [LW ORK], [ $\mathbb{N F O}$ ])
$\mathbb{N} T E G E R:: M, N, N R H S, L D A, L D B, R A N K, L W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{J V T}$
REAL ::RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::W ORK
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A,B

```

SU BROUTINE GELSY_64 (M) \(\mathbb{M}], \mathbb{N} R H S], A,[L D A], B,[L D B], \mathbb{P V T}\), RCOND, RANK, [W ORK], [LW ORK ], [ \(\mathbb{N F O}]\) )
\(\mathbb{N} T E G E R(8):: M, N, N R H S, L D A, L D B, R A N K, L W O R K, \mathbb{N F O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P V} T\)
REAL ::RCOND
REAL,DIM ENSION (:) ::W ORK
REAL,D IM ENSION (:,:) ::A, B

\section*{C INTERFACE}
\#include < sunperfh>
void sgelsy (intm, intn, intnrhs, float *a, int lda, float *b, int ldb, int *jpt, float rcond, int *rank, int*info);
void sgelsy_64 (long m, long n, long nrhs, float *a, long lda, float*b, long ldb, long * pivt, float rcond, long *rank, long *info);

\section*{PURPOSE}
sgelsy com putes the \(m\) inim um norm solution to a real linear least squares problem :
\(m\) inim ize \(\|A * X-B\|\)
using a com plete orthogonal factorization ofA. A is an M by \(-\mathrm{N} m\) atrix \(w\) hich \(m\) ay be rank-deficient.

Several righthand side vectors b and solution vectors \(x\) can be handled in a single call; they are stored as the colum ns of the M by-NRHS righthand side m atrix \(B\) and the \(N\) by-NRHS solution \(m\) atrix \(X\).

The routine firstcom putes \(a Q R\) factorization \(w\) th \(c o l u m n\) pivoting:
\(A * P=Q *[R 11 R 12]\)
[ 0 R22]
w ith R 11 defined as the largest leading subm atrix whose estim ated condition num ber is less than 1RCOND. The order ofR11, RAN \(K\), is the effective rank ofA.

Then, R 22 is considered to be negligible, and R 12 is anninilated by orthogonal transform ations from the right, amiving at the com plete orthogonal factorization:
\[
\begin{gathered}
A * P=Q *[T 110] * Z \\
{\left[\begin{array}{ll}
0 & 0
\end{array}\right]}
\end{gathered}
\]

Them inim um norm solution is then
\(\mathrm{X}=\mathrm{P} * \mathrm{Z}^{\prime}[\operatorname{inv}(\mathrm{T} 11) * \mathrm{Q} 1\) *B]
[ 0 ]
where Q 1 consists of the firstRANK collum ns of Q .

This routine is basically identical to the original xG ELSX except three differences:
o The call to the subroutine xGEQPF has been substituted by the
the call to the subroutine xG EQP3.This subroutine is a B las-3
version of the \(Q R\) factorization \(w\) ith colum \(n\) pivoting.
0 M atrix B (the righthand side) is updated w ith B las -3 . - The perm utation ofm atrix B (the right hand side) is fasterand m ore sim ple.

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of collm ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns ofm atrices B and X.NRHS \(>=0\).

A (input/output)
On entry, the M -by -N m atrix A. On exit, A has been overw rilten by details of its com plete orthogonal factorization.

LD A (input)
The leading dim ension of the array A. LDA >= max (1,M).

B (input/output)
On entry, the M \(-b y-N\) RH S righthand side m atrix B . On exit, the N -by-N RH S solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B. LD B >= max (1, M N \()\).

JPVT (input/output)
On entry, if JPV T (i) ne. 0, the i-th collm n of A is perm uted to the frontof \(P\), otherw ise colum \(n\) i is a free colum n. On exit, if JPV T (i) = k, then the \(i\)-th colum \(n\) of AP \(w\) as the \(k\)-th colum \(n\) of \(A\).

RCOND (input)
RCOND is used to determ ine the effective rank of A, which is defined as the order of the largest
leading triangularsubm atrix R 11 in the Q R factorization w ith pivoting ofA , w hose estim ated condition num ber<1/RCOND.

RANK (output)
The effective rank ofA, ie., the order of the subm atrix R11. This is the sam e as the order of the subm atrix T11 in the com plete orthogonal factorization of A.

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, \mathrm{~W}\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the aray W ORK. The unblocked strategy requires that: LW ORK \(>=\mathrm{MAX}(\mathrm{MN}+3 * \mathrm{~N}+1\), \(2 * M N+N R H S)\), where \(M N=m\) in ( \(M, N\) ). The block algorithm requires that: LWORK >= MAX ( \(\mathrm{M} N+2 * N+N B *(\mathbb{N}+1), 2 * \mathrm{MN}+\mathrm{NB} * \mathrm{NRHS}\) ), where NB is an upper bound on the blocksize retumed by HAENV for the routines SGEQP3, STZRZF, STZRQF, SORMQR, and SORM RZ .

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK aray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) If \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue.

\section*{FURTHER DETAILS}
```

B ased on contributions by
A .Petitet, C om puterS cience D ept., U niv . ofTenn ., K nox-
ville, U SA
E .Q uintana-O nti, D epto.de Inform atica, U niversidad Jaim e
I, Spain
G . Q uintana-O rti, D epto. de Inform atica, U niversidad Jaim e
I, Spain

```

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgem \(m\)-perform one of the \(m\) atrix-m atrix operations \(C:=\) alpha*op (A ) *op (B ) + beta*C

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGEMM (TRANSA,TRANSB,M,N,K,ALPHA,A,LDA,B,LDB,}
BETA,C,LDC)
CHARACTER * 1 TRANSA,TRANSB
INTEGERM,N,K,LDA,LDB,LDC
REALALPHA,BETA
REALA([LDA **),B (LD B,*),C (LDC,*)
SUBROUTINE SGEMM _64 (TRANSA,TRANSB,M,N,K,ALPHA,A,LDA,B,LDB,
BETA,C,LDC)

```
CHARACTER * 1 TRANSA, TRANSB
\(\mathbb{N} T E G E R * 8 M, N, K, L D A, L D B, L D C\)
REALALPHA,BETA
REALA (LDA ,*), B (LDB,*), C (LDC , *)

\section*{F95 INTERFACE}

SU BROUTINE GEM M ([TRANSA], [TRANSB], \(\mathbb{M}], \mathbb{N}], \mathbb{K}], A L P H A, A,[L D A]\), B, [LD B],BETA, C , [LD C ])

CHARACTER (LEN=1) ::TRANSA,TRANSB
\(\mathbb{N} T E G E R:: M, N, K, L D A, L D B, L D C\)
REAL ::ALPHA,BETA
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A, B, C
SUBROUTINE GEMM_64 ([TRANSA], [TRANSB], \(\mathbb{M}], \mathbb{N}],[K], A L P H A, A,[L D A]\), B, [LD B],BETA, C, [LDC])

CHARACTER ( \(L E N=1\) ) :: TRANSA , TRAN SB
\(\mathbb{N}\) TEGER (8) :: M , N , K , LDA , LD B , LD C
REAL ::ALPHA,BETA
REAL, D \(\mathbb{M}\) ENSION (: : : : : A, B, C

\section*{C INTERFACE}
\#include <sunperfh>
void sgem m (chartransa, chartransb, intm, int \(n\), int \(k\), float alpha, float *a, int lda, float *b, int ldb, floatbeta, float *c, int ldc);
void sgem m _64 (chartransa, chartransb, long m, long n, long k, float alpha, float *a, long lda, float *b, long ldlo, floatbeta, float * c, long ldc);

\section*{PURPOSE}
sgem \(m\) perform sone of the \(m\) atrix-m atrix operations \(C:=\) alpha*op (A )*op (B) + beta*C where op (X ) is one of
\[
o p(X)=X \quad \text { or } o p(X)=X^{\prime},
\]
alpha and beta are scalars, and A , B and C are matrices, with op (A ) an m by kmatrix, op (B) a k by \(n m a t r i x\) and \(C\) an \(m\) by \(n m\) atrix.

\section*{ARGUMENTS}

TRANSA (input)
On entry, TRANSA specifies the form of op (A ) to be used in the \(m\) atrix \(m\) ultiplication as follow \(s\) :

TRANSA \(=N^{\prime}\) or \(h^{\prime}\), op (A) \(=A\).

TRANSA = T'or \(t^{\prime}, ~ o p(A)=A '\).

TRANSA = C 'or \(C^{\prime}\), op (A) \(=\) A '.

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF95 \(\mathbb{N}\) TERFACE.

TRANSB (input)
O n entry, TRAN SB specifies the form ofop (B) to be used in the \(m\) atrix \(m\) ultiplication as follow \(s\) :

TRANSB = N 'or h', op (B) = B.
TRANSB \(=T\) 'or \(\mathrm{t}^{\prime}, \mathrm{op}(\mathrm{B})=\mathrm{B}\) '.
TRANSB = C'or \(\mathrm{t}^{\prime}, \mathrm{op}(\mathrm{B})=\mathrm{B}\) '.
U nchanged on exit.
TRANSB is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

M (input)
On entry, M specifies the num ber of rows of the matrix op (A ) and of the matrix C.M \(m\) ust be at least zero. U nchanged on exit.
N (input)
O n entry, N specifies the num ber of colum ns of the \(m\) atrix \(o p(B)\) and the num ber of colum ns of the m atrix C. . m ustbe at least zero. U nchanged on exit.

K (input)
On entry, \(K\) specifies the num berof colum ns of the \(m\) atrix \(o p\) ( \(A\) ) and the num ber of row s of the \(m\) atrix op ( B ). K must be at least zero. U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
REAL array ofD \(\mathbb{M}\) ENSION (LDA, ka), where ka isk when TRANSA = N 'or \(h\) ', and is \(m\) otherw ise. Before entry w ith TRANSA = N 'or \(h\) ', the leading \(m\) by \(k\) part of the array A mustcontain the \(m\) atrix \(A\), otherw ise the leading \(k\) by \(m\) part of the array \(A\) must contain the matrix A. U nchanged on exit.

LDA (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen TRANSA \(=N\) 'or \(h\) 'then LDA \(>=m a x(1, m)\), otherw ise LDA \(>=\max (1, k)\). U nchanged on exit.

B (input)
REAL array ofD \(\mathbb{I M}\) ENSION (LD B , kb ), where kb is n w hen TRANSB = N 'or h ', and is k otherw ise.
Before entry \(w\) ith TRANSB \(=N\) 'or \(h\) ', the lead-
ing \(k\) by \(n\) partof the anay \(B\) mustcontain the \(m\) atrix \(B\), otherw ise the leading \(n\) by \(k\) part of the array \(B\) must contain the \(m\) atrix \(B\). U nchanged on exit.

LD B (input)
On entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program . W hen TRANSB \(=N^{\prime}\) 'or h'then LD B \(>=m a x(1, k)\), otherw ise LD \(B>=m a x(1, n)\). U nchanged on exit.
BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then C need notbe set on input. U nchanged on exit.

C (input/output)
REAL array ofD \(\mathbb{M}\) ENSION (LD C, n). Before entry, the leading \(m\) by \(n\) partof the anay \(C\) ust contain them atrix \(C\), except when beta is zero, in which case \(C\) need notbe seton entry. On exit, the array \(C\) is overw ritten by the \(m\) by n matrix (alpha*op (A )*op (B ) + beta*C ).

LD C (input)
On entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program . LD C
\(>=m a x(1, m)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgem v -perform one of the m atrix-vectoroperations \(\mathrm{y}:=\) alpha*A * \(\mathrm{X}+\) beta* y ory \(:=\) alpha*A *x + beta* y

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGEMV (TRANSA,M,N,ALPHA,A,LDA,X, INCX,BETA,Y, INCY)}
CHARACTER * 1 TRANSA
\mathbb{NTEGERM,N,LDA,INCX,INCY}
REAL ALPHA,BETA
REAL A (LDA,*),X (*),Y (*)
SU BROUTINE SGEM V_64(TRANSA,M,N,ALPHA,A,LDA,X, INCX,BETA,Y,
\mathbb{NCY})
CHARACTER * 1 TRANSA
\mathbb{NTEGER*8M,N,LDA, INCX, INCY}
REALALPHA,BETA
REALA (LDA,*),X (*),Y(*)

```

\section*{F95 INTERFACE}

SUBROUTINE GEMV ([TRANSA], \(\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A], X,[\mathbb{N C X}], B E T A\), Y, [ \(\mathbb{N C Y}\) )

CHARACTER ( \(4 E N=1\) ) ::TRANSA
\(\mathbb{N} T E G E R:: M, N, L D A, \mathbb{N} C X, \mathbb{N} C Y\)
REAL ::ALPHA,BETA
REAL,D IM ENSION (:) :: X,Y
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A
SU BROUTINE GEM V_64 ([TRANSA], \(\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A], X,[\mathbb{N C X}]\), BETA, Y, [ \(\mathbb{N C Y}\) ])

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{LD} \mathrm{A}, \mathbb{N} C X, \mathbb{N C Y}\)
REAL ::ALPHA,BETA
REAL,D \(\mathbb{I M}\) ENSION (:) :: X,Y
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void sgem v (char transa, intm, intn, floatalpha, float*a, int lda, float *x, int incx, floatbeta, float *y, intincy);
void sgem v_64 (chartransa, long m, long n, float alpha, float *a, long lda, float*x, long incx, float beta, float *y, long incy);

\section*{PURPOSE}
sgem v perform sone of the \(m\) atrix-vector operations \(y:=\)
 and beta are scalars, \(x\) and \(y\) are vectors and \(A\) is an \(m\) by \(n\) \(m\) atrix.

\section*{ARGUMENTS}

TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=\mathrm{N}\) 'or h ' \(\mathrm{y}:=\) alpha*A * \(\mathrm{x}+\) beta* y .
TRANSA \(=\) ' 'or \(t^{\prime} y=a l p h a * A ~ * x+b e t a * y\).
TRANSA \(=\) C'ort' \(y:=a l p h a * A\) * \(x+\) beta* \(y\).

U nchanged on exit.
TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

M (input)
O n entry, M specifies the num ber of row s of the \(m\) atrix \(A . M>=0\). U nchanged on exit.

N (input)
O n entry, \(N\) specifies the num ber of colum ns of the \(m\) atrix A. \(N>=0\). U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha.
U nchanged on exit.

A (input)
B efore entry, the leading \(m\) by \(n\) part of the array
A must contain the \(m\) atrix of coefficients.
U nchanged on exit.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A >= \(\max (1, m)\). U nchanged on exit.

X (input)
\((1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C X))\) when TRANSA \(=\mathrm{N}\) 'or \(h^{\prime}\) and at least \((1+(m-1) * a b s(\mathbb{N} C X))\)
otherw ise. Before entry, the increm ented array \(X\) \(m\) ustcontain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X<>0\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then \(Y\) need notbe set on input. U nchanged on exit.

Y (input/output)
\((1+(m-1) \star \operatorname{abs}(\mathbb{N} C Y))\) when TRANSA \(=\mathrm{N}\) 'or
\(h^{\prime}\) and at least ( \(\left.1+(\mathrm{n}-1)^{\star} \operatorname{abs}(\mathbb{N} C Y)\right)\)
otherw ise. Before entry \(w\) ith BETA non-zero, the increm ented array \(Y\) m ust contain the vectory. On exit, \(Y\) is overw rilten by the updated vectory.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of. \(\mathbb{N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgeqlf - com pute \(a \operatorname{QL}\) factorization of a realM -by -N m atrix A

\section*{SYNOPSIS}

```

\mathbb{NTEGERM,N,LDA,LDW ORK,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REALA (LDA,*),TAU (*),W ORK (*)
SUBROUT\mathbb{NESGEQLF_64M,N,A,LDA,TAU,W ORK,LDW ORK, INFO)}
\mathbb{NTEGER*8M,N,LDA,LDW ORK,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REALA (LDA,*),TAU (*),W ORK (*)

```
F95 INTERFACE
    SU BROUTINE GEQLF ( \(\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N F O}])\)
    \(\mathbb{N} T E G E R:: M, N, L D A, L D W\) ORK, \(\mathbb{N} F O\)
    REAL,D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
    REAL,D \(\mathbb{M}\) ENSION (:,:) ::A
    SU BROUT \(\left.\left.\mathbb{N E} G E Q L F \_64(\mathbb{M}], \mathbb{N}\right], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N} F O]\right)\)
    \(\mathbb{N}\) TEGER (8) ::M , N ,LDA,LDW ORK, \(\mathbb{N} F O\)
    REAL,D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
    REAL,D IM ENSION (: :: : ::A
C INTERFACE
    \#include <sunperfh>
void sgeqlf(intm, intn, float *a, int lda, float *tau, int
*info);
void sgeqlf_64 (long m, long n, float *a, long lda, float *tau, long *info);

\section*{PURPOSE}
sgeqlf com putes a Q L factorization of a real M boy-N m atrix \(A: A=Q\) * \(L\).

\section*{ARGUMENTS}

M (input) The num ber of row s of the matrix A. M >=0.
N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the M -by N m atrix A . On exit, if \(\mathrm{m}>=\) \(n\), the low er triangle of the subanray A ( \(m-\) \(\mathrm{n}+1 \mathrm{~m}, 1 \mathrm{~m}\) ) contains the N -by -N low er triangular \(m\) atrix L; ifm <= n, the elem ents on and below the ( \(n-m\) ) -th superdiagonal contain the \(M\) boy- N low er trapezoidalm atrix \(L\); the rem aining elem ents, w ith the array TA \(U\), represent the orthogonal \(m\) atrix \(Q\) as a product ofelem entary reflectors (see Further D etails).

LD A (input)
The leading dim ension of the array A. LDA >= \(m\) ax ( \(1, M\) ).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails) .

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LD W ORK.

LDW ORK (input)
The dim ension of the array \(W\) ORK. LDW ORK >= \(m a x(1, N)\). Foroptim um perform ance LDW ORK \(>=N * N B\), w here N B is the optim alblocksize.

If LD W ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N F O}\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{I N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(k) \ldots H(2) H(1), \text { where } k=m \text { in }(m, n) .
\]

Each \(H\) (i) has the form
\[
\mathrm{H}(\mathrm{i})=\mathrm{I}-\tan * \mathrm{~V}^{*} \mathrm{~V}^{\prime}
\]
where tau is a real scalar, and \(v\) is a realvectorw ith \(v(m-k+i+1 m)=0\) and \(v(m-k+i)=1\); \(v(1 m-k+i-1)\) is stored on exitin A ( \(1 \mathrm{~m}-\mathrm{k}+\mathrm{i}-1, \mathrm{n}-\mathrm{k}+\mathrm{i}\) ), and tau in TAU (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgeqp3 - com pute \(a \operatorname{QR}\) factorization \(w\) th colum \(n\) pivoting of a m atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGEQP3M,N,A,LDA,JPVT,TAU,W ORK,LW ORK,INFO)}
INTEGERM,N,LDA,LW ORK,INFO
INTEGER JPVT (*)
REALA (LDA,*),TAU (*),W ORK (*)

```

```

INTEGER*8M,N,LDA,LW ORK,NNFO
INTEGER*8 JPVT (*)
REALA (LDA,*),TAU (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GEQP3 (M ], \(\mathbb{N}], A,[L D A], J P V T, T A U,[\mathbb{W}\) ORK ], [LW ORK ], [ \(\mathbb{N} F O\) ])
\(\mathbb{N}\) TEGER ::M,N,LDA,LW ORK, \(\mathbb{N} F O\) \(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: JPVT
REAL,D \(\mathbb{I M}\) ENSION (:) ::TAU,WORK
REAL,D \(\mathbb{I}\) ENSION (:,:) ::A
SU BROUTINE GEQP3_64 (M) \(\mathbb{M}, \mathbb{N}], A,[L D A], J P V T, T A U,[W O R K],[L W ~ O R K]\), [ \(\mathbb{N}\) FO ])
\(\mathbb{N} T E G E R(8):: M, N, L D A, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) :: JPVT

REAL,D \(\mathbb{I M} E N S I O N(:):\) TAU ,W ORK
REAL,D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void sgeqp3 (intm , intn, float *a, int lda, int * jpvt, float *tau, int *info);
void sgeqp3_64 (long m, long n, float *a, long lda, long
* jpvt, float *tau, long *info);

\section*{PURPOSE}
sgeqp3 com putes a Q R factorization \(w\) ith colum n pivoting ofa \(m\) atrix \(A: A * P=Q * R\) using Level3 BLA \(S\).

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the M -by \(-N m\) atrix A. On exit, the upper triangle of the amay contains the \(m\) in \((M, N)\)-by \(-N\) upper trapezoidalm atrix R ; the elem ents below the diagonal, together w ith the aray TA \(U\), represent the orthogonalm atrix \(Q\) as a product of \(m\) in \((M, N)\) elem entary reflectors.

LDA (input)
The leading dim ension of the array A. LDA >= max (1, M).

JPV T (input/output)
On entry, if \(J P V T(J)\) ne. 0 , the Jth colum \(n\) of \(A\) is perm uted to the frontof \(A\) *P (a leading colum \(n\) ); if JPVT \((J)=0\), the \(J\) th column of \(A\) is a free column. On exit, if \(\mathbb{P V V T}(J)=K\), then the \(J\) th colum \(n\) of A *P w as the the \(K\) th colum \(n\) of A.

TAU (output)
The scalar factors of the elem entary reflectors.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al

LW ORK.

LW ORK (input)
The din ension of the anray \(W\) ORK. LW ORK \(>=3 * N+1\). For optim al perform ance LW ORK \(>=2 * N+(N+1) * N B\), w here NB is the optim alblocksize.

IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N F O}\) (output)
\(=0\) : successfulexit.
< 0 : if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue.

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(1) H(2) \ldots H(k), \text { where } k=m \text { in }(m, n) .
\]

Each H (i) has the form
H (i) \(=I-\tan * V^{*} V^{\prime}\)
w here tau is a real/com plex scalar, and \(v\) is a real/com plex vectorw th \(v(1: i-1)=0\) and \(v(i)=1 ; v(i+1 \mathrm{~m})\) is stored on exitin A ( \(+1+1 \mathrm{~m}, \mathrm{i})\), and tau in TAU (i).

B ased on contributions by
G. Q uintana-O rti, D epto. de Inform atica, U niversidad Jaim e I, Spain
X . Sun, C om puterS Science D ept., D uke U niversity, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgeqpf-routine is deprecated and has been replaced by routine SGEQP3

\section*{SYNOPSIS}

\(\mathbb{N} T E G E R M, N, L D A, \mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{J P I V}\) ( \({ }^{( }\))
REALA (LDA,*),TAU (*),WORK (*)

SU BROUTINE SGEQPF_64 M,N,A,LDA, JPIVOT,TAU,WORK, \(\mathbb{N} F O\) )
\(\mathbb{N} T E G E R * 8 M, N, L D A, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{U} \mathbb{V} O T(\star)\)
REALA (LDA, *), TAU (*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GEQPF ( \(\mathbb{M}], \mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V O T}, T A U,[\mathbb{W}\) ORK ], [ \(\mathbb{N F O}]\) )
\(\mathbb{N} T E G E R:: M, N, L D A, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL,D IM ENSION (: : : : : A
SUBROUTINE GEQPF_64 (M ], \(\mathbb{N}], A,[L D A], \mathbb{N} \mathbb{I V O T}, T A U,[\mathbb{W} O R K],[\mathbb{N} F O])\)
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, L D A, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{J} \mathbb{V} O T\)
REAL,D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{I M}\) ENSION (: : : : : A

\section*{C INTERFACE}
\#include < sunperfh>
void sgeqpf(intm, intn, float *a, int lda, int * jpivot, float*tau, int*info);
void sgeqpf_64 (long m, long n, float *a, long lda, long
* p ivot, float *tau, long *info);

\section*{PURPOSE}
sgeqpf routine is deprecated and has been replaced by routine SGEQP3.

SG EQPF com putes a QR factorization \(w\) ith colum \(n\) pivoting of a realM -by \(-N\) m atrix \(A: A * P=Q * R\).

\section*{ARGUMENTS}

M (input) The num ber of row sof the \(m\) atrix \(A . M>=0\).
N (input) The num ber of colum ns of the m atrix \(\mathrm{A} \cdot \mathrm{N}>=0\)

A (input/output)
On entry, the M by -N m atrix A. On exit, the upper triangle of the aray contains the \(m\) in \((M, N)\)-by \(-N\) upper triangularm atrix \(R\); the elem ents below the diagonal, together \(w\) ith the array TA \(U\), represent the orthogonalm atrix \(Q\) as a product of \(m\) in \((m, n)\) elem entary reflectors.

LD A (input)
The leading dim ension of the array A. LDA >= max (1, M).

JPIVOT (input/output)
On entry, if \(\mathbb{P} \mathbb{I V O T}\) (i) ne.0, the i-th colum n of \(A\) is perm uted to the front ofA *P (a leading colum n); if \(\mathbb{P} \mathbb{I V O T}\) (i) \(=0\), the i-th column ofA is a free column. On exit, if JPIVOT (i) \(=k\), then the i-th column of A *P was the \(k\)-th colum \(n\) of A.

TAU (output)
The scalar factors of the elem entary reflectors.

W ORK (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{I N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{I N}\) FO \(=-i\), the -th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

Them atrix Q is represented as a product of elem entary reflectors
```

Q = H (1)H(2) ...H (n)

```

Each H (i) has the form \(\mathrm{H}=\mathrm{I}-\tan { }^{*} \mathrm{v}^{*} \mathrm{v}^{\prime}\)
\(w\) here tau is a realscalar, and \(v\) is a real vectorw ith \(\mathrm{v}(1: i-1)=0\) and \(\mathrm{v}(\mathrm{i})=1 ; \mathrm{v}(\mathrm{i}+1 \mathrm{~m})\) is stored on exit in A (i+1 \(m, ~ i)\).

Them atrix \(P\) is represented in jpvtas follow \(s\) : If jut( 7 ) \(=i\)
then the th colum \(n\) ofP is the ith canonicalunitvector.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgeqrf-com pute \(a \mathrm{Q}\) R factorization of a realM -by -N m atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGEQRFM,N,A,LDA,TAU,W ORK,LDW ORK,INFO)}
INTEGER M,N,LDA,LDW ORK,NNFO
REALA (LDA,*),TAU (*),W ORK (*)

```

```

\mathbb{NTEGER*8M,N,LDA,LDW ORK,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REALA (LDA,*),TAU (*),W ORK (*)

```

\section*{F95 INTERFACE}
\(\operatorname{SUBROUT\mathbb {NE}GEQRF}(\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N F O}])\)
\(\mathbb{N} T E G E R:: M, N, L D A, L D W\) ORK, \(\mathbb{N} F O\)
REAL,D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{I}\) ENSION (:,:) ::A
SUBROUTINE GEQRF_64 (M) \(\mathbb{N} \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER (8) ::M , N ,LDA,LDW ORK, \(\mathbb{N} F O\)
REAL,D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
REAL,D \(\mathbb{I M}\) ENSION (: : : : : A

\section*{C INTERFACE}
\#include < sunperfh>
void sgeqrff(intm , intn, float*a, int lda, float *tau, int
*info);
void sgeqrf_ 64 (long m , long n, float *a, long lda, float
*tau, long *info);

\section*{PURPOSE}
sgeqrf com putes a \(Q R\) factorization of a real \(M\)-by-N m atrix \(A: A=Q * R\).

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of \(c o l u m\) ns of the \(m\) atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the M -by -N m atrix A. On exit, the ele\(m\) ents on and above the diagonal of the array contain the \(m\) in \(M, N\) )-by \(-N\) uppertrapezoidalm atrix \(R\) \((R\) is upper triangular if \(m>=n\) ); the elem ents below the diagonal, w ith the amay TAU, represent the orthogonal \(m\) atrix \(Q\) as a productofm in ( \(m, n\) ) elem entary reflectors (see FurtherD etails).

LDA (input)
The leading dim ension of the array A. LD A >= max (1, M).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the amay \(W\) ORK. LDW ORK >= \(m\) ax \((1, N)\). Foroptim um perform ance LDW ORK \(>=N * N B\), where NB is the optim alblocksize.

If LD W ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim al size of
theW ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LDW ORK is issued by X ERBLA.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\(Q=H(1) H(2) \ldots H(k)\), where \(k=m\) in \((m, n)\).
Each \(H\) (i) has the form
H (i) \(=I-\tan * V^{*} V^{\prime}\)
where tau is a real scalar, and \(v\) is a realvectorw ith
\(\mathrm{v}(1: \mathrm{i}-1)=0\) and \(\mathrm{v}(\mathrm{i})=1\); \(\mathrm{v}(\mathrm{i}+1 \mathrm{~m})\) is stored on exit in A ( \(i+1 m, i)\), and tau in TAU (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sger-perform the rank 1 operation A: alpha* \(X^{*} y^{\prime}+A\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGER M ,N,ALPHA,X,INCX,Y,INCY,A,LDA)}
\mathbb{NTEGERM,N,\mathbb{NCX,}\mathbb{NCY,LDA}}\mathbf{N}=1
REAL ALPHA
REALX (*),Y (*),A (LDA,*)
SU BROUT\mathbb{NE SGER_64M ,N,ALPHA,X, INCX,Y, INCY,A,LDA)}
\mathbb{NTEGER*8 M ,N,INCX,INCY,LDA}
REAL ALPHA
REALX (*),Y (*),A (LDA,*)

```

\section*{F95 INTERFACE}

SUBROUTINE GER ( \(\mathbb{M}], \mathbb{N}], A L P H A, X,[\mathbb{N} C X], Y,[\mathbb{N} C Y], A,[L D A])\)
\(\mathbb{N} T E G E R:: M, N, \mathbb{N C X}, \mathbb{N C Y}, L D A\)
REAL ::ALPHA
REAL,D IM ENSION (:) :: X,Y
REAL,D \(\mathbb{I M}\) ENSION (: : : : : A
SUBROUTINE GER_64 (M ], \(\mathbb{N}], A \operatorname{LPHA}, X,[\mathbb{N C X}], Y,[\mathbb{N} C Y], A,[L D A])\)
\(\mathbb{N} T E G E R(8):: M, N, \mathbb{N C X}, \mathbb{N} C Y, L D A\)
REAL ::ALPHA
REAL,D IM ENSION (:) :: X,Y
REAL,D \(\mathbb{I}\) ENSION (:,:) ::A
void sger(intm, intn, floatalpha, float *x, int incx, float *y, int incy, float *a, int lda);
void sger_64 (long m, long n, float alpha, float *x, long incx, float *y, long incy, float *a, long lda);

\section*{PURPOSE}
sgerperform sthe rank 1 operation \(A:=a l p h{ }^{*} x^{*} y^{\prime}+A\), \(w\) here alpha is a scalar, \(x\) is an \(m\) elem entvector, \(y\) is an \(n\) elem entvector and \(A\) is an \(m\) by \(n m\) atrix.

\section*{ARGUMENTS}

M (input)
O n entry, \(M\) specifies the num ber of row \(s\) of the \(m\) atrix \(A . M>=0\). U nchanged on exit.

N (input)
O n entry, \(N\) specifies the num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\). U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
\((1+(m-1) \star \operatorname{abs}(\mathbb{N C X}))\). B efore entry, the increm ented aray \(X\) m ust contain the \(m\) elem ent vectorx. U nchanged on exit.
\(\mathbb{N} C X\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{I N C X}<>0\). U nchanged on exit.

Y (input)
\((1+(n-1) \star a b s(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y \mathrm{~m}\) ust contain the \(n\) elem ent vectory. U nchanged on exit.
\(\mathbb{N} C Y\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(. \mathbb{N} C Y<>0\). U nchanged on exit.

A (input/output)

Before entry, the leading \(m\) by \(n\) part of the anray
A must contain the matrix ofcoefficients. On exit, A is overw rilten by the updated \(m\) atrix.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program . LD A \(>=\) \(\max (1, m)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

sgerfs -im prove the com puted solution to a system of linear
equations and provides emorbounds and backw ard emoresti-
m}\mathrm{ ates forthe solution

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGERFS (TRANSA,N,NRHS,A,LDA,AF,LDAF, PPIVOT,B,LDB,}
X,LDX,FERR,BERR,W ORK,W ORK2,\mathbb{NFO)}

```
CHARACTER * 1 TRANSA
\(\mathbb{N}\) TEGER N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{P} \mathbb{I} O T(*), W\) ORK \(2(*)\)
REAL A (LDA, \(\left.{ }^{\star}\right)\), AF (LDAF,*), B (LDB , \(\left.{ }^{\star}\right)\), X (LDX,\(\left.^{\star}\right)\), FERR (*),
BERR (*),W ORK (*)
SU BROUTINE SGERFS_64 (TRANSA,N,NRHS,A,LDA,AF,LDAF, \(\mathbb{P} \mathbb{I V O T}, B\),
    LD \(B, X, L D X, F E R R, B E R R, W\) ORK,W ORK 2, \(\mathbb{N} F O\) )
CHARACTER * 1 TRANSA
\(\mathbb{N}\) TEGER*8 N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{I V O T}(*), \mathrm{W}\) ORK \(2(*)\)

BERR (*), WORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GERFS ([TRANSA], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F], \mathbb{P} \mathbb{I V O T}\), B, [LDB], \(\mathrm{X},[\operatorname{LDX}], F E R R, B E R R,[\mathbb{O}\) ORK], \([\mathbb{W} O R K 2],[\mathbb{N F O}])\)

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V O T}, \mathrm{W} O R K 2\)
REAL,D IM ENSION (:) ::FERR,BERR,WORK

SU BROUTINE GERFS_64 ([TRANSA], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F]\), \(\mathbb{P} \mathbb{V} O T, B,[L D B], X,[L D X], F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T, W\) ORK 2
REAL,D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK
REAL,D \(\mathbb{I}\) ENSION (:,:) ::A,AF,B,X

\section*{C INTERFACE}
\#include <sunperfh>
void sgerfs (chartransa, intn, intnrhs, float *a, int lda, float *af, int ldaf, int*ipivot, float*b, int ldb, float *x, int ldx, float * ferr, float *berr, int*info);
void sgerfs_64 (chartransa, long n, long nrhs, float *a, long lda, float *af, long ldaf, long *ijivot, float*b, long ldb, float *x, long ldx, float
* ferr, float *berr, long *info);

\section*{PURPOSE}
sgerfs im proves the com puted solution to a system of linear equations and provides errorbounds and backw ard erroresti\(m\) ates for the solution.

\section*{ARGUMENTS}

TRANSA (input)
Specifies the form of the system of equations:
\(=\mathrm{N}: A * X=B \quad\) (Notranspose)
\(=T\) ': A **T * \(\mathrm{X}=\mathrm{B} \quad\) ( T ranspose)
\(=C^{\prime}: A * * H * X=B \quad\) (C onjugate transpose \(=T\) ran spose)

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the m atrices B and X. NRH S >=0.

A (input) The original \(N\) foy N m atrix A .

\section*{LD A (input)}

The leading dim ension of the array A. LDA >= \(\max (1, N)\).

\section*{AF (input)}

The factors L and U from the factorization \(\mathrm{A}=\) \(\mathrm{P} * \mathrm{~L} * \mathrm{U}\) as com puted by SGETRF.

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, \mathbb{N})\).
\(\mathbb{P I V O T}\) (input)
The pivotindioes from SGETRF ; for \(1<=i<=N\), row \(i\) of the \(m\) atrix \(w\) as interchanged \(w\) th row \(\mathbb{P}\) IV OT (i).
\(B\) (input) The righthand side m atrix \(B\).
LD B (input)
The leading dim ension of the array \(B\). LD B >= \(\max (1, N)\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by SGETRS. On exit, the im proved solution \(m\) atrix \(X\).

\section*{LD X (input)}

The leading dim ension of the array X . LD X >= \(\max (1, \mathbb{N})\).

\section*{FERR (output)}

The estim ated forw ard errorbound for each solution vector \(X()\) (the \(j\) th column of the solution \(m\) atrix X). If XTRUE is the true solution comesponding to \(X(\mathcal{O})\), FERR ( \()\) ) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{\nu})-X\) TRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

BERR (output)
The com ponentw ise relative backw ard error of each solution vector X (j) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension \((3 * N)\)

W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgerqf-com pute an \(R Q\) factorization of a reall -by N m atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGERQFM,N,A,LDA,TAU,W ORK,LDW ORK,INFO)}
\mathbb{NTEGERM,N,LDA,LDW ORK,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REALA (LDA,*),TAU (*),W ORK (*)

```

```

\mathbb{NTEGER*8M,N,LDA,LDW ORK,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REALA (LDA,*),TAU (*),W ORK (*)

```

\section*{F95 INTERFACE}
\(\operatorname{SUBROUT\mathbb {NE}GERQF}(\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N F O}])\)
\(\mathbb{N} T E G E R:: M, N, L D A, L D W\) ORK, \(\mathbb{N} F O\)
REAL,D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{I}\) ENSION (:,:) ::A
SU BROUTINE GERQF_64 (M ], \(\mathbb{N}], A,[L D A], T A U,[\mathbb{N} O R K],[L D W O R K],[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER (8) ::M ,N,LDA,LDW ORK, \(\mathbb{N}\) FO
REAL,D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
REAL,D \(\mathbb{I M}\) ENSION (: : : : : A

\section*{C INTERFACE}
\#include < sunperfh>
void sgerqf(intm, intn, float*a, int lda, float *tau, int *info);
void sgerqf_64 (long m, long n, float *a, long lda, float
*tau, long *info);

\section*{PURPOSE}
sgerqf com putes an RQ factorization of realM -by N m atrix \(A: A=R * Q\).

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the M -by -N m atrix A. On exil, if \(m<=\) n , the upper triangle of the subarray \(A(1 \mathrm{~m}, \mathrm{n}\) \(\mathrm{m}+1 \mathrm{~m}\) ) contains the M boy M upper triangularm atrix \(R\); if \(m>=n\), the elem ents on and above the \(m\) n )-th subdiagonalcontain the M -by -N upper trapezoidal \(m\) atrix \(R\); the rem aining elem ents, \(w\) ith the anay TA \(U\), represent the orthogonal \(m\) atrix \(Q\) as a product of \(m\) in ( \(m, n\) ) elem entary reflectors (see FurtherD etails).

LD A (input)
The leading dim ension of the array A. LD A >= max (1, M).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, \mathrm{~W}\) ORK (1) retums the optim al LDW ORK.

LD W ORK (input)
The dim ension of the array W ORK. LDW ORK >= m ax \((1, M)\). Foroptim um perform ance LDW ORK \(>=M * N B\), w here NB is the optim alblocksize.

IfLD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N N}\) FO = -i, the i-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\(Q=H(1) H(2) \ldots H(k)\), where \(k=m\) in \((m, n)\).
Each \(H\) (i) has the form

H (i) \(=I-\tan * V^{*} V^{\prime}\)
where tau is a real scalar, and \(v\) is a realvectorw ith \(v(n-k+i+1 m)=0\) and \(v(n-k+i)=1 ; v(1 m-k+i-1)\) is stored on exitin \(A(m-k+i, 1 n-k+i-1)\), and tau in TA U (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgesdd - com pute the singularvahue decom position (SV D ) of a real M -by-N m atrix A, optionally com puting the leftand rightsingularvectors

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGESDD(OBZ,M,N,A,LDA,S,U,LDU,VT,LDVT,W ORK,}
LW ORK,INORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ
\mathbb{NTEGERM,N,LDA,LDU,LDVT,LW ORK,NNFO}
INTEGER IV ORK (*)
REALA (LDA,*),S (*),U (LDU ,*),VT (LDVT,*),W ORK (*)
SU BROUT\mathbb{NE SGESDD_64(0)BZ,M,N,A,LDA,S,U,LDU,VT,LDVT,W ORK,}
LW ORK,\mathbb{N ORK,\mathbb{NFO)}}\mathbf{N}=(
CHARACTER * 1 JOBZ
INTEGER*8M,N,LDA,LDU,LDVT,LW ORK,INFO
INTEGER*8 IN ORK (*)
REALA (LDA,*),S (*),U (LDU ,*),VT (LDVT,*),W ORK (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE GESDD (JOBZ, M ], $\mathbb{N}], A,[L D A], S, U,[L D U], V T,[L D V T]$, [W ORK ], [LW ORK ], [ $\mathbb{W}$ ORK ], [ $\mathbb{N F O}$ ])
CHARACTER (LEN=1)::JOBZ
$\mathbb{N} T E G E R:: M, N, L D A, L D U, L D V T, L W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{I N}$ ORK
REAL,D IM ENSION (:) ::S,W ORK
REAL,D IM ENSION (:,:) ::A,U,VT

```

SU BROUTINE GESDD_64 (OBBZ, \(\mathbb{M}], \mathbb{N}], A,[L D A], S, U,[L D U], V T,[L D V T]\), [W ORK ], [LW ORK ], [IW ORK ], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1): : JOBZ
\(\mathbb{N} T E G E R(8):: M, N, L D A, L D U, L D V T, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
REAL,D IM ENSION (:) ::S,W ORK
REAL,D IM ENSION (:,:) ::A,U,VT

\section*{C INTERFACE}
\#include < sunperfh>
void sgesdd (char jobz, intm, int \(n\), float *a, int lda, float *s, float *u, int ldu, float *vt, int ldvt, int*info);
void sgesdd_64 (char jobz, long m, long \(n\), float *a, long lda, float *s, float *u, long ldu, float *vt, long ldvt, long *info);

\section*{PURPOSE}
sgesdd com putes the singular value decom position (SVD ) of a real \(\mathrm{M}-\mathrm{by}-\mathrm{N} \mathrm{m}\) atrix A , optionally com puting the left and right singularvectors. If singular vectors are desired, it uses a divide-and-conquer algorithm .

The SVD isw ritten
\(=\mathrm{U}\) * SIG M A * transpose ( V )
where S IG M A is an M -by \(-\mathrm{N} m\) atrix which is zero except for its \(m\) in ( \(m, n\) ) diagonal elem ents, \(U\) is an \(M\) by \(-M\) orthogonal \(m\) atrix, and \(V\) is an \(N\) by \(-N\) orthogonalm atrix. The diagonal elements of SIGMA are the singular values of A ; they are real and non-negative, and are retumed in descending order. The firstm in ( \(\mathrm{m}, \mathrm{n}\) ) colum ns of \(U\) and \(V\) are the left and right singular vectors of .

N ote that the routine retums \(\mathrm{VT}=\mathrm{V} * * \mathrm{~T}\), notV .

The divide and conquer algorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on \(m\) achines w th a guard digit in add/subtract, or on those binary \(m\) achines \(w\) thout guard digits w hich subtract like the \(C\) ray X M P , C ray Y M P , C ray C -90, orC ray-2. It could conceivably fail on hexadecim al or decim al \(m\) achines \(w\) ithout guard digits, butw e know of none.

\section*{ARGUMENTS}

JOBZ (input)
Specifies options for com puting allorpart of the
\(m\) atrix \(U\) :
= A ': allM colum ns of U and all N row sof V **T are retumed in the arrays \(U\) and \(V T ;=S\) : the firstm in \((M, N)\) colum ns of \(U\) and the firstm in \((M, N)\) row S of V **T are retumed in the arrays U and VT ;
\(=\mathrm{O}^{\prime}\) : If \(\mathrm{M}>=\mathrm{N}\), the first N colum ns of U are overw ritten on the array \(A\) and all row sofV **T are retumed in the amay VT ; otherw ise, all colum \(n s\) of \(U\) are retumed in the array \(U\) and the firstM row SofV **T are overw ritten in the amay \(\mathrm{VT} ;=\mathrm{N}^{\prime}\) : no colum ns of U or row sof \(\mathrm{V} * * \mathrm{~T}\) are com puted.

M (input) The num ber of row s of the inputm atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of colum ns of the inputm atrix \(\mathrm{A} . \mathrm{N}>=\) 0.

A (input/output)
On entry, the M -by -N m atrix A . On exit, if \(\mathrm{OB} \mathrm{Z}=\)
\(O^{\prime}, A\) is overw ritten \(w\) ith the first \(N\) colum ns of
U (the leftsingularvectors, stored colum nw ise)
if \(\mathrm{M}>=\mathrm{N}\); A is overw rilten w ith the firstM row s of \(\mathrm{V} * * \mathrm{~T}\) the right singularvectors, stored row wise) otherw ise. if OBZ ne. 0 ', the contents of A are destroyed.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, M)\).

S (output)
The singularvalues ofA, sorted so that \(S\) (i) \(>=\) \(S(i+1)\) 。

U (output)
\(\mathrm{UCOL}=\mathrm{M}\) iff \(\mathrm{OBZ}=\mathrm{A}^{\prime}\) or \(\mathrm{OBZ}=\mathrm{D}^{\prime}\) 'and \(\mathrm{M}<\mathrm{N}\);
\(\mathrm{UCOL}=\mathrm{m}\) in \((\mathrm{M}, \mathrm{N})\) if \(J O B Z=S^{\prime}\). If \(\mathrm{OBZ}=\mathrm{A}^{\prime}\) or
JOBZ \(=0\) 'and \(M<N\), U contains the \(M\) boy \(M\) orthogonalm atrix U ; if \(\mathrm{OBZ}=\mathrm{S}\) ', U contains the firstm in \((M, N)\) colum ns of \(U\) (the left singular vectors, stored colum nw ise); if OB BZ \(=0\) 'and \(M\) \(>=\mathrm{N}\), or \(\mathrm{OBZ}=\mathrm{N}^{\prime}\), U is notreferenced.

LD U (input)
The leading dim ension of the aray \(U . L D U>=1\);
if \(\mathrm{OBZ}=\mathrm{S}^{\prime}\) or \(\mathrm{A}^{\prime}\) 'or \(\mathrm{OBZ}=\mathrm{O}^{\prime}\) 'and \(\mathrm{M}<\mathrm{N}, \mathrm{LDU}\) \(>=\mathrm{M}\).

VT (output)
If \(\mathrm{OBZ}=\mathrm{A}^{\prime}\) or \(\mathrm{OBBZ}=\mathrm{D}^{\prime}\) 'and \(\mathrm{M}>=\mathrm{N}, \mathrm{VT}\) contains the N boy N orthogonalm atrix \(\mathrm{V} * * \mathrm{~T}\); if \(\mathrm{OB} \mathrm{Z}=\) \(S\) ', V T contains the firstm in \((\mathrm{M}, \mathrm{N})\) row S of \(\mathrm{V} * * T\) (the right singularvectors, stored row w ise); if \(\mathrm{JOBZ}=\mathrm{O}\) 'and \(\mathrm{M}<\mathrm{N}\), orJOBZ \(=\mathrm{N}^{\prime}\) ', VT is not referenced.

LDVT (input)
The leading dim ension of the amay V T. LDV T >=1; if \(\mathrm{OOBZ}=\mathrm{A}\) 'or \(\mathrm{OBBZ}=\mathrm{O}^{\prime}\) 'andM \(>=\mathrm{N}, \mathrm{LDVT}>=\mathrm{N}\); if \(J O B Z=S^{\prime}, L D V T>=m\) in \((M)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al
LW ORK;

LW ORK (input)
The dim ension of the aray \(W\) ORK.LW ORK \(>=1\). If JOBZ \(=\mathrm{N}^{\prime}\), LWORK \(>=3 \star_{\mathrm{m}} \operatorname{in}(\mathrm{M}, N) \quad+\) \(\max (m \operatorname{ax}(\mathbb{M}, N), 6 \star m\) in \((M, N))\). If \(\mathrm{OBBZ}=\mathrm{O}^{\prime}, \mathrm{LW}\) ORK \(>=\) \(3{ }^{*} m\) in \((M, N){ }^{2} m\) in \((M, N)+\max \left(m a x(M, N), 5 \star_{m}\right.\) in \((M, N) \star\) \(m\) in \((M, N)+4{ }^{*} m\) in \((M N)\) ). If OBZ \(=S^{\prime}\) or \(A^{\prime}\) LW ORK \(>=3 *_{m}\) in \((M, N) \star_{m}\) in \((M, N)+m \operatorname{ax}\left(m \operatorname{ax}(M, N), 4 *_{m}\right.\) in \((M, N) *\) \(m\) in \((M, N)+4 * m\) in \((M, N))\). Forgood perform ance, LW ORK should generally be larger. If LW ORK < O but other input argum ents are legal, W ORK (1) retums optim alLW ORK .

IW ORK (w orkspace)
dim ension \((8 * \mathrm{M} \mathbb{N} \mathbb{M}, \mathbb{N}))\)

INFO (output)
\(=0\) : successfulexit.
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue. > 0: SBD SD C did not converge, updating process failed.

\section*{FURTHER DETAILS}

B ased on contributions by
M ing Gu and H uan Ren, C om puterScience D ívision, U niversity of

C alifomia atB erkeley, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgesv -com pute the solution to a real system of linear equations \(A * X=B\),

\section*{SYNOPSIS}

```

\mathbb{NTEGERN,NRHS,LDA,LDB,INFO}
INTEGER \mathbb{PIVOT (*)}
REALA (LDA,*),B (LDB,*)

```

```

\mathbb{N}TEGER*8N,NRHS,LDA,LDB,\mathbb{NFO}
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
REALA (LDA,*),B (LDB,*)

```

\section*{F95 INTERFACE}

SUBROUT \(\mathbb{N E E G E S V}(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I} O T, B,[L D B],[\mathbb{N} F O])\)
\(\mathbb{I N}\) TEGER ::N,NRHS,LDA,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A, B
SU BROUTINE GESV_64 ( \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LD} \mathrm{B}],[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDA,LDB, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T\)
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void sgesv (intn, intnins, float *a, intlda, int *ipivot, float*b, int lalb, int *info);
void sgesv_64 (long n, long nrhs, float*a, long lda, long *ípivot, float *b, long ldb, long *info);

\section*{PURPOSE}
sgesv com putes the solution to a real system of linear equations
\(A * X=B, w h e r e A\) is an \(N\) boy \(-N m\) atrix and \(X\) and \(B\) are N -by -N R H S m atrices.

The LU decomposition w ith partial pivoting and row interchanges is used to factorA as
\(A=P * L * U\),
\(w\) here \(P\) is a perm utation \(m\) atrix, \(L\) is unit low er triangular, and \(U\) is upper triangular. The factored form of \(A\) is then used to solve the system ofequations \(A * X=B\).

\section*{ARGUMENTS}

N (input) The num ber of linear equations, ie., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the m atrix B. NRHS \(>=0\).

A (input/output)
O n entry, the N boy -N coefficient m atrix A . On exit, the factors \(L\) and \(U\) from the factorization \(A\) \(=\mathrm{P} * \mathrm{~L} * \mathrm{U}\); the unitdiagonalelem ents of \(L\) are not stored.

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, N)\).

IPIVOT (output)
The pivot indices that define the perm utation \(m\) atrix \(P\); row \(i\) of the \(m\) atrix \(w\) as interchanged w th row \(\mathbb{P}\) IVOT (i).

B (input/output)
On entry, the N -by-N RH S m atrix of righthand side
\(m\) atrix \(B\). On exit, if \(\mathbb{N} F O=0\), the \(N\) boy-NRHS solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray \(B\). LD B \(>=\) \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO = -i, the i-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{U}(i, i)\) is exactly zero. The factorization has been com pleted, but the factor \(U\) is exactly singular, so the solution could not be com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgesvd - com pute the singularvalue decom position (SV D ) of a real \(M\) by -N m atrix \(A\), optionally com puting the left and/or right singularvectors

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGESVD(JOBU,NOBVT,M,N,A,LDA,SNNG,U,LDU,VT,LDVT,} W ORK,LDW ORK, $\mathbb{N} F O$ )

```

CHARACTER * 1 JOBU, JOBVT
\(\mathbb{N}\) TEGER M, N,LDA,LDU,LDVT,LDW ORK, \(\mathbb{N} F O\) REALA (LDA, \(), S \mathbb{N} G(*), U(\mathbb{L D} U, \star), V T(L D V T, \star), W\) ORK (*)

SU BROUTINE SGESVD_64 (JOBU, JOBVT,M,N,A,LDA,SING,U,LDU,VT, LDVT,W ORK,LDWORK, \(\mathbb{N} F O)\)

CHARACTER * 1 JOBU, JOBVT
\(\mathbb{N} T E G E R * 8 M, N, L D A, L D U, L D V T, L D W O R K, \mathbb{N} F O\)
REALA ( \(\mathbb{L D A}, \star), S \mathbb{N} G(*), \mathrm{U}(\mathrm{LD} \mathrm{U}, \star), \mathrm{VT}\left(\mathrm{LDVT},{ }^{\star}\right), \mathrm{W} O R K\left({ }^{\star}\right)\)

\section*{F95 INTERFACE}

SU BROUTINE GESVD (JOBU, JOBVT, \(\mathbb{M}], \mathbb{N}], A,[L D A], S \mathbb{N} G, U,[L D U], V T\), [LDVT], [W ORK], [LDW ORK], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1)::JOBU, JOBVT
\(\mathbb{N} T E G E R:: M, N, L D A, L D U, L D V T, L D W O R K, \mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::SING,W ORK
REAL,D \(\mathbb{M}\) ENSION (: : : : : A, U,VT

SU BROUTINE GESVD_64 (JOBU, JOBVT, M ], \(\mathbb{N}], A,[L D A], S \mathbb{N} G, U,[L D U]\), VT, [LDVT], [W ORK], [LDW ORK], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) :: JOBU, JOBVT
\(\mathbb{N} T E G E R(8):: M, N, L D A, L D U, L D V T, L D W O R K, \mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::SING,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:)::A,U,VT

\section*{C INTERFACE}
\#include <sunperfh>
void sgesvd (char jobu, char jobvt, intm , intn, float *a, int lda, float *sing, float *u, int ldu, float *vt, int ldvt, int *info);
void sgesvd_64 (char jंbu, char j̀jbvt, long m, long n, float
*a, long lda, float *sing, float *u, long ldu, float*vt, long lavt, long *info);

\section*{PURPOSE}
sgesvd com putes the singularvalue decom position (SV D ) of a real M -by -N m atrix A , optionally com puting the leftand/or right singularvectors. The SVD isw rilten
\(=U *\) SIGMA * transpose \((V)\)
where SIG M A is an M by N m atrix which is zero except for its \(m\) in ( \(m, n\) ) diagonal elem ents, \(U\) is an \(M\) by \(M\) orthogonal \(m\) atrix, and \(V\) is an \(N\) by \(N\) orthogonalm atrix. The diagonal elem ents of SIGM A are the singular values of A ; they are real and non-negative, and are retumed in descending order. The firstm in ( \(m, n\) ) colum ns of \(U\) and \(V\) are the left and right singularvectors of A.
\(N\) ote that the routine retums \(V\) **T, not \(V\).

\section*{ARGUMENTS}

\section*{JOBU (input)}

Specifies options for com puting allor part of the \(m\) atrix U :
= A ': all M colum ns of U are retumed in amay
U :
\(=S\) ': the firstm in \((m, n)\) colum ns ofU the left singular vectors) are retumed in the array \(U\); \(=\) \(O^{\circ}\) : the firstm in \((m, n)\) colum ns of \(U\) the left singular vectors) are overw ritten on the array A; \(=\mathrm{N}\) ': no colum ns of U (no left singularvectors) are com puted.

JOBVT (input)
Specifies options for com puting allorpart of the m atrix V **T:
\(=\mathrm{A}\) ': alln rowsof \(\mathrm{V} * * \mathrm{~T}\) are retumed in the amay VT;
\(=S\) : the firstm in \((m, n)\) row sofV \(* * T\) the right singular vectors) are retumed in the array VT ; \(=\) \(\mathrm{O}^{\prime}\) : the firstm in \((m, n)\) row s of \(V * * T\) the right singular vectors) are overw ritten on the array A; \(=\mathrm{N}\) ': no row sof \(\mathrm{V}^{* *} \mathrm{~T}\) (no right singular vectors) are com puted.

JO BV T and JOBU cannotboth be \(0^{\prime}\) '.

M (input) The num ber of row s of the inputm atrix \(A . M>=0\).
N (input) The num ber of colum ns of the inputm atrix \(\mathrm{A} . \mathrm{N}>=\) 0.

A (input/output)
On entry, the \(M-b y-N\) m atrix A. On exit, if \(J O B U=\) \(O^{\prime}\), A is overw rilten w ith the firstm in \((m, n)\)
colum ns of (the left singular vectors, stored colum nw ise); if JOBVT = O',A is overw rilten w th the firstm in \((m, n)\) row sof \(V * * T\) the right singularvectors, stored row w ise); if JO BU ne. 0 'and JOBVT ne. O', the contents of A are destroyed.

LDA (input)
The leading dim ension of the anay A. LDA >= \(m a x(1, M)\).

\section*{SING (output)}

The singularvalues of A, sorted so that SIN G (i) \(>=S \mathbb{N} G(i+1)\).

U (input) ( \(\mathrm{LD} \mathrm{U}, \mathrm{M}\) ) if \(\mathrm{JOBU}=\mathrm{A}\) 'or ( \((\mathrm{LD} \mathrm{U}, \mathrm{m}\) in \((\mathrm{M}, \mathbb{N})\) ) if \(\mathrm{JOBU}=\) \(S^{\prime}\). If \(\mathrm{JOBU}=A\) ', U contains the M -by -M orthogonalm atrix \(U\); if \(J O B U=S\) ', \(U\) contains the first \(m\) in ( \(m, n\) ) colum ns of \(U\) (the left singular vectors, stored colum nw ise); if \(\mathrm{JOBU}=\mathrm{N}\) 'or \(\mathrm{O}^{\prime}\) ' U is not referenced.

LD U (input)
The leading dim ension of the array \(U . L D U>=1\); if JO BU = S'or A',LDU >= M.

VT (input)
If \(\mathrm{OOBVT}=\mathrm{A}\) ', VT contains the N -by-N orthogonal m atrix \(\mathrm{V} * * \mathrm{~T}\); if \(\mathrm{JO} \mathrm{BVT}=\mathrm{S}^{\prime}, \mathrm{VT}\) contains the first \(m\) in \((m, n)\) row s of \(V * * T\) (the right singularvectors,
stored row w ise); if \(\mathrm{JOBVT}=\mathrm{N}\) 'or \(\mathrm{D}^{\prime}, \mathrm{VT}\) is not referenced.

LDVT (input)
The leading dim ension of the array V T. LD V T >=1;
if JOBVT = A',LDVT >= N ; if JOBVT = S', LDVT >= \(m\) in \(M, N\) ).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK ; if \(\mathbb{N} F O>0, W\) ORK ( \(2 \mathbb{M} \mathbb{N} M, N\) )) contains the unconverged superdiagonal elem ents of an upper bidiagonalm atrix B whose diagonal is in SIN G (not necessarily sorted). B satisfies \(\mathrm{A}=\mathrm{U} * \mathrm{~B} * \mathrm{VT}\), so it has the sam e singular values as A, and singular vectors related by U and V T.

LD W ORK (input)
The dim ension of the array W ORK. LDW ORK >= 1.
 Forgood perform ance, LDW ORK should generally be larger.

If LD W ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0: successfulexit.
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the i-th argum enthad an illegalvalue.
> 0 : if \(S B D S Q R\) did not converge, \(\mathbb{N F O}\) specifies
how \(m\) any superdiagonals of an interm ediate bidiagonal form B did not converge to zero. See the description ofW ORK above fordetails.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgesvx -use the LU factorization to com pute the solution to a realsystem of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUTINE SGESVX (FACT,TRANSA,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,}
EQUED,R,C,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,
W ORK2,\mathbb{NFO)}
CHARACTER * 1FACT,TRANSA,EQUED
\mathbb{NTEGER N,NRHS,LDA,LDAF,LDB,LDX,}\mathbb{N}FO
\mathbb{NTEGER \mathbb{PIVOT (*),W ORK2(*)}}\mathbf{(})
REALRCOND
REALA (LDA,*),AF (LDAF,*),R (*),C (*), B (LDB,*), X (LDX,*),
FERR (*),BERR (*),W ORK (*)

```

```

    EQUED,R,C,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,
    WORK2, \mathbb{NFO)}
    CHARACTER * 1FACT,TRANSA,EQUED
INTEGER*8N,NRHS,LDA,LDAF,LDB,LDX,}\mathbb{N}F
\mathbb{NTEGER*8 \mathbb{PIVOT (*),W ORK2 (*)}}\mathbf{(})
REALRCOND
REALA (LDA,*),AF (LDAF,*),R (*),C (*), B (LDB,*), X (LDX,*),
FERR (*),BERR (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GESVX (FACT, [TRANSA], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F]\), IPIVOT,EQUED,R,C,B,[LDB],X, [LDX],RCOND,FERR, BERR, [W ORK], [W ORK2], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::FACT,TRANSA, EQUED
\(\mathbb{N}\) TEGER :: N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V O T}, \mathrm{W} O R K 2\)
REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::R,C,FERR,BERR,W ORK
REAL,D IM ENSION (: :: : ::A,AF,B,X
SU BROUTINE GESVX_64 (FACT, [TRANSA], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F]\), IPIVOT,EQUED,R,C,B,[LDB],X,[LDX],RCOND,FERR, BERR, [W ORK], [W ORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::FACT,TRANSA,EQUED
\(\mathbb{N}\) TEGER (8) :: N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}, \mathrm{W}\) ORK2
REAL ::RCOND
REAL,D IM ENSION (:) ::R,C,FERR,BERR,W ORK
REAL,D \(\mathbb{M}\) ENSION (: : : : : A, AF, B, X

\section*{C INTERFACE}
\#include <sunperfh>
void sgesvx (char fact, char transa, intn, int nrhs, float
*a, int lda, float *af, int ldaf, int *ipivot, char equed, float *r, float * c , float *b, int ladb, float *x, int ldx, float *roond, float *ferr, float *berr, int *info);
void sgesvx_64 (char fact, chartransa, long n, long nihs, float *a, long lda, float *af, long ldaf, long *ipivot, charequed, float *r, float *c, float *b, long ldb, float *x, long ldx, float * roond, float * ferr, float *berr, long *info);

\section*{PURPOSE}
sgesvx uses the LU factorization to com pute the solution to a realsystem of linear equations
\(A * X=B\), where \(A\) is an \(N\) by \(-N m\) atrix and \(X\) and \(B\) are N -by-N R H S m atrices.

Errorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT = E', real scaling factors are com puted to equilibrate
the system:
TRANS = \(\mathrm{N}^{\prime}: \operatorname{diag}(\mathrm{R}) * A * \operatorname{diag}(\mathrm{C}) \quad * \operatorname{inv}(\operatorname{diag}(\mathrm{C})) * \mathrm{X}=\) \(\operatorname{diag}(\mathbb{R}) * B\)

TRANS \(=T:(\operatorname{diag}(\mathbb{R}) \star A * \operatorname{diag}(C)) * * T * \operatorname{inv}(\operatorname{diag}(\mathbb{R})) * X=\) diag (C)*B

TRANS \(=C\) ': \((\operatorname{diag}(\mathbb{R}) * A * \operatorname{diag}(C)) * * H * \operatorname{inv}(\operatorname{diag}(\mathbb{R})) * X=\) diag (C)*B

W hether or not the system w illbe equilibrated depends on the
scaling of the m atrix A , but ifequilibration is used, A is
overw rilten by diag \((\mathbb{R}) \star A\) *diag \((C)\) and \(B\) by diag \((\mathbb{R}) \star B\) (if TRANS = N ) ordiag (C)*B (ifTRANS = T'or C).
2. IfFACT = N 'or E', the LU decomposition is used to factor the
\(m\) atrix A (afterequilibration ifFACT = E ) as
\[
A=P * L * U
\]
\(w\) here \(P\) is a perm utation \(m\) atrix, \(L\) is a unit low er triangular
\(m\) atrix, and \(U\) is upper triangular.
3. If som e \(U(i, i)=0\), so that \(U\) is exactly singular, then the routine
retums w ith \(\mathbb{N} F O=\) i. O therw ise, the factored form of \(A\) is used
to estim ate the condition num ber of the \(m\) atrix A. If the reciprocal of the condition num ber is less than m achine precision,
\(\mathbb{N} F O=N+1\) is retumed as a waming, but the routine stillgoes on
to solve for \(X\) and compute error bounds as described below.
4. The system ofequations is solved forX using the factored form
of A.
5. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for it.
6. If equilibration \(w\) as used, the \(m\) atrix \(X\) is prem ultiplied by
diag (C) (ifTRANS = N ) ordiag \((\mathbb{R})\) (ifTRANS \(=T^{\prime}\) or C) so
that it solves the originalsystem before equilibration.

\section*{ARGUMENTS}

FACT (input)
Specifies w hether ornotthe factored form of the \(m\) atrix \(A\) is supplied on entry, and ifnot, w hether them atrix A should be equilibrated before it is factored. = F': On entry, AF and IPIV OT contain the factored form of \(A\). IfEQUED is not \(N^{\prime}\), the \(m\) atrix A has been equilibrated \(w\) ith scaling factors given by R and \(\mathrm{C} . \mathrm{A}, \mathrm{AF}\), and \(\mathbb{P}\) IV OT are not m odified. \(=\mathrm{N}\) : Them atrix A w illbe copied to A F and factored.
\(=\mathrm{E}\) : The matrix A will be equilibrated if necessary, then copied to AF and factored.

TRANSA (input)
Specifies the form of the system of equations:
\(=N^{\prime}: A * X=B \quad\) N o transpose)
\(=T\) ': \(A * * T * X=B \quad\) ( ranspose)
\(=C: A * * H * X=B \quad\) (Transpose)

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE. N (input) The num ber of linearequations, ie., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atriges B and X. NRHS \(>=0\).

A (input/output)
O n entry, the N -by -N m atrix A . IfFA CT \(=\mathrm{F}^{\prime}\) and EQUED is not \(N\) ', then A m usthave been equilibrated by the scaling factors in R and/orC. A is not modified if \(\mathrm{FACT}=\mathrm{F}^{\prime}\) or \(\mathrm{N}^{\prime}\), or if \(\mathrm{FACT}=\) E'and EQU ED = N 'on exit.

On exit, ifEQ UED ne. \(N\) ', A is scaled as follow s: \(\operatorname{EQUED}=\mathrm{R}: A:=\operatorname{diag}(\mathbb{R}) * A\)
\(E Q U E D=C\) ': A \(=A * \operatorname{diag}(C)\)
\(E Q U E D=B \prime A:=\operatorname{diag}(R) * A * \operatorname{diag}(C)\).

LDA (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

AF (input/output)
If FACT \(=F^{\prime}\), then \(A F\) is an inputargum entand on entry contains the factors \(L\) and \(U\) from the factorization \(A=P * L * U\) as com puted by SGETRF. If EQUED ne. \(N^{\prime}\) ', then \(A F\) is the factored form of the equilibrated \(m\) atrix A.

IfFACT \(=N\) ', then AF is an output argum ent and on exit retums the factors \(L\) and \(U\) from the factorization \(A=P * L * U\) of the originalm atrix \(A\).

If \(F A C T=E\) ', then \(A F\) is an output argum ent and on exit retums the factors \(L\) and \(U\) from the factorization \(\mathrm{A}=\mathrm{P} * \mathrm{~L} * \mathrm{U}\) of the equilibrated m atrix A (see the description of \(A\) for the form of the equilibrated \(m\) atrix) .

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, N)\).

PIVOT (inputoroutput)
IfFACT = \(\mathrm{F}^{\prime}\), then \(\mathbb{P I V O T}\) is an input argum ent and on entry contains the pivot indioes from the factorization \(A=P * L * U\) as com puted by SGETRF ; row \(i\) of the matrix was interchanged with row \(\mathbb{P I V O T}\) (i).

IfFACT = N', then \(\mathbb{P I V O T}\) is an output argum ent and on exit contains the pivot indices from the factorization \(A=P * L * U\) of the originalm atrix \(A\).

IfFACT = E', then \(\mathbb{P I V O T}\) is an output argum ent and on exit contains the pivot indices from the factorization \(\mathrm{A}=\mathrm{P} * \mathrm{~L} * \mathrm{U}\) of the equilibrated m atrix A.

EQUED (input/output)
Specifies the form of equilibration thatw as done. \(=\mathrm{N}\) ': No equilibration (alw ays true iffACT = \(\mathrm{N})\).
\(=R\) ': Row equilibration, ie., A has been prem ultiplied by diag \((R)\). = C ': C olum n equilibration, ie., A has been postm ultiplied by diag (C ). = B': B oth row and colum n equilibration, ie., A has been replaced by diag \((\mathbb{R})\) * A * diag (C). EQUED is an inputargum entifFACT= F '; otherw ise, it is an output argum ent.

R (input/output)
The row scale factors for \(A\). IfEQUED \(=R^{\prime}\) or B', A is multiplied on the left.by diag \((\mathbb{R})\); if EQUED = N 'or C', R is notaccessed. \(R\) is an input argum ent ifFACT = \(F\) '; otherw ise, \(R\) is an output argum ent. IfFACT = F'andEQUED = R'or \(B\) ',each elem entofR m ustbe positive.

C (input/output)
The colum n scale factors for \(A\). IfEQ UED = C 'or B', A ismultiplied on the rightby diag (C ) ; if EQUED \(=N\) 'or \(R\) ', \(C\) is notaccessed. \(C\) is an input argum ent ifFACT \(=F\) '; otherw ise, \(C\) is an outputargum ent. IfFACT = F'and EQUED = C'or \(B\) ', each elem entofC \(m\) ust.be positive.

B (input/output)
On entry, the N -by-NRHS righthand side m atrix \(B\). On exit, if EQUED = \(N\) ', \(B\) is notm odified; if TRANSA \(=N^{\prime}\) and EQUED \(=R^{\prime}\) or \(B^{\prime}, B\) is overw rilten by diag \((R) * B\); if TRANSA \(=T\) 'or \(C^{\prime}\) and EQUED \(=C^{\prime}\) or \(B^{\prime}, B\) is overw ritten by diag (C) *B.

LD B (input)
The leading dim ension of the aray \(B\). LD B \(>=\) \(\max (1, N)\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=N+1\), the \(N\)-by-NRH \(S\) solution
\(m\) atrix \(X\) to the original system ofequations.
\(N\) ote that \(A\) and \(B\) are \(m\) odified on exit if EQUED
ne. N ', and the solution to the equilibrated
system is inv (diag (C))*X ifTRANSA = N 'and EQUED
\(=C\) 'or \(B\) ', orinv (diag \((R)) * X\) ifTRANSA \(=T\) 'or \(C^{\prime}\) 'and \(E Q U E D=R\) 'or \(B\) '.

LD X (input)
The leading dim ension of the aray X . LD X >= \(\max (1, N)\).

RCOND (output)
The estim ate of the reciprocal condition num berof the matrix A after equilibration (if done). If
RCOND is less than the \(m\) achine precision (in particular, ifRCOND \(=0\) ), them atrix is singular to w orking precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO \(>0\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X()\) ) the \(j\) th colum n of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{H}), \operatorname{FERR}(1)\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{\nu})-X\) TRU \(E\) ) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost
alw ays a slight overestim ate of the true error.

BERR (output)
The com ponentw ise relative backw ard error of each solution vectorX ( \(j\) ) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exact solution).

W ORK (w orkspace)
dim ension ( \(4 * \mathrm{~N}\) ) On exit, W ORK (1) contains the reciprocal pivot grow th factornom (A)/norm (U). The "m ax absolute elem ent" norm is used. If W ORK (1) is m uch less than 1 , then the stability of the LU factorization of the (equilibrated) matrix A could be poor. This also \(m\) eans that the solution X, condition estim atorRCOND, and forw ard error bound FERR could be unreliable. If factorization fails w ith \(0<\mathbb{N} F O<=N\), then \(W\) ORK (1) contains the reciprocal pivot grow th factor forthe leading \(\mathbb{N} F O\) colum nsofA.

W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{N}\) FO \(=-\)-i, the \(i\)-th argum enthad an illegalvalue
> 0 : if \(\mathbb{N} F O=i\), and \(i\) is
\(<=\mathrm{N}: \mathrm{U}(i, i)\) is exactly zero. The factorization has been completed, but the factor \(U\) is exactly singular, so the solution and error bounds could not be com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1: \mathrm{U}\) is nonsingular, butRCOND is less than machine precision, \(m\) eaning that the \(m\) atrix is singular to w orking precision. N evertheless, the solution and error bounds are com puted because there are a num berof situations w here the com puted solution can bem ore accurate than the value ofRC O ND w ould suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgetf2 -com pute an LU factorization of a general \(m\)-by-n \(m\) atrix A using partialpivoting \(w\) th row interchanges

\section*{SYNOPSIS}

```

\mathbb{NTEGERM,N,LDA,INFO}
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(*)}
REALA (LDA,*)
SUBROUTINE SGETF2_64(M,N,A,LDA,\mathbb{P}\mathbb{IV},\mathbb{N}FO)
\mathbb{N}TEGER*8M,N,LDA,\mathbb{NFO}
\mathbb{NTEGER*8 \mathbb{PIV (*)}}\mathbf{(*)}
REALA (LDA,*)

```
F95 INTERFACE
    SU BROUTINE GETF2 ( \(\mathbb{M}], \mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I},[\mathbb{N} F O])\)
    \(\mathbb{N}\) TEGER ::M,N,LDA, \(\mathbb{N} F O\)
    \(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}\)
    REAL,D \(\mathbb{M}\) ENSION (:,:) ::A
    SU BROUTINE GETF2_64 (노 ], \(\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I},[\mathbb{N} F O])\)
    \(\mathbb{N}\) TEGER (8) ::M , N ,LDA , \(\mathbb{N}\) FO
    \(\mathbb{N} T E G E R(8), D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}\)
    REAL,D \(\mathbb{I}\) ENSION (:,:) ::A
C INTERFACE
    \#include <sunperfh>
void sgetf2 (intm, intn, float *a, intlda, int*ipiv, int *info);
void sgetf2_64 (long m, long n, float *a, long lda, long *ipiv, long *info);

\section*{PURPOSE}
sgetff com putes an LU factorization of a general \(m\)-by- \(n\) \(m\) atrix A using partialpivoting \(w\) ith row interchanges.

The factorization has the form
\[
A=P * L * U
\]
\(w\) here \(P\) is a perm utation \(m\) atrix, \(L\) is low ertriangular \(w\) ith unit diagonal elem ents (low ertrapezoidalifm > n), and U is uppertriangular (uppertrapezoidalifm < n).

This is the right-looking Level2 B LA S version of the algorithm .

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, them by \(n m\) atrix to be factored. On
exit, the factors \(L\) and \(U\) from the factorization \(A\)
\(=\mathrm{P} * \mathrm{~L} * \mathrm{U}\); the unitdiagonalelem ents of L are not stored.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, M)\).
\(\mathbb{P} \mathbb{I} V\) (output)
The pivotindioes; for \(1<=i<=m\) in \(M N\) ), row i of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P} \mathbb{I V}\) (i).
\(\mathbb{I N}\) FO (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\mathrm{k}\), the \(k\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=k, U(k, k)\) is exactly zero. The factorization has been com pleted, but the factor \(U\) is
exactly singular, and division by zero will occur
if it is used to solve a system of equations.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgetrf-com pute an LU factorization of a general \(M\)-by-N \(m\) atrix A using partialpivoting \(w\) th row interchanges

\section*{SYNOPSIS}

\(\mathbb{N}\) TEGER \(\mathrm{M}, \mathrm{N}, \mathrm{LD} A, \mathbb{N} F \mathrm{O}\)
\(\mathbb{N} T E G E R \mathbb{P} \mathbb{I} O T(*)\)
REALA (LDA,*)
SU BROUTINE SGETRF_64 \(M, N, A, L D A, \mathbb{P} \mathbb{I} O T, \mathbb{N} F O)\)
\(\mathbb{N}\) TEGER*8 M , N,LDA, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{V} O T(*)\)
REALA (LDA,*)

\section*{F95 INTERFACE}

SUBROUTINE GETRF ( \(\mathbb{M}], \mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I} O T,[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER ::M,N,LDA, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T\)
REAL,D IM ENSION (:,:) ::A
SUBROUTINE GETRF_64 (M ], \(\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V O T},[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER (8) ::M , N,LDA , \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT}\)
REAL,D \(\mathbb{I}\) ENSION (: : : : ::A

\section*{C INTERFACE}
\#include < sunperfh>
void sgetrf(intm, intn, float *a, int lda, int *ipivot, int*info);
void sgetrf_64 (long m, long n, float *a, long lda, long *ịívot, long *info);

\section*{PURPOSE}
sgetrf com putes an LU factorization of a general \(M\)-by -N \(m\) atrix A using partialpivoting \(w\) ith row interchanges.

The factorization has the form
\[
A=P * L * U
\]
\(w\) here \(P\) is a perm utation \(m\) atrix, \(L\) is low er triangular \(w\) ith unit diagonal elem ents (low ertrapezoidal ifm > n), and U is upper triangular (uppertrapezoidal ifm <n).

This is the right-looking Level3 B LA S version of the algorithm .

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(A . M>=0\).
N (input) The num ber of collm ns of the \(m\) atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the \(M\)-by -N matrix to be factored. On exit, the factors \(L\) and \(U\) from the factorization \(A\)
\(=P * L * U\); the unit diagonalelem ents of \(L\) are not stored.

LD A (input)
The leading din ension of the array A. LDA >= \(\max (1, M)\).

\section*{IPIVOT (output)}

The pivotindioes; for \(1<=i<=m\) in \((M, N)\), row i of the \(m\) atrix \(w\) as interchanged \(w\) th row \(\mathbb{P} \mathbb{I V O T}\) (i).
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{U}(\mathrm{i}, \mathrm{i})\) is exactly zero. The factorization has been com pleted, but the factor U
is exactly singular, and division by zero will occur if it is used to solve a system ofequations.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgetri-com pute the inverse of a m atrix using the LU factorization com puted by SG ETRF

\section*{SYNOPSIS}

SU BROUTINE SGETRIN,A,LDA, \(\mathbb{P} \mathbb{I V O T}, \mathrm{W} O R K, L D W O R K, \mathbb{N} F O\) )
\(\mathbb{N}\) TEGER N,LDA,LDW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{P} \mathbb{V} O T(*)\)
REALA (LDA, \(\left.{ }^{*}\right), \mathrm{W} O R K(*)\)
SUBROUTINESGETRI_64 \(\mathbb{N}, A, L D A, \mathbb{P} \mathbb{I} O T, W\) ORK,LDW ORK, \(\mathbb{N} F O)\)
\(\mathbb{N} T E G E R * 8 N, L D A, L D W\) ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{V} O T\left({ }^{*}\right)\)
REALA (LDA, \(\left.{ }^{*}\right), \mathrm{W} O R K(\star)\)

\section*{F95 INTERFACE}

SUBROUTINE GETRI( \(\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V O T}, \mathbb{W}\) ORK \(]\), [LDW ORK], [ \(\mathbb{N F O}])\)
\(\mathbb{N}\) TEGER :: N,LDA,LDW ORK, \(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER,D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL,D IM ENSION (:) ::W ORK
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A
SU BROUTINE GETRI_64 (N ],A, [LDA], \(\mathbb{P} \mathbb{V} O T,[\mathbb{N} O R K],[L D W\) ORK \(],[\mathbb{N F O}])\)
\(\mathbb{N} T E G E R(8):: N, L D A, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V O T}\)
REAL,D \(\mathbb{M}\) ENSION (:) ::W ORK
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void sgetri(intn, float *a, int lda, int *ipivot, int *info);
void sgetri_ 64 (long n, float *a, long lda, long *ịìvot, long *info);

\section*{PURPOSE}
sgetricom putes the inverse of a m atrix using the LU factorization com puted by SGETRF .

Thism ethod inverts \(U\) and then com putes inv (A) by solving the system \(\operatorname{inv}(A) * L=\operatorname{inv}(U)\) for inv (A).

\section*{ARGUMENTS}

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the factors \(L\) and \(U\) from the factoriza-
tion \(A=P * L * U\) as com puted by SGETRF. On exit, if \(\mathbb{N} F O=0\), the inverse of the originalm atrix \(A\).

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, N)\).

IPIVOT (input)
The pivotindiges from \(S G E T R F\); for \(1<=i<=N\), row \(i\) of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P}\) IV OT (i).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0\), then \(W\) ORK (1) retums the optim alLD W ORK .

LDW ORK (input)
The dimension of the array \(W\) ORK. LDW ORK >= \(m\) ax \((1, N)\). Foroptim alperform ance LDW ORK \(>=N * N B\), where NB is the optim al blocksize retumed by แAENV.

If LD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK aray, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i, U(i, i)\) is exactly zero; the \(m\) atrix is singular and its inverse could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgetrs - solve a system of linearequations \(A * X=B\) orA '
* \(\mathrm{X}=\mathrm{B}\) w th a generall -by -N m atrix A using the LU factorization com puted by SG ETRF

\section*{SYNOPSIS}

```

CHARACTER * 1 TRANSA
\mathbb{NTEGER N,NRHS,LDA,LDB,}\mathbb{N}FO
INTEGER \mathbb{PIVOT (*)}
REALA (LDA,*),B (LDB,*)
SU BROUT\mathbb{NE SGETRS_64(TRANSA,N,NRHS,A,LDA,\mathbb{PIVOT,B,LDB,INFO)}}\mathbf{N}\mathrm{ (IN,}
CHARACTER * 1 TRANSA
INTEGER*8N,NRHS,LDA,LDB,INFO
\mathbb{NTEGER*8 P\mathbb{IVOT (*)}}\mathbf{(})
REALA (LDA,*),B (LDB,*)

```

\section*{F95 INTERFACE}

SU BROUTINE GETRS ([TRANSA], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I V O T}, \mathrm{B},[\operatorname{LD} B]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, \mathbb{N F O}\)
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
REAL,D \(\mathbb{M}\) ENSION (: : : : : A, B

SU BROUTINE GETRS_64 ([TRANSA], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I V O T}, B,[L D B]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDA ,LDB, \(\mathbb{N}\) FO
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL,D \(\mathbb{M}\) ENSION (: : : : : A , B

\section*{C INTERFACE}
\#include <sunperfh>
void sgetrs (chartransa, intn, intnrhs, float *a, int lda, int *ioivot, float *b, int ldl , int *info);
void sgetrs_64 (char transa, long n, long nrhs, float *a, long lda, long *ipivot, float*b, long ldb, long *info);

\section*{PURPOSE}
sgetrs solves a system of linear equations
\(A * X=B\) or \(A^{\prime *} X=B\) w ith a generalN boy \(N \mathrm{~N}\) matrix \(A\) using the LU factorization com puted by SGETRF .

\section*{ARGUMENTS}

TRANSA (input)
Specifies the form of the system of equations:
\(=N: A * X=B \quad\) N otranspose)
\(=T\) ': \(A * X=B \quad\) (Transpose)
\(=C^{\prime}: A^{*} \mathrm{X}=\mathrm{B}\) (C onjugate transpose \(=\) Transpose)

TRANSA is defaulted to \(N\) 'forF95 \(\mathbb{N}\) TERFACE.

N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS \(>=0\).

A (input) The factors \(L\) and \(U\) from the factorization \(A=\) \(\mathrm{P} * \mathrm{~L} * \mathrm{U}\) as com puted by SG ETRF .

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, N)\).

IPIVOT (input)
The pivotindiaes from \(\operatorname{SGETRF}\); for \(1<=i<=N\), row \(i\)
of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P}\) IV OT (i).

B (input/output)
On entry, the right hand side \(m\) atrix \(B\). On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B . LD B >= \(\max (1, N)\).

IN FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N N}\) FO \(=-i\), the \(i\) th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sggbak - form the rightor lefteigenvectors of a real gen-
eralized eigenvalue problem A *x = lam bda*B*x, by backw ard transform ation on the com puted eigenvectors of the balanced
pair ofm atrices outputby SG G BA L

\section*{SYNOPSIS}

```

    \mathbb{NFO)}
    CHARACTER * 1 JOB,SIDE
\mathbb{NTEGER N,}\mathbb{LO},\mathbb{H}\textrm{I},\textrm{M},LDV,\mathbb{NFO}
REAL LSCALE (*),RSCALE (*),V (LDV,*)

```

```

    LDV,\mathbb{NFO)}
    CHARACTER * 1 JOB,SDE

```

```

REAL LSCALE (*),RSCALE (*),V (LDV,*)

```

\section*{F95 INTERFACE}

SU BROUTINE GGBAK (JOB,SIDE, \(\mathbb{N}], \mathbb{L O}, \mathbb{H} I, L S C A L E, R S C A L E, ~ M], V\), [LDV], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1):: JOB,SDE
\(\mathbb{N} T E G E R:: N, \mathbb{L} O, \mathbb{H} I, M, L D V, \mathbb{N} F O\)
REAL,D IM ENSION (:) ::LSCALE,RSCALE
REAL,D IM ENSION (:,:) ::V

SU BROUTINE GGBAK_64 (JOB,SDE, N ], \(\mathbb{L} O, \mathbb{H} I, L S C A L E, R S C A L E, ~ M ~], V\),
[LDV], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1):: OB , SDE
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathbb{I} \mathrm{O}, \mathbb{H} \mathrm{I}, \mathrm{M}, \mathrm{LDV}, \mathbb{N}\) FO
REAL,D \(\mathbb{I M}\) ENSION (:) ::LSCALE,RSCALE
REAL,D \(\mathbb{M}\) ENSION (:,:) ::V

\section*{C INTERFACE}
\#include <sunperfh>
void sggbak (char jंb, charside, intn, int 17 , int ini, float *lscale, float *rscale, intm , float*v, int ldv, int*info);
void sggbak_64 (char job, char side, long n, long ilo, long
ihi, float *lscale, float *rscale, long m, float
*V, long ldv, long *info);

\section*{PURPOSE}
sggbak form sthe rightor lefteigenvectors of a real generalized eigenvalue problem \(A{ }^{*} x=\) lam bda*B *x, by backw ard transform ation on the com puted eigenvectors of the balanced pair ofm atrioes outputby SG G BA L .

\section*{ARGUMENTS}

JO B (input)
Specifies the type of backw ard transform ation
required:
\(=\mathrm{N}\) ': do nothing, retum im m ediately;
\(=P\) ': do backw ard transform ation forperm utation
only;
= \(\mathrm{S}^{\prime}\) : do backw ard transform ation for scaling
only;
\(=\mathrm{B}:\) do backw ard transform ations forboth per\(m\) utation and scaling. JOB m ustbe the sam e as the argum ent 0 B supplied to SG G BA L .

SID E (input)
\(=R\) : V contains righteigenvectors;
\(=\mathrm{L}: \mathrm{V}\) contains lefteigenvectors.

N (input) The num ber of row s of the m atrix \(\mathrm{V} . \mathrm{N}>=0\).

IIO (input)
The integers \(\mathbb{I} O\) and \(\mathbb{H}\) I determ ined by SGGBAL 1
\(<=\mathbb{H} O<=\mathbb{H} I<=N\), if \(N>0\); \(\mathbb{H} O=1\) and \(\mathbb{H} \mathrm{I}=0\), if \(\mathrm{N}=0\).

IH I (input)
See the description for \(\mathbb{I} \mathrm{O}\).
LSCALE (input)
D etails of the perm utations and/or scaling factors applied to the left side of \(A\) and \(B\), as retumed by SG GBAL .

RSCALE (input)
D etails of the perm utations and/or scaling factors applied to the right side of \(A\) and \(B\), as retumed by SG GBAL.

M (input) The num ber of colum ns of the m atrix \(\mathrm{V} . \mathrm{M}>=0\).

V (input/output)
O \(n\) entry, the \(m\) atrix of rightor lefteigenvectors to be transform ed, as retumed by STGEVC. On exit, \(V\) is overw ritten by the transform ed eigenvectors.

LD V (input)
The leading din ension of the \(m\) atrix \(V\). LD \(V>=\) \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0: successfulexit.
<0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvahue.

\section*{FURTHER DETAILS}

See R C.W ard, B alancing the generalized eigenvalue problem, SIAM J.Sci.Stat.C omp. 2 (1981),141-152.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
```

sggbal-balance a pair of general realm atrices (A ,B)

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGGBAL (JOB,N,A,LDA,B,LD B, ILO,\mathbb{H I, LSCA LE,RSCA LE,}}\mathbf{N},
W ORK,INFO)
CHARACTER * 1 JOB

```

```

REALA (LDA,*),B (LDB,*),LSCALE (*),RSCALE (*),W ORK (*)
SU BROUTINE SGGBAL_64 (JO B ,N,A ,LD A ,B,LD B , ILO,\mathbb{H I, LSCALE,}
RSCALE,WORK,INFO)
CHARACTER * 1 JOB

```

```

REALA (LDA,*),B (LDB ,*),LSCALE (*),RSCALE (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GGBAL (JOB, \(\mathbb{N}], A,[\operatorname{LDA}], B,[L D B], \mathbb{I} O, \mathbb{H} I, L S C A L E\), RSCALE, [W ORK ], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) :: JOB
\(\mathbb{N} T E G E R:: N, L D A, L D B, \mathbb{L}, \mathbb{H} I, \mathbb{N} F O\)
REAL,D \(\mathbb{I M} E N S I O N(:):: L S C A L E, R S C A L E, W\) ORK
REAL,D \(\mathbb{M}\) ENSION (: : : : ::A, B

SU BROUTINE GGBAL_64 (JOB, N ],A, [LDA ], B, [LD B ], \(\mathbb{L O}, \mathbb{H} I, L S C A L E\), RSCALE, [W ORK], [NFO])

CHARACTER (LEN=1) :: J B
\(\mathbb{N}\) TEGER (8) ::N,LDA,LDB, \(\mathbb{L}, \mathbb{H} I, \mathbb{N} F O\)
REAL,D \(\mathbb{I M}\) ENSION (:) ::LSCA LE,RSCALE,W ORK
REAL,D IM ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void sggbal(char j̀b, intn, float *a, int lda, float *b, int ldb, int*ilo, int *ihi, float *lscale, float *rscale, int *info);
void sggbal_64 (char job, long n, float*a, long lda, float *b, long ldb, long *ilo, long *ihi, float *lscale, float *rscale, long *info);

\section*{PURPOSE}
sggbalbalances a pair of general realm atrioes ( \(A, B\) ). This involves, first, perm uting A and B by sim ilarity transform ations to isolate eigenvalues in the first1 to \(\mathbb{H O} \$-\$ 1\) and last IH I+ 1 to N elem ents on the diagonal; and second, applying a diagonal sim ilarity transform ation to row \(s\) and colum ns IIO to \(\mathbb{H}\) I to \(m\) ake the row \(s\) and colum ns as close in norm as possible. B oth steps are optional.

B alancing \(m\) ay reduce the 1 -norm of the \(m\) atrices, and im prove the accuracy of the com puted eigenvalues and/oreigenvectors in the generalized eigenvalue problem \(A * x=1 a m\) bda*B \({ }^{*} x\).

\section*{ARGUMENTS}
\(J O B\) (input)
Specifies the operations to be perform ed on \(A\) and
B :
\(=\mathrm{N}\) ': none: smply set \(\mathbb{L} \mathrm{O}=1, \mathbb{H} \mathrm{I}=\mathrm{N}\), \(\operatorname{LSCALE}(\mathbb{I})=1.0\) and RSCALE (I) \(=1.0\) for \(i=\)
\(1, \ldots, N .=P\) ': perm ute only;
= S': scale only;
= \(\mathrm{B}:\) : both perm ute and scale.

N (input) The order of the m atriges A and B. \(\mathrm{N}>=0\).

A (input/output)
On entry, the inputm atrix A. On exit, A is overw rilten by the balanced \(m\) atrix. If \(J 0 B=N\) ', A is notreferenced.

LDA (input)
The leading dim ension of the array A. LD A >= \(\max (1, \mathbb{N})\).
\(B\) (input) On entry, the inputm atrix \(B\). On exit, \(B\) is overw ritten by the balanced \(m\) atrix. If \(\mathrm{JO} \mathrm{B}=\mathrm{N}\) ', \(B\) is notreferenced.

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

ㅍㅇ (output)
IO Ond \(\mathbb{H}\) Iare set to integers such thaton exit
\(A(i, j)=0\) and \(B(i, 7)=0\) if i> jand \(j=\) \(1, \ldots\), ILIO -1 or \(i=\mathbb{H}\) I \(+1, \ldots, N\). If \(J 0 B=N\) ' or \(S^{\prime}, \mathrm{HO}=1\) and \(\mathbb{H} \mathrm{I}=\mathrm{N}\).

IH I (output) See the description for IIO.

LSCALE (input)
D etails of the perm utations and scaling factors applied to the left side of \(A\) and \(B\). IfP \((\mathcal{O})\) is the index of the row interchanged with row \(j\) and D ( 7 ) is the scaling factor applied to row \(j\) then \(\operatorname{LSCALE}(\mathcal{O})=\mathrm{P}(\mathrm{O})\) for \(J=1, \ldots\), ILO \(-1=\mathrm{D}(\mathrm{y})\) for \(J=\mathbb{L} O, \ldots, \mathbb{H} I=P(j) \quad\) for \(J=\mathbb{H} I+1, \ldots, N\).
The order in which the interchanges are m ade is N to \(\mathbb{H} \mathrm{I}+1\), then 1 to \(\mathrm{IO} \mathrm{O}-1\).

\section*{RSCALE (input)}

D etails of the perm utations and scaling factors applied to the right side of \(A\) and \(B\). IfP \((\mathcal{I})\) is the index of the colum \(n\) interchanged \(w\) ith colum \(n\) \(j\) and \(D(j)\) is the scaling factorapplied to column \(j\) then LSCALE \((j)=P(\mathcal{i})\) for \(J=\) \(1, \ldots, \mathbb{H} O-1=D(\mathcal{i})\) for \(J=\mathbb{H}, \ldots, \mathbb{H} I=P(\mathcal{J})\) for \(J=\mathbb{H} \mathrm{I}+1, \ldots, \mathrm{~N}\). The order in which the interchanges are m ade is N to \(\mathrm{IH} \mathrm{I}+1\), then 1 to


W ORK (w orkspace)
dim ension ( \(6 * \mathrm{~N}\) )
\(\mathbb{N F O}\) (output)
\(=0\) : successfulexit
<0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvahue.

\section*{FURTHER DETAILS}

See R C.W A RD , B alancing the generalized eigenvalue problem, SIAM J.Sci.Stat.Comp. 2 (1981),141-152.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgges - com pute for a pair of \(N\) by -N real nonsym \(m\) etric \(m\) atrices ( \(A, B\) ),

\section*{SYNOPSIS}

SU BROUTINE SGGES (JOBVSL, JOBVSR, SORT, SELCTG,N, A,LDA, B, LDB, SD \(\mathbb{I}\), A LPHAR, ALPHAI, BETA, VSL,LDVSL,VSR,LDVSR,W ORK,LW ORK, BW ORK, \(\mathbb{N} F O\) )

CHARACTER * 1 JOBVSL, JOBVSR, SORT
\(\mathbb{N}\) TEGER N,LDA,LDB,SD \(\mathbb{I M}, L D V S L, L D V S R, L W O R K, \mathbb{N} F O\) LOG ICAL SELCTG
LO G ICAL BW ORK (*)
REAL A (LDA , *), B (LDB,*), ALPHAR (*), ALPHAI(*), BETA (*), VSL (LDVSL, *), VSR (LDVSR, \(\left.{ }^{\star}\right), \mathrm{W}\) ORK (*)
 SD \(\mathbb{I}\), ALPHAR,ALPHAI,BETA,VSL,LDVSL,VSR,LDVSR,W ORK,LW ORK, BW ORK, \(\mathbb{N} F O\) )

CHARACTER * 1 JOBVSL, JO BVSR, SORT
\(\mathbb{N} T E G E R * 8 N, L D A, L D B, S D \mathbb{M}, L D V S L, L D V S R, L W O R K, \mathbb{N} F O\)
LOG ICAL*8 SELCTG
LO G ICAL*8 BW ORK (*)

VSL (LDVSL, , ), VSR (LDVSR, , \(), W\) ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GGES (OOBVSL, \(\mathcal{O}\) BVSR, SORT, [SELCTG], \(\mathbb{N}], A,[L D A], B,[L D B]\), SD \(\mathbb{I}\), ALPHAR,A LPHA I, BETA, V SL, [LDVSL], V SR, [LDVSR], [W ORK ], [LW ORK], BW ORK ], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) :: JOBVSL , JOBV SR , SO RT
\(\mathbb{N}\) TEGER :: N,LDA,LDB,SD \(\mathbb{I M}, L D V S L, L D V S R, L W O R K, \mathbb{N F O}\)
LOG ICAL :: SELCTG
LOG ICAL,D IM ENSION (:) ::BW ORK
REAL,D IM ENSION (:) ::ALPHAR,ALPHAI,BETA,W ORK
REAL,D IM ENSION (:,:) ::A,B,VSL,VSR
SU BROUTINE G GES_64 (JOBVSL, JOBV SR ,SORT, [SELCTG], \(\mathbb{N}\) ], A, [LDA ],B, [LD B],SD \(\mathbb{I M}, A L P H A R, A L P H A I, B E T A, V S L,[L D V S L], V S R,[L D V S R]\), [ \(\mathbb{W}\) ORK ], [LW ORK], \(B W\) ORK], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1) :: JO BVSL, JO BV SR , SO RT
\(\mathbb{N}\) TEGER (8) ::N,LDA,LDB,SD \(\mathbb{I M}, L D V S L, L D V S R, L W O R K, \mathbb{N F O}\)
LOG ICAL (8) :: SELCTG
LOG ICAL (8), D IM ENSIO N (:) ::BW ORK
REAL,D IM ENSION (:) ::ALPHAR,ALPHAI,BETA,W ORK
REAL,D \(\mathbb{M}\) ENSION (: : : : : A, B, VSL,VSR

\section*{C INTERFACE}
\#include <sunperfh>
void sgges(char jobvsl, char jobvss, char sort, int(*selctg) (float,float,float), intn, float *a, int lda, float *b, int ldb, int *sdim, float *alphar, float *alphai, float *beta, float *vsl, int ldvsl, float *vsr, int ldvsr, int *info);
void sgges_64 (char jobvsl, char jobvss, char sort, long (*selctg) (float,float,float), long n, float
*a, long lda, float *b, long lolb, long *sdim, float *alphar, float *alphai, float *beta, float *vsl, long ldvsl, float *vsr, long ldvsr, long *info);

\section*{PURPOSE}
sgges com putes for a pair of N -by -N real nonsym \(m\) etric \(m\) atrices ( \(A, B\) ), the generalized eigenvalues, the generalized realSchur form ( \(\mathrm{S}, \mathrm{T}\) ), optionally, the left and/or right \(m\) atrices of Schurvectors (V SL and V SR ). This gives the generalized Schur factorization
\[
(A, B)=(N S L) * S * N S R) * * T,(N S L) * T * N S R) * * T)
\]

Optionally, it also orders the eigenvalues so that a selected chuster of eigenvalues appears in the leading diagonalblocks of the upperquasi-triangularm atrix \(S\) and the upper triangularm atrix \(T\).The leading colum ns of V SL and V SR then form an orthonorm albasis for the comesponding left and righteigenspaces (deflating subspaces).
(If only the generalized eigenvalues are needed, use the driverSG G EV instead, which is faster.)

A generalized eigenvalue for a pair ofm atrices \((A, B)\) is a scalar \(w\) or a ratio alphabeta \(=w\), such that \(A-w * B\) is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation for beta=0 orboth being zero.

A pairofm atrices \((S, T)\) is in generalized real Schur form if \(T\) is upper triangularw ith non-negative diagonaland \(S\) is block uppertriangularw ith 1 Hoy -1 and 2 -by -2 blocks. 1-by-1 blocks correspond to real generalized eigenvalues, while \(2-b y-2\) blocks ofS w illbe "standardized" by making the corresponding elem ents of \(T\) have the form :
\(\left[\begin{array}{lll}a & 0\end{array}\right]\)
[ 0 b ]
and the pair of corresponding 2 -by -2 blocks in \(S\) and \(T \mathrm{w}\) ill have a com plex conjugate pair of generalized eigenvalues.

\section*{ARGUMENTS}

JOBVSL (input)
\(=\mathrm{N}^{\prime}\) : do notcom pute the leftSchurvectors;
\(=\mathrm{V}\) ': com pute the leftSchurvectors.

JO BV SR (input)
\(=\mathrm{N}\) : do notcom pute the rightSchurvectors;
\(=\mathrm{V}^{\prime}\) : com pute the rightSchurvectors.

SORT (input)
Specifies w hether or not to order the eigenvalues
on the diagonal of the generalized Schur form . =
N ': Eigenvalues are notordered;
\(=S\) ': Eigenvalues are ordered (see SELCTG);

SELCTG (input)
SELCTG m ustbe declared EXTERNAL in the calling
subroutine. If \(\mathrm{SORT}=\mathrm{N}^{\prime}\) ', SELCTG is notrefer-
enced. IfSORT = S', SELCTG is used to select
eigenvalues to sort to the top leftof the Schur
form. A n eigenvalue (A LPHAR ( \()+A L P H A I(\mathcal{J})\) ) BETA ( \()\)
is selected ifSELCTG (A LPHAR ( ) , A LPHAI ( ) , BETA ( 7 )
is true; ie. if either one of a com plex conjugate
pair of eigenvalues is selected, then both com plex
eigenvahues are selected.

N ote that in the ill-conditioned case, a selected complex eigenvalue may no longer satisfy \(\operatorname{SELCTG}(A L P H A R(\mathcal{j}) A L P H A I(\lambda)\), BETA \((\mathcal{j})=\operatorname{TRUE}\).
after ordering. \(\mathbb{N} F O\) is to be set to \(N+2\) in this case.

N (input) The order of the m atrices A, B, V SL, and VSR. N
\[
>=0 .
\]

A (input/output)
O \(n\) entry, the firstof the pair of \(m\) atrices. On
exit, A has been overw ritten by its generalized Schur form \(S\).

LD A (input)
The leading dim ension ofA. LD \(A>=m\) ax \((1, N)\).

B (input/output)
O n entry, the second of the pair ofm atrices. On exit, B has been overw ritten by its generalized Schur form T.

LD B (input)
The leading dim ension ofB. LD B \(>=\max (1, \mathbb{N})\).
SD \(\mathbb{I M}\) (output)
If \(S O R T=N^{\prime}, S D \mathbb{I M}=0\). IfSORT \(=S^{\prime}, S D \mathbb{M}=\)
num ber of eigenvalues (aftersorting) forw hich
SELCTG is true. (Complex conjugate pairs for which SELCTG is true foreither eigenvalue count as 2.)

ALPHAR (output)
On exil, (ALPHAR ( ) + ALPHAI ( \() * i)\) BETA ( \()\), \(\dot{F} 1, \ldots, N\), will be the generalized eigenvalues.
 are the diagonals of the com plex Schur form ( \(\mathrm{S}, \mathrm{T}\) ) thatw ould result if the 2 -by-2 diagonalblocks of the real.Schur form of ( \(A, B\) ) w ere further reduced to triangular form using 2 -by-2 complex unitary transform ations. If A LPHA I \((\underset{)}{ })\) is zero, then the \(j\) th eigenvalue is real; if positive, then the \(j\) th and ( \(j+1\) )-steigenvahues are a com plex con \(j u-\) gate pair, w ith A LPH A I(j+1) negative.

Note: the quotients ALPHAR ( ) BETA ( ) \()\) and A LPHAI( ) BETA ( ) m ay easily over-or underflow, andBETA ( \(\mathcal{O}\) ) may even be zero. Thus, the user should avoid naively com puting the ratio. H ow -
ever, A LPH AR and A LPH A Iw illbe alw ays less than and usually com parable w ith norm (A) in m agnitude, and BETA alw ays less than and usually com parable w ith norm ( \(B\) ).

\section*{A LPH A I (output)}

See the description forA LPH AR.

BETA (output)
See the description forA LPH AR .
VSL (input)
If JOBVSL = V',VSL willcontain the left Schur vectors. N ot referenced if \(\mathrm{OBVSL}=\mathrm{N}^{\prime}\).

LD V SL (input)
The leading dim ension of the \(m\) atrix VSL. LDV SL \(>=1\), and if \(\mathrm{OBVSL}=\mathrm{V}^{\prime}, \mathrm{LDV}\) SL \(>=\mathrm{N}\).

VSR (input)
If \(J O B V S R=V\) ', VSR willcontain the right Schur vectors. N ot referenced if \(\mathrm{OBVSR}=\mathrm{N}^{\prime}\).

LDVSR (input)
The leading dim ension of the \(m\) atrix \(V\) SR .LD V SR \(>=\) 1 , and if \(\mathrm{OBVSR}=\mathrm{V}\) ', LD V SR \(>=\mathrm{N}\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the option al LW ORK.

LW ORK (input)
The dim ension of the anray W ORK. LW ORK \(>=8 * \mathrm{~N}+16\).

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

BW ORK (w orkspace)
dim ension \((\mathbb{N}) N\) ot referenced if \(S O R T=N^{\prime}\).
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an illegalvalue.
\(=1, \ldots, N\) : The Q Z iteration failed. ( \(\mathrm{A}, \mathrm{B}\) ) are not in Schur form, butA LPHAR ( \(\mathcal{I}\), A LPHAI ( \(\mathcal{I}\) ) , and BETA ( 1 ) should be correct for \(\mathcal{F} \mathbb{N}\) FO \(+1, \ldots, N .>\)
\(\mathrm{N}:=\mathrm{N}+1\) : other than Q Z iteration failed in

SHGEQ Z.
\(=\mathrm{N}+2\) : after reordering, roundoff changed values of some complex eigenvalues so that leading eigenvalues in the \(G\) eneralized Schur form no longer satisfy SELCTG=TRUE. This could also be caused due to scaling. \(=\mathrm{N}+3\) : reordering failed in STG SEN .

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sggesx -com pute fora pair of \(N\)-by -N real nonsym \(m\) etric \(m\) atrices \((A, B)\), the generalized eigenvalues, the realSchur form ( \(\mathrm{S}, \mathrm{T}\) ), and,

\section*{SYNOPSIS}
```

SU BROUTINE SGGESX (JOBVSL,JOBVSR,SORT,SELCTG,SENSE,N,A,LDA,B,
LDB,SD IM,ALPHAR,A LPHA I,BETA,VSL,LDVSL,VSR,LDVSR,RCONDE,
RCONDV,W ORK,LW ORK,IN ORK,LIN ORK,BW ORK,INFO)
CHARACTER * 1 JOBVSL,JOBVSR,SORT,SENSE
INTEGERN,LDA,LDB,SD IM,LDVSL,LDVSR,LW ORK,L\mathbb{IN ORK,INFO}
INTEGER IN ORK (*)
LOG ICAL SELCTG
LOG ICAL BW ORK (*)
REAL A (LDA,*), B (LDB,*), ALPHAR (*), ALPHAI(*), BETA (*),
VSL (LDVSL,*),VSR (LDVSR,*),RCONDE (*),RCONDV (*),W ORK (*)
SU BROUT\mathbb{NE SGGESX_64 (JOBVSL,NOBVSR,SORT,SELCTG ,SEN SE,N ,A,LDA,}
B,LDB,SD IM ,A LPHAR,A LPHA I,BETA,VSL,LDVSL,VSR,LDVSR,
RCONDE,RCONDV,W ORK,LW ORK,IN ORK,LIW ORK,BW ORK,\mathbb{NFO)}

```
CHARACTER * 1 JOBVSL, JOBVSR, SORT, SENSE
\(\mathbb{N} T E G E R * 8 N, L D A, L D B, S D \mathbb{M}, L D V S L, L D V S R, L W O R K, L \mathbb{I N} O R K\),
\(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER * \(8 \mathbb{I N}\) ORK (*)
LO G ICAL*8 SELCTG
LOG ICAL*8BWORK (*)

VSL (LDVSL,*),VSR (LDVSR,*),RCONDE (*),RCONDV (*),W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE G GESX (JOBVSL, OBVSR, SORT, [SELCTG ], SENSE, \(\mathbb{N}], A,[L D A]\), \(B\), [LD B ], SD \(\mathbb{I}\), A LPH AR, A LPHAI, BETA , V SL, [LDV SL], V SR, [LDV SR], RCONDE,RCONDV, [W ORK], [LW ORK], [IW ORK], [LIWORK], [BW ORK], [ \(\mathbb{N} \mathrm{FO}\) ])

CHARACTER (LEN=1) :: JOBVSL, JOBV SR , SORT, SEN SE
\(\mathbb{N} T E G E R:: N, L D A, L D B, S D \mathbb{M}, L D V S L, L D V S R, L W O R K, L \mathbb{N} O R K\),
\(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER,D \(\mathbb{I M} E N S I O N(:):: \mathbb{I W}\) ORK
LOGICAL :: SELCTG
LOG ICAL, D \(\mathbb{I M} E N S I O N(:):\) BW ORK
REAL, D \(\mathbb{I M} E N S I O N\) (:) ::ALPHAR,ALPHAI,BETA,RCONDE, RCONDV,
W ORK
REAL,D \(\mathbb{M}\) ENSION (: : : : : A , B, VSL,VSR
 \(B,[L D B], S D \mathbb{M}, A L P H A R, A L P H A I, B E T A, V S L,[L D V S L], V S R,[L D V S R]\), RCONDE,RCONDV, \(\mathbb{W}\) ORK \(],[L W O R K],[\mathbb{N} O R K],[L \mathbb{W} O R K],[B W O R K]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) :: JOBVSL, JOBVSR , SORT, SEN SE
\(\mathbb{N} T E G E R(8):: N\), LDA, LDB, SD \(\mathbb{M}\), LDVSL, LDVSR, LW ORK,
LIW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N} O R K\)
LOG ICAL (8) :: SELCTG
LOG ICAL (8), D \(\mathbb{M}\) ENSION (:) ::BW ORK
REAL, D \(\mathbb{M} E N S I O N(:):: A L P H A R, A L P H A I, B E T A, R C O N D E, R C O N D V\),
W ORK
REAL,D IM ENSION (:,:) ::A , B , V SL , V SR

\section*{C INTERFACE}
\#include <sunperfh>
void sggesx (char jojbvsl, char jobvsr, char sort, int(*selctg) (float,float,float), char sense, int n, float*a, int lda, float *b, int ldb, int *sdim, float*alphar, float*alphai, float*beta, float*vsl, int ldvsl, float *Vsr, int ldvsr, float *roonde, float *roondv, int *info);
void sggesx_64 (char jobvsl, char jobvsr, char sort, long (* selctg) (float,float,float), char sense, long n, float*a, long lda, float*b, long ldlo, long *sdim, float*alphar, float*alphai, float*beta, float*vsl, long ldvsl, float *vsr, long ldvsr, float *roonde, float *roondv, long *info);

\section*{PURPOSE}
\(m\) atrices ( \(A, B\) ), the generalized eigenvalues, the realSchur form ( \(S, T\) ), and, optionally, the leftand/or right \(m\) atrioes of Schurvectors (V SL and V SR). This gives the generalized Schur factorization
\(A, B)=((N S L) S(N S R) \star \star T,(N S L) T(V S R) * * T)\)

Optionally, it also orders the eigenvalues so that a selected cluster of eigenvalues appears in the leading diagonalblocks of the upperquasi-triangularm atrix \(S\) and the upper triangular \(m\) atrix \(T\); com putes a reciprocalcondition num ber for the average of the selected eigenvalues (RCONDE); and com putes a reciprocal condition num ber for the right and left deflating subspaces corresponding to the selected eigenvalues (RCONDV). The leading colum ns ofV SL andVSR then form an orthonorm albasis for the corresponding left and righteigenspaces (deflating subspaces).

A generalized eigenvalue for a pair ofm atrices \((A, B)\) is a scalar \(w\) or a ratio alpha/beta \(=w\), such that \(A-w * B\) is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation for beta=0 or forboth being zero.

A pairofm atrices ( \(\mathrm{S}, \mathrm{T}\) ) is in generalized real Schur form if \(T\) is upper triangularw ith non-negative diagonaland \(S\) is block uppertriangularw ith \(1-b y-1\) and \(2-b y-2\) blocks. 1-by-1 blocks correspond to real generalized eigenvalues, while \(2-b y-2\) blocks ofS w illbe "standardized" by making the corresponding elem ents of \(T\) have the form :
\(\left[\begin{array}{lll}a & 0\end{array}\right]\)
[ 0 b ]
and the pair of comesponding 2 -by -2 blocks in \(S\) and \(T\) w ill have a com plex conjugate pair of generalized eigenvalues.

\section*{ARGUMENTS}

JOBVSL (input)
\(=\mathrm{N}^{\prime}\) : do notcom pute the leftSchurvectors;
\(=\mathrm{V}^{\prime}:\) com pute the leftSchurvectors.

JOBVSR (input)
\(=\mathrm{N}\) : do notcom pute the rightSchurvectors;
\(=\mathrm{V}^{\prime}\) : com pute the rightSchurvectors.

SORT (input)
Specifies w hether or not to order the eigenvalues on the diagonal of the generalized Schur form . =

N ': Eigenvalues are notordered;
= S ': Eigenvalues are ordered (see SELCTG).

SELCTG (input)
SELCTG m ustbe declared EXTERNAL in the calling
subroutine. If \(\mathrm{SORT}=\mathrm{N}^{\prime}\) ', SELCTG is notrefer-
enced. IfSORT = S', SELCTG is used to select
eigenvalues to sort to the top leftof the Schur
form. A n eigenvalue (ALPHAR ( \(\mathfrak{j})+A\) LPHAI( \()\) ) BETA ( \()\)
is selected ifSELCTG (A LPHAR ( 7 ) A LPHAI( ) , BETA ( \(\mathfrak{j}\) )
is true; i.e. if either one of a com plex conjugate
pair ofeigenvalues is selected, then both com plex
eigenvalues are selected. N ote that a selected
complex eigenvalue may no longer satisfy
SELCTG (ALPHAR ( \()\), ALPHAI ( \()\), BETA ( \(\mathcal{j})\) ) = TRUE. after
ordering, since ordering \(m\) ay change the value of
com plex eigenvalues (especially if the eigenvalue
is ill-conditioned), in this case \(\mathbb{N ~ F O ~ i s ~ s e t ~ t o ~}\)
\(\mathrm{N}+3\).

SEN SE (input)
D eterm ines which reciprocal condition num bers are com puted. \(=\mathrm{N}^{\prime}: \mathrm{N}\) one are com puted;
\(=E^{\prime}:\) C om puted for average of selected eigen-
values only;
\(=V^{\prime}: C\) om puted forselected deflating subspaces
only;
\(=B^{\prime}:\) Com puted forboth. IfSENSE = E', \(V\) ',
or B', SO RT m ustequal S'.

N (input) The order of the m atrioes A , B , V SL, and V SR. N
\[
>=0 .
\]

A (input/output)
O \(n\) entry, the first of the pair of \(m\) atrices. O \(n\)
exit, A has been overw rilten by its generalized Schurform \(S\).

LD A (input)
The leading dim ension ofA. LD \(A>=\max (1, N)\).

B (input/output)
O \(n\) entry, the second of the pair ofm atrices. On exit, B has been overw rilten by its generalized Schurform \(T\).

LD B (input)
The leading dim ension ofB. LD B \(>=m\) ax \((1, N)\).

SD \(\mathbb{I M}\) (output)

If \(S O R T=N^{\prime}, S D \mathbb{M}=0\). IfSORT \(=S^{\prime}, S D \mathbb{M}=\) num ber of eigenvahues (aftersorting) forw hich SELCTG is tue. (Complex conjugate pairs for which SELCTG is tue foreithereigenvalue count as 2.)

ALPHAR (output)
On exit, (ALPHAR ( ) + ALPHAI ( 1 *i) BETA ( ) ,于1,...,N, w ill be the generalized eigenvalues.
 the diagonals of the com plex Schur form \((S, T)\) that w ould result if the 2 -by - 2 diagonalblocks of the real Schur form of ( \(\mathrm{A}, \mathrm{B}\) ) w ere further reduced to triangular form using 2 -by-2 complex unitary transform ations. If A LPHA I( \()\) is zero, then the \(j\) th eigenvalue is real; ifpositive, then the \(j\) th and ( \(j+1\) )-steigenvalues are a com plex conjugate pair, w ith A LPH A I( \(\mathfrak{j}+1\) ) negative.

N ote: the quotients ALPHAR ( \(\mathcal{I}\) ) BETA ( \(\mathcal{O}\) ) and A LPHAI (Э) BETA (ㄱ) may easily over-orunderflow, and BETA ( 1 ) m ay even be zero. Thus, the user should avoid naively com puting the ratio. H ow ever, A LPH AR and A LPH A Iw illbe alw ays less than and usually com parable w ith norm (A) in m agnitude, and BETA alw ays less than and usually com parable with norm ( \(B\) ).

\section*{A LPH A I (output)}

See the description forA LPH AR .

\section*{BETA (output)}

See the description forA LPHAR.

VSL (input)
If JOBVSL = V',VSL willcontain the left Schur
vectors. N ot referenced if \(\mathrm{OBVSL}=\mathrm{N}^{\prime}\).

LDVSL (input)
The leading dim ension of the \(m\) atrix VSL. LDVSL
\(>=1\), and if \(\mathrm{OBVSL}=\mathrm{V}^{\prime}, \mathrm{LDVSL}>=\mathrm{N}\) 。

VSR (input)
If \(J O B V S R=V ', V S R w\) ill contain the right Schur
vectors. N ot referenced if OBV BR \(=\mathrm{N}\) '.

LDVSR (input)
The leading dim ension of the \(m\) atrix \(V\) SR.LD V SR \(>=\) 1 , and if \(\mathrm{OBVSR}=\mathrm{V}\) ', LDVSR \(>=\mathrm{N}\).

RCONDE (output)
IfSENSE = E'or B', RCONDE (1) and RCONDE (2)
contain the reciprocalcondition num bers for the average of the selected eigenvalues. N ot referenced ifSEN SE = N 'or V'.

RCONDV (output)
If SENSE = V'or B', RCONDV (1) and RCONDV (2)
contain the reciprocalcondition num bers for the selected deflating subspaces. N ot referenced if SENSE = N'or E'.

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, \mathrm{~W}\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the array W ORK. LW ORK >= 8* \((\mathbb{N}+1)+16\). IfSEN \(S E=E\) ', V', or B', LW ORK >= \(\left.\left.\operatorname{MAX}\left(8^{*} \mathbb{N}+1\right)+16,2 * S D \mathbb{M} * \mathbb{N}-S D \mathbb{M}\right)\right)\).

IN ORK (w orkspace)
N ot referenced if SEN SE \(=\mathrm{N}\) '.
LIV ORK (input)
The dim ension of the amay \(W\) ORK. LIN ORK >=N+6.
BW ORK (w orkspace)
dim ension (N) N ot referenced ifSORT = N'.
\(\mathbb{I N} F O\) (output)
= 0 : successfinlexit
<0: if \(\mathbb{N N}\) FO \(=-\) i, the i-th argum enthad an illegalvahue.
\(=1, \ldots, N\) : The Q Z teration failed. ( \(\mathrm{A}, \mathrm{B}\) ) are not in Schurform, butA LPHAR ( \(j\) ) A LPHA \(I(\mathcal{j})\), and BETA ( \(j\) ) should be comect for \(\ddagger \mathbb{N} F O+1, \ldots, N\). >
\(\mathrm{N}:=\mathrm{N}+1\) : other than Q Z teration failed in SH G EQ Z \(=\mathrm{N}+2\) : after reordering, roundoff changed values of some complex eigenvalues so that leading eigenvalues in the \(G\) eneralized Schur form no longer satisfy SELCTG=.TRUE. This could also be caused due to scaling. \(=\mathrm{N}+3\) : reordering failed in STG SEN .

Further details \(==============\)

A \(n\) approxim ate (asym ptotic) bound on the average absolute emror of the selected eigenvalues is

EPS * norm \((A, B)) / \operatorname{RCONDE}(1)\).
A \(n\) approxim ate (asym ptotic) bound on the \(m\) axim um angular error in the com puted deflating subspaces is

EPS * norm ( \(\mathrm{A}, \mathrm{B})\) ) /RCONDV(2).

Se LAPACK U ser's Guide, section 4.11 for more inform ation.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sggev - com pute for a pair of \(N\)-by -N real nonsym \(m\) etric \(m\) atrices ( \(A, B\) )

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SGGEV (JOBVL,JOBVR,N,A,LDA,B,LD B,ALPHAR,A LPHA I,}
BETA,VL,LDVL,VR,LDVR,WORK,LW ORK,NNFO)
CHARACTER * 1 0 BVL,JOBVR
\mathbb{NTEGERN,LDA,LDB,LDVL,LDVR,LW ORK,\mathbb{NFO}}\mathbf{N},\mp@code{L}
REAL A (LDA,*), B (LDB,*), ALPHAR (*), ALPHAI(*), BETA (*),
VL (LDVL,*),VR (LDVR,*),W ORK (*)
SU BROUTINE SGGEV_64(OD BVL,JO BVR,N,A ,LDA,B,LD B,ALPHAR,ALPHA I,
BETA,VL,LDVL,VR,LDVR,W ORK,LW ORK,INFO)

```

CHARACTER * 1 JOBVL, JOBVR
\(\mathbb{N} T E G E R * 8 N, L D A, L D B, L D V L, L D V R, L W O R K, \mathbb{N} F O\)
REAL A (LDA,\(\star), \operatorname{B}(\operatorname{LDB}, \star), \operatorname{ALPHAR}(\star), \operatorname{ALPHAI}(*)\), BETA \(\left(^{*}\right)\),
VL (LDVL,*),VR (LDVR,*),W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GGEV (JOBVL, \(\mathcal{J} 0 \mathrm{BVR}, \mathbb{N}], A,[L D A], B,[L D B], A L P H A R\), A LPHAI, BETA,VL, [LDVL],VR, [LDVR], [W ORK ], [LW ORK ], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) :: JOBVL, JOBVR
\(\mathbb{N}\) TEGER ::N,LDA,LDB,LDVL,LDVR,LW ORK, \(\mathbb{N} F O\)
REAL,D \(\mathbb{I M} E N S I O N(:):: A L P H A R, A L P H A I, B E T A, W\) ORK
REAL,D IM ENSION (:,:) ::A,B,VL,VR
SU BROUTINE GGEV_64 (OOBVL, JOBVR, \(\mathbb{N}], A,[L D A], B,[L D B], A L P H A R\), A LPHAI, BETA, VL, [LDVL],VR, [LDVR], [WORK], [LW ORK], [ \(\mathbb{N F O}])\)

CHARACTER ( \(L E N=1\) ) :: JOBVL, OOBVR
\(\mathbb{N} \operatorname{TEGER}(8):: N, L D A, L D B, L D V L, L D V R, L W O R K, \mathbb{N} F O\)
REAL, D \(\mathbb{M} E N S I O N(:):: A L P H A R, A L P H A I, B E T A, W O R K\)
REAL, D \(\mathbb{M}\) ENSION (: : : : : A, B, VL, VR

\section*{C INTERFACE}
\#include <sunperfh>
void sggev (char jobvl, char j.jbvr, intn, float *a, int lda, float *b, int ldb, float *alphar, float *alphai, float *beta, float * vl, int ldvl, float *vr, int ldvr, int*info);
void sggev_64 (char jंbvl, char jobvr, long n, float *a, long
lda, float *b, long ldb, float*alphar, float
*alphai, float *beta, float *vl, long ldvl, float
*vr, long ldvr, long *info);

\section*{PURPOSE}
sggev com putes for a pair of \(N\) Hoy N real nonsymm etric \(m\) atrices ( \(A, B\) ) the generalized eigenvalues, and optionally, the leftand/or right generalized eigenvectors.

A generalized eigenvalue for pair ofm atrices \((A, B)\) is a scalar lam bda or a ratio alpha/beta = lam boda, such thatA lam boda*B is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation for beta \(=0\), and even forboth being zero.

The righteigenvectorv (i) corresponding to the eigenvalue lam bda ( \()\) ) of ( \(A, B\) ) satisfies
\[
A * v(\mathcal{J})=\operatorname{lam} \operatorname{bda}(\mathcal{J}) * B * v(\mathcal{j})
\]

The lefteigenvectoru ( \(\mathcal{I}\) ) corresponding to the eigenvalue lam bda ( \()\) ) of ( \(A, B\) ) satisfies
\[
u(j) \star * H * A=\operatorname{lam} \operatorname{bda}(j) * u(j) * * H * B .
\]
where \(u(\mathfrak{\jmath}) * * H\) is the conjugate-transpose ofu ( \()\).

\section*{ARGUMENTS}

\section*{JOBVL (input)}
\(=\mathrm{N}\) ': do not com pute the leftgeneralized eigenvectors;
= \(\mathrm{V}^{\prime}\) : com pute the left generalized eigenvectors.
JO BVR (input)
\(=\mathrm{N}^{\prime}\) : do not com pute the right generalized eigenvectors;
\(=\mathrm{V}\) ': com pute the right generalized eigenvectors.

N (input) The order of the m atrices \(\mathrm{A}, \mathrm{B}, \mathrm{VL}\), and V R. N >= 0.

A (input/output)
On entry, them atrix \(A\) in the pair \((A, B)\). On exit, A has been overw rilten.

\section*{LD A (input)}

The leading dim ension ofA. LD A \(>=\max (1, N)\).
B (input/output)
On entry, them atrix \(B\) in the pair \((A, B)\). On
exit, B has been overw rilten.
LD B (input)
The leading dim ension ofB. LD B \(>=m\) ax \((1, N)\).

\section*{ALPHAR (output)}

On exit, (ALPHAR ( ) + ALPHAI ( ) *i) BETA ( \(\mathcal{7}\), \(\dot{j} 1, \ldots, N, w i l l\) be the generalized eigenvalues. If A LPHAI \((\mathcal{J})\) is zero, then the \(j\) th eigenvalue is real; if positive, then the \(j\) th and (j+1)-st eigenvalues are a com plex conjugate pair, w ith A LPH A I(j+1 ) negative.

Note: the quotients ALPHAR ( ) BETA ( ) \()\) and A LPHAI ( ) BETA ( ) m ay easily over-or underflow, and BETA ( \()\) may even be zero. Thus, the user should avoid naively computing the ratio alpha/beta. H ow ever, A LPHAR and A LPHA I w ill be alw ays less than and usually com parable w ith norm (A) in \(m\) agnitude, and BETA alw ays less than and usually com parable w ith norm (B).

\section*{A LPHA I (output)}

See the description forA LPHAR.

\section*{BETA (output)}

See the description forA LPHAR.
VL (input)
If \(\mathrm{JOBVL}=\mathrm{V}\) ', the left eigenvectors \(\mathrm{u}(\mathcal{)}\) are
stored one after another in the colum ns of \(V L\) ，in the sam e order as theireigenvalues．If the jth eigenvalue is real，then \(u(\mathcal{I})=V L(:, \mathcal{J})\) ，the \(j\) th colum \(n\) ofV L．If the \(j\) th and（ \(j+1\) ）－th eigenvalues form a complex conjugate pair，then \(u(1)=\) \(V L(:\rceil+,i^{\star} V L(:, j+1)\) and \(u(j+1)=V L(:, j)-\) i＊V L（：ュj＋1）．Each eigenvectorw illbe scaled so the largest component have abs（real part）\(+a b s\)（＇m ag．part）\(=1\) ．N otreferenced if 30 BVL \(=\mathrm{N}^{\prime}\) ．

LDVL（input）
The leading dim ension of the \(m\) atrix \(\mathrm{V} \mathrm{L} . \mathrm{LD} V \mathrm{~L}>=1\) ， and if \(\mathrm{JOBVL}=\mathrm{V}^{\prime}, \mathrm{LDVL}>=\mathrm{N}\) 。

VR（input）
If \(\mathrm{OBVR}=\mathrm{V}\)＇，the right eigenvectors \(\mathrm{V}(\mathcal{I})\) are stored one after another in the colum ns of VR，in the sam e order as theireigenvalues．If the jth eigenvalue is real，then \(v(\mathcal{I})=\mathrm{VR}(:, 7)\) ，the \(j\) th colum \(n\) ofVR．If the \(j\) th and（ \(j+1\) ）－th eigenvalues form a complex conjugate pair，then \(v(1)=\) \(\operatorname{VR}(:, j)+i \star V R(:, j+1)\) and \(v(j+1)=\operatorname{VR}(:, \jmath)-\) i＊VR（：,\(j+1\) ）．Each eigenvectorw illbe scaled so the largest component have abs（real part）+ abs（im ag．part）\(=1\) ．N otreferenced if O BVR \(=\mathrm{N}^{\prime}\) ．

LDVR（input）
The leading dim ension of the \(m\) atrix \(V R . L D V R>=1\) ， and if \(\mathrm{JOBVR}=\mathrm{V}\)＇，LDVR \(>=\mathrm{N}\) 。

W ORK（w orkspace）
On exit，if \(\mathbb{N F O}=0, W\) ORK（1）retums the optim al
LW ORK．

\section*{LW ORK（input）}

The dim ension of the amay W ORK．LW ORK＞＝ \(\max (1,8 * N)\) ．Forgood perform ance，LW O RK m ustgen－ erally be larger．

If LW ORK \(=-1\) ，then a w orkspace query is assum ed；
the routine only calculates the optim alsize of the W ORK array，retums this value as the first entry of the W ORK array，and no errorm essage related to LW ORK is issued by XERBLA．
\(\mathbb{N}\) FO（output）
＝0：successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\) ，the \(i\) th argum enthad an ille－
galvalue.
\(=1, \ldots, N\) : The Q Z iteration failed. N o eigenvectors have been calculated, but ALPHAR ( \(\mathcal{I}\), A LPHAI( \()\), and BETA ( 1 ) should be comect for \(\dot{F} \mathrm{NFO}+1, \ldots, N .>N:=\mathrm{N}+1\) : other than Q Z iteration failed in SH G EQ Z .
\(=\mathrm{N}+2\) : error retum from STGEVC.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sggevx - com pute fora pair of \(N\) by -N real nonsym \(m\) etric \(m\) atrices ( \(A, B\) )

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGGEVX BALANC,NOBVL,JOBVR,SEN SE,N,A,LDA,B,LDB,}
A LPHAR,ALPHAI,BETA,VL,LDVL,VR,LDVR, \#O, \#HI,LSCALE,
RSCALE,ABNRM,BBNRM,RCONDE,RCONDV,W ORK,LW ORK,IN ORK,BW ORK,
\mathbb{NFO)}

```
CHARACTER * 1 BALANC, JOBVL, JOBVR,SENSE
\(\mathbb{N}\) TEGER N,LDA,LDB,LDVL,LDVR, \(\mathbb{L O}, \mathbb{H} I, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R\) IV ORK (*)
LOG ICAL BW ORK (*)
REAL ABNRM,BBNRM

VL (LDVL, \(\left.{ }^{\star}\right), \operatorname{VR}\left(\operatorname{LDVR},^{\star}\right), \operatorname{LSCALE}\left({ }^{\star}\right), \operatorname{RSCALE}\left({ }^{\star}\right), \operatorname{RCONDE}\left(^{\star}\right)\),
RCONDV \(\left(^{*}\right), \mathrm{W} O R K(*)\)

    A LPHAR, A LPHA I, BETA, VL, LDVL,VR,LDVR, \(\amalg 1, \mathbb{H} I, L S C A L E\),
    RSCALE,ABNRM,BBNRM,RCONDE,RCONDV,W ORK,LW ORK, IN ORK,BW ORK,
    \(\mathbb{N} F O\) )

CHARACTER * 1 BALANC, JOBVL, JOBVR,SENSE
\(\mathbb{I N} T E G E R * 8 N, L D A, L D B, L D V L, L D V R, \mathbb{L} O, \mathbb{H} I, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{I N}\) ORK (*)
LO G ICAL*8 BW ORK (*)
REALABNRM,BBNRM
REAL A (LDA,\(\star), \operatorname{B}(\operatorname{LDB}, \star), \operatorname{ALPHAR}(\star), \operatorname{ALPHAI}(*), \operatorname{BETA}(\star)\),
VL (LDVL, \(\left.{ }^{\star}\right), \operatorname{VR}\left(\operatorname{LDVR},^{\star}\right), \operatorname{LSCALE}\left(^{\star}\right), \operatorname{RSCALE}\left({ }^{\star}\right), \operatorname{RCONDE}\left(^{\star}\right)\),

\section*{F95 INTERFACE}

SU BROUTINE G GEVX BALANC, JOBVL, JOBVR,SENSE, \(\mathbb{N}], A,[L D A], B,[L D B]\), A LPHAR, ALPHAI, BETA, VL, [LDVL],VR, [LDVR], \(\mathbb{H} 0, \mathbb{H} I, L S C A L E\), RSCALE,ABNRM,BBNRM,RCONDE,RCONDV, [W ORK], [LW ORK], [IN ORK], [BW ORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::BALANC, JOBVL, JOBVR, SEN SE
\(\mathbb{N} T E G E R:: N, L D A, L D B, L D V L, L D V R, \mathbb{L}, \mathbb{H} I, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{I N}\) ORK
LOG ICAL,D IM ENSION (:) ::BW ORK
REAL ::ABNRM,BBNRM
REAL,D \(\mathbb{M}\) ENSION (:) ::ALPHAR,ALPHAI,BETA,LSCALE, RSCALE,
RCONDE,RCONDV,W ORK
REAL,D IM ENSION (:,:) ::A,B,VL,VR
SU BROUTINE GGEVX_64 (BALANC, JOBVL, JOBVR, SENSE, \(\mathbb{N}\) ], A, [LDA], B,
\([\mathrm{LD} B], \mathrm{A} L P H A R, A \operatorname{LPHA}, B E T A, V L,[L D V L], V R,[L D V R], \mathbb{L} O, \mathbb{H} I\), LSCALE,RSCALE,ABNRM,BBNRM,RCONDE,RCONDV, [W ORK], [LW ORK], [ \(\mathbb{I N}\) ORK], \([B W\) ORK], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::BALANC, JOBVL, JOBVR,SEN SE
\(\mathbb{N} T E G E R(8):: N, L D A, L D B, L D V L, L D V R, \mathbb{I} O, \mathbb{H} I, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
LOG ICAL (8),D IM ENSION (:) ::BW ORK
REAL ::ABNRM,BBNRM
REAL,D \(\mathbb{M}\) ENSION (:) ::ALPHAR,ALPHAI,BETA,LSCALE, RSCALE,
RCONDE,RCONDV,W ORK
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A,B,VL,VR

\section*{C INTERFACE}
\#include < sunperfh>
void sggevx (charbalanc, char j̀bvl, char jobvr, charsense, intn, float *a, int lda, float *b, int ldb, float *alphar, float*alohai, float *beta, float *vl, int ldvl, float *vr, int ldvr, int *ilo, int *ihi, float *lscale, float *rscale, float *abnm, float *bbnm, float * rconde, float *rcondv, int *info);
void sggevx_64 (charbalanc, char jobvl, char jobvr, char sense, long n, float *a, long lda, float *b, long ldb, float *alphar, float *alphai, float *beta, float *vl, long ldvl, float *vr, long ldvr, long *ilo, long *ihi, float *lscale, float *rscale, float *abnım, float*bbnrm, float*roonde, float *rcondv, long *info);

\section*{PURPOSE}
sggevx com putes for a pair of N -by -N real nonsym \(m\) etric \(m\) atrices ( \(A, B\) ) the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors.

O ptionally also, it com putes a balancing transform ation to im prove the conditioning of the eigenvalues and eigenvectors ( \(\mathbb{H} O, \mathbb{H} I, L S C A L E, R S C A L E, A B N R M\), and BBNRM ), reciprocal condition num bers for the eigenvalues (RCONDE), and reciprocalcondition num bers for the righteigenvectors (RCONDV).

A generalized eigenvalue for a pairofm atrices ( \(A, B\) ) is a scalar lam bda or a ratio alpha/beta \(=\) lam bda, such that A lam bda*B is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation for beta= 0 , and even forboth being zero. The righteigenvectorv ( \(\mathcal{j}\) ) comesponding to the eigenvalue \(\operatorname{lam}\) bda ( \()\) ) of ( \(A, B\) ) satisfies
\[
A * v(\mathcal{I})=\operatorname{lam} \operatorname{bda}(\mathcal{I}) * B * v(\mathcal{I}) .
\]

The lefteigenvectoru ( \(\mathcal{j}\) ) corresponding to the eigenvalue lam bda ( \()\) ) of ( \(A, B\) ) satisfies
\[
\mathrm{u}(\boldsymbol{j}) * \star_{\mathrm{H}} * \mathrm{~A}=\operatorname{lam} \operatorname{bda}(\mathcal{j}) * \mathrm{u}(\boldsymbol{j}) * * \mathrm{H} * \mathrm{~B} .
\]
where \(u()^{* *}{ }^{*}\) is the conjugate-transpose ofu ( ) .

\section*{ARGUMENTS}

BALANC (input)
Specifies the balance option to be perform ed. =
N ': do not diagonally scale orperm ute;
= P ': perm ute only;
= S : scale only;
= B ': both perm ute and scale. C om puted reciprocal condition num bers \(w i l l\) be forthe \(m\) atrices afterperm uting and/orbalancing. Perm uting does not change condition num bers (in exactarithm etic), but.balancing does.

JOBVL (input)
\(=\mathrm{N}^{\prime}\) : do not com pute the left generalized eigenvectors;
= V ': com pute the left generalized eigenvectors.

JO BVR (input)
\(=\mathrm{N}^{\prime}\) : do not com pute the right generalized
eigenvectors;
\(=\mathrm{V}\) : com pute the right generalized eigenvec-
tors.

SEN SE (input)
D eterm ines which reciprocal condition num bers are com puted. = N ': none are com puted;
= E ': com puted for eigenvalues only;
= V ': com puted foreigenvectors only;
= \(\mathrm{B}^{\prime}\) : com puted foreigenvalues and eigenvectors.

N (input) The order of the m atrioes A , B , V L , and VR. N >= 0.

A (input/output)
Onentry, them atrix A in the pair \((A, B)\). On
exit, \(A\) has been overw rilten. If \(\mathrm{JO} \mathrm{BVL}=\mathrm{V}\) 'or
JO BVR=V 'orboth, then A contains the first part
of the realSchur form of the "balanced" versions
of the input \(A\) and \(B\).
LD A (input)
The leading dim ension ofA. LDA \(>=\max (1, \mathbb{N})\).
B (input/output)
On entry, them atrix B in the pair \((A, B)\). On exit, B has been overw ritten. If \(\mathrm{JO} \mathrm{BVL}=\mathrm{V}\) 'or JO BVR=V 'orboth, then B contains the second part of the realSchur form of the "balanced" versions of the inputA and \(B\).

LD B (input)
The leading dim ension ofB . LD B \(>=\mathrm{max}(1, \mathbb{N})\).
ALPHAR (output)
On exil, (ALPHAR ( \()\) + ALPHAI ( \()\) *i) BETA ( \(\mathcal{(})\), \(\dot{j} 1, \ldots, N, w i l l\) be the generalized eigenvalues. IfA LPHAI( \(\mathcal{I})\) is zero, then the \(j\) th eigenvalue is real; if positive, then the \(j\) th and (j+1)-st eigenvalues are a com plex conjugate pair, w ith A LPHA I (j+1) negative.

Note: the quotients ALPHAR ( ) BETA ( ) ( and A LPHAI ( \()\) BETA ( \()\) m ay easily over-orunderflow, andBETA ( \(\mathcal{O}\) ) may even be zero. Thus, the user should avoid naively computing the ratio A LPHA BETA.How ever, ALPHAR and ALPHAI w ill be alw ays less than and usually com parable w th norm (A) in \(m\) agnitude, and BETA alw ays less than and usually com parable w ith norm (B).

See the description of A LPH AR .
BETA (output)
See the description of LP LP AR .
VL (output)
If \(J O B V L=V\) ', the left eigenvectors \(u(y)\) are stored one after another in the colum ns of V L, in the sam e order as theireigenvalues. If the \(j\) th eigenvalue is real, then \(u(j)=V L(:, ~)\), the \(j\) th colum \(n\) ofV L. If the \(j\) th and ( \(j+1\) )-th eigenvalues form a complex conjugate pair, then \(u(1)=\) VL \((:, 1)+i \star V L(:, j+1)\) and \(u(j+1)=V L(:, i)-\) i*VL (:, jł1). Each eigenvectorw illbe scaled so the largest com ponent have abs(real part) + abs(in ag. part) \(=1 . \mathrm{N}\) ot referenced if \(\mathrm{JOBVL}=\) N '.

LDVL (input)
The leading dim ension of the \(m\) atrix \(V \mathrm{~L} . \mathrm{LD} V \mathrm{~L}>=1\), and if \(\mathrm{JOBVL}=\mathrm{V}\) ', LDVL >= N .

\section*{VR (output)}

If JO BVR \(=\mathrm{V}\) ', the right eigenvectors \(\mathrm{v}(\mathrm{f})\) are stored one after another in the colum ns of \(V R\), in the sam e order as theireigenvalues. If the \(j\) th eigenvalue is real, then \(v(i)=\operatorname{VR}(:, 7)\), the \(j\) th colum \(n\) ofVR. If the \(j\) th and ( \(j+1\) )-th eigenvalues form a complex conjugate pair, then \(\mathrm{v}(\mathcal{)})=\) \(\operatorname{VR}(:, 1)+i \star V R(:, j+1)\) and \(V(j+1)=V R(:, j)-\) i*VR (:, jł1). Each eigenvectorw illbe scaled so the largest com ponent have abs(real part) + abs(im ag. part) \(=1 . \mathrm{N}\) ot referenced if JO BV R \(=\) N '.

LDVR (input)
The leading dim ension of the \(m\) atrix \(V R\).LD V \(\mathrm{P}>=1\), and if \(J 0 B V R=V\) ', LDVR \(>=N\).

ㅍO (output)
HO and \(\mathbb{H}\) Iare integervalues such that on exit
\(A(i, j)=0\) and \(B(i, j)=0\) if i> jand \(j=\)
\(1, \ldots, \mathbb{L} O-1\) ori \(=\mathbb{H} \mathrm{I}+1, \ldots, \mathrm{~N}\). IfBALANC \(=\mathrm{N}^{\prime}\) or \(S^{\prime}, \mathbb{L} O=1\) and \(\mathbb{H} I=N\).

IH I (output)
See the description of IIO .

D etails of the perm utations and scaling factors applied to the left side of A and B. IfPL ( \()\) ) is the index of the row interchanged with row \(j\) and D L ( \()\) is the scaling factor applied to row \(j\), then \(\operatorname{LSCALE}(\mathcal{j})=\mathrm{PL}(\mathcal{j})\) for \(j=1, \ldots\), ILO \(-1=\mathrm{DL}(\mathcal{j})\) for \(j=\mathbb{H}, \ldots, \mathbb{H} I=P L(j)\) for \(j=\mathbb{H} I+1, \ldots, N\). The order in \(w\) hich the interchanges are \(m\) ade is \(N\) to \(\mathbb{H} \mathrm{I}+1\), then 1 to \(\mathbb{L O} \mathrm{-}\).

\section*{RSCALE (output)}

D etails of the perm utations and scaling factors applied to the right side of \(A\) and \(B\). IfPR \((\mathcal{)})\) is the index of the colum \(n\) interchanged with column \(j\) and \(D R(j)\) is the scaling factor applied to
column \(j\) then RSCALE \((j)=\operatorname{PR}(\mathcal{i})\) for \(j=\) \(1, \ldots\), IHO-1 \(=\operatorname{DR}(j)\) for \(j=\mathbb{H O}, \ldots\), H \(I=\operatorname{PR}(\mathcal{j})\) for \(j=\mathbb{H} \mathrm{I}+1, \ldots, \mathrm{~N}\) The order in which the interchanges are \(m\) ade is \(N\) to \(\mathbb{H} \mathrm{I}+1\), then 1 to \(\mathbb{L} \mathrm{O}-1\).

\section*{ABNRM (output)}

The one-norm of the balanced \(m\) atrix A.

BBNRM (output)
The one-norm of the balanced \(m\) atrix \(B\).

RCONDE (output)
IfSENSE = E'or B', the reciprocal condition num bers of the selected eigenvalues, stored in consecutive elem ents of the array. For a com plex conjugate pair of eigenvalues tw o consecutive ele\(m\) ents of RCONDE are set to the sam e value. Thus RCONDE ( \()\), RCONDV ( 1 ), and the \(j\) th colum ns of L and VR allcorrespond to the same eigenpair but not in general the jth eigenpair, unless all eigenpairs are selected). IfSENSE = V', RCONDE is not referenced.

RCONDV (output)
If SEN SE = V 'or B ', the estim ated reciprocal
condition num bers of the selected eigenvectors, stored in consecutive elem ents of the array. For a complex eigenvector tw o consecutive elem ents of RCONDV are set to the sam evalue. If the eigenvalues cannot be reordered to com pute RCONDV ( 7 ), RCONDV \((\mathcal{j})\) is set to 0 ; this can only occur when the true value would be very sm allanyw ay. If SENSE = E',RCONDV is not referenced.

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, \mathrm{~W}\) ORK (1) retums the optim al

LW ORK.

LW ORK (input)
The dimension of the array W ORK. LW ORK >= \(\max (1,6 * \mathrm{~N})\). If SENSE = E',LW ORK \(>=12 \star \mathrm{~N}\). If SEN SE \(=\mathrm{V}\) 'or B ', LW ORK \(>=2 * \mathrm{~N} * \mathrm{~N}+12 * \mathrm{~N}+16\).

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IN ORK (w orkspace)
dim ension \((N+6)\) IfSEN \(S E=E\) ', IV ORK is not referenced.

BW ORK (w orkspace)
dim ension \((\mathbb{N})\) If SEN SE \(=N\) ', BW ORK is not referenced.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO \(=-\) i, the \(i\) th argum enthad an illegalvalue.
\(=1, \ldots, N\) : The Q Z iteration failed. N o eigenvec-
tors have been calculated, but A LPHAR ( \(\mathcal{7}\),
A LPHAI ( \()\), and BETA ( \(\mathcal{j}\) ) should be comect for
\(\dot{F} \operatorname{FO}+1, \ldots, N .>N:=N+1\) : other than \(Q Z\) iteration failed in SH G EQ Z.
\(=\mathrm{N}+2\) : error retum from STGEVC.

\section*{FURTHER DETAILS}

B alancing a m atrix pair ( \(A, B\) ) includes, first, perm uting row \(s\) and colum ns to isolate eigenvalues, second, applying diagonalsim ilarity transform ation to the row s and colum ns to \(m\) ake the row sand colum ns as close in norm as possible. The com puted reciprocal condition num bers comespond to the balanced \(m\) atrix. Perm uting row \(s\) and colum ns will not change the condition num bers (in exact arithm etic) but diagonal scaling will. For further explanation ofbalancing, see section 4.11 .12 of LA PA CK U sers'G uide.

A n approxim ate errorbound on the chordal distance betw een the \(i\)-th computed generalized eigenvalue \(w\) and the comesponding exacteigenvalue lam bda is hord (w, lam bda) <= EPS * norm (ABNRM, BBNRM) /RCONDE (I) A \(n\) approxim ate errorbound forthe angle betw een the \(i\)-th com puted eigenvectorV L (i) orVR (i) is given by

PS * norm (ABNRM,BBNRM)/D \(\mathbb{F}\) (i).
For firtherexplanation of the reciprocalcondition num bers RCONDE and RCONDV , see section 4.11 of LA PACK U sers G uide.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sggglm -solve a general G auss M arkov linear model (G LM )
problem

\section*{SYNOPSIS}
```

SUBROUTINE SGGGLM N,M,P,A,LDA,B,LDB,D,X,Y,W ORK,LDW ORK,
\mathbb{NFO)}
\mathbb{NTEGERN,M,P,LDA,LDB,LDW ORK, INFO}
REAL A (LDA ,*),B (LD B ,*),D (*),X (*),Y (*),W ORK (*)
SUBROUT\mathbb{NE SGGGLM _64 N,M,P,A,LDA,B,LDB,D,X,Y,W ORK,LDW ORK,}
INFO)

```
\(\mathbb{N} T E G E R * 8 N, M, P, L D A, L D B, L D W O R K, \mathbb{N} F O\)
REALA (LDA , *), B (LDB,*),D (*), X (*), Y (*),WORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GGGLM ( \(\mathbb{N}], \mathbb{M}], \mathbb{P}], A,[\operatorname{LDA}], B,[L D B], D, X, Y,[\mathbb{W}\) ORK \(]\), [LDW ORK], [ \(\mathbb{N F O}])\)
\(\mathbb{N} T E G E R:: N, M, P, L D A, L D B, L D W O R K, \mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::D, X,Y,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A,B
SU BROUTINE GGGLM _64 ( \(\mathbb{N}], \mathbb{M}], \mathbb{P}], A,[L D A], B,[\operatorname{LD}], D, X, Y, \mathbb{W}\) ORK \(]\), [LDW ORK], [ \(\mathbb{N F O}]\) )
\(\mathbb{N} T E G E R(8):: N, M, P, L D A, L D B, L D W O R K, \mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::D,X,Y,W ORK
REAL,D IM ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void sggglm (intn, intm, intp, float*a, int lda, float
*b, int ldb, float *d, float *x, float *y, int
*info);
void sggglm _64 (long n, long m , long p, float *a, long lda, float *b, long ldb, float *d, float *x, float *y, long *info);

\section*{PURPOSE}
sggglm solves a general Gauss -M arkov linear model (G LM ) problem :
\(m\) inim ize \(\|y\| 2\) subject to \(d=A *_{x}+B^{*} y\)
X
where \(A\) is an \(N\) boy \(-M m\) atrix, \(B\) is an \(N\) boy \(P m\) atrix, and \(d\) is a given N -vector. It is assum ed that \(\mathrm{M}<=\mathrm{N}<=\mathrm{M}+\mathrm{P}\), and
\[
\operatorname{rank}(A)=M \quad \text { and } \quad \operatorname{rank}(A B)=N
\]

U nder these assum ptions, the constrained equation is alw ays consistent, and there is a unique solution \(x\) and a \(m\) inim al 2-nom solution \(y\), which is obtained using a generalized \(Q R\) factorization of \(A\) and \(B\).

In particular, ifm atrix \(B\) is square nonsingular, then the problem G LM is equivalent to the follow ing w eighted linear least squares problem
\(m\) inim ize \(\|\operatorname{inv}(B) *(d-A * x)\| 2\)
X
w here inv ( \(B\) ) denotes the inverse of \(B\).

\section*{ARGUMENTS}

\section*{N (input) The num ber of row s of the m atriges A and \(\mathrm{B} . \mathrm{N}>=\)} 0 .
\(M\) (input) The num ber of colum ns of them atrix A. \(0<=\mathrm{M}<=\) N .
\(P\) (input) The num ber of colum ns of the m atrix \(B . P>=N M\).

A (input/output)
On entry, the \(N\)-by \(-M\) m atrix A. On exit, A is destroyed.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

B (input/output)
On entry, the N -by P m atrix B. On exit, B is destroyed.

LD B (input)
The leading dim ension of the array \(\mathrm{B} . \mathrm{LD} \mathrm{B}>=\) \(\max (1, N)\).

D (input/output)
On entry, \(D\) is the lefthand side of the G LM equation. On exit, D is destroyed.

X (output)
On exit, X and Y are the solutions of the G LM problem.

Y (output)
See the description ofX .

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al
LDW ORK.

LDW ORK (input)
The dimension of the anay W ORK. LDW ORK >= \(\max (1, N+M+P)\). Foroptim um perform ance, LD W ORK \(>=\) \(M+m\) in \((N, P)+m\) ax \((N, P) * N B\), where \(N B\) is an upperbound for the optim al blocksizes forSGEQRF, SGERQF, SORM QR and SORMRQ .

If LD W ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0: successfulexit.
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the i-th argum enthad an illegalvahue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sgghrd -reduce a pair of realm atrices \((A, B)\) to generalized upper H essenberg form using orthogonal transform ations, \(w\) here \(A\) is a generalm atrix and \(B\) is upper triangular

\section*{SYNOPSIS}
```

SUBROUTINESGGHRD (COMPQ,COMPZ,N, $\mathbb{L O}, \mathbb{H} I, A, L D A, B, L D B, Q, L D Q$,
Z, LD Z, $\mathbb{N}$ FO)

```
CHARACTER * 1 COMPQ, COMPZ
\(\mathbb{N} T E G E R N, \mathbb{L} O, \mathbb{H} I, L D A, L D B, L D Q, L D Z, \mathbb{N F O}\)
REAL A (LDA,*), B (LDB,*), Q (LD Q ,*), Z (LD Z ,*)
SU BROUTINE SGGHRD_64 (COMPQ, COMPZ,N, \(\mathbb{H} O, \mathbb{H} I, A, L D A, B, L D B, Q\),
    LD Q , Z, LD Z, \(\mathbb{N}\) FO)
CHARACTER * 1 COMPQ,COMPZ
\(\mathbb{N}\) TEGER*8N, \(\mathbb{L} O, \mathbb{H} \mathrm{I}, \mathrm{LD} A, L D B, L D Q, L D Z, \mathbb{N} F O\)
REALA (LDA,*), B (LDB,*), Q (LD Q , *), Z (LD Z ,*)

\section*{F95 INTERFACE}

SU BROUTINE GGHRD (COMPQ,COMPZ, \(\mathbb{N}], \mathbb{H}, \mathbb{H} I, A,[L D A], B,[L D B], Q\), [LDQ], Z, [LD Z], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1): : COMPQ,COM PZ
\(\mathbb{N} T E G E R:: N, \mathbb{N}, \mathbb{H} I, L D A, L D B, L D Q, L D Z, \mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (: : : : : A, B, \(\mathrm{Q}, \mathrm{Z}\)
SU BROUTINE GGHRD_64 (COMPQ, COMPZ, \(\mathbb{N}], \mathbb{H}, \mathbb{H} I, A,[L D A], B,[L D B]\), \(Q,[\operatorname{LD} Q], Z,[L D Z],[\mathbb{N F O}])\)

CHARACTER (LEN=1): :COMPQ,COMPZ
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathbb{L O}, \mathbb{H} \mathrm{I}, \mathrm{LD} A, L D B, L D Q, L D Z, \mathbb{N} F O\) REAL,D \(\mathbb{I}\) ENSION (: : : : : A, B, \(\mathrm{Q}, \mathrm{Z}\)

\section*{C INTERFACE}
\#include < sunperfh>
void sgghrd (charcom pq, char com pz, intn, int ilo, int ini, float *a, int lda, float*b, int ldb, float *q, int ldq, float * \(z\), int ldz, int*info);
void sgghrd_64 (char com pq, charcom pz, long n, long ilo, long ihi, float*a, long lda, float *b, long ldb, float*q, long ldq, float *z, long ldz, long *info);

\section*{PURPOSE}
sgghrd reduces a pair of realm atrioes \((A, B)\) to generalized upper H essenberg form using orthogonal transform ations, \(w\) here \(A\) is a generalm atrix and \(B\) is upper triangular: \(Q\) '* \(\mathrm{A} * \mathrm{Z}=\mathrm{H}\) and \(\mathrm{Q}{ }^{\prime} * \mathrm{~B} * \mathrm{Z}=\mathrm{T}\), where H is upper H essenberg, T is uppertriangular, and \(Q\) and \(Z\) are orthogonal, and ' \(m\) eans transpose.

The orthogonalm atrices \(Q\) and \(Z\) are determ ined as products of \(G\) ivens rotations. They \(m\) ay eitherbe form ed explicitly, or they \(m\) ay be postm ultiplied into inputm atrices Q 1 and Z1, so that
\(1 * A * Z 1{ }^{\prime}=\left(Q 1^{*}\right)^{*} H^{*}(Z 1 * Z)^{\prime}\)

\section*{ARGUMENTS}

COMPQ (input)
\(=\mathrm{N}\) ': do not com pute Q ;
\(=I^{\prime}: \mathrm{Q}\) is initialized to the unit m atrix, and the orthogonalm atrix \(Q\) is retumed; \(=V\) ': Q m ust contain an orthogonalm atrix Q 1 on entry, and the product \(\mathrm{Q} \mathrm{I}^{*} \mathrm{Q}\) is retumed.

COMPZ (input)
\(=\mathrm{N}\) ': do notcom pute Z ;
\(=I^{\prime}: Z\) is initialized to the unit matrix, and the orthogonalm atrix Z is retumed; = \(\mathrm{V}^{\prime}: \mathrm{Z} \mathrm{m}\) ust contain an orthogonalm atrix Z1 on entry, and the product Z1*Z is retumed.

N (input) The order of the m atrices A and \(\mathrm{B} . \mathrm{N}>=0\).

ㅍO (input)
It is assum ed thatA is already upper triangular
in row sand colum ns \(1: \mathbb{H O}-1\) and \(\mathbb{H} \mathrm{I}+1 \mathbb{N} . \mathbb{I} \mathrm{O}\) and
Hi I are norm ally setby a previous call to SG G BA L; otherw ise they should be setto 1 and \(N\) respec-
tively. \(1<=\mathbb{H O}<=\mathbb{H} I<=N\), if \(N>0 ; \mathbb{H O}=1\) and \(\mathbb{H} \mathrm{I}=0\), \(\dot{\text { if }} \mathrm{N}=0\).

IH I (input)
See the description of IIO .
A (input/output)
On entry, the N -by -N generalm atrix to be reduced.
On exit, the upper triangle and the first subdiagonalofA are overw ritten w th the upper \(H\) essenberg \(m\) atrix \(H\), and the rest is set to zero.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

B (input/output)
On entry, the \(\mathrm{N}-\) by -N upper triangular \(m\) atrix B .
On exit, the upper triangularm atrix \(T=Q\) ' Z . The elem ents below the diagonal are set to zero.

LD B (input)
The leading dim ension of the array \(B\). LD B \(>=\) \(\max (1, N)\).

Q (input/output)
If \(C O M P Q=N\) : \(Q\) is not referenced.
If COM PQ = 'I': on entry, Q need notbe set, and on exitit contains the orthogonalm atrix \(Q\), where \(Q\) ' is the product of the G ivens transform ations which are applied to \(A\) and \(B\) on the left. IfCOM \(P Q=V\) ': on entry, Q m ust contain an orthogonalm atrix Q 1 , and on exit this is overw rilten by \(Q 1 * Q\).

LD Q (input)
The leading dim ension of the array \(Q . L D Q>=N\) if \(\mathrm{COMPQ}=\mathrm{V}\) 'or I '; LD Q >= 1 otherw ise.

Z (input/output)
If \(C O M P Z=N^{\prime}: Z\) is not referenced.
If COM PZ= I': on entry, Z need notbe set, and on exititcontains the orthogonalm atrix \(Z\), which is the product of the G ivens transform ations which
are applied to \(A\) and \(B\) on the right. If \(\mathrm{COMPZ=V}\) : on entry, Z m ustcontain an orthogonal \(m\) atrix Z1, and on exit this is overw rilten by Z1*Z.

LD \(Z\) (input)
The leading dim ension of the array \(Z\). LD \(Z>=N\) if \(C O M P Z=V\) 'or I'; LD Z >= 1 otherw ise.
\(\mathbb{N F O}\) (output)
= 0: successfulexit.
<0: if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvahue.

\section*{FURTHER DETAILS}

This routine reduces \(A\) to \(H\) essenberg and \(B\) to triangular form by an unblocked reduction, as described in _M atrix_C om putations_, by G olub and V an Loan (Johns H opkins Press.)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgglse - solve the linearequality-constrained least squares (LSE) problem

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGGLSE M ,N,P,A,LDA,B,LDB,C,D,X,W ORK,LDW ORK,}
\mathbb{NFO)}
\mathbb{NTEGERM,N,P,LDA,LDB,LDW ORK,INFO}
REAL A (LDA ,*),B (LDB,*),C (*),D (*),X (*),W ORK (*)
SUBROUTINE SGGLSE_64M,N,P,A,LDA,B,LDB,C,D,X,W ORK,LDW ORK,
INFO)

```
\(\mathbb{N} T E G E R * 8 M, N, P, L D A, L D B, L D W O R K, \mathbb{N F O}\)
REALA (LDA \(\left.{ }^{\star}\right), \mathrm{B}(\mathrm{LDB}, \star), \mathrm{C}\left({ }^{\star}\right), \mathrm{D}\left({ }^{\star}\right), \mathrm{X}\left({ }^{*}\right), \mathrm{W} O R K\left({ }^{*}\right)\)

\section*{F95 INTERFACE}

SU BROUTINE GGLSE (M) \(\mathbb{M}],[\mathbb{P}], A,[L D A], B,[L D B], C, D, X,[W\) ORK \(]\), [LDW ORK], [ \(\mathbb{N F O}])\)
\(\mathbb{N} T E G E R:: M, N, P, L D A, L D B, L D W O R K, \mathbb{N} F O\)
REAL,D IM ENSION (:) ::C,D,X,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A,B
SUBROUTINE GGLSE_64 ( \(\mathbb{M}], \mathbb{N}], \mathbb{P}], A,[\operatorname{LDA}], B,[\operatorname{LDB}], C, D, X,[\mathbb{O}\) ORK \(]\), [LDW ORK], [ \(\mathbb{N F O}]\) )
\(\mathbb{N} T E G E R(8):: M, N, P, L D A, L D B, L D W O R K, \mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::C,D ,X,W ORK
REAL,D \(\mathbb{I M}\) ENSION (: : : : : : A, B

\section*{C INTERFACE}
\#include <sunperfh>
void sgglse (intm, intn, intp, float*a, int lda, float
*b, int ldb, float * C , float *d, float * x , int
*info);
void sgglse_64 (long m, long n, long p, float *a, long lda, float *b, long ldb, float * C , float *d, float *x, long *info);

\section*{PURPOSE}
sgglse solves the linear equality-constrained least squares (LSE ) problem :
\(m\) inim ize \(\left\|C-A *_{X}\right\|_{2}\) subject to \(B *_{X}=d\)
where \(A\) is an \(M\) boy \(N m\) atrix, \(B\) is a \(P\) boy \(-N m\) atrix, \(c\) is a given \(M\)-vector, and \(d\) is a given \(P\)-vector. It is assum ed that \(\mathrm{P}<=\mathrm{N}<=\mathrm{M}+\mathrm{P}\), and
\(\operatorname{rank}(B)=P\) and \(\operatorname{rank}((A))=N\). ( (B) )

These conditions ensure that the LSE problem has a unique solution, which is obtained using a G RQ factorization of the \(m\) atrices \(B\) and A.

\section*{ARGUMENTS}

M (input) The num ber of row s of the matrix \(A . M>=0\).

N (input) The num ber of colum ns of the m atrioes A and B. N \(>=0\).

P (input) The num ber of row s of the m atrix B. \(0<=\mathrm{P}<=\mathrm{N}<=\) \(\mathrm{M}+\mathrm{P}\).

A (input/output)
On entry, the M boy- N m atrix A. On exit, A is destroyed.

LD A (input)
The leading dim ension of the array A.LDA >= \(\max (1, M)\).

B (input/output)
On entry, the \(P-b y-N\) m atrix B. On exit, B is destroyed.

LD B (input)
The leading dim ension of the aray B . LD B >= \(\max (1, \mathrm{P})\).

C (input/output)
On entry, \(C\) contains the right hand side vector for the least squares part of the LSE problem. On exit, the residual sum of squares for the solution is given by the sum of squares ofelem ents \(\mathrm{N}-\mathrm{P}+1\) to \(M\) ofvector \(C\).

D (input/output)
O n entry, \(D\) contains the right hand side vector for the constrained equation. On exit, \(D\) is destroyed.

X (output)
On exit, X is the solution of the LSE problem.
W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dim ension of the anay W ORK. LDW ORK >= \(\max (1, M+N+P)\). For optim um perform ance LD \(W\) ORK \(>=\) \(P+m\) in \((M, N)+m \operatorname{ax}(M, N) * N B\), where \(N B\) is an upperbound for the optim al blocksizes forSGEQRF,SGERQF, SORM QR and SORMRQ.

If LD W ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LD W ORK is issued by XERBLA.
\(\mathbb{I N F O}\) (output)
= 0: successfulexit.
<0: if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvahue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sggqrf - com pute a generalized \(Q R\) factorization of an \(N-b y+M\) \(m\) atrix \(A\) and an \(N\) by \(P\) m atrix B .

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGGQRFN,M,P,A,LDA,TAUA,B,LDB,TAUB,W ORK,LW ORK,}
\mathbb{NFO)}

```
\(\mathbb{N}\) TEGER N, M, P,LDA, LDB,LWORK, \(\mathbb{N} F O\)
REALA (LDA \(\left.{ }^{\star}\right)\), TAUA (*), B (LDB, \(\left.{ }^{\star}\right)\),TAUB (*), W ORK (*)
SU BROUTINE SGGQRF_64 \(\mathbb{N}, \mathrm{M}, \mathrm{P}, \mathrm{A}, \mathrm{LDA}, \mathrm{TAUA}, \mathrm{B}, \mathrm{LD} \mathrm{B}, \mathrm{TA} \mathrm{B}, \mathrm{W}\) ORK,
    LW ORK, \(\mathbb{N} F O\) )
\(\mathbb{N}\) TEGER*8N,M,P,LDA,LDB,LW ORK, \(\mathbb{N} F O\)
REALA (LDA,\(\star)\),TAUA (*), B (LDB,\(\star)\),TAUB (*),WORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GGQRF ( \(\mathbb{N}], \mathbb{M}], \mathbb{P}], A,[L D A], T A U A, B,[L D B], T A U B,[\mathbb{O} O R]\), [LW ORK], [ \(\mathbb{N F O}\) ])
\(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathrm{M}, \mathrm{P}, \mathrm{LDA}, \mathrm{LD} \mathrm{B}, \mathrm{LW}\) ORK, \(\mathbb{N} F \mathrm{O}\)
REAL,D \(\mathbb{I M} E N S I O N(:):: T A U A, T A U B, W\) ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A,B
SU BROUTINEGGQRF_64 ( \(\mathbb{N}], \mathbb{M}],[\mathbb{P}], A,[L D A], T A U A, B,[L D B], T A U B\), [W ORK], [LW ORK], [NFO])
\(\mathbb{N}\) TEGER (8) ::N,M,P,LDA,LDB,LW ORK, \(\mathbb{N} F O\)
REAL,D IM ENSION (:) ::TAUA,TAUB,W ORK
REAL,D \(\mathbb{M}\) ENSION (: :: : ::A, B

\section*{C INTERFACE}
\#include < sunperfh>
void sggqrf(intn, intm, intp, float *a, int lda, float
*taua, float *b, int ldb, float * taub, int *info);
void sggqrf_64 (long \(n\), long \(m\), long \(p\), float *a, long lda, float *taua, float *b, long ldb, float *taub, long *info);

\section*{PURPOSE}
sggqrf com putes a generalized \(Q R\) factorization of an \(N\)-by \(-M\) \(m\) atrix \(A\) and an \(N\) by \(P m\) atrix \(B\) :
\[
A=Q * R, \quad B=Q * T * Z,
\]
where \(Q\) is an \(N\) boy -N orthogonal \(m\) atrix, \(Z\) is a \(P-b y-P\) orthogonalm atrix, and \(R\) and \(T\) assum e one of the form \(s\) :
```

ifN >=M, R = (R11)M , orifN < M, R = (R11 R12

```
) N,
( 0 ) N M
N \(\quad \mathrm{M}-\mathrm{N}\)
M
where R11 is upper triangular, and
if \(\mathrm{N}<=\mathrm{P}, \mathrm{T}=(0 \mathrm{~T} 12) \mathrm{N}\), orif \(\mathrm{N}>\mathrm{P}, \mathrm{T}=(\mathrm{T} 11)\)
\(\mathrm{N} P\),
P-N N (T21)P
P
where T12 orT21 is uppertriangular.
In particular, if \(B\) is square and nonsingular, the \(G Q R\) factorization of \(A\) and \(B\) im plicitly gives the \(Q R\) factorization of inv ( \(B\) ) *A :
\[
\operatorname{inv}(B) \star A=Z \text { * }(\operatorname{inv}(T) * R)
\]
where inv \((B)\) denotes the inverse of the \(m\) atrix \(B\), and \(Z\) ' denotes the transpose of the \(m\) atrix \(Z\).

\section*{ARGUMENTS}

N (input) The num ber of row sof the m atrioes A and \(\mathrm{B} . \mathrm{N}>=\) 0.

M (input) The num ber of collm ns of the \(m\) atrix \(A . M>=0\).
\(P\) (input) The num ber of colum ns of the \(m\) atrix \(B . P>=0\).

A (input/output)
On entry, the N -by M matrix A. On exit, the ele\(m\) ents on and above the diagonalof the amay contain the \(m\) in \((\mathbb{N}, M)-b y-M\) uppertrapezoidalm atrix \(R\) \((R\) is upper triangular if \(N>=M\) ); the elem ents below the diagonal, w ith the amay TA U A, represent the orthogonal matrix \(Q\) as a productofm in \((\mathbb{N}, M\) ) elem entary reflectors (see FurtherD etails).

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

TAUA (output)
The scalar factors of the elem entary reflectors which represent the orthogonal matrix \(Q\) (see FurtherD etails).

B (input/output)
On entry, the \(N\) by P m atrix B. On exit, if \(N\) <= \(P\), the upper triangle of the subarray \(B(1, N, P-\) \(\mathrm{N}+1 \mathrm{P}\) ) contains the N -by -N upper triangularm atrix T ; if \(\mathrm{N}>\mathrm{P}\), the elem ents on and above the \((\mathbb{N} \mathrm{P})\) th subdiagonal contain the N -by P upper trapezoidal \(m\) atrix \(T\); the rem aining elem ents, \(w\) ith the array TAUB, represent the orthogonalm atrix Z as a product ofelem entary reflectors (see Further D etails).

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).

TAUB (output)
The scalar factors of the elem entary reflectors which represent the orthogonal matrix Z (see FurtherD etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the aray \(W\) ORK. LW ORK >= \(\max (1, N, M, \mathbb{P})\). For optim um perform ance LW ORK \(>=\)
\(\max (\mathbb{N}, \mathbb{M}, \mathbb{A})_{\max }(\mathbb{N B} 1, N B 2, N B 3)\), where NB1 is the optim al blocksize forthe \(Q R\) factorization of an N -by M m atrix, NB 2 is the optim al blocksize for the \(R Q\) factorization of an \(N\) by \(P m\) atrix, and NB3 is the optim alblocksize for a callof \(S O R M Q R\).

If LW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{N}\) FO \(=-\) i, the i-th argum enthad an illegalvahue.

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(1) H(2) \ldots H(k), w \text { here } k=m \text { in }(n, m) .
\]

Each H (i) has the form
\[
\mathrm{H}(\mathrm{i})=\mathrm{I}-\operatorname{tana} * \mathrm{~V}^{*} \mathrm{~V}^{\prime}
\]
where taua is a real scalar, and \(v\) is a realvectorw ith \(v(1: i-1)=0\) and \(v(i)=1 ; v(i+1 n)\) is stored on exit in \(A(i+1 m, i)\), and taua in TAUA (i). To form \(Q\) explicitly, use LAPACK subroutine \(S O R G Q R\). To use \(Q\) to update another \(m\) atrix, use LAPACK subroutine SORMQR.

Them atrix \(Z\) is represented as a product of elem entary reflectors
\[
Z=H(1) H(2) \ldots H(k), w h e r e k=m \text { in }(n, p) .
\]

Each H (i) has the form
H (i) = I-taub * v * v'
\(w\) here taub is a real scalar, and \(v\) is a realvectorw ith \(v(p-k+i+1: p)=0\) and \(v(p-k+i)=1 ; v(1: p-k+i-1)\) is stored on exit in B ( \(n-k+i, 1\) p \(-k+i-1\) ), and taub in TAUB (i).
To form Z explicitly, use LAPACK subroutine SORGRQ. To use \(Z\) to update another matrix, use LAPACK subroutine SORMRQ.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sggrqf-com pute a generalized \(R Q\) factorization of an \(M\)-by \(-N\) \(m\) atrix \(A\) and \(a P-\) by \(-N m\) atrix \(B\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGGRQFM,P,N,A,LDA,TAUA,B,LDB,TAUB,W ORK,LW ORK,}
\mathbb{NFO)}

```
\(\mathbb{N}\) TEGERM, \(\mathrm{P}, \mathrm{N}\), LDA, LDB,LW ORK, \(\mathbb{N} F O\)
REALA (LDA \(\left.{ }^{\star}\right)\),TAUA (*), B (LDB, \(\left.{ }^{\star}\right)\),TAUB (*), W ORK (*)
SU BROUTINE SGGRQF_64M,P,N,A,LDA,TAUA,B,LDB,TAUB,WORK,
    LW ORK, \(\mathbb{N} F O\) )
\(\mathbb{N}\) TEGER*8M, \(\mathrm{P}, \mathrm{N}, \mathrm{LD} A, L D B, L W O R K, \mathbb{N} F O\)
REALA (LDA,\(\star)\),TAUA (*), B (LDB,\(\star)\),TAUB (*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GGRQF ( \(\mathbb{M}],[\mathbb{P}], \mathbb{N}], A,[L D A], T A U A, B,[L D B], T A U B,[\mathbb{O}\) OR ], [LW ORK], [ \(\mathbb{N F O}\) ])
\(\mathbb{N}\) TEGER :: M, P,N,LDA,LDB,LW ORK, \(\mathbb{N} F O\)
REAL,D \(\mathbb{I M} E N S I O N(:):: T A U A, T A U B, W\) ORK
REAL,D IM ENSION (:,:) ::A,B
SU BROUTINE GGRQF_64 (M) \(\mathbb{M}, \mathbb{P}], \mathbb{N}], A,[L D A], T A U A, B,[L D B], T A U B\), \(\left[\begin{array}{l}\text { W ORK ], [LW ORK ], [ } \mathbb{N F O}])\end{array}\right.\)
\(\mathbb{N}\) TEGER (8) ::M, P, N,LDA,LDB,LW ORK, \(\mathbb{N} F O\)
REAL,D IM ENSION (:) ::TAUA,TAUB,W ORK
REAL,D \(\mathbb{M}\) ENSION (: :: : ::A, B

\section*{C INTERFACE}
\#include < sunperfh>
void sggrqf(intm, intp, intn, float*a, int lda, float
*taua, float *b, int ldb, float * taub, int *info);
void sggrqf_64 (long m, long p, long n, float*a, long lda, float *taua, float *b, long ldb, float *taub, long *info);

\section*{PURPOSE}
sggrqf com putes a generalized RQ factorization of an \(M\)-by \(-\mathbb{N}\) \(m\) atrix \(A\) and \(a P-b y-N\) matrix \(B\) :
\[
A=R * Q, \quad B=Z * T * Q,
\]
where \(Q\) is an \(N\) by \(-N\) orthogonal \(m\) atrix, \(Z\) is a \(P\)-by \(P\) orthogonalm atrix, and \(R\) and \(T\) assum e one of the form \(s\) :
```

ifM <=N, R = (0 R12)M, orifM > N, R = (R11 )

```
\(\mathrm{M}-\mathrm{N}\),
\[
\begin{equation*}
\mathrm{N} \rightarrow \mathrm{M} \quad \mathrm{M} \tag{R21}
\end{equation*}
\]

\section*{N}
where R12 orR21 is upper triangular, and
if \(P>=N, T=(T 11) N, \quad\) orif \(P<N, T=(T 11 \quad T 12\) ) P,
\((0) P-N\)
\(N\)
where T11 is upper triangular.
In particular, ifB is square and nonsingular, the GRQ factorization ofA and \(B\) im plicitly gives the \(R Q\) factorization ofA *inv (B):
\[
A * \operatorname{inv}(B)=(R * \operatorname{inv}(T)) * Z^{\prime}
\]
where inv \((B)\) denotes the inverse of the \(m\) atrix \(B\), and \(Z\) ' denotes the transpose of the \(m\) atrix \(Z\).

\section*{ARGUMENTS}
\(M\) (input) The num ber of row sof the \(m\) atrix \(A . M>=0\).
\(P\) (input) The num ber of row sof the \(m\) atrix \(B . P>=0\).

N (input) The num ber of collm ns of the m atrioes A and B. N \(>=0\).

A (input/output)
O n entry, the M -by -N m atrix A. On exit, if M <=
N , the upper triangle of the subarray A ( \(1 \mathrm{M}, \mathrm{N}\) -
\(M+1 \mathbb{N}\) ) contains the \(M\)-by \(-M\) upper triangularm atrix
\(R\); if \(M>N\), the elem ents on and above the ( \(M-N\) )th subdiagonal contain the \(\mathrm{M}-\) by -N upper trapezoidal \(m\) atrix \(R\); the rem aining elem ents, \(w\) ith the array TAUA, represent the orthogonalm atrix \(Q\) as a product of elem entary reflectors (see Further D etails).
LD A (input)
The leading dim ension of the array A. LD A >= \(m a x(1, M)\).

TAUA (output)
The scalar factors of the elem entary reflectors which represent the orthogonal matrix Q (see FurtherD etails).

B (input/output)
On entry, the P-by-N m atrix B. On exit, the ele\(m\) ents on and above the diagonal of the aray contain the \(m\) in \((\mathbb{P} N)\)-by -N uppertrapezoidalm atrix T ( \(T\) is upper triangular if \(P>=N\) ); the elem ents below the diagonal, w th the array TA U B, represent the orthogonalm atrix \(Z\) as a product of elem entary reflectors (see FurtherD etails).

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, \mathrm{P})\).

TAUB (output)
The scalar factors of the elem entary reflectors which represent the orthogonal matrix Z (see FurtherD etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al
LW ORK.
LW ORK (input)
The dim ension of the array W ORK. LW ORK >= \(m a x(1, N, M, \mathbb{R})\). For optim um perform ance LW ORK >= \(\max (\mathbb{N}, \mathrm{M}, \mathbb{P})^{\star} \max (\mathbb{N} 1, N B 2, N B 3)\), where \(N B 1\) is the
optim al blocksize forthe \(R Q\) factorization of an M by -N m atrix, NB2 is the optim al blocksize for the QR factorization of a \(P\) by \(-\mathrm{N} m\) atrix, and NB3 is the optim alblocksize for a call of \(S O R M R Q\).

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N} F \mathrm{O}\) (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N} F 0=-i\), the \(i\)-th argum enthad an illegalvahue.

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(1) H(2) \ldots H(k), \text { where } k=m \text { in }(m, n) .
\]

Each H (i) has the form
H (i) \(=I-\operatorname{tana} * V^{*} V^{\prime}\)
where taua is a real scalar, and \(v\) is a realvectorw ith \(v(n-k+i+1 n)=0\) and \(v(n-k+i)=1 ; v(1 n-k+i-1)\) is stored on exit in \(A(m-k+i, 1 m-k+i-1)\), and taua in TAUA (i).
To form \(Q\) explicitly, use LAPACK subroutine SORGRQ. To use \(Q\) to update another \(m\) atrix, use LAPACK subroutine SORMRQ.

Them atrix Z is represented as a product of elem entary reflectors
\[
\mathrm{Z}=\mathrm{H}(1) \mathrm{H}(2) \ldots \mathrm{H}(\mathrm{k}), \text { where } k=m \text { in }(\mathrm{p}, \mathrm{n}) .
\]

Each H (i) has the form
H (i) \(=I-\operatorname{tanb} * V^{*} V^{\prime}\)
where taub is a real scalar, and \(v\) is a realvectorw ith \(\mathrm{v}(1: i-1)=0\) and \(\mathrm{v}(\mathrm{i})=1\); \(\mathrm{v}(\mathrm{i}+1 \mathrm{p})\) is stored on exit in B (i+1 \(\mathrm{p}, \mathrm{i})\), and taub in TAUB (i).
To form Z explicitly, use LAPACK subroutine SORGQR. To use \(Z\) to update another \(m\) atrix, use LAPACK subroutine SORM QR.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sggsvd - com pute the generalized singular value decom position ( \(G\) SVD) of an \(M\) by \(-N\) real \(m\) atrix \(A\) and \(P-b y-N\) real \(m\) atrix B

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SGGSVD(JOBU,NOBV,NOBQ,M,N,P,K,L,A,LDA,B,LDB,}
ALPHA,BETA,U,LDU,V,LDV,Q,LDQ,W ORK,IN ORK 3, INFO)

```
CHARACTER * 1 JOBU, JOBV , JOBQ
\(\mathbb{N}\) TEGER M, N, P, K, L, LDA ,LD B, LD U ,LDV ,LD Q , \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R \mathbb{I N}\) ORK 3 (*)
REAL A (LDA, *), B (LDB,*), ALPHA (*), BETA (*), U (LDU,*),
V (LDV,*), Q (LDQ,*), W ORK (*)

    A LPHA, BETA, U, LDU, \(V, L D V, Q, L D Q, W\) ORK, \(\mathbb{I N} O R K 3, \mathbb{N} F O)\)
CHARACTER * 1 JOBU, 0 BV , JOBQ
\(\mathbb{N}\) TEGER*8 M , N, P, K, L, LD A ,LD B, LD U ,LDV ,LD Q , \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R * 8 \mathbb{I N}\) ORK 3 (*)
REAL A (LDA, *), B (LDB,*), ALPHA (*), BETA (*), U (LDU,*),
V (LDV , *), Q (LDQ , *), W ORK (*)

\section*{F95 INTERFACE}
 [LD B],A LPHA, BETA, U , [LDU ], V , [LDV], Q, [LD Q], [W ORK], IV ORK 3, [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1)::JOBU ,JOBV,JOBQ
\(\mathbb{N} T E G E R:: M, N, P, K, L, L D A, L D B, L D U, L D V, L D Q, \mathbb{N} F O\)
\(\mathbb{I N T E G E R , D} \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK 3

REAL,D \(\mathbb{M}\) ENSION (:) ::ALPHA,BETA,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A, B , U , V , Q

SU BROUTINE G G SVD_64 (JOBU, JOBV , JO BQ, M ], \(\mathbb{N}],[P], K, L, A,[L D A]\),
 \(\mathbb{I} W\) ORK 3, [ \(\mathbb{N} F O\) ])

CHARACTER ( \(\llcorner E N=1\) ) :: JOBU , \(\mathrm{OBV}, \mathrm{OB}\) B
\(\mathbb{N}\) TEGER (8) :: M , N , P , K , L, LDA , LD B , LD U ,LDV , LD Q , \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N} O R K 3\)
REAL,D \(\mathbb{M}\) ENSION (:) ::ALPHA,BETA,W ORK
REAL, D \(\mathbb{M}\) ENSION (:,:) ::A, B , U , V , Q

\section*{C INTERFACE}
\#include <sunperfh>
void sggsvd (char j̀bu, char j̀bv, char j̀bq, intm, int n, int \(p\), int \({ }^{*} \mathrm{k}\), int \({ }^{*} \mathrm{l}\), float * a , int lda, float *b, int ldlb, float * alpha, float *beta, float *u, int ldu, float *v, int ldv, float *q, int ldq, int *íiv ork3, int *info);
void sggsvd_64 (char j̀jbu, char j̀bv, char joboq, long m , long n, long p, long *k, long *l, float *a, long lda, float*b, long ldb, float *alpha, float *beta, float *u, long ldu, float *v, long ldv, float *q, long ldq, long *íw ork3, long *info);

\section*{PURPOSE}
sggsvd com putes the generalized singularvalue decom position ( \(G\) SV D ) of an \(M\) by \(-N\) realm atrix \(A\) and \(P\) boy \(-N\) realm atrix \(B\) :
\(U{ }^{*} A * Q=D 1^{*}(0 R), \quad V * B * Q=D 2^{*}(0 R)\)
\(w\) here \(\mathrm{U}, \mathrm{V}\) and Q are orthogonal \(m\) atrioes, and Z ' is the transpose of \(Z . L e t K+L=\) the effective num erical rank of them atrix ( \(A\) ' \(B\) ' ', then \(R\) is a \(K+L-b y-K+L\) nonsingular upper triangularm atrix, \(D 1\) and \(D 2\) are \(M\)-by \(-(K+L)\) and \(P-b y-\) \((\mathbb{K}+L)\) "diagonal" \(m\) atrices and of the follow ing structures, respectively:

IfM \(-\mathrm{K}->=0\),
\begin{tabular}{|c|}
\hline \multirow[t]{4}{*}{} \\
\hline \\
\hline \\
\hline \\
\hline
\end{tabular}

K L
```

    D2 = L (0 S )
    P-工 (0 0)
        N-K\dashv K L
    (0R ) = K (0 R11 R12 )
L(0 0 R22)

```
where
\[
\begin{aligned}
& C=\operatorname{diag}(A L P H A(K+1), \ldots, \text { A LPH }(\mathbb{K}+L)), \\
& S=\operatorname{diag}(\operatorname{BETA}(\mathbb{K}+1), \ldots, \operatorname{BETA}(\mathbb{K}+L)), \\
& C \star * 2+S * * 2=I .
\end{aligned}
\]
\(R\) is stored in \(A(1: K+L, N-K-1 \mathbb{N})\) on exit.

IfM \(\mathrm{K}-\mathrm{L}<0\),
\[
\begin{aligned}
& \text { K M K K + L } \mathrm{M} \\
& D 1=K\left(\begin{array}{ll}
I & 0
\end{array}\right) \\
& M K\left(\begin{array}{lll}
0 & C
\end{array}\right) \\
& \text { K M K K + L }-\mathrm{M} \\
& D 2=M K(0 S 0) \\
& \mathrm{K}+\mathrm{L} \mathrm{M} \text { ( } 0 \text { O I ) } \\
& \text { P }-\left(\begin{array}{lll}
0 & 0
\end{array}\right)
\end{aligned}
\]
```

                N-K\dashv K M K K +L M
    (0R ) = K (0 R11 R12 R13 )
M K (0 0 R22 R23 )
K+L-M (0 0 0 R33)

```
where
```

$C=\operatorname{diag}(A L P H A(K+1), \ldots, A L P H A(M))$,
$S=\operatorname{diag}(B E T A(K+1), \ldots, B E T A(M))$,
$C * * 2+S * * 2=I$.

```
(R11 R12 R13) is stored in A ( \(1 \mathrm{M}, \mathrm{N} \mathrm{K}-\mathrm{L}+1 \mathbb{N}\) ), and R 33 is stored
(0 R 22 R 23 )
in \(B(M-1: L, N+M-K+1 \mathbb{N})\) on exit.

The routine com putes \(C, S, R\), and optionally the orthogonal transform ation \(m\) atrices \(U, V\) and \(Q\).

In particular, if B is an N boy N nonsingular m atrix, then the G SVD ofA and B im plicitly gives the SVD ofA *inv \((B)\) :
\[
A * \operatorname{inv}(B)=U *(D 1 * \operatorname{inv}(D 2)) * V{ }^{\prime} .
\]

If ( \(A\) 'B )'has orthonorm alcolum ns, then the G SVD of A and \(B\) is also equal to the \(C S\) decom position of \(A\) and B.Further-
m ore, the G SVD can be used to derive the solution of the eigenvalue problem :

A *A \(\mathrm{x}=\operatorname{lam}\) bda* B *B x .
In som e literature, the G SVD of \(A\) and \(B\) is presented in the form

U *A * \(\mathrm{X}=(0 \mathrm{D} 1), \quad \mathrm{V}{ }^{*} \mathrm{~B} * \mathrm{X}=(0 \mathrm{D} 2)\)
\(w\) here \(U\) and \(V\) are orthogonal and \(X\) is nonsingular, \(D 1\) and \(D 2\) are "diagonal". The form erG SVD form can be converted to the latter form by taking the nonsingularm atrix X as
\[
\begin{aligned}
X= & Q *\left(\begin{array}{ll}
I & 0
\end{array}\right) \\
& (0 \operatorname{inv}(\mathbb{R})) .
\end{aligned}
\]

\section*{ARGUMENTS}
\(J O B U\) (input)
\(=\mathrm{U}:\) : O rthogonalm atrix U is com puted;
\(=\mathrm{N}\) : U is notcom puted.
JO BV (input)
\(=\mathrm{V}:\) : O rthogonalm atrix V is com puted;
\(=\mathrm{N}: \mathrm{V}\) is not com puted.
\(J O B Q\) (input)
= Q ': Orthogonalm atrix \(Q\) is com puted;
\(=\mathrm{N}\) ': Q is notcom puted.

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrioes A and B. N \(>=0\).
\(P\) (input) The num ber of row sof the \(m\) atrix \(B . P>=0\).

K (output)
On exit, \(K\) and L specify the dim ension of the subblocks described in the Pupose section. \(\mathrm{K}+\mathrm{L}=\) effective num erical rank of (A 'B )'.

L (output)
See the description of K .

A (input/output)
On entry, the M -by-N m atrix A. On exit, A contains the triangularm atrix \(R\), orpartofR. See
Purpose fordetails.

LD A (input)
The leading dim ension of the array A. LDA >= max (1, M).

B (input/output)
On entry, the \(P-b y-N\) m atrix B. On exit, B contains the triangularm atrix \(R\) if \(M K-4<0\). See Purpose fordetails.

LD B (input)
The leading dim ension of the array B. LD A >= \(\max (1, P)\).
ALPHA (output)
On exi, ALPHA and BETA contain the generalized singularvahue pairs of \(A\) and ; A LPHA \((1: K)=1\),
\(\operatorname{BETA}(1: K)=0\), and ifM \(K-\Psi=0\), ALPHA \((\mathbb{K}+1 \mathbb{K}+L)\)
= C,
BETA \((K+1 K+L)=S\), or if \(M-K-0\),
A LPHA \((\mathbb{K}+1 \mathrm{M})=\mathrm{C}, \mathrm{A}\) LPHA \((\mathrm{M}+1: \mathrm{K}+\mathrm{L})=0\)
\(\operatorname{BETA}(K+1 M)=S, \quad B E T A(M+1 \pi+L)=1\) and
A LPHA \((\mathbb{K}+\mathrm{L}+1 \mathbb{N})=0\)
\(\operatorname{BETA}(\mathbb{K}+\mathrm{L}+1 \mathbb{N})=0\)

\section*{BETA (output)}

See the description ofA LPH A .
U (output)
If \(\mathrm{JOBU}=\mathrm{U}\) ', U contains the M -by M orthogonal
\(m\) atrix \(U\). If \(J O B U=N ', U\) is not referenced.

LD U (input)
The leading dim ension of the array \(U\). LD \(U\) >= \(m a x(1, M)\) if \(J 0 B U=U ' ; L D U>=1\) otherw ise.

V (output)
If \(\mathrm{JOBV}=\mathrm{V}\) ', V contains the P -by P orthogonal \(m\) atrix \(V\). If \(J O B V=N ', V\) is not referenced.

LDV (input)
The leading dim ension of the array V . LDV >= \(\max (1, \mathrm{P})\) if \(\mathrm{JOBV}=\mathrm{V} ; \mathrm{LDV}>=1\) otherw ise.

Q (output)
If \(\mathrm{OOBQ}=\mathrm{Q}\) ', Q contains the N by-N orthogonal \(m\) atrix \(Q\). If \(J O B Q=N\) ', \(Q\) is not referenced.

LD Q (input)
The leading dimension of the array \(Q . L D Q>=\) \(\max (1, N)\) if \(J O B Q=Q ; L D Q>=1\) otherw ise.

W ORK (w orkspace)
dim ension \((\max (3 * N, M, P)+N)\)

IW ORK 3 (output)
dim ension \((\mathbb{N})\) On exit, \(\mathbb{I V}\) ORK 3 stores the sorting
inform ation. M ore precisely, the follow ing loop
w ill.sortALPHA for \(I=K+1\), \(m\) in \((M, K+L) ~ s w a p\)
A LPHA (I) and ALPHA ( \(\mathbb{I N}\) ORK 3 (I)) endfor such that
\(A \operatorname{LPHA}(1)>=A \operatorname{LPHA}(2)>=\ldots>=A L P H A(N)\).
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvahue.
\(>0\) : if \(\mathbb{N F O}=1\), the Jacobi-type procedure failed to converge. For further details, see subroutine STG SJA.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sggsvp - com pute orthogonalm atrioes \(\mathrm{U}, \mathrm{V}\) and Q such that \(\mathrm{NH} \mathrm{K} \mathrm{K} \mathrm{L} \mathrm{U} * \mathrm{~A} * \mathrm{Q}=\mathrm{K}\) (0A12A13) if \(\mathrm{M} \mathrm{K}-\mathrm{L}>=0\)

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SGG SVP (OOBU,NOBV,NOBQ ,M ,P,N,A ,LDA,B,LD B,TOLA,}
TOLB,K,L,U,LDU,V,LDV,Q,LDQ,IN ORK,TAU,W ORK,\mathbb{NFO )}
CHARACTER * 1 JOBU,NOBV,NOBQ
\mathbb{NTEGERM,P,N,LDA,LDB,K,L,LDU,LDV,LDQ,INFO}
INTEGER IN ORK (*)
REAL TOLA,TOLB
REAL A (LDA,*), B (LDB,*), U (LDU ,*), V (LDV,*), Q (LDQ,*),
TAU(*),W ORK (*)
SU BROUT\mathbb{NE SGGSVP_64(JOBU,NOBV,JOBQ,M ,P,N,A,LDA,B,LD B,TOLA,}
TOLB,K,L,U,LDU,V,LDV,Q,LDQ,IN ORK,TAU,W ORK,INFO)

```
CHARACTER * 1 JOBU, JOBV , JOBQ
\(\mathbb{N}\) TEGER*8M, \(\mathrm{P}, \mathrm{N}, \mathrm{LD} A, L D B, K, L, L D U, L D V, L D Q, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER*8 \(\mathbb{I N}\) ORK (*)
REAL TOLA,TOLB
REAL A (LDA,*), B (LDB,*), U (LDU,*), V (LDV,*), Q (LDQ,*),
TAU ( \({ }^{\star}\) ), W ORK ( \({ }^{*}\) )

\section*{F95 INTERFACE}

SU BROUTINE G GSVP (JOBU, \(\mathrm{JOBV}, \mathcal{O B Q}, \mathbb{M}],[\mathbb{P}], \mathbb{N}], A,[L D A], B,[L D B]\), TO LA, TO LB , K, L, U , [LDU ], V , [LDV ], Q , [LD Q ], [IW ORK ], [TAU ], [ \(\mathrm{W} O \mathrm{RK}\) ], [ \(\mathbb{N} F \mathrm{O}]\) )

CHARACTER (LEN=1) :: \(0 \mathrm{OBU}, J 0 \mathrm{BV}, \mathcal{J O B Q}\)
\(\mathbb{N}\) TEGER ::M, \(\mathrm{P}, \mathrm{N}, \mathrm{LD} A, L D B, K, L, L D U, L D V, L D Q, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
REAL ::TOLA,TOLB
REAL,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A, B, U, V, Q
SU BROUTINE G GSVP_64 (JOBU, \(\mathcal{J O B V}, \mathcal{J O B Q}, \mathbb{M}], \mathbb{P}], \mathbb{N}], A,[L D A], B\), [LDB],TOLA,TOLB, \(\mathrm{K}, \mathrm{L}, \mathrm{U},[\mathrm{LD} \mathrm{U}], \mathrm{V},[\mathrm{LDV}], \mathrm{Q},[\mathrm{LD} \mathrm{Q}],[\mathbb{I V} \mathrm{ORK}]\), [TAU], [W ORK], [NFO])

CHARACTER (LEN =1) :: JO BU , JOBV , 0
\(\mathbb{N}\) TEGER (8) ::M, \(\mathrm{P}, \mathrm{N}, \mathrm{LDA}, \mathrm{LD} B, \mathrm{~K}, \mathrm{~L}, \mathrm{LD} \mathrm{U}, \mathrm{LDV}, \mathrm{LD} Q, \mathbb{N} F \mathrm{O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{I N}\) ORK
REAL ::TOLA,TOLB
REAL,D IM ENSION (:) ::TAU,W ORK
REAL,D IM ENSION (:,:) ::A,B,U,V,Q

\section*{C INTERFACE}
\#include <sunperfh>
 intn, float *a, int lda, float*b, int ldb, float tola, float toll, int * \(k\), int *l, float *u, int ldu, float *v, int ldv, float *q, int ldq, int *info);
void sggsvp_64 (char jobu, char jobv, char jobo, long m, long
p , long n , float *a, long lda, float *b, long ldb, float tola, floattolb, long *k, long *l, float
*u, long ldu, float *v, long ldv, float *q, long ldq, long *info);

\section*{PURPOSE}
sggsvp com putes orthogonalm atrices \(\mathrm{U}, \mathrm{V}\) and Q such that
```

            L (0 0 A23)
        MK工(0}0000
            N-K-L K L
        = K (0 A12 A13) ifM K- < 0;
        MK(0 0 A 23)
            N-K-Ł K L
    V *B*Q = L (0 0 B13)
P-((0 0 0 )

```
where the K -by-K m atrix A 12 and L -by-L m atrix B13 are nonsingular uppertriangular; A 23 is L-by-L upper triangular if M K-L >=0, otherw ise A 23 is \(M-K\) )-by-L upper trapezoidal. \(K+L=\) the effective num erical rank of the \(M+P)\)-by \(-N \mathrm{~N}\) atrix
(A ', B )'. Z 'denotes the transpose of Z .

This decom position is the preprocessing step for com puting the Generalized Singular V alue D ecom position (GSVD), see subroutine SG G SV D .

\section*{ARGUMENTS}
\(J 0 \mathrm{BU}\) (input)
\(=\mathrm{U}:\) : O rthogonalm atrix U is com puted;
\(=N^{\prime}: U\) is notcom puted.
JO BV (input)
\(=\mathrm{V}\) : O rthogonalm atrix V is com puted;
= N ': V is not com puted.
\(J O B Q\) (input)
\(=Q\) : O rthogonalm atrix Q is com puted;
\(=N^{\prime}: Q\) is notcom puted.
M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
\(P\) (input) The num ber of row sof the \(m\) atrix \(B . P>=0\).

N (input) The num ber of collm ns of the m atrioes A and B. N \(>=0\).

A (input/output)
On entry, the M by-N matrix A. On exit, A contains the triangular (or trapezoidal) matrix described in the Purpose section.

LDA (input)
The leading dim ension of the array A. LDA >= max (1,M).

B (input/output)
On entry, the \(\mathrm{P}-\) by-N m atrix B. On exit, B contains the triangularm atrix described in the Purpose section.

LD B (input)
The leading dim ension of the aray \(B . L D B>=\) \(\max (1, P)\).

TO LA (input)
TOLA and TOLB are the thresholds to determ ine the effective num erical rank ofm atrix B and a subblock ofA. Generally, they are set to TO LA =
\(\operatorname{MAXM}(\mathbb{N}) \star\) norm (A)*MACHEPS, TOLB =
MAX \((P, N) \star\) norm ( \(B\) )*M ACHEPS. The size of TOLA and TO LB \(m\) ay affect the size of backw ard emors of the decom position.

\section*{TO LB (input)}

See the description of TO LA .

K (output)
O n exit, \(K\) and \(L\) specify the dim ension of the subblocks described in Punpose. \(\mathrm{K}+\mathrm{L}=\) effective num ericalrank of (A 'B )'.

L (output)
See the description of .
U (input) If \(\mathrm{OB} \mathrm{BU}=\mathrm{U}\) ', U contains the orthogonalm atrix U . If \(\mathrm{JO} \mathrm{BU}=\mathrm{N}\) ', U is not referenced.

LD U (input)
The leading dim ension of the array \(U\). LD U >= \(\mathrm{max}(1, \mathrm{M})\) if \(\mathrm{JOBU}=\mathrm{U}\) '; LD U >= 1 otherw ise.

V (input) If \(\mathrm{O} \mathrm{BV}=\mathrm{V}\) ', V contains the orthogonalm atrix V . If \(\mathrm{JO} \mathrm{BV}=\mathrm{N}\) ', V is not referenced.

\section*{LD V (input)}

The leading dim ension of the array V . LDV >= \(\max (1, \mathrm{P})\) if \(\mathrm{OOBV}=\mathrm{V}\) '; LDV \(>=1\) otherw ise.
\(Q\) (input) If \(J O B Q=Q\) ', \(Q\) contains the orthogonalm atrix \(Q\). If \(\mathrm{JOBQ}=\mathrm{N}, \mathrm{Q}\) is not referenced.

LD Q (input)
The leading dim ension of the aray \(\mathrm{Q} . \mathrm{LD} \mathrm{Q}\) >= \(\max (1, N)\) if \(J O B Q=Q ; L D Q>=1\) otherw ise.

IN ORK (w orkspace)
dim ension \((\mathbb{N})\)

TAU (w orkspace)
dim ension (N)

W ORK (w orkspace)
dim ension MAX ( \(\left.3{ }^{\star} \mathrm{N}, \mathrm{M}, \mathrm{P}\right)\) )
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvahue.

\section*{FURTHER DETAILS}

The subroutine uses LA PACK subroutine SGEQPF for the QR factorization \(w\) ith \(c o l u m n\) pivoting to detect the effective num erical rank of the a \(m\) atrix. It \(m\) ay be replaced by a better rank determ ination strategy.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
sgssco -G eneral sparse solvercondition num berestim ate.

\section*{SYNOPSIS}

SUBROUTINE SGSSCO (COND,HANDLE, \(\mathbb{E R}\) )
\begin{tabular}{lc}
\(\mathbb{N} T E G E R\) & \(\mathbb{E R}\) \\
REAL & COND \\
DOUBLE PRECISION & \\
\end{tabular}

\section*{PURPOSE}

SGSSCO -C ondition num berestim ate.

\section*{PARAMETERS}

COND -REAL
On exit, an estim ate of the condition num ber of the factored \(m\) atrix. M ust.be called after the num erical factorization subroutine, SGSSFA 0 ).

HANDLE (150) -D OUBLE PREC IS IO N anay
On entry, HANDLE ( \(*\) ) is an array containing inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine.
M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Enrornum ber. If no errorencountered, unchanged on
exit. If emorencountered, it is set to a non-zero
integer. Errornum bers setby this subroutine:
-700 : Invalid calling sequence - need to callSG SSFA first.
-710 : C ondition num ber estim ate not available (notim plem ented for this H A N D LE sm atix type).

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
sgssda -D eallocate w orking storage for the general sparse solver.

\section*{SYNOPSIS}

SUBROUTINE SGSSDA (HANDLE, \(\mathbb{E R}\) )
\(\mathbb{N}\) TEGER \(\quad \mathbb{R}\)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

SGSSDA -D eallocate dynam ically allocated w orking storage.

\section*{PARAMETERS}

HANDLE (150) -D OUBLE PREC IS IO N array
On entry, HANDLE (*) is an array containing inform ation needed by the solver, and \(m\) ustbe passed unchanged to each sparse solver subroutine. M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Errornum ber. If no errorencountered, unchanged on exit. If errorencountered, it is set to a non-zero integer. Errornum bers set.by this subroutine:
none

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
sgssfa -G eneral sparse solvernum eric factorization.

\section*{SYNOPSIS}

SUBROUTINE SGSSFA (NEQNS,COLSTR,ROW \(\mathbb{N D}, V A L U E S, H A N D L E, \mathbb{E R}\) )
\(\mathbb{N}\) TEGER \(\quad\) NEQNS, COLSTR (*), ROW \(\mathbb{N D}(*), \mathbb{E R}\)
REAL VALUES (*)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

SGSSFA -N um eric factorization of a sparse m atrix.

\section*{PARAMETERS}

NEQNS - \(\mathbb{N}\) TEGER
On entry, NEQNS specifies the num ber of equations in coefficientm atrix. U nchanged on exit.
\(\operatorname{COLSTR}\left(^{*}\right)-\mathbb{N}\) TEG ER array
On entry, \(\operatorname{COLSTR}\left(^{*}\right)\) is an array of size \((\mathbb{N} E \mathrm{~N}+1\) ), containing the pointers of them atrix structure.
U nchanged on exit.
ROWIND ( \(\left.{ }^{( }\right)-\mathbb{N}\) TEGER aray
On entry, ROWIND ( \({ }^{*}\) ) is an array of size
CO LSTR \(\mathbb{N} E Q N S+1)-1\), containing the indices of the \(m\) atrix structure. U nchanged on exit.

VALUES (*) -REA L array
On entry, VALUES (*) is an array of size
CO LSTR \(\mathbb{N E Q N S}+1\) )-1, containing the num eric values of
the sparse \(m\) atrix to be factored. U nchanged on exit.

HANDLE (150) -D OUBLE PRECISIO N array On entry, HANDLE (*) is an array containing inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine. M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Errornum ber. If no emrorencountered, unchanged on exit. If errorencountered, it is set to a non-zero integer. Enrornum bers set.by this subroutine:
-300 : Invalid calling sequence - need to callSG SSO R first.
-301 : Failure to dynam ically allocate \(m\) em ory. -666 : Intemalerror.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
sgssfs -G eneral.sparse solver one callinterface.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGSSFS(M TXTYP,PIVOT,NEQNS,COLSTR,ROW IND,}
VALUES,NRHS ,RHS ,LDRHS,ORDMTHD,
OUTUNT,MSGLVL,HANDLE,\mathbb{ER)}

```
CHARACTER*2 M TXTYP
CHARACTER*1 PIVOT
\(\mathbb{N}\) TEGER NEQNS,COLSTR (*),ROW \(\mathbb{N} D\left({ }^{*}\right)\),NRHS,LDRHS,
    OUTUNT,MSGLVL, ER
CHARACTER*3 ORDMTHD
REAL VALUES (*),RHS (*)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

SGSSFS -G eneral sparse solver one call interface.

\section*{PARAMETERS}
```

MTXTYP -CHARACTER*2
On entry, M TX TY P specifies the coefficientm atrix type. Specifically, the valid options are:

```

Sp 'or SP '-sym m etric structure, positive-definite values
ss'or SS '-sym m etric structure, sym m etric values
su 'or SU '-sym m etric structure, unsym \(m\) etric values
uu 'or UU '-unsym \(m\) etric structure, unsym \(m\) etric values

U nchanged on exit.

\section*{PIVOT -CHARACTER*1}

On entry, pivot specifies w hether ornotpivoting is used in the course of the num eric factorization.
The valid options are:
h'or N '-no pivoting is used
(Pivoting is not supported forthis release).
U nchanged on exit.

NEQNS - \(\mathbb{N}\) TEGER
On entry, NEQN S specifies the num ber ofequations in the coefficientm atrix. NEQ NS m ustbe at leastone. U nchanged on exit.
\(\operatorname{COLSTR}\left(^{\star}\right)-\mathbb{N}\) TEGER array
On entry, \(\operatorname{COLSTR}\) ( \({ }^{*}\) ) is an array of size \((\mathbb{N} E Q \mathrm{~N}+1\) ), containing the pointers of them atrix structure.
U nchanged on exit.
ROWIND ( \({ }^{\star}\) ) - \(\mathbb{N}\) TEG ER aray
On entry, ROWIND (*) is an array of size COLSTR \(\mathbb{N} E Q N S+1)-1\), containing the indices of the \(m\) atrix structure. U nchanged on exit.

VALUES (*) -REA L array
O n entry, VALUES (*) is an array of size
CO LSTR \(\mathbb{N} E Q N S+1)-1\), containing the non-zero num eric
values of the sparse \(m\) atrix to be factored.
U nchanged on exit.

NRHS - \(\mathbb{N}\) TEGER
On entry, NRH S specifies the num ber of righthand sides to solve for. U nchanged on exit.

RHS (*) -REAL amay
On entry, RH S (LDRHS,NRHS) contains the NRHS right hand sides. On exit, itcontains the solutions.

\section*{LDRHS - \(\mathbb{N}\) TEGER}

On entry, LD RH S specifies the leading dim ension of the RH S array. U nchanged on exit.

ORDMTHD -CHARACTER*3
On entry, ORDM THD specifies the fill-reducing ordering to be used by the sparse solver. Specifically, the valid options are:
hat'or NAT'-naturalordering (no ordering) mmd'or M M D '-m ultiplem inim um degree gnd 'or GND '-general nested dissection
uso 'or U SO '-user specified ordering (see SG SSU O )

U nchanged on exit.
OUTUNT - \(\mathbb{N}\) TEGER
O utputunit. U nchanged on exit.
M SGLVL - \(\mathbb{N}\) TEGER
M essage level.
0 -no output from solver.
N o m essages supported for this release.)
U nchanged on exit.

HANDLE (150) -D OUBLE PREC IS IO N array
On entry, HANDLE (*) is an array of containing inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine.
M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
E rrornum ber. If no errorencountered, unchanged on
exit. If errorencountered, it is set to a non-zero
integer. Enrornum bers setby this subroutine:
-101 : Failure to dynam ically allocate m em ory.
-102 : Invalid m atrix type.
-103 : Invalid pivotoption.
-104:N um berof nonzeros is less than NEQ N S .
-105: NEQNS < 1
-201: Failure to dynam ically allocate \(m\) em ory.
-301 : Failure to dynam ically allocate \(m\) em ory.
-401: Failure to dynam ically allocate \(m\) em ory.
-402 : NRHS < 1
-403 :NEQNS > LDRHS
-666 : Intemalemor.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
sgssin - Initialize the general sparse solver.

\section*{SYNOPSIS}
```

SUBROUTINESGSSIN(MTXTYP,PIVOT,NEQNS,COLSTR,ROW IND,OUTUNT,
M SGLVL,HANDLE,\mathbb{ER )}
CHARACTER*2 M TXTYP
CHARACTER*1 PIVOT
\mathbb{NTEGER NEQNS,COLSTR (*),ROW IND (*),OUTUNT,MSGLVL,\mathbb{ER}}\mathbf{N}\mathrm{ (*)}
DOUBLE PRECISION HANDLE (150)

```

\section*{PURPOSE}

SGSSIN -Initialize the sparse solver and input the m atrix
structure.

\section*{PARAMETERS}
```

MTXTYP -CHARACTER*2
On entry,M TX TY P specifies the coefficientm atrix
type. Specifically, the valid options are:
sp 'or SP '-symm etric structure, positive-definite values
ss'or SS'-symm etric structure, sym m etric values
su'or SU '-symm etric structure, unsym m etric values
uu 'or UU '-unsymm etric structure, unsym m etric values
U nchanged on exit.
PIVOT -CHARACTER*1
On entry,PIV OT specifies w hetherornotpivoting is
used in the course of the num eric factorization.

```

The valid options are:
h'or \(\mathrm{N}^{\text {' }}\)-no pivoting is used
(Pivoting is not supported for this release).
U nchanged on exit.

NEQNS - \(\mathbb{N}\) TEGER
On entry, NEQNS specifies the num ber of equations in the coefficientm atrix. NEQNS m ustibe at leastone. U nchanged on exit.
\(\operatorname{COLSTR}\) ( \(\left.^{( }\right)-\mathbb{N}\) TEGER aray
On entry, \(\operatorname{COLSTR}\left({ }^{*}\right)\) is an array of size \((\mathbb{N} E Q \mathrm{~N}+1)\), containing the pointers of them atrix structure. U nchanged on exit.

\section*{ROWIND (*) - \(\mathbb{N}\) TEG ER anay}

On entry, ROWIND ( \({ }^{*}\) ) is an array of size
CO LSTR \(\mathbb{N} E Q N S+1)-1\), containing the indices of the \(m\) atrix structure. Unchanged on exit.

HANDLE (150) -D OUBLE PREC IS IO N array
On entry, HANDLE (*) is an array containing
inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine.
M odified on exit.

OUTUNT - \(\mathbb{N} T E G E R\)
O utputunit. U nchanged on exit.
M SGLVL - \(\mathbb{N}\) TEGER
M essage level.
0 -no output from solver.
(N om essages supported for this release.)

U nchanged on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
E rrornum ber. If no emrorencountered, unchanged on exit. If emorencountered, it is set to a non-zero integer. Enrornum bers setby this subroutine:
-101 : Faihure to dynam ically allocate \(m\) em ory.
-102 : Invalid m atrix type.
-103 : Invalid pivotoption.
-104 :N um berofnonzeros less than N EQ N S .
-105: NEQNS < 1

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
sgssor-G eneral sparse solverordering and sym bolic factorization.

\section*{SYNOPSIS}
```

SUBROUTINE SGSSOR(ORDMTHD,HANDLE,\mathbb{ER})
CHARACTER*3 ORDMTHD
\mathbb{NTEGER ER}
DOUBLE PRECISION HANDLE (150)

```

\section*{PURPOSE}

SGSSOR -O rders and sym bolically factors a sparse m atrix.

\section*{PARAMETERS}

ORDMTHD -CHARACTER*3
On entry, ORDM THD specifies the fill-reducing ordering to be used by the sparse solver. Specifically, the valid options are:
hat'or NAT '-naturalordering (no ordering)
mmd'or M M D '-m ultiplem inim um degree
gnd 'or GND '-generalnested dissection
Uso 'or U SO '-user specified ordering (see SG SSU O )
U nchanged on exit.
HANDLE (150) -DOUBLE PRECISIO N aray
On entry, HANDLE (*) is an array containing
inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine.

M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Errornum ber. If no errorencountered, unchanged on
exit. If emrorencountered, it is set to a non-zero
integer. Enrornum bers set.by this subroutine:
-200 : Invalid calling sequence - need to callSG SS IN first.
-201 : Failure to dynam ically allocate \(m\) em ory.
-666 : Intemalerror.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
sgssps -Print general.sparse solver statics.

\section*{SYNOPSIS}

SUBROUTINE SGSSPS (HANDLE, \(\mathbb{E R}\) )
\(\mathbb{N} T E G E R \quad \mathbb{E R}\)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

SGSSPS -Print solver statistics.

\section*{PARAMETERS}

HANDLE (150) -DOUBLE PRECISION aray
On entry, HANDLE ( \(*\) ) is an array containing
inform ation needed by the solver, and \(m\) ustbe passed
unchanged to each sparse solver subroutine.
M odified on exit.

ER
- \(\mathbb{N}\) TEGER

Errornum ber. If no errorencountered, unchanged on
exit. If errorencountered, it is set to a non-zero
integer. Enrornum bers set.by this subroutine:
-800 : Invalid calling sequence - need to callSG SSSL first.
-899 : Printed solver statistics not supported this release.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
sgssip - R etum perm utation used by the general sparse solver.

\section*{SYNOPSIS}

SUBROUTINE SGSSRP (PERM,HANDLE, \(\mathbb{E R}\) )
\(\mathbb{N}\) TEGER PERM (*), \(\mathbb{E R}\)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

SGSSRP -Retums the perm utation used by the solver for the fill-reducing ordering.

\section*{PARAMETERS}

PERM \(\mathbb{N E Q N S}\) ) - \(\mathbb{N}\) TEGER amay
U ndefined on entry. PERM \(\mathbb{N E Q N S}\) ) is the perm utation array used by the sparse solver for the fillreducing ordering. M odified on exit.

HANDLE (150) -D OUBLE PRECIS IO N array
On entry, HANDLE (*) is an array containing
inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine.
M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Enrornum ber. If no errorencountered, unchanged on exit. If errorencountered, it is set to a non-zero
integer. Errornum bers set.by this subroutine:

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
sgsssl-Solve routine for the general sparse solver.

\section*{SYNOPSIS}

SUBROUTINE SGSSSL (NRHS,RHS,LDRHS,HANDLE, 正R)
\(\mathbb{N}\) TEGER \(\quad\) NRHS,LDRHS, \(\mathbb{E R}\)
REAL \(\quad\) RHS (LDRHS,NRHS)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

SGSSSL -Triangular solve of a factored sparse m atrix.

\section*{PARAMETERS}

NRHS - \(\mathbb{N}\) TEGER
On entry, N RH S specifies the num ber of righthand
sides to solve for. U nchanged on exit.

RHS (LDRHS,*) -REA L array
On entry, RH S (LDRHS,NRHS) contains the NRHS right hand sides. On exit, itcontains the solutions.

LDRHS - \(\mathbb{N}\) TEGER
On entry, LD RH S specifies the leading dim ension of the RH S array. U nchanged on exit.

HANDLE (150) -D OUBLE PREC ISIO N aray
O n entry, HANDLE ( \({ }^{\star}\) ) is an array containing
inform ation needed by the solver, and \(m\) ustbe passed
unchanged to each sparse solver subroutine.
M odified on exit.

ER
- \(\mathbb{N}\) TEGER

Errornum ber. If no errorencountered, unchanged on
exit. If errorencountered, it is set to a non-zero
integer. Errornum bers setby this subroutine:
-400 : Invalid calling sequence - need to callSG SSFA first.
-401 : Failure to dynam ically allocate \(m\) em ory.
-402 : NRHS < 1
-403 : NEQN S > LD RHS

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
sgssuo -U ser supplied perm utation for ordering used in the general sparse solver.

\section*{SYNOPSIS}

SUBROUTINE SGSSUO (PERM,HANDLE, \(\mathbb{E R}\) )
\(\mathbb{N}\) TEGER PERM (*), \(\mathbb{E R}\)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

SGSSUO -U ser supplied perm utation for ordering. M ust.be called afterSGSS IN () (sparse solver initialization) and before SGSSOR () (sparse solver ordering).

\section*{PARAMETERS}

PERM \(\mathbb{N} E Q N S\) ) - \(\mathbb{N}\) TEGER array
On entry, PERM (NEQNS) is a perm utation array supplied by the user for the fill-reducing ordering.
U nchanged on exit.

HANDLE (150) -D OUBLE PRECISION aray
On entry, HANDLE (*) is an array containing
inform ation needed by the solver, and \(m\) ust.be passed
unchanged to each sparse solver subroutine.
M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Errornum ber. If no errorencountered, unchanged on
exit. If emorencountered, it is set to a non-zero
integer. Errornum bers setby this subroutine:
-500 : Invalid calling sequence - need to callSG SS \(\mathbb{N}\) first.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgtoon - estim ate the reciprocal of the condition num ber of a real tridiagonal \(m\) atrix A using the LU factorization as com puted by SG TTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NESGTCON NORM,N,LOW,DIAG,UP1,UP2, \mathbb{PIVOT,ANORM,RCOND,}}\mathbf{N},
W ORK,\mathbb{N ORK2,INFO)}
CHARACTER * 1 NORM
\mathbb{NTEGER N,}\mathbb{N}FO
\mathbb{NTEGER \mathbb{PIVOT (*), IN ORK2 (*)}}\mathbf{(*)}
REALANORM,RCOND
REAL LOW (*),D IAG (*),UP1 (*),UP2 (*),W ORK (*)

```

```

        RCOND,W ORK,INORK2,INFO)
    CHARACTER * 1 NORM
INTEGER*8 N, INFO
\mathbb{NTEGER*8 \mathbb{PIVOT (*), IN ORK2 (*)}}\mathbf{(*)}
REALANORM,RCOND
REAL LOW (*),D IA G (*),UP1 (*),UP2 (*),W ORK (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE GTCON $\mathbb{N} O R M, \mathbb{N}], L O W, D I A G, U P 1, U P 2, \mathbb{P} \mathbb{I V O T}, A N O R M$, RCOND, [W ORK ], [ $\mathbb{W}$ ORK 2], [ $\mathbb{N F O}]$ )
CHARACTER (LEN=1) ::NORM
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V} O T, \mathbb{I}$ ORK2
REAL ::ANORM,RCOND

```

REAL,D \(\mathbb{M}\) ENSION (:) ::LOW ,DIAG,UP1,UP2,W ORK

SUBROUTINEGTCON_64 \(\mathbb{N} O R M, ~ \mathbb{N}], L O W, D \mathbb{A} G, U P 1, U P 2, \mathbb{P} \mathbb{I} O T, A N O R M\), RCOND, [WORK], [IW ORK2], [ \(\mathbb{N} F O])\)

CHARACTER (LEN=1) ::NORM
\(\mathbb{N}\) TEGER (8) :: N , \(\mathbb{N}\) FO
\(\mathbb{N} \operatorname{TEGER}\) (8),D \(\mathbb{M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT,\mathbb {IW}\text {ORK}2~}\)
REAL ::ANORM,RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::LOW ,DIAG,UP1,UP2,W ORK

\section*{C INTERFACE}
\#include < sunperfh>
void sgtcon (char norm , intn, float *low , float *diag, float
*up1, float*up2, int*ipijot, floatanorm, float
*rcond, int*info);
void sgtcon_64 (charnorm , long n, float *low , float *diag, float*up1, float*up2, long *ipivot, floatanorm , float*roond, long *info);

\section*{PURPOSE}
sgtoon estim ates the reciprocal of the condition num ber of a realtridiagonalm atrix A using the LU factorization as com puted by SG TTRF .

An estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND \(=1\) / (ANORM * norm (inv (A))).

\section*{ARGUMENTS}

NORM (input)
Specifies w hether the 1-norm condition num ber or the infinity-norm condition num ber is required:
= ' 'or \(\mathrm{D}^{\prime}\) : 1-norm;
\(=\mathrm{I}^{\prime}: \quad\) Infinity-norm.

N (input) The order of the m atrix \(A . N>=0\).

LOW (input)
The \((n-1) \mathrm{m}\) ultipliers that define the \(m\) atrix \(L\) from the LU factorization of \(A\) as com puted by SGTTRF.

D IA G (input)

The n diagonalelem ents of the upper triangular \(m\) atrix \(U\) from the \(L U\) factorization ofA.

UP1 (input)
The ( \(n-1\) ) elem ents of the first superdiagonal of U.

UP2 (input)
The ( \(n-2\) ) elem ents of the second superdiagonal of U.
\(\mathbb{P I V O T}\) (input)
The pivotindioes; for \(1<=i<=n\), row \(i\) of the matrix was interchanged with row PIVOT (i). IPIVOT (i) will always be either \(i\) or i+1; PIVOT (i) = iindicates a row interchange \(w\) as not required.

ANORM (input)
IfNORM = I'orD', the 1-norm of the original \(m\) atrix \(A\). IfNORM = I', the infinity-nom of the originalm atrix A .

RCOND (output)
The reciprocal of the condition number of the \(m\) atrix \(A\), com puted as RCOND \(=1 /(A N O R M * A \mathbb{N} V N M)\), where \(A \mathbb{N} V N M\) is an estim ate of the 1 -nom of \(\operatorname{inv}(A)\) com puted in this routine.

W ORK (w orkspace)
dim ension \((2 * N)\)

IV ORK 2 (w orkspace)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N N}\) FO = -i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgthr-G athers specified elem ents from \(y\) into \(x\).

\section*{SYNOPSIS}
```

SUBROUTINE SGTHR\mathbb{NZ,Y,X,NNDX)}

```

REALY(*), X (*)
\(\mathbb{N}\) TEGER NZ
\(\mathbb{N} T E G E R \mathbb{N} D X(*)\)
SUBROUTINE SGTHR_64 \(\mathbb{N} Z, Y, X, \mathbb{N} D X)\)
REALY (*), X (*)
\(\mathbb{N} T E G E R * 8 N Z\)
\(\mathbb{I N} T E G E R * 8 \mathbb{N} D X(*)\)
F95 \(\mathbb{I N}\) TERFACE
SUBROUTINE GTHR ( \(\mathbb{N} Z], Y, X, \mathbb{N} D X)\)
REAL,D IM ENSION (:) :: Y,X
\(\mathbb{I N T E G E R}:: \mathrm{NZ}\)
\(\mathbb{N}\) TEGER,D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{N} D \mathrm{X}\)
SUBROUTINE GTHR_64( \(\mathbb{N} Z], Y, X, \mathbb{N} D X)\)
REAL,D IM ENSION (:) :: Y, X
\(\mathbb{N} T E G E R(8):: N Z\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{N D X}\)

\section*{PURPOSE}

SG THR -G athers the specified elem ents from a vectory in fullstorage form into a vectorx in com pressed form. Only
the elem ents of \(y\) w hose indices are listed in indx are referenced.
do \(i=1, n\) \(x(i)=y(\) indx \((i))\)
enddo

\section*{ARGUMENTS}

N Z (input) - \(\mathbb{N}\) TEGER
\(N\) um ber of elem ents in the com pressed form .
U nchanged on exit.
\(Y\) (input)
V ectorin fullstorage form . U nchanged on exit.
X (output)
V ector in com pressed form. C ontains elem ents ofy
whose indices are listed in indx on exit.
\(\mathbb{I N D X}\) (input) - \(\mathbb{N}\) TEGER
\(V\) ector containing the indices of the com pressed
form. It is assum ed that the elem ents in \(\mathbb{N} D \mathrm{X}\) are
distinct and greater than zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgthrz -G ather and zero.

\section*{SYNOPSIS}

SUBROUTINE SGTHRZ \(\mathbb{N} Z, Y, X, \mathbb{N} D X)\)
REALY(*), X (*)
\(\mathbb{N}\) TEGER NZ
\(\mathbb{N} T E G E R \mathbb{N} D X(*)\)
SUBROUTINE SGTHRZ_64 \(\mathbb{N} Z, Y, X, \mathbb{N} D X)\)
REALY (*), X (*)
\(\mathbb{N} T E G E R * 8 N Z\)
\(\mathbb{I N} T E G E R * 8 \mathbb{N} D X(*)\)
F95 \(\mathbb{I N}\) TERFACE
SUBROUTINE GTHRZ ( \(\mathbb{N} Z], Y, X, \mathbb{N D X}\) )
REAL,D \(\mathbb{I M}\) ENSION (:) :: Y,X
\(\mathbb{I N T E G E R}:: \mathrm{NZ}\)
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{N D X}\)
SUBROUTINEGTHRZ_64(NZ],Y,X, \(\mathbb{N} D X)\)
REAL,D IM ENSION (:) :: Y, X
\(\mathbb{N} T E G E R(8):: N Z\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{I} \operatorname{ENSION(:)::\mathbb {N}D\mathrm {X}}\)

\section*{PURPOSE}

SG THRZ -G athers the specified elem ents from a vectory in fullstorage form into a vectorx in com pressed form. The
gathered elem ents ofy are set to zero. O nly the elem ents ofy w hose indices are listed in indx are referenced.
```

do i=1,n
x (i) = y (indx (i))
y(indx (i)) = 0
enddo

```

\section*{ARGUMENTS}

N Z (input) - \(\mathbb{N}\) TEGER
N um ber of elem ents in the com pressed form .
U nchanged on exit.

Y (input/output)
V ector in fullstorage form. G athered elem ents are setto zero.
X (output)
V ector in com pressed form. C ontains elem ents ofy w hose indices are listed in indx on exit.
\(\mathbb{N} D X\) (input) - \(\mathbb{N} T E G E R\)
V ector containing the indiges of the com pressed form. It is assum ed that the elem ents in \(\mathbb{N D} X\) are distinctand greater than zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgtrifs - im prove the com puted solution to a system of linear equations when the coefficientm atrix is tridiagonal, and provides errorbounds and backw ard error estim ates for the solution

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SGTRFS (TRANSA,N,NRHS,LOW ,D IA G,UP,LOW F,D IA GF,UPF1,}
UPF2,\mathbb{PIVOT,B,LDB,X,LDX,FERR,BERR,W ORK,W ORK 2,INFO)}
CHARACTER * 1 TRANSA
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*),W ORK2(*)}
REAL LOW (*),DIAG (*), UP (*), LOW F (*), D IAGF (*), UPF1 (*),
UPF2 (*),B (LDB,*),X (LDX ,*),FERR (*),BERR (*),W ORK (*)
SU BROUT\mathbb{NE SGTRFS_64(TRANSA,N,NRHS,LOW ,D IAG,UP,LOW F,D IAGF,}
UPF1,UPF2,\mathbb{PIVOT,B,LDB,X,LDX,FERR,BERR,W ORK,W ORK2,}
\mathbb{NFO)}
CHARACTER * 1 TRANSA
\mathbb{NTEGER*8N,NRHS,LDB,LDX,}\mathbb{N}FO
INTEGER*8 \mathbb{PIVOT (*),W ORK2 (*)}
REAL LOW (*),DIAG (*), UP (*), LOW F (*), D IAGF (*), UPF1 (*),
UPF2 (*),B (LDB,*),X (LDX ,*),FERR (*),BERR (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GTRFS ([TRANSA], \(\mathbb{N}], \mathbb{N} R H S], L O W, D \mathbb{A} G, U P, L O W F, D \mathbb{A} G F\),
 [W ORK2], [ \(\mathbb{N F O}\) ])
\(\mathbb{N}\) TEGER : \(: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} \mathrm{B}, \mathrm{LD} \mathrm{X}, \mathbb{N} \mathrm{FO}\)
\(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}, \mathrm{W} O R K 2\)
REAL, D \(\mathbb{M} E N S I O N\) (:) :: LOW , DIAG, UP, LOW F, D IAGF, UPF1,
UPF2,FERR, BERR, W ORK
REAL,D \(\mathbb{M}\) ENSION (: : : : : B , X

SU BROUTINE GTRFS_64 ([TRANSA ], \(\mathbb{N}], \mathbb{N} R H S\) ],LOW, D IA G, UP, LOW F, D \(\mathbb{I A} G F, U P F 1, U P F 2, \mathbb{P} \mathbb{I} O T, B,[L D B], X,[L D X], F E R R, B E R R,[W O R K]\), [ W ORK2], [ \(\mathbb{N} \mathrm{FO}\) ])

CHARACTER (LEN=1) ::TRANSA
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDB,LDX , \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T, W\) ORK2
REAL, D \(\mathbb{M}\) ENSION (:) ::LOW , DIAG, UP, LOW F, D IAGF, UPF1,
UPF2,FERR, BERR, W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::B,X

\section*{C INTERFACE}
\#include <sunperfh>
void sgtrfs (chartransa, intn, intnrhs, float*low, float *diag, float *up, float *low f, float *diagf, float *upfl, float*upf2, int *ipivot, float *b, int ldb, float *x, int ldx, float *ferr, float *berr, int*info);
void sgtrfs_64 (chartransa, long n, long nrhs, float *low, float * diag, float *up, float*low f, float *diagf, float *upfl, float*upf2, long *ípivot, float *b, long ldb, float*x, long ldx, float *ferr, float *berrr, long *info);

\section*{PURPOSE}
sgtrfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is tridiagonal, and provides emorbounds and backw ard error estim ates for the solution.

\section*{ARGUMENTS}

TRANSA (input)
Specifies the form of the system of equations:
\(=N^{\prime}: A * X=B \quad\) Notranspose)
\(=T\) ': A**T * \(\mathrm{X}=\mathrm{B} \quad\) (Transpose)
\(=C\) : \(A * * H * X=B \quad\) (C onjugate transpose \(=T\) ran -
spose)

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE .

N (input) The order of the matrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of them atrix B. NRHS \(>=0\).

LOW (input)
The \((n-1)\) subdiagonalelem ents of \(A\).

D IA G (input)
The diagonalelem ents ofA.

UP (input)
The ( \(n-1\) ) superdiagonalelem ents ofA .
LOW F (input)
The \((n-1) \mathrm{m} u l t i p l e r s\) that define the \(m\) atrix \(L\) from the LU factorization of \(A\) as com puted by SG TTRF .

D IA GF (input)
Then diagonalelem ents of the upper triangular \(m\) atrix \(U\) from the \(L U\) factorization ofA.

UPF1 (input)
The ( \(n-1\) ) elem ents of the first superdiagonal of U.

UPF2 (input)
The ( \(n-2\) ) elem ents of the second superdiagonal of U.

PIVOT (input)
The pivotindices; for \(1<=i<=n\), row \(i\) of the \(m\) atrix \(w\) as interchanged \(w\) th row \(\mathbb{P I V O T}\) (i). IPIVOT (i) will always be either \(i\) or i+1;
PIVOT (i) = iindicates a row interchange was not required.

B (input) The righthand side m atrix \(B\).
LD B (input)
The leading dim ension of the anay B . LD B >= \(\max (1, \mathbb{N})\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by SG TTRS. On exit, the im proved solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the array X . LD X >= \(\max (1, N)\).

FERR (output)
The estim ated forw ard error bound for each solution vector \(X(\neg\) (the \(j\) th colum \(n\) of the solution matrix \(X)\). IfXTRUE is the true solution comesponding to \(X(\mathcal{O}), \operatorname{FERR}(\mathcal{O})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(\mathrm{X}(\mathcal{\nu})-\mathrm{X}\) TRU E ) divided by the \(m\) agnitude of the largestelem ent in \(X(j)\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emror.

BERR (output)
The com ponentw ise relative backw ard error of each solution vectorX ( \(j\) ) (ie., the sm allest relative change in any elem entof \(A\) or \(B\) thatm akes \(X(\mathcal{J})\) an exactsolution).

W ORK (w orkspace)
dim ension \((3 * N)\)
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

sgtsv - solve the equation A *X = B,

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGTSV N,NRHS,LOW,DIAG,UP,B,LDB,INFO)}
\mathbb{NTEGERN,NRHS,LDB,NNFO}
REAL LOW (*),DIAG (*),UP (*),B (LDB,*)
SU BROUT\mathbb{NE SGTSV_64 N ,NRHS,LOW ,D IAG,UP,B,LDB,INFO)}
INTEGER*8N,NRHS,LDB,INFO
REAL LOW (*),DIAG (*),UP (*),B (LDB,*)

```

\section*{F95 INTERFACE}

SUBROUTINEGTSV ( \(\mathbb{N}], \mathbb{N R H S}], L O W, D \mathbb{I A G}, \mathrm{UP}, \mathrm{B},[\mathrm{LDB}],[\mathbb{N} F O]\) )
\(\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O\)
REAL,D IM ENSION (:) ::LOW ,D IAG,UP
REAL,D \(\mathbb{M}\) ENSION (:,:) ::B
SU BROUTINE GTSV_64 ( \(\mathbb{N}\) ], \(\mathbb{N} R H S], L O W, D \mathbb{A} G, U P, B,[L D B],[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LD B, \(\mathbb{N} F O\)
REAL,D IM ENSION (:) ::LOW ,D IAG,UP
REAL,D \(\mathbb{M}\) ENSION (:,:) ::B

\section*{C INTERFACE}
\#include <sunperfh>
void sgtsv (intn, intnrhs, float * low , float *diag, float
*up, float *b, int ldb, int*info);
void sgtsv_64 (long n, long nrhs, float *low , float *diag, float *up, float *b, long ldb, long *info);

\section*{PURPOSE}
sgtsv solves the equation
where \(A\) is an \(n\) by \(n\) tridiagonalm atrix, by \(G\) aussian elm ination \(w\) ith partialpivoting.

N ote that the equation \(A\) *X \(=\mathrm{B}\) m ay be solved by interchanging the order of the argum ents \(D U\) and \(D L\).

\section*{ARGUMENTS}

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
NRHS (input)
The num berof righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS >=0.

LOW (input/output)
On entry, LOW m ust contain the ( \(n-1\) ) sub-diagonal elem ents ofA.

On exit, LOW is overw ritten by the ( \(n-2\) ) elem ents of the second super-diagonal of the upper triangularm atrix \(U\) from the \(L U\) factorization of \(A\), in LOW (1), .., LOW (n-2).

D IA G (input/output)
On entry, D IA G m ustcontain the diagonal elem ents of A.

On exit, D IA G is overw rilten by the \(n\) diagonal elem ents of \(U\).

UP (input/output)
O \(n\) entry, UP m ust contain the ( \(n-1\) ) super-diagonal elem ents of .

On exit, UP is overw ritten by the ( \(n-1\) ) elem ents of the first super-diagonal of \(U\).

B (input/output)
On entry, the N by NRH S m atrix of righthand side
matrix B. On exit, if \(\mathbb{N} F O=0\), the \(N\) by NRHS solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i, U(i, i)\) is exactly zero, and the solution has notbeen com puted. The factorization has notbeen com pleted unless \(i=N\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgtsvx - use the LU factorization to com pute the solution to a realsystem of linearequations \(A * X=B\) orA \({ }^{*}{ }^{*} T\) * \(X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NESGTSVX (FACT,TRANSA,N,NRHS,LOW,DIAG,UP,LOW F,DIAGF,}
UPF1,UPF2,\mathbb{PIVOT,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,}
W ORK2,\mathbb{NFO)}
CHARACTER * 1FACT,TRANSA
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*),W ORK2(*)}
REALRCOND
REAL LOW (*),DIAG (*), UP (*), LOW F (*), D IAGF (*), UPF1 (*),
UPF2 (*),B (LDB,*),X (LDX,*),FERR (*),BERR (*),W ORK (*)
SUBROUT\mathbb{NE SGTSVX_64(FACT,TRANSA,N,NRHS,LOW ,D IAG,UP,LOW F,}
D IAGF,UPF1,UPF2,\mathbb{PIVOT,B,LDB,X,LDX,RCOND,FERR,BERR,}
WORK,W ORK 2, INFO)
CHARACTER * 1 FACT,TRANSA
INTEGER*8N,NRHS,LDB,LDX,\mathbb{NFO}
INTEGER *8 \mathbb{PIVOT (*),W ORK2 (*)}
REAL RCOND
REAL LOW (*),DIAG (*), UP (*), LOW F (*), DIAGF (*), UPF1 (*),
UPF2 (*),B (LDB ,*),X (LDX ,*),FERR (*),BERR (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GTSVX (FACT, [TRANSA], \(\mathbb{N}], \mathbb{N} R H S]\), LOW, D IAG, UP, LOW F, D IA GF, UPF1, UPF2, \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LD} \mathrm{B}], \mathrm{X},[\mathrm{LD} \mathrm{X}], R C O N D, F E R R, B E R R\), [W ORK], [W ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::FACT,TRANSA
\(\mathbb{N}\) TEGER : \(: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LDB}, \mathrm{LDX}, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{I} O T, W O R K 2\)
REAL ::RCOND
REAL, D \(\mathbb{M}\) ENSION (:) :: LOW , DIAG, UP, LOW F, DIAGF, UPF1,
UPF2,FERR, BERR, WORK
REAL,D \(\mathbb{M}\) ENSION (: :) ::B,X

SUBROUTINE GTSVX_64 (FACT, [TRANSA], \(\mathbb{N}], \mathbb{N R H S}], L O W, D \mathbb{A} G, U P, L O W E\), D \(\mathbb{A} G F, U P F 1, U P F 2, \mathbb{P} \mathbb{V} O T, B,[L D B], X,[L D X], R C O N D, F E R R, B E R R\), \([\mathbb{W} O R K],[W O R K 2],[\mathbb{N F O}])\)

CHARACTER (LEN=1) :: FACT,TRANSA
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T, W\) ORK 2
REAL ::RCOND
REAL, D \(\mathbb{M}\) ENSION (:) :: LOW , DIAG, UP, LOW F, DIAGF, UPF1,
UPF2,FERR, BERR, W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::B,X

\section*{C INTERFACE}
\#include <sunperfh>
void sgtsvx (char fact, chartransa, intn, int nrhs, float
*low, float *diag, float*up, float*low f, float
*diagf, float *upfl, float *upf2, int *ipivot, float *b, int ldb, float *x, int ldx, float *rcond, float * ferr, float *berr, int *info);
void sgtsvx_64 (char fact, chartransa, long n, long nrhs, float *low, float*diag, float*up, float*low f, float *diagf, float *upfl, float *upf2, long *ípivot, float *b, long ldlo, float *x, long ldx, float *rcond, float *ferr, float *berr, long *info);

\section*{PURPOSE}
sgtsvx uses the LU factorization to com pute the solution to a realsystem of linearequations \(A * X=B\) orA \(* * T * X=B\), \(w\) here \(A\) is a tridiagonalm atrix of order \(N\) and \(X\) and \(B\) are N -by-N R H S m atrices.

Emorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT \(=\mathrm{N}\) ', the LU decom position is used to factor the
\(m\) atrix \(A\)
as \(A=L * U\),where \(L\) is a product of perm utation and unitlow er
bidiagonal \(m\) atrices and \(U\) is upper triangular \(w\) ith nonzeros in
only the \(m\) ain diagonal and first tw o superdiagonals.
2. If som e \(U(i, i)=0\), so that \(U\) is exactly singular, then the routine
retums w ith \(\mathbb{N N F O}=\) i. O therw ise, the factored form of \(A\) is used
to estim ate the condition num ber of the \(m\) atrix A. If the reciprocal of the condition num ber is less than m achine precision,
\(\mathbb{N} F O=N+1\) is retumed as a w aming, but the routine stillgoes on
to solve for \(X\) and com pute error bounds as described below.
3.The system ofequations is solved forX using the factored form of A.
4. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for it.

\section*{ARGUMENTS}
```

FACT (input)
Specifies w hether ornot the factored form of $A$ has been supplied on entry $.=F ': L O W$ F , D IA GF, $\mathrm{UPF} 1, \mathrm{UPF} 2$, and $\mathbb{P} \mathbb{I V O T}$ contain the factored form of $A$; LOW , DIAG, UP,LOW F, DIAGF, UPF1, UPF2 and $\mathbb{P}$ IV O T w illnotbem odified. $=\mathrm{N}$ ': The m atrix
w ill be copied to LOW F , D IA GF, and UPF1 and factored.
TRAN SA (input)
Specifies the form of the system ofequations:
$=N$ : A * X $=\mathrm{B} \quad$ (N o transpose)
$=T$ ': $A * * T * X=B \quad$ ( ranspose)
$=C^{\prime}: A * * H * X=B \quad$ (C onjugate transpose $=T$ ran spose)

```

TRANSA is defaulted to \(N\) 'forF95 \(\mathbb{I N T E R F A C E . ~}\)

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS \(>=0\).

LOW (input)
The \((n-1)\) subdiagonalelem ents of \(A\).
D IA G (input)
The \(n\) diagonalelem ents of A.

UP (input/output)
The ( \(n-1\) ) superdiagonalelem ents of .
LOW F (input/output)
IfFACT = F', then LOW F is an inputargum ent and on entry contains the ( \(n-1\) ) multipliers that define the \(m\) atrix \(L\) from the \(L U\) factorization of \(A\) as com puted by SG TTRF .

IfFACT \(=\mathrm{N}^{\prime}\), then LOW F is an output argum ent and on exit contains the ( \(n-1\) ) m ultipliers that define them atrix \(L\) from the LU factorization of A.

D IA GF (input/output)
If FACT = F ', then D IAGF is an input argum ent and on entry contains the \(n\) diagonalelem ents of the upper triangularm atrix \(U\) from the LU factorization ofA.

IfFACT \(=N\) ', then \(D I A G F\) is an output argum ent and on exit contains the \(n\) diagonalelem ents of the uppertriangularm atrix \(U\) from the \(L U\) factorization ofA.

UPF1 (input/output)
IfFACT = \(\mathrm{F}^{\prime}\), then UPF 1 is an inputargum ent and on entry contains the ( \(n-1\) ) elem ents of the first superdiagonalofU .

IfFACT \(=\mathrm{N}^{\prime}\), then UPF1 is an output argum ent and on exit contains the ( \(n-1\) ) elem ents of the first superdiagonalof \(U\).

UPF2 (input/output)
IfFACT=F', then UPF2 is an input argum ent and on entry contains the ( \(n-2\) ) elem ents of the second superdiagonalofU .

IfFACT \(=N\) ', then UPF2 is an outputargum ent and
on exit contains the ( \(n-2\) ) elem ents of the second superdiagonalof \(U\).

PIVOT (input/output)
If \(F A C T=F '\), then \(\mathbb{P I V O T}\) is an input argum ent and on entry contains the pivotindioes from the LU factorization of A as com puted by SG TTRF.

IfFACT = N', then \(\mathbb{P} \mathbb{I V O T}\) is an output argum ent and on exit contains the pivot indices from the LU factorization of ; row iof the \(m\) atrix \(w\) as interchanged w ith row \(\mathbb{P} \mathbb{I V O T}\) (i). \(\mathbb{P} \mathbb{I V O T}\) (i) w illalw ays be either ior i+ \(1 ; \mathbb{P} \mathbb{I V O T}(i)=\) indicates a row interchange \(w\) as not required.

B (input) The N by -N RH S righthand side m atrix B .

LD B (input)
The leading dim ension of the array B . LD B >= \(\max (1, N)\).

X (output)
If \(\mathbb{N F O}=0\) or \(\mathbb{N} F O=\mathrm{N}+1\), the N -by-NRH S solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the aray X . LD X >= \(\max (1, N)\).

\section*{RCOND (output)}

The estim ate of the reciprocal condition num berof the \(m\) atrix \(A\). IfRCOND is less than the \(m\) achine precision (in particular, if RCOND \(=0\) ), the \(m\) atrix is singular to working precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO > 0.

FERR (output)
The estim ated forw ard enrorbound for each solution vector \(X(\mathcal{)}\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(\mathrm{X}(\mathcal{)}\), FERR ( \()\) ) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(\mathrm{X}(\mathcal{1})\)-X TRU E) divided by the magnitude of the largestelem entin \(X(\mathcal{)}\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

BERR (output)
The com ponentw ise relative backw ard error of each
solution vectorX (i) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension ( \(3 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue
> 0 : if \(\mathbb{N F O}=i\), and \(i\) is
\(<=\mathrm{N}: \mathrm{U}(i, i)\) is exactly zero. The factorization has notbeen com pleted unless \(i=N\), but the factor \(U\) is exactly singular, so the solution and error bounds could notbe com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1: \mathrm{U}\) is nonsingular, but RCOND is less than \(m\) achine precision, \(m\) eaning that the m atrix is singular to working precision. N evertheless, the solution and errorbounds are com puted because there are a num ber of situations where the com puted solution can be \(m\) ore accurate than the value of RCOND w ould suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sgttrf-com pute an LU factorization of a real tridiagonal \(m\) atrix \(A\) using elim ination \(w\) th partialpivoting and row interchanges

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SGTTRF N,LOW ,D IAG,UP1,UP2, PIVOT,INFO)}
\mathbb{NTEGER N,\mathbb{NFO}}\mathbf{N}\mathrm{ (})
INTEGER \mathbb{PIVOT (*)}
REAL LOW (*),DIAG (*),UP1 (*),UP2 (*)

```

```

\mathbb{NTEGER*8N,\mathbb{NFO}}\mathbf{N}\mathrm{ ( }
NNTEGER*8 \mathbb{PIVOT (*)}
REAL LOW (*),D IAG (*),UP1 (*),UP2 (*)
F95 INTERFACE

```

```

    \mathbb{NTEGER ::N,\mathbb{NFO}}0=0,
    \mathbb{NTEGER,D IM ENSION (:) :: \mathbb{PIVOT}}\mathbf{T}\mathrm{ (%)}
    REAL,D\mathbb{M ENSION (:) ::LOW ,D IAG,UP1,UP2}
    ```

```

    \mathbb{NTEGER (8)::N,\mathbb{NFO}}\mathbf{N}=0
    ```

```

    REAL,D IM ENSION (:) ::LOW ,D IAG,UP1,UP2
    \#include <sunperfh>
void sgttrf(intn, float *low, float *diag, float *up1, float *up2, int*ipivot, int*info);
void sgturf_64 (long n, float*low , float *diag, float *up1, float *up2, long *ipìvot, long *info);

## PURPOSE

sgturfcom putes an LU factorization of a real tridiagonal $m$ atrix A using elim ination $w$ ith partialpivoting and row interchanges.

The factorization has the form

$$
A=L \star U
$$

where $L$ is a productofperm utation and unit low er bidiagonalm atrioes and $U$ is uppertriangularw ith nonzeros in only the $m$ ain diagonal and firsttw o superdiagonals.

## ARGUMENTS

N (input) The order of the $m$ atrix $A$.

LOW (input/output)
On entry, LOW m ustcontain the ( $\mathrm{n}-1$ ) sub-diagonal elem ents ofA.

On exit, LOW is overw rilten by the ( $n-1$ ) multipliers that define the $m$ atrix $L$ from the $L U$ factorization of A.

D IA G (input/output)
O n entry, D IA G m ust contain the diagonal elem ents of A.

On exit, D IA G is overw rilten by the $n$ diagonal elem ents of the upper triangularm atrix $U$ from the LU factorization of A.

UP1 (input/output)
On entry, UP1 must contain the $(n-1)$ superdiagonalelem ents of .

O n exit, UP1 is overw ritten by the ( $\mathrm{n}-1$ ) elem ents of the first super-diagonal of $U$.

UP2 (output)
On exit, UP2 is overw rilten by the ( $n-2$ ) elem ents of the second super-diagonalofU.

IPIVOT (output)
The pivotindioes; for $1<=i<=n$, row $i$ of the matrix was interchanged w th row $\mathbb{P I V O T}$ (i). IPIVOT (i) will always be either $i$ or i+1; PIVOT (i) = iindicates a row interchange was not required.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0$ : if $\mathbb{N} F O=-k$, the $k$-th argum enthad an illegalvalue
> 0 : if $\mathbb{N} F O=k, U(k, k)$ is exactly zero. The factorization has been com pleted, but the factor $U$ is exactly singular, and division by zero will occur if it is used to solve a system ofequations.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sgturs - solve one of the system sofequations $A * X=B$ or A * $\mathrm{X}=\mathrm{B}$,

## SYNOPSIS

```
SU BROUT\mathbb{NE SGTTRS (TRANSA,N,NRHS,LOW ,D IA G,UP1,UP2, PP\mathbb{NOT,B,}}\mathbf{N},\textrm{N},
    LDB, \mathbb{NFO)}
CHARACTER * 1 TRANSA
\mathbb{NTEGERN,NRHS,LDB,INFO}
\mathbb{NTEGER \mathbb{PIVOT (*)}}\mathbf{(})
REAL LOW (*),D IAG (*),UP1 (*),UP2 (*),B (LD B ,*)
SU BROUT\mathbb{NE SGTTRS_64 (TRANSA,N,NRHS,LOW ,D IAG,UP1,UP2, \mathbb{PIVOT,B,}}\mathbf{I}=,
    LDB,\mathbb{NFO)}
```

CHARACTER * 1 TRANSA
$\mathbb{N}$ TEGER*8N,NRHS,LDB, $\mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{V} O T(*)$
REAL LOW (*), DIAG (*), UP1 (*), UP2 (*), B (LDB,*)

## F95 INTERFACE

SUBROUTINE GTTRS ([TRANSA], $\mathbb{N}], \mathbb{N} R H S], L O W, D \mathbb{I A}, \operatorname{UP} 1, U P 2, \mathbb{P} \mathbb{I} O T$, B, [LDB], [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1) ::TRANSA
$\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$
REAL,D $\mathbb{M}$ ENSION (:) ::LOW ,DIAG,UP1,UP2
REAL,D $\mathbb{I M}$ ENSION (:,:) ::B
SU BROUTINE GTTRS_64 ([TRANSA], $\mathbb{N}], \mathbb{N} R H S], L O W, D \mathbb{A} G, U P 1, U P 2$,
$\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}],[\mathbb{N} F O])$

CHARACTER (LEN=1) ::TRANSA
$\mathbb{N}$ TEGER (8) :: N , NRHS,LD B, $\mathbb{N}$ FO
$\mathbb{N} \operatorname{TEGER}$ (8), D $\mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$
REAL,D $\mathbb{M}$ ENSION (:) ::LOW ,DIAG ,UP1, UP2
REAL,D $\mathbb{I}$ ENSION (:,: : : B

## C INTERFACE

\#include <sunperfh>
void sgturs (chartransa, intn, intnrhs, float*low, float *diag, float*up1, float*up2, int*ipivot, float *b, int ldb, int *info);
void sgttrs_64 (char transa, long n, long nrhs, float *low , float *diag, float *up1, float *up2, long *ipivot, float *b, long ldb, long *info);

## PURPOSE

sgtters solves one of the system s of equations
$A * X=B$ or $A * X=B, w$ th a tridiagonalm atrix $A$ using the LU factorization com puted by SG TTRF.

## ARGUMENTS

TRANSA (input)
Specifies the form of the system of equations. =
N': A * X = B ( $\circ$ otranspose)
$=T$ ': A * $\mathrm{X}=\mathrm{B} \quad$ (Transpose)
$=C$ ': A * $\mathrm{X}=\mathrm{B}$ (C onjugate transpose $=$ Transpose)

TRANSA is defaulted to $N$ 'forF $95 \mathbb{I N}$ TERFACE.
N (input) The order of the $m$ atrix A.

NRHS (input)
The num ber of righthand sides, i.e., the num ber
of colum ns of them atrix B. NRHS $>=0$.
LOW (input)
The $(n-1) m$ ultipliers that define the $m$ atrix $L$ from the LU factorization ofA.

D IA G (input)
Then diagonalelem ents of the upper triangular
$m$ atrix $U$ from the $L U$ factorization ofA .

## UP1 (input)

The $(n-1)$ elem ents of the first super-diagonal of U .

UP2 (input)
The ( $n-2$ ) elem ents of the second super-diagonal of U.

IPIVOT (input)
The pivotindioes; for $1<=i<=n$, row $i$ of the $m$ atrix $w a s$ interchanged $w$ th row $\mathbb{P} \mathbb{I V O T}(i)$. IPIVOT (i) w ill always be either $i$ or i+1; IPIVOT (i) = iindicates a row interchange was not required.

B (input/output)
O n entry, the m atrix of righthand side vectors B . On exit, B is overw rilten by the solution vectors X .

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.

INFO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the i-th argum enthad an illegalvałue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

shgeqz - im plem enta single-/double-shiftversion of the Q Z
$m$ ethod for finding the generalized eigenvalues $\mathrm{w}(\boldsymbol{j})=(\mathrm{A}$ LPHAR $(\boldsymbol{j})+\mathrm{i}$ A LPHA $(\mathcal{j})$ )BETAR $(\mathcal{j})$ of the equation $\operatorname{det}(A-w(i) B)=0$ In addition, the pairA, B may be reduced to generalized Schur form

## SYNOPSIS

```
SUBROUT\mathbb{NE SHGEQZ (JOB,COMPQ,COMPZ,N, HO, HHI,A,LDA,B,LDB,}
    A LPHAR,ALPHA , BETA,Q,LDQ,Z,LD Z,W ORK,LWORK,INFO)
```

CHARACTER * $1 \mathrm{JOB}, \mathrm{COMPQ}, \mathrm{COMPZ}$
$\mathbb{N} T E G E R N, \mathbb{L} O, \mathbb{H} I, L D A, L D B, L D Q, L D Z, L W O R K, \mathbb{N} F O$

Q (LD Q , $)^{\prime}$, Z (LD Z , *), W ORK (*)
SU BROUTINE SHGEQZ_64 (JOB,COM PQ, COMPZ,N, $\mathbb{L} O, \mathbb{H} I, A, L D A, B, L D B$,
A LPHAR,A LPHA I, BETA, $Q, L D Q, Z, L D Z, W O R K, L W O R K, \mathbb{N} F O$ )
CHARACTER * $1 \mathrm{JOB}, \mathrm{COMPQ}, \mathrm{COMPZ}$
$\mathbb{N} T E G E R * 8 N, \mathbb{L O}, \mathbb{H} I, L D A, L D B, L D Q, L D Z, L W O R K, \mathbb{N} F O$
REAL A (LDA ,*), B (LDB,*), ALPHAR (*), ALPHAI(*), BETA (*),
$\mathrm{Q}\left(\mathrm{LD} Q,^{\star}\right), \mathrm{Z}(\mathrm{LD} Z, \star), \mathrm{W} O R K(\star)$

F95 INTERFACE
SU BROUTINE HGEQZ (JOB , COMPQ, COMPZ, $\mathbb{N}], \mathbb{L} O, \mathbb{H} I, A,[L D A], B,[L D B]$, ALPHAR,ALPHAI, BETA, $Q,[L D Q], Z,[L D Z],[W O R K],[L W O R K],[\mathbb{N} F O])$

CHARACTER (LEN=1):: JOB,COMPQ,COMPZ
$\mathbb{N} T E G E R:: N, \mathbb{L O}, \mathbb{H} I, L D A, L D B, L D Q, L D Z, L W O R K, \mathbb{N} F O$
REAL,D $\mathbb{I}$ ENSION (:) ::ALPHAR,ALPHAI,BETA,W ORK

REAL,D $\mathbb{I}$ ENSION (: : : : : A, B, Q, Z

SU BROUTINE H GEQ Z_64 (OB , COMPQ, COMPZ, $\mathbb{N}], \mathbb{L} O, \mathbb{H} I, A,[L D A], B$, [LD B ], A LPHAR, ALPHAI, BETA, Q , [LDQ ], Z, [LD Z], [WORK], [LWORK], [ $\mathbb{N} \mathrm{FO}]$ )

CHARACTER (LEN=1): : JOB, COMPQ, COMPZ
$\mathbb{N}$ TEGER (8) :: N , ILO , $\mathbb{H} I, L D A, L D B, L D Q, L D Z, L W O R K, \mathbb{N} F O$
REAL, D $\mathbb{M} E N S I O N(:):: A L P H A R, A L P H A I, B E T A, W O R K$
REAL,D $\mathbb{M} E N S I O N(:,:): A, B, Q, Z$

## C INTERFACE

\#include <sunperfh>
void shgeqz (char job, char com pq, char com pz, int $n$, int
ilo, int ini, float *a, intlda, float*b, int ldlb, float * alphar, float *alphai, float *beta, float * $q$, int ldq, float * $z$, int ld $z$, int *info);
void shgeqz_64 (char job, charcom pq, char com pz, long n, long ilo, long ini, float*a, long lda, float*b, long lalo, float *ahphar, float *alphai, float *beta, float *q, long ldq, float*z, long ldz, long *info);

## PURPOSE

shgeqz im plem ents a single-/double-shiftversion of the Q Z $m$ ethod for finding the generalized eigenvalues B is upper triangular, and A isblock upper triangular, w here the diagonal blocks are either 1 foy-1 or 2 foy- 2 , the 2 foy- 2 blocks having com plex generalized eigenvalues (see the description of the argum entJOB.)

If $J O B=S$ ', then the pair $(A, B)$ is sim ultaneously reduced to Schur form by applying one orthogonaltranform ation (usually called Q ) on the left and another (usually called Z) on the right. The $2-$ boy -2 uppertriangular diagonalblocks ofB corresponding to 2 -by-2 blocks of A w illbe reduced to positive diagonal $m$ atrices. ( $1 . e .$, if $A(j+1, j)$ is non-zero, then $B(j+1, j)=B(j \mathfrak{j}+1)=0$ and $B(j)$ and $B(j+1, j+1) w$ ill be posilive.)

If $J O B=E$ ', then ateach iteration, the sam e transform ations are com puted, but they are only applied to those parts of A and $B$ which are needed to com pute $A$ LPH AR , ALPH A I, and BETAR.

If $J O B=S$ 'and $C O M P Q$ and $C O M P Z$ are $V^{\prime}$ or $I$ ', then the orthogonal transform ations used to reduce ( $\mathrm{A}, \mathrm{B}$ ) are accum u-
lated into the arays $Q$ and $Z$ s.t.:
(in) A (in) Z (in) ${ }^{\star}=\mathrm{Q}$ (out) A (out) Z (out)*

Ref: C B.M oler \& G IN . Stew art, "A n A lgorithm for Generalized M atrixigenvalue Problem s", SIAM J. Num er. A nal, 10 (1973) р. 241-256.

## ARGUMENTS

JOB (input)
$=E$ ': com pute only A LPHAR,A LPHAI, and BETA. A and $B$ w illnotnecessarily be put into generalized Schurform . = $S^{\prime}:$ putA and B into generalized Schur form, as wellas com puting A LPHAR, A LPHA I, and BETA .

COMPQ (input)
$=\mathrm{N}$ : do notm odify Q .
$=\mathrm{V}$ ':m ultiply the anay Q on the right by the transpose of the orthogonaltranform ation that is applied to the left side of A and B to reduce them to Schur form . = 'I': like C OM PQ = V ', except that Q w illbe initialized to the identity first.

COMPZ (input)
$=\mathrm{N}$ ': do notm odify Z.
$=\mathrm{V}$ ':m ultiply the array Z on the right by the orthogonal tranform ation that is applied to the rightside of $A$ and $B$ to reduce them to Schur form . = 'I': like C OM PZ = V ', except thatZ will be initialized to the identily first.

N (input) The order of the $m$ atrices $A, B, Q$, and $Z . N>=0$.

IIO (input)
It is assum ed thatA is already upper triangular in row s and colum ns $1: \mathbb{H} O-1$ and $\mathbb{H} \mathrm{I}+1 \mathrm{~N} .1<=\mathbb{L} 0$ $<=\mathbb{H} I<=N$, if $N>0 ; \mathbb{H}=1$ and $\mathbb{H} I=0$, if $N=0$.

IH I (input)
See the description of IIO .

A (input) On entry, the $N$ boy-N upper H essenberg $m$ atrix A.
Elem ents below the subdiagonalm ustbe zero. If $\mathrm{JOB}=\mathrm{S}$ ', then on exit $A$ and $B \mathrm{w}$ ill have been sim ultaneously reduced to generalized Schur form . If $J O B=E$ ', then on exitA w ill have been destroyed. The diagonalblocks w ill.be corect, but
the off-diagonalportion $w$ illbe $m$ eaningless.

LD A (input)
The leading din ension of the array A. LD A $>=\max$ ( $1, \mathrm{~N})$.
$B$ (input) On entry, the $N$ boy -N upper triangular $m$ atrix $B$. Elem ents below the diagonalm ustbe zero. 2 -by- 2 blocks in $B$ corresponding to 2 -by-2 blocks in A w illbe reduced to positive diagonal form . (I.e., if $(j+1, \gamma)$ is non-zero, then $B(j+1, j)=B(j j+1)=0$ and $B(j)$ ) and $B(j+1, j+1) w$ illlbe positive.) If JO $B=S$ ', then on exit $A$ and $B$ will have been sim ultaneously reduced to Schur form. If JO B=E', then on exith w illhave been destroyed. Elem ents corresponding to diagonal blocks of A w illlbe conect, but the off-diagonal portion w ill be m eaningless.

LD B (input)
The leading dim ension of the array $B . L D B>=m a x($ 1,N ).

ALPHAR (output)
A LPHAR ( $1 \mathbb{N}$ ) w illlbe set to realparts of the diagonalelem ents of $A$ thatw ould result from reducing $A$ and $B$ to Schur form and then further reducing them both to triangular form using unitary transform ations s.t.the diagonalof B was nonnegative real. Thus, if A ( $j, j$ is in a 1 by-1 block (i.e., $A(j+1, j)=A(j j+1)=0)$, then A LPHAR $(\mathcal{j})=A(j)$. N ote that the (realor com plex) values (ALPHAR ( $\mathcal{j}$ ) $+\mathrm{i}^{\star} A \operatorname{LPHAI}(\mathcal{j})$ )BETA ( $\mathcal{j}$, $j 1, \ldots, N$, are the generalized eigenvalues of the $m$ atrix pencill $-w B$.

## A LPHA I (output)

A LPH A I( $1: \mathbb{N}$ ) w illbe set to im aginary parts of the diagonal elem ents of A that would result from reducing $A$ and $B$ to Schur form and then further reducing them both to triangular form using unitary transform ations s.t. the diagonal of $B \mathrm{w}$ as non-negative real. Thus, ifA ( 7 ) is in a 1-by-1 block (i.e., $A(j+1, j)=A(j j+1)=0)$, then A LPHAR ( $\mathcal{j}=0$. N ote that the (real orcomplex)
 $\dot{F} 1, \ldots, N$, are the generalized eigenvalues of the $m$ atrix pencilA $-w B$.
$\operatorname{BETA}(1 \mathbb{N})$ w illbe set to the (real) diagonal ele$m$ ents of $B$ thatw ould result from reducing $A$ and $B$ to Schur form and then further reducing them both to triangular form using unitary transform ations s.t. the diagonal of $B$ was non-negative real. Thus, if A (j) is in a 1-by-1 block (ie., $A(j+1, j)=A(j j+1)=0)$, then BETA $(j=B(j)$. N ote that the (real or complex) values (A LPHAR ( $\mathcal{j}$ ) i*ALPHAI ( $\mathcal{I}$ ) ABETA ( $\mathcal{j}$, $\bar{j} 1, \ldots, N$, are the generalized eigenvalues of the $m$ atrix pencil $A-w B$. N ote thatBETA ( $1: \mathbb{N}$ ) w illalw ays be non-negative, and no BETA I is necessary .)
Q (input/output)
If $C O M P Q=N$ ', then $Q \mathrm{w}$ illlnotbe referenced. If $C O M P Q=V$ ' or ' $I$ ', then the transpose of the orthogonal transform ationsw hich are applied to $A$ and $B$ on the leftw illbe applied to the array $Q$ on the right.

LD Q (input)
The leading dim ension of the array $\mathrm{Q} . \mathrm{LD} \mathrm{Q}>=1$. If $C O M P Q=V$ 'or $I$ ', then $L D Q>=N$.

Z (input/output)
If $\mathrm{COM} \mathrm{MZ}=\mathrm{N}$ ', then Z w ill notbe referenced. If
$\mathrm{COM} P Z=\mathrm{V}^{\prime}$ or ' I ', then the orthogonaltransform ations w hich are applied to $A$ and $B$ on the right w ill be applied to the array $Z$ on the right.

LD $Z$ (input)
The leading $d i m$ ension of the array $Z . L D Z>=1$. If COM PZ=V 'or I', then LD Z >=N .

W ORK (w orkspace)
On exit, if $\mathbb{N}$ FO >= 0 , W ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the array W ORK. LW ORK >= $\max (1, N)$.

If LW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
$\mathbb{N} F O$ (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the $i$-th argum enthad an ille-
galvalue
$=1, \ldots, N$ : the QZ iteration did not converge. ( $\mathrm{A}, \mathrm{B}$ ) is not in Schur form, but ALPHAR (i), A LPHAI(i), and BETA (i), $i=\mathbb{N} F O+1, \ldots, N$ should be correct. $=\mathrm{N}+1, \ldots, 2 * \mathrm{~N}$ : the shiftcalculation failed. ( $A, B$ ) is not in Schur form, but ALPHAR (i), ALPHAI(i), and BETA (i), i= $\mathbb{N F O -}$ $\mathrm{N}+1, \ldots, \mathrm{~N}$ should be correct. > $2 \star \mathrm{~N}$ : various
"im possible" errors.

## FURTHER DETAILS

## Iteration counters:

JITER - counts iterations.
ITER - counts iterations nun since ILA ST w as last changed. This is therefore resetonly when a 1-by-1 or
$2-b y-2$ block deflates off the bottom .

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

shsein-use inverse tieration to find specified right and/or lefteigenvectors of a real upper $H$ essenberg $m$ atrix $H$

## SYNOPSIS

```
SU BROUT\mathbb{NE SHSE\mathbb{N (SDDE,EIGSRC,IN ITV,SELECT,N,H,LDH,W R,W I,VL,}}\mathbf{N},\textrm{N},\textrm{N},
    LDVL,VR,LDVR,MM,M,W ORK,\mathbb{FA}\mathbb{H},\mathbb{FA}\mathbb{H},\mathbb{N}FO)
```

CHARACTER * 1 SIDE,EIGSRC, $\mathbb{N} \operatorname{ITV}$
$\mathbb{N}$ TEGER N, LD H, LDVL,LDVR,MM, M, $\mathbb{N} F O$
$\mathbb{N} T E G E R \mathbb{F} A \mathbb{L}(*)$, $\mathbb{F A} \mathbb{H} R(*)$
LO G ICAL SELECT (*)

SU BROUTINE SHSEIN_64 (SDE,EIGSRC, $\mathbb{N} \mathbb{I} \mathbb{V}, \operatorname{SELECT}, N, H, L D H, W R, W I$,
VL,LDVL,VR,LDVR,MM,M,WORK, $\mathbb{F A} \mathbb{H}, \mathbb{F} A \mathbb{H}, \mathbb{N} F O)$
CHARACTER * $1 \mathrm{SDE} \mathrm{E}, \mathrm{EIG} \mathrm{SRC}, \mathbb{N} \operatorname{ITV}$
$\mathbb{N}$ TEGER*8N,LDH,LDVL,LDVR,MM,M, $\mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{F} A \mathbb{L}(*)$, $\mathbb{F A} \mathbb{H} R(*)$
LOG ICAL*8SELECT (*)
REAL H (LD H ,*), WR (*), W I(*),VL (LDVL,*),VR (LDVR,*),W ORK (*)

## F95 INTERFACE

SU BROUTINE HSEIN (SDE,EIGSRC, $\mathbb{N} \mathbb{I T V}, \operatorname{SELECT}, \mathbb{N}], H,[L D H], W R, W$ I, VL, [LDVL], VR, [LDVR], MM, M, [W ORK], $\mathbb{F} A \mathbb{L}, \mathbb{F} A \mathbb{L} R,[\mathbb{N} F O])$

CHARACTER (LEN=1)::SDE,EIGSRC, IN ITV
$\mathbb{N} T E G E R:: N, L D H, L D V L, L D V R, M M, M, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: $\mathbb{F} A \mathbb{L} L, \mathbb{F} A \mathbb{I} R$
LOG ICAL,D IM ENSION (:) ::SELECT

SU BROU T $\mathbb{N} E \operatorname{HE} \operatorname{SE} \_64(S \mathbb{D} E, E \operatorname{G} \operatorname{SRC}, \mathbb{N} \mathbb{I} V, S E L E C T,[N], H,[L D H], W R$, W I, VL, [LDVL], VR, [LDVR],MM,M,[WORK], $\mathbb{F} A \mathbb{L}, \mathbb{F} A \mathbb{L} R,[\mathbb{N F O}])$

CHARACTER (LEN=1) ::SIDE,EIG SRC, $\mathbb{N} \operatorname{ITV}$
$\mathbb{N}$ TEGER (8) :: N, LD H ,LDVL,LDVR, M M, M, $\mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{F} A \mathbb{I}, \mathbb{F} A \mathbb{I}$
LOGICAL (8), D $\mathbb{M} E N S I O N(:):$ SELECT
REAL,D $\mathbb{M} E N S I O N(:):: W R, W I, W O R K$
REAL,D $\mathbb{I M}$ ENSION (:,:) ::H,VL,VR

## C INTERFACE

\#include <sunperfh>
void shsein (char side, chareigsre, char initv, int *select, int $n$, float *h, int ldh, float *W $r$, float *W i, float * Vl , int ldvl, float * Vr , int ldvr, int m m , int *m, int *ifaill, int *ifailr, int *info);
void shsein_64 (charside, char eigsrc, char initv, long
*select, long n, float*h, long ldh, float *w r, float *W i, float* vl , long ldvl, float *Vr, long ldvi, long mm, long *m, long *ifaill, long *ifailr, long *info);

## PURPOSE

shsein uses inverse teration to find specified rightand/or lefteigenvectors of a real upperH essenberg m atrix H.

The righteigenvectorx and the lefteigenvector $y$ of the $m$ atrix $H$ comesponding to an eigenvalue $w$ are defined by:

$$
H^{*} x=w^{*} x, \quad y^{\star * h} * H=w^{\star} y^{\star *} h
$$

where $y^{* *}$ h denotes the conjugate transpose of the vectory.

## ARGUMENTS

```
SID E (input)
\(=R\) ': com pute righteigenvectors only;
\(=\mathbb{L}\) ': com pute lefteigenvectors only;
\(=\mathrm{B}:\) com pute both right and lefteigenvectors.
```

E IG SRC (input)
Specifies the source of eigenvalues supplied in
(WR,WI):
$=Q$ ': the eigenvalues w ere found using SH SEQR; thus, if H has zero subdiagonalelem ents, and so is block-triangular, then the $j$ th eigenvalue can be assum ed to be an eigenvalue of the block containing the jth row /oolum n. Thisproperty allow s SHSEIN to perform inverse iteration on justone diagonalblock. $=N$ ': no assum ptions are $m$ ade on the comespondence betw een eigenvalues and diagonalblocks. In this case, SH SE $\mathbb{N}$ m ustalw ays perform inverse teration using the w holem atrix $H$.
$\mathbb{N} \operatorname{ITV}$ (input)
$=\mathrm{N}$ ': no initial vectors are supplied;
$=\mathrm{U}$ ': user-supplied initial vectors are stored in the arays $V L$ and/orV R.

## SELECT (input/output)

Specifies the eigenvectors to be com puted. To select the real eigenvector corresponding to a realeigenvalue $W$ R ( 1 ) , SELECT ( 7 ) m ust be set to .TRUE.. To select the complex eigenvector corresponding to a complex eigenvalue (WR $(\mathcal{j}), W I(j))$, w th complex conjugate (WR $(j+1), W I(j+1))$, eitherSELECT ( $\mathfrak{j}$ ) orSELECT ( $\mathfrak{j}+1$ )
orboth $m$ ust.be setto

N (input) The order of the m atrix $\mathrm{H} . \mathrm{N}>=0$.

H (input) The upperH essenberg $m$ atrix $H$.

## LD H (input)

The leading dim ension of the aray H . LD H >= $\max (1, N)$.

W R (input/output)
On entry, the real and im aginary parts of the eigenvalues of H ; a complex conjugate pair of eigenvalues m ustibe stored in consecutive elem ents
of W R and W I. On exit, W R m ay have been altered since close eigenvalues are perturbed slightly in searching for independenteigenvectors.

W I (input)
See the description of R R .

VI (input/output)
On entry, if $\mathbb{N} \mathbb{I T V}=\mathrm{U}$ 'and $S \mathbb{D} E=\mathrm{L}$ 'or $B$ ', VL $m$ ust contain starting vectors for the inverse iteration for the lefteigenvectors; the starting vector for each eigenvectorm ust.be in the sam e
colum $n(s)$ in which the eigenvectorw illibe stored. On exit, if $S \mathbb{D} E=$ L'or B', the lefteigenvectors specified by SELEC T w illbe stored consecutively in the colum ns of $V L$, in the sam e order as their eigenvalues. A complex eigenvector corresponding to a com plex eigenvalue is stored in tw o consecutive colum ns, the first holding the real part and the second the im aginary part. If $S D E=R ', V L$ is not referenced.

LDVL (input)
The leading dim ension of the array VL. LDVL >= $\max (1, N)$ if $S \mathbb{D} E=L$ 'or $B$ '; LDVL >= 1 otherw ise.

VR (input/output)
On entry, if $\mathbb{N} \mathbb{T V}=\mathrm{U}$ 'and $S \mathbb{D} E=R$ 'or $B$ ', VR $m$ ust contain starting vectors for the inverse iteration for the righteigenvectors; the starting vector for each eigenvectorm ustbe in the sam e colum n(s) in which the eigenvectorw illbe stored. On exit, if $S \mathbb{D E}=\mathrm{R}$ 'or B ', the righteigenvectors specified by SELEC T w illbe stored consecutively in the colum ns of VR, in the sam e order as their eigenvahues. A complex eigenvector comesponding to a com plex eigenvalue is stored in tw o consecutive colum ns, the first holding the real part and the second the im aginary part. If $S \mathbb{D E}=\mathrm{L}, \mathrm{VR}$ is notreferenced.

LDVR (input)
The leading dim ension of the array VR. LDVR >= $\max (1, N)$ if $S \mathbb{D} E=R$ 'or $B$ '; LDVR $>=1$ otherw ise.

M M (input)
The num berof colum ns in the arrays VL and/or VR. M M >= M .

M (output)
The num ber of colum ns in the anays VL and/or VR required to store the eigenvectors; each selected realeigenvector occupies one column and each selected com plex eigenvector occupies tw o colum ns.

W ORK (w orkspace)
dim ension $(\mathbb{N}+2) * \mathrm{~N}$ )

FFA ILL (output)
If $S \mathbb{D} E=L$ 'or $B ', \mathbb{F A}$ ILL $(i)=j>0$ if the
left eigenvector in the ith column of VL (comesponding to the eigenvalue $\mathrm{w}(\mathrm{j})$ failed to converge; $\mathbb{F A} \mathbb{I L L}(i)=0$ if the eigenvectorconverged satisfactorily. If the $i$-th and (i+1)th colum ns of VL hold a com plex eigenvector, then $\mathbb{F A} \mathbb{I L L}(i)$ and $\mathbb{F} A \mathbb{L} L(i+1)$ are set to the sam e value. If $S \mathbb{D} E=R ', \mathbb{F} A \mathbb{I} L$ is not referenced.
FFA ILR (output)
If $S \mathbb{D} E=R$ 'or $B$ ', $\mathbb{F A}$ ILR $(i)=j>0$ if the right eigenvector in the i-th colum $n$ of VR (comesponding to the eigenvalue w( $\mathcal{J}$ ) failed to converge; $\mathbb{F A} \mathbb{I L} R(i)=0$ if the eigenvectorconverged satisfactorily. If the $i$-th and (i+1)th colum ns of VR hold a com plex eigenvector, then $\mathbb{F A} \mathbb{I} R(i)$ and $\mathbb{F} A \mathbb{I} R(i+1)$ are set to the sam e value. IfSID $\mathrm{E}=\mathrm{L}^{\prime}$, , $\mathbb{F} A \mathbb{I} \mathrm{R}$ is not referenced.
$\mathbb{N} F O$ (output)
= 0 : successfiulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the i-th argum enthad an illegalvalue
> 0 : if $\mathbb{N} F O=i$, $i$ is the num ber of eigenvectors which failed to converge; see $\mathbb{F} A \mathbb{I} L$ and $\mathbb{F} A \mathbb{I} R$ for further details.

## FURTHER DETAILS

E ach eigenvector is nom alized so that the elem ent of largest $m$ agnitude has $m$ agnitude 1 ; here the $m$ agnitude of a com plex num ber $(x, y)$ is taken to be $|x+t y|$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

shseqr-com pute the eigenvalues of a real upper H essenberg $m$ atrix $H$ and, optionally, the $m$ atrices $T$ and $Z$ from the Schurdecom position $H=Z \mathrm{~T} \mathrm{Z}^{* *} \mathrm{~T}$, where T is an upper quasi-triangular $m$ atrix (the Schur form ), and $Z$ is the orthogonalm atrix of Schurvectors

## SYNOPSIS



```
    W ORK,LW ORK,\mathbb{NFO)}
CHARACTER * 1 JOB,COMPZ
\mathbb{NTEGERN, ILO,\mathbb{H I,LDH,LD Z,LW ORK,INFO}}\mathbf{N},\mp@code{L}
REALH (LDH ,*),W R (*),W I(*),Z (LD Z ,*),W ORK (*)
SU BROUT\mathbb{NE SHSEQR_64(JOB,COM PZ,N,#O, IHI,H,LD H,W R,W I, Z,LD Z,}
    W ORK,LW ORK,\mathbb{NFO)}
CHARACTER * 1 JOB,COMPZ
NNTEGER*8N,\mathbb{LO,}\mathbb{H}I,LDH,LDZ,LW ORK,\mathbb{NFO}
REALH (LDH ,*),W R (*),W I(*),Z (LD Z ,*),W ORK (*)
```


## F95 INTERFACE

```
SU BROUTINE HSEQR (JOB,COMPZ,N, \(\mathbb{H} O, \mathbb{H} I, H,[L D H], W R, W I, Z,[L D Z]\), [ \(\mathbb{N}\) ORK ], [LW ORK ], [ \(\mathbb{N F O}]\) )
```

CHARACTER (LEN=1)::JOB,COMPZ
$\mathbb{N} T E G E R:: N, \mathbb{L} O, \mathbb{H} I, L D H, L D Z, L W O R K, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::WR,WI,W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) :: H,Z
SU BROUTINE HSEQR_64 (JOB,COMPZ,N, $\mathbb{L} O, \mathbb{H} I, H,[L D H], W R, W I, Z$,
[LD Z], [W ORK ], [LW ORK ], [ $\mathbb{N} F \mathrm{FO}$ ])

CHARACTER (LEN=1)::JOB,COMPZ
$\mathbb{N}$ TEGER (8) ::N, $\mathbb{N} O, \mathbb{H} I, L D H, L D Z, L W O R K, \mathbb{N} F O$ REAL,D $\mathbb{M}$ ENSION (:) ::WR,W I,W ORK
REAL,D IM ENSION (:,:) :: H , Z

## C INTERFACE

\#include < sunperfh>
void shseqr(char job, char com pz, intn, int ilo, int ihi, float *h, int ldh, float *w r, float *w i, float *z, int ldz, int *info);
void shseqr_64 (char j̀b, charcom pz, long n, long ilo, long ihi, float *h, long ldh, float*w r, float *w i, float *z, long ldz, long *info);

## PURPOSE

shseqr com putes the eigenvalues of a real upper H essenberg $m$ atrix $H$ and, optionally, the $m$ atrices $T$ and $Z$ from the Schurdecomposition $H=Z \mathrm{~T} \mathrm{Z**T}$, where T is an upper quasi-triangular $m$ atrix the Schur form ), and $Z$ is the orthogonalm atrix of Schurvectors.

O ptionally Z m ay be postm ultiplied into an input orthogonal $m$ atrix $Q$, so that this routine can give the Schur factorization of a m atrix A which has been reduced to the $H$ essenberg form $H$ by the orthogonal $m$ atrix $Q: A=Q * H * Q * T=$ (Q Z)*T* $\mathrm{Q} Z)^{\star *} \mathrm{~T}$.

## ARGUMENTS

JOB (input)
= E ': com pute eigenvalues only;
$=S$ ': com pute eigenvalues and the Schur form T .

COM PZ (input)
$=\mathrm{N}$ ': no Schurvectors are com puted;
$=I^{\prime}: \mathrm{Z}$ is in inialized to the unit m atrix and
the m atrix Z of Schurvectors of H is retumed; $=$ V ': Z mustcontain an orthogonal matrix Q on entry, and the product Q * Z is retumed.

N (input) The order of the m atrix $\mathrm{H} . \mathrm{N}>=0$.

It is assum ed that H is already upper triangular in row s and colum ns 1: $\mathbb{T}-1$ and $\mathbb{H} \mathrm{I}+1 \mathrm{~N} . \mathbb{I} \mathrm{O}$ and $\mathbb{H}$ I are norm ally setby a previous call to SG EBA L, and then passed to SGEHRD w hen them atrix output by SG EBA $L$ is reduced to $H$ essenberg form. O therw ise HO and $\mathbb{H} I$ should be set to 1 and $N$ respectively. $1<=\mathbb{H O}<=\mathbb{H} \mathrm{I}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{H} \mathrm{O}=1$ and $\mathbb{H} \mathrm{I}=0$, if $\mathrm{N}=0$.

IH I (input)
See the description of IIO .
H (input/output)
On entry, the upper H essenberg $m$ atrix H . On exit, if $J O B=S ', H$ contains the upper quasitriangularm atrix $T$ from the Schur decom position (the Schur form); 2-by-2 diagonal blocks (comesponding to com plex conjugate pairs of eigenvalues) are retumed in standard form, w ith $\mathrm{H}(i, i)=H(i+1, i+1)$ and $H(i+1, i) * H(i, i+1)<0$. If $\mathrm{JOB}=\mathrm{E}$ ', the contents of H are unspecified on exit.

LD H (input)
The leading din ension of the array H. LD H >= $\max (1, N)$.

W R (output)
The real and im aginary parts, respectively, of the com puted eigenvalues. Iftw o eigenvalues are com puted as a com plex conjugate pair, they are stored in consecutive elem ents of $W R$ and $W I$, say the $i$-th and (i+1)th, with W I(i) > 0 and $W$ I(i+1) < 0 . If $J 0 B=S$ ', the eigenvalues are stored in the sam e order as on the diagonal of the Schur form retumed in $H$, w ith $W$ R (i) $=\mathrm{H}(i, i)$ and, if H ( $i: i+1, i: i+1$ ) is a 2 -by -2 diagonalblock, $W$ I(i) $=$ squt $(\mathrm{H}(i+1, i) * H(i, i+1))$ and $W I(i+1)=-W I(i)$.

W I (output)
See the description ofW R .

Z (input) IfCOM PZ = N ': Z is not referenced.
If COM PZ = I': on entry, Z need notbe set, and on exit, $Z$ contains the orthogonalm atrix $Z$ of the Schurvectors of H . IfCOM PZ = V ': on entry Z m ust contain an N -by -N m atrix Q , which is assum ed to be equal to the unitm atrix except for the sub$m$ atrix $Z$ ( $\mathbb{H} O: \mathbb{H} I, \mathbb{I} O: \mathbb{H} I$ ); on exitZ contains $Q$ *Z . N orm ally $Q$ is the orthogonalm atrix generated by SORGHR after the call to SGEHRD which form ed the

H essenberg $m$ atrix $H$.

LD Z (input)
The leading dim ension of the array Z . LD $\mathrm{Z}>=$ $\max (1, N)$ if COMPZ = I'orV';LD $Z>=1$ otherwise.
W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the array W ORK. LW ORK >= $\max (1, N)$.

IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
$\mathbb{N} F O$ (output)
= 0: successfulexit
<0: if $\mathbb{I N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue
> 0 : if $\mathbb{N} F O=i, S H$ SEQR failed to com pute allof the eigenvalues in a total of 30* ( $\mathbb{H} \mathrm{I}-\mathbb{H O}+1$ ) iterations; elem ents 1 :ilo- 1 and i+ 1 m of $\mathrm{W} R$ and W I contain those eigenvalues w hich have been successfully com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

singb - synthesize a Fourier sequence from its representation in term sofa sine series w th odd w ave num bers. The $S \mathbb{N}$ Q operations are unnorm alized inverses of them selves, so a call to $S \mathbb{N} Q F$ follow ed by a call to $S \mathbb{N} Q B$ w illm ultiply the inputsequence by 4 * $N$.

## SYNOPSIS

```
SUBROUT\mathbb{NE S\mathbb{NQB N,X,W SAVE)}}\mathbf{N}=()
INTEGER N
REALX (*),W SAVE (*)
SUBROUTINESINQB_64(N,X,W SAVE)
INTEGER*8 N
REALX (*),W SAVE (*)
```

F95 INTERFACE
SU BROUTINESINQB $\mathbb{N}, X, W$ SAVE)
$\mathbb{N} T E G E R:: N$
REAL,D $\mathbb{I M}$ ENSION (:) ::X,W SAVE
SU BROUTINE SINQB_64 $\mathbb{N}, \mathrm{X}, \mathrm{W}$ SAVE)
$\mathbb{N} T E G E R(8):: N$
REAL,D $\mathbb{M}$ ENSION (:) ::X,W SAVE

## C INTERFACE

\#include <sunperfh>
void singb (intn, float *x, float *w save);

## ARGUMENTS

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product
of sm allprim es. $\mathrm{N}>=0$.
X (input/output)
On entry, an array of length N containing the sequence to be transform ed. On exit, the quarterw ave sine synthesis of the input.
W SAVE (input)
On entry, an array with dim ension of at least (3 *
$\mathrm{N}+15$ ) for scalar subroutines, initialized by
$S \mathbb{N} Q \mathrm{I}$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

sinqf-com pute the Fourier coefficients in a sine series
representation w ith only odd w ave num bers. The $S \mathbb{N} Q$ operations are unnorm alized inverses of them selves, so a call to $S \mathbb{N} Q F$ follow ed by a call to $S \mathbb{N} Q B$ w illm ultiply the input sequence by 4 * $N$.

## SYNOPSIS

```
SUBROUT\mathbb{NE SINQF(N,X,W SAVE)}
```

$\mathbb{N}$ TEGER N
REALX $(*), \mathrm{W} \operatorname{SAVE}\left({ }^{*}\right)$
SU BROUTINE SINQF_64 $\mathbb{N}, \mathrm{X}, \mathrm{W}$ SAVE)
$\mathbb{N}$ TEGER*8N
REALX (*), W SAVE (*)

F95 INTERFACE
SUBROUTINE SINQF $\mathbb{N}, \mathrm{X}, \mathrm{W}$ SAVE)
$\mathbb{N} T E G E R:: N$
REAL,D $\mathbb{I M}$ ENSION (:) ::X,W SAVE
SUBROUTINESTNQF_64 $\mathbb{N}, \mathrm{X}, \mathrm{W}$ SAVE)
$\mathbb{N} T E G E R(8):: N$
REAL,D $\mathbb{I M}$ ENSION (:) ::X,W SAVE

## C INTERFACE

\#include <sunperfh>
void sinqf(intn, float *x, float *w save);

## ARGUMENTS

N (input) Length of the sequence to be transform ed. These subroutines are $m$ ostefficientw hen $N$ is a product
of sm allprim es. $\mathrm{N}>=0$.
X (input/output)
Onentry, an aray of length N containing the sequence to be transform ed. On exit, the quarterw ave sine transform of the input.
W SAVE (input)
On entry, an array with dim ension of at least (3

* $N+15$ ) forscalar subroutines, in inialized by
$S \mathbb{N} Q$.


## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

sinqi-initialize the array $\times W$ SA VE, which is used in both $S \mathbb{N} Q F$ and $S \mathbb{N} Q B$.

## SYNOPSIS

SU BROUTINESINQIN,W SAVE)
$\mathbb{N}$ TEGER N
REALW SAVE (*)
SUBROUTINESINQI_64 $\mathbb{N}, \mathrm{W}$ SAVE)
$\mathbb{N}$ TEGER*8 N
REALW SAVE (*)
F95 INTERFACE
SU BROUTINESINQIN,W SAVE)
$\mathbb{N} T E G E R:: N$
REAL,D $\mathbb{M}$ ENSION (:) ::W SAVE

SUBROUTINESINQI_64 (N,W SAVE)
$\mathbb{N} T E G E R(8):: N$
REAL,D $\mathbb{I M}$ ENSION (:) ::W SAVE

## C INTERFACE

\#include <sunperfh>
void sinqi(intn, float *W save);
void sinqi_ 64 (long n, float *w save);

## ARGUMENTS

N (input) Length of the sequence to be transform ed. The m ethod is m ost efficientw hen N is a product of sm allprim es.

W SAVE (input)
On entry, an array ofdim ension ( 3 * $\mathrm{N}+15$ ) or greater. SIN Q I needs to be called only once to intialize W SA VE before calling $S \mathbb{N} Q F$ and/orS $\mathbb{N} Q B$ if N and W SAVE rem ain unchanged betw een these calls. Thus, subsequent transform s or inverse transforms of same size can be obtained faster than the first since they do not require initialization of the w orkspace.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

sint-com pute the discrete F ourier sine transform of an odd sequence. The SIN T transform s are unnorm alized inverses of them selves, so a callofS $\mathbb{N} T$ follow ed by another call of SIN $T$ w illm ultiply the input sequence by 2 * $(\mathbb{N}+1)$.

## SYNOPSIS

SU BROUTINESINT $\mathbb{N}, \mathrm{X}, \mathrm{W}$ SAVE)
$\mathbb{N}$ TEGER N
REALX ${ }^{*}$ ) , W SAVE (*)
SU BROUTINE SNT_64 $\mathbb{N}, \mathrm{X}, \mathrm{W}$ SAVE)
$\mathbb{N}$ TEGER*8 N
REALX (*), W SAVE (*)

## F95 INTERFACE

SU BROUTINE SNT $\mathbb{N}$, X, W SAVE)
$\mathbb{N} T E G E R:: N$
REAL,D $\mathbb{M}$ ENSION (:) ::X,W SAVE
SU BROUTINESTNT_64 $\mathbb{N}, \mathrm{X}, \mathrm{W}$ SAVE)
$\mathbb{N}$ TEGER (8) :: N
REAL,D $\mathbb{M}$ ENSION (:) ::X,W SAVE

## C INTERFACE

\#include <sunperfh>
void $\operatorname{sint}$ (intn, float *x, float *w save);

## ARGUMENTS

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen $\mathrm{N}+1$ is a productof sm allprim es. $\mathrm{N}>=0$.
$X$ (input/output)
Onentry, an aray of length N containing the sequence to be transform ed. On exit, the sine transform of the input.
W SAVE (input/output)
On entry, an array w ith dimension of at least int(2.5*N + 15) initialized by SIN TI.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS


## NAME

sinti-initialize the anray W SA VE, which is used in subroutine $S \mathbb{N} T$.

## SYNOPSIS

SU BROUTINE SINTIN,W SAVE)
$\mathbb{N}$ TEGER N
REALW SAVE (*)
SU BROUTINESINTI_64N,W SAVE)
$\mathbb{N}$ TEGER*8N
REALW SAVE (*)
F95 INTERFACE
SUBROUTINESINTIN,W SAVE)
$\mathbb{N} T E G E R:: N$
REAL,D $\mathbb{M}$ ENSION (:) ::W SAVE

SU BROUTINESINTI_64 N,W SAVE)
$\mathbb{N}$ TEGER (8) :: N
REAL,D $\mathbb{M}$ ENSION (:) ::W SAVE

## C INTERFACE

\#include <sunperfh>
void sinti(intn, float *W save);
void sinti_ 64 (long n, float *w save);

## ARGUMENTS

N (imput) Length of the sequence to be transform ed. $\mathrm{N}>=0$.

W SAVE (input/output)
On entry, an array ofdim ension ( $2 \mathrm{~N}+\mathrm{N} / 2+15$ ) or greater. SIN TI is called once to initializeW SA VE before calling $S \mathbb{N} T$ and need notbe called again betw een calls to SINT if N and W SAVE rem ain unchanged. Thus, subsequent transform s of sam e size can be obtained faster than the first since they do not require initialization of the w orkspace.

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

sfadm m -Jagged diagonalm atrix-m atrix m ultiply (m odified Ellpack)

## SYNOPSIS

```
SUBROUTINESJADMM(TRANSA,M,N,K,ALPHA,DESCRA,
* VAL,\mathbb{NDX,PNTR,MAXNZ,\mathbb{PERM,}}\mathbf{N}=
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,M,N,K,DESCRA (5),MAXNZ,}
* LDB,LDC,LW ORK
\mathbb{NTEGER INDX NNZ),PNTR MAXNZ+1),\mathbb{PERM M)}}\mathbf{M}\mathrm{ (N)}
REAL ALPHA,BETA
REAL VAL (NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NESJADMM_64(TRANSA,M,N,K,A LPHA,DESCRA,}
* VAL,\mathbb{NDX,PNTR,MAXNZ,\mathbb{PERM,}}+\mathbf{M}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,M,N,K,DESCRA (5),MAXNZ,
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 INDX NNZ),PNTR MAXNZ+1),\mathbb{PERM M)}}\mathbf{M}\mathrm{ (1)}
REAL ALPHA,BETA
REAL VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
```

where NN Z=PN TR M A XN Z +1)-PN TR (1)+1 is the num berofnon-zero elem ents

## F95 INTERFACE

SUBROUTINE JADMM (TRANSA, M, $\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N} D X$, * PNTR,MAXNZ, $\mathbb{P E R M}, \mathrm{B},[\operatorname{LDB}], B E T A, C,[L D C],[W$ ORK], [LW ORK]) $\mathbb{N} T E G E R$ TRANSA, M, K, MAXNZ
$\mathbb{I N}$ TEGER,D $\mathbb{M}$ ENSION (:) :: DESCRA, $\mathbb{N D D X , P N T R , ~ \mathbb { P E R M }}$
REAL ALPHA,BETA
REAL,D $\mathbb{I}$ ENSION (:) :: VAL

REAL,D $\mathbb{I M}$ ENSION (:, :) :: B , C

SUBROUTINE JADMM_64 (TRANSA, M, $\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N D X}$,

$\mathbb{N}$ TEGER*8 TRANSA, M, K, MAXNZ
$\mathbb{N}$ TEGER*8, D $\mathbb{M}$ ENSION (:) :: DESCRA, $\mathbb{N} D X, P N T R, \mathbb{P E R M}$
REAL ALPHA,BETA
REAL, D $\mathbb{M}$ ENSION (:) :: VAL
REAL,D $\mathbb{I}$ ENSION (:, :) :: B, C

## DESCRIPTION

C <-alpha op (A ) B + beta C
where A LPHA and BETA are scalar, $C$ and $B$ are dense $m$ atrioes, $A$ is a m atrix represented in jagged-diagonal form at and op (A ) is one of $o p(A)=A$ or $o p(A)=A^{\prime}$ or $o p(A)=\operatorname{conjg}\left(A^{\prime}\right)$.
( 'indicates m atrix transpose)

## ARGUMENTS

TRANSA Indicates how to operate $w$ th the sparse $m$ atrix
0 : operate $w$ ith $m$ atrix
1 : operate $w$ ith transpose $m$ atrix
2 : operate $w$ ith the conjugate transpose ofm atrix. 2 is equivalentto 1 ifm atrix is real.

M $\quad N$ um berof row $s$ in matrix A

N $N$ um ber of colum ns in $m$ atrix $C$

K $\quad$ Num berof colum ns in matrix A

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem ent integer anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( $\mathrm{A}=\mathrm{A}$ )
2 : Herm itian ( $A=C O N J(A)$ )
3 :Triangular
4 : Skew (A nti)-Symm etric ( $\mathrm{A}=-\mathrm{A}$ )
5 :D iagonal
6 : Skew Herm itian ( $A=-C O N J(A)$ )
D ESCRA (2) upper/low er triangular indicator
1 : low er

2 :upper
DESCRA (3) main diagonal type
0 :non-unit
1 :unit
DESCRA (4) A ray base $\mathbb{N} O T \mathbb{I}$ PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
D ESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 :no repeated indices

VAL () array of length NN Z consisting of entries of A. VA L can be view ed as a colum $n m$ ajorordering of a row perm utation of the Ellpack representation of , where the Ellpack representation is perm uted so that the row s are non-increasing in the num ber of nonzero entries. V alues added forpadding in E llpack are not included in the Jagged-D iagonal form at.
$\mathbb{I N D X}$ () array of length NNZ consisting of the colum n indices of the comesponding entries in VAL.
PN TR () array of length M AXNZ+1, where PNTR (I)-PN TR (1)+1 points to the location in VAL of the firstelem ent in the row -perm uted E llpack represenation of $A$.

MAXNZ max num berofnonzeros elem ents per row .
$\mathbb{P E R M} 0$ integer array of length $M$ such that $I=\mathbb{P E R M}$ ( $I$ ), where row I in the originalE llpack representation corresponds to row I' in the perm uted representation. If $\operatorname{PERM}(1)=0$, it is assum ed by convention that $\mathbb{P E R M}(\mathbb{I})=I . \mathbb{P E R M}$ is used to determ ine the order in which row s ofC are updated.

B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension of $B$
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension of $C$
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is notreferenced in the current version.

## SEE ALSO

N IST FO RTRA N Sparse B las U ser's G uide available at: http://m ath nistgov/n cso/Staff/k Rem ington/Ispblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

sjadrp - rightperm utation of a jagged diagonalm atrix

## SYNOPSIS

```
SUBROUT\mathbb{NE SJADRP(TRANSP,M,K,VAL, INDX,PNTR,MAXNZ,}
* IPERM,WORK,LW ORK )
INTEGER TRANSP,M,K,MAXNZ,LW ORK
INTEGER INDX(*),PNTR(MAXNZ+1),\mathbb{PERM (K),WORK (LW ORK)}
REAL VAL (*)
SUBROUT\mathbb{NE SJADRP_64(TRAN SP,M,K,VAL, INDX,PNTR,M AXNZ,}
* IPERM,WORK,LW ORK )
INTEGER*8 TRANSP,M,K,MAXNZ,LW ORK
```



```
REAL VAL(*)
```


## F95 INTERFACE

```
SUBROUTINE JADRP(TRANSP,M,K,VAL, INDX,PNTR,MAXNZ,
* IPERM,[WORK],[LW ORK])
INTEGER TRANSP,M,K,MAXNZ
```



```
REAL,DIM ENSION (:) ::VAL
SUBROUT\mathbb{NE JADRP_64(TRANSP,M,K,VAL,INDX,PNTR,MAXNZ,}
* \quadPERM,[W ORK],[LW ORK])
INTEGER*8 TRANSP,M,K,M AXNZ
```



```
REAL,DIM ENSION (:) ::VAL
```

A $<-A P$
$A<-A P^{\prime}$
( 'indicates m atrix transpose)
$w$ here perm utation $P$ is represented by an integervector $\mathbb{P} E R M$, such that $\mathbb{P E R M}(I)$ is equal to the position of the only nonzero elem entin row Iofperm utation $m$ atrix $P$.

N O TE : In orderto get a sym etrically perm uted jagged diagonal $m$ atrix P A P', one can explicitly perm ute the colum ns P A by calling

SJADRP ( $0, M, M, V A L, \mathbb{N} D X, P N T R, M A X N Z, \mathbb{P} E R M, W$ ORK,LW ORK)
where param eters $V A L, \mathbb{N D X}, P N T R, M A X N Z, \mathbb{P E R M}$ are the representation of $A$ in the jagged diagonal form at. The operation $m$ akes sense if the originalm atrix $A$ is square.

## ARGUMENTS

TRAN SP Indicates how to operate $w$ ith the perm utation $m$ atrix
0 : operate w ith m atrix
1 : operate $w$ ith transpose $m$ atrix

M $\quad \mathrm{N}$ um berof row $s$ in matrix A

K $\quad \mathrm{N}$ um ber of colum ns in matrix A

VAL () amay of length PNTR MAXNZ+1)-PNTR (1) consisting of entries ofA. VA L can be view ed as a colum $n m$ ajor ordering of a row perm utation of the E llpack representation of A, w here the Ellpack representation is perm uted so that the row $s$ are non-increasing in the num ber of nonzero entries. $V$ alues added for padding in Ellpack are not included in the Jagged - D iagonal form at.

INDX () array of length PN TR MAXNZ+1)-PNTR (1) consisting of the colum $n$ indices of the corresponding entries in VAL.

PNTR () array of length M AXNZ+1, where PNTR (I) PNTR (1)+1 points to the location in VA L of the firstelem ent in the row -perm uted E lhpack represenation of .

M A X N Z max num ber ofnonzeros elem ents per row.
$\mathbb{P} E R M$ ( integeramay of length $K$ such that $I=\mathbb{P} E R M$ ( $I$ ).

A ray $\mathbb{P} E R M$ represents a perm utation $P$, such that $\mathbb{P E R M}$ ( I ) is equal to the position of the only nonzero elem ent in row Iofperm utation $m$ atrix $P$.
Forexam ple, if
|001|
$\mathrm{P}=\left|\begin{array}{lll}1 & 0 & 0\end{array}\right|$
|010|
then $\mathbb{P E R M}=(3,1,2)$.

W ORK () scratch array of length LW ORK. LW ORK should be at leastK.

LW ORK length ofW ORK aray

If LW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

## SEE ALSO

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/K Rem ington/tspblas/
"D ocum ent for the B asic $L$ inearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib org/utk/papers/sparse _ps

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

sfádsm - Jagged-diagonal form at triangular solve

## SYNOPSIS

```
SUBROUTINESJADSM(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,
* VAL,\mathbb{NDX,PNTR,MAXNZ,\mathbb{PERM,}}\mathbf{N},
* B,LDB,BETA,C,LDC,W ORK,LWORK)
INTEGER TRANSA,M,N,UNITD,DESCRA (5),MAXNZ,
* LDB,LDC,LW ORK
\mathbb{NTEGER INDX NNZ),PNTR MAXNZ+1),\mathbb{PERM M)}}\mathbf{M}\mathrm{ (N)}
REAL ALPHA,BETA
REAL DV M),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NESJAD SM _64(TRANSA,M,N,UNITD,DV,A LPHA,DESCRA,}
* VAL,NDX,PNTR,MAXNZ,\mathbb{PERM,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,M,N,UNITD,DESCRA (5),MAXNZ,}
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 INDX NNZ),PNTR MAXNZ+1), IPERM M)}
REAL ALPHA,BETA
REAL DV M),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
```

where NN Z=PN TR M A XNZ+1)-PN TR (1)+1 is the num berofnon-zero elem ents

## F95 INTERFACE

SUBROUTINE JADSM (TRANSA, M, $\mathbb{N}], U N \mathbb{I T D}, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X$, * PNTR,MAXNZ, $\mathbb{P E R M}, \mathrm{B},[\mathrm{LD} \mathrm{B}], \mathrm{BETA}, \mathrm{C},[\mathrm{LD} \mathrm{C}],[W \mathrm{ORK}]$, [LW ORK ]) $\mathbb{I N T E G E R}$ TRANSA, M, MAXNZ
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: DESCRA, $\mathbb{N} D \mathrm{X}, \operatorname{PNTR}, \mathbb{P E R M}$
REAL ALPHA,BETA
REAL,D $\mathbb{M}$ ENSION (:) :: VAL,DV
REAL,D $\mathbb{I M}$ ENSION (: : : :: B, C

SUBROUTINE JAD SM _64 (TRANSA, M, $\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X$,

* $\operatorname{PNTR}, \mathrm{MAXNZ}, \mathbb{P E R M}, \mathrm{B},[\mathrm{LDB}], \mathrm{BETA}, \mathrm{C},[\mathrm{LD} \mathrm{C}],[\mathrm{W} O R K],[L W O R K])$
$\mathbb{N} T E G E R * 8$ TRANSA, M, MAXNZ
$\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O}(:):: \operatorname{DESCRA}, \mathbb{N} D X, P N T R, \mathbb{P} E R M$
REAL ALPHA,BETA
REAL,D $\mathbb{I M}$ ENSION (:) :: VAL, DV
REAL,D $\mathbb{I M} E N S I O N(:,:): B, C$


## DESCRIPTION

$$
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { Op (A) B + BETA C } \\
& C<-A L P H A \text { Op (A)D B + BETA } C
\end{aligned}
$$

where ALPHA and BETA are scalar, $C$ and $B$ are $m$ by $n$ dense $m$ atrices, $D$ is a diagonalscaling $m$ atrix, $A$ is a unit, ornon-unit, upper or low ertriangularm atrix represented in jagged-diagonal form at and $o p(A)$ is one of op (A) $)=\operatorname{inv}(A)$ or op (A $)=\operatorname{inv}(A)$ or op (A) $=\operatorname{inv}\left(\infty n \dot{g}\left(A^{\prime}\right)\right)$ (inv denotesm atrix inverse, 'indicates m atrix transpose)

## ARGUMENTS

TRAN SA Indicates how to operate $w$ th the sparse $m$ atrix 0 : operate $w$ ith $m$ atrix 1 : operate $w$ th transpose $m$ atrix
2 : operate $w$ ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M $\quad \mathrm{N}$ um berof row s in $m$ atrix $A$
$N \quad N$ um berof colum $n s$ in $m$ atrix $C$

UN ITD Type of scaling:
1 : Identity $m$ atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)
4 : A utom atic row scaling (see section N O TES for further details)

DV () A rray of length M containing the diagonalentries of the scaling diagonalm atrix D .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay D ESCRA (1) m atrix structure

0 : general
1 : symm etric ( $A=A$ )
2 : Herm ( $\mathrm{A}=\mathrm{CONJG}$ (A))
3 :Triangular
4 : Skew (A nti)-Symm etric ( $A=-A$ )
5 :D iagonal
6 : Skew Herm itian ( $\mathrm{A}=-\mathrm{CON}$ J ( A ) )
N ote:For the routine, DESCRA (1)=3 is only supported.
DESCRA (2) upper/low er triangular indicator
1: low er
2 : upper
DESCRA (3) m ain diagonaltype
0 :non-unit
1 : unit
DESCRA (4) A ray base $\mathbb{N} O T \mathbb{M}$ PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? $\mathbb{N} O T \mathbb{I}$ PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL 0 array of length NNZ consisting of entries of A. VA L can be view ed as a colum $n m$ ajorordering of a row perm utation of the Ellpack representation of A, where the Ellpack representation is perm uted so that the row s are non-increasing in the num ber of nonzero entries. V alues added forpadding in E llpack are not included in the Jagged-D iagonal form at.
$\mathbb{N} D \mathrm{X} 0 \quad$ array of length NNZ consisting of the colum n indices of the comesponding entries in VAL.

PNTR 0) array of length M AXNZ +1 , where PNTR ( 1 ) PNTR (1) +1 points to the location in VA L of the firstelem ent in the row -perm uted $E$ lipack represenation of $A$.

MAXNZ max num berofnonzeros elem ents per row .
$\mathbb{P E R M}$ ) integer array of length M such that $\mathrm{I}=\mathbb{P} E R M$ ( I ), where row I in the originalE llpack representation corresponds to row I' in the perm uted representation. If $\operatorname{PERM}(\mathbb{1})=0$, its assum ed by convention that $\mathbb{P E R M}(\mathbb{I})=I . \mathbb{P E R M}$ is used to determ ine the order in which row s ofC are updated.

B 0 rectangular array w th first dim ension LD B.

LD B leading dim ension ofB
BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension of $C$

W ORK () scratch array of length LW ORK.
On exit, ifLW ORK $=-1, W$ ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK aray. LW ORK should be at least2*M.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on multiple processors, LW ORK $>=2 * \mathrm{M}$ *N_CPUS where N_CPUS is the maxim um num berof processors available to the program.

If LW ORK $=0$, the routine is to allocate $w$ orkspace needed.
IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

## SEE ALSO

N IST FORTRAN Sparse B las U ser's G uide available at:
http://m ath nistgov/m csd/Staff/k Rem ington/Ispblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

## NOTES/BUGS

1. No test for singularity ornear-singularity is included in this routine. Such tests $m$ ust.be perform ed before calling this routine.
2. If U N ITD $=4$, the routine scales the row s ofA such that their 2 -norm s are one. The scaling $m$ ay im prove the accuracy of the com puted solution. C orresponding entries ofV A L are changed only in the particular case. On retum DV m atrix stored as a vector contains the diagonalm atrix by w hich the row s have been scaled. U N ITD = 2 should be used for the next calls to the routine w ith overw ritten VAL and DV.

WORK (1)=0 on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here $i$ is the row num berw hich 2 -norm is exactly zero.
3. If $\operatorname{DESCRA}(3)=1$ and UN ITD < 4, the unitdiagonalelem ents $m$ ightorm ightnotbe referenced in the JA D representation of a sparse m atrix. They are notused anyw ay in these cases. ButifUN ITD=4, the unit diagonalelem ents M U ST be referenced in the $\sqrt{A} D$ representation.
4.The routine can be applied for solving triangular system $s$ w hen the upper or low er triangle of the general sparse m atrix A is used. H ow ever DESCRA (1) m ust.be equal to 3 in this case.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

slagtf - factorize the $m$ atrix ( T -lam bda*I), where T is an n by n tridiagonal m atrix and lam bda is a scalar, as T lam bda*I = PLU

## SYNOPSIS

```
SUBROUT\mathbb{NE SLAGTF N,A,LAM BDA,B,C,TOL,D, IN, INFO)}
INTEGER N,\mathbb{NFO}
\mathbb{NTEGER \mathbb{N (*)}}\mathbf{(})
REAL LAMBDA,TOL
REAL A (*),B(*),C (*),D (*)
```



```
\mathbb{NTEGER*8 N,\mathbb{NFO}}\mathbf{N}=0
\mathbb{NTEGER*8 \mathbb{N (*)}}\mp@subsup{}{(}{*})
REAL LAMBDA,TOL
REAL A (*),B(*),C (*),D (*)
```


## F95 INTERFACE

SU BROUTINE LAGTF ( $\mathbb{N}], A, L A M B D A, B, C, T O L, D, \mathbb{N},[\mathbb{N} F O])$
$\mathbb{N}$ TEGER ::N, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{N}$
REAL ::LAM BDA,TOL
REAL,D $\mathbb{M}$ ENSION (:) ::A,B,C,D

SU BROUTINE LAGTF_64 ( $\mathbb{N}], A, L A M B D A, B, C, T O L, D, \mathbb{N},[\mathbb{N} F O])$
$\mathbb{N}$ TEGER ( 8 ) :: N, $\mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{N}$

REAL ::LAMBDA,TOL
REAL,D $\mathbb{M}$ ENSION (:) ::A,B,C,D

## C INTERFACE

\#include <sunperfh>
void slagtf(intn, float*a, float lam boda, float *b, float *C, floattol, float *d, int *in, int *info);
void slagtf_64 (long n, float *a, float lam bda, float *b, float *C, float tol, float *d, long *in, long *info);

## PURPOSE

slagtf factorizes the $m$ atrix ( $T$-lam bda* $I$ ), where $T$ is an $n$ by $n$ tridiagonalm atrix and lam bda is a scalar, as w here $P$ is a perm utation $m$ atrix, $L$ is a unit low er tridiagonal $m$ atrix $w$ ith atm ostone non-zero sub-diagonalelem ents per colum $n$ and $U$ is an uppertriangularm atrix $w$ ith atm ost tw o non-zero super-diagonalelem ents percolum $n$.

The factorization is obtained by G aussian elim ination with partialpivoting and im plicitrow scaling.

The param eterLAMBDA is included in the routine so that SLAGTF m ay be used, in conjunction w ith SLA GTS, to obtain eigenvectors of $T$ by inverse iteration.

## ARGUMENTS

N (input) The orderof the m atrix T .

A (input/output)
O $n$ entry, A m ust contain the diagonalelem ents of T.

On exit, A is overw rilten by the $n$ diagonal ele$m$ ents of the upper triangularm atrix $U$ of the factorization of $T$.

LAM BDA (input)
O n entry, the scalar lam bda.
B (input/output)
O $n$ entry, B m ustcontain the ( $n-1$ ) super-diagonal elem ents of $T$.

On exit, $B$ is overw rilten by the $(n-1)$ superdiagonal elem ents of the $m$ atrix $U$ of the factorization of $T$.

C (input/output)
On entry, C m ustcontain the ( $n-1$ ) sub-diagonal elem ents of $T$.

On exit, $C$ is overw ritten by the ( $n-1$ ) subdiagonal elem ents of the $m$ atrix $L$ of the factorization ofT.
TOL (input/output)
On entry, a relative tolerance used to indicate whether ornot the $m$ atrix ( $I$-lam bda*I) is nearly singular. TO L should norm ally be chose as approxi$m$ ately the largest relative error in the elem ents of T.For exam ple, if the elem ents of T are correct to about 4 significant figures, then TO L should be set to about $5 * 10 * *(-4)$. If TOL is supplied as less than eps, where eps is the relative m achine precision, then the value eps is used in place of TO L.

D (output)
On exit, $D$ is overw ritten by the ( $n-2$ ) second super-diagonal elem ents of the $m$ atrix $U$ of the factorization of $T$.
$\mathbb{N}$ (output)
On exit, $\mathbb{N}$ contains details of the perm utation $m$ atrix $P$. If an interchange occurred at the $k$ th step of the elm ination, then $\mathbb{N}(k)=1$, otherv ise $\mathbb{N}(k)=0$. The elem ent $\mathbb{N}(n)$ retums the sm allest positive integer jsuch that
abs(u) (j) ).le.norm ( ( I -lam bda*I) (ㄱ) )*TO L,
w here norm (A ( $)$ ) denotes the sum of the absolute values of the th row of them atrix $A$. If no such jexists then $\mathbb{I N}(n)$ is retumed as zero. If $\mathbb{I N}(n)$ is retumed as positive, then a diagonalelem ent of $U$ is $s m$ all, indicating that ( $(1-l a m b d a * I$ ) is singular ornearly singular,
$\mathbb{I N} F O$ (output)
= 0 : successfulexit

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

slam rg -w ill create a perm utation listw hich w illm erge the elem ents of A (which is com posed of tw o independently sorted sets) into a single setw hich is sorted in ascending order

## SYNOPSIS

```
SU BROUT\mathbb{NE SLAM RG N1,N2,A,TRD 1,TRD 2, INDEX)}
\mathbb{NTEGER N1,N2,TRD 1,TRD2}
\mathbb{NTEGER }\mathbb{N}DEX (*)
REALA (*)
SU BROUT\mathbb{NE SLAM RG_64 N 1,N2,A,TRD 1,TRD 2, INDEX)}
IN TEGER*8N1,N 2,TRD 1,TRD 2
INTEGER*8 \mathbb{NDEX (*)}
REALA (*)
F95 INTERFACE
    SU BROUT\mathbb{NE LAM RG N 1,N2,A,TRD 1,TRD 2,INDEX)}
    \mathbb{NTEGER ::N1,N2,TRD 1,TRD2}
    INTEGER,D IM ENSION (:) :: \mathbb{NDEX}
    REAL,D IM ENSION (:) ::A
    SUBROUT\mathbb{NE LAM RG_64 N 1,N 2,A,TRD 1,TRD 2, INDEX)}
    \mathbb{NTEGER (8)::N1,N2,TRD 1,TRD 2}
```



```
    REAL,D IM ENSION (:) ::A
void slam rg (intn1, intn2, float *a, int trod1, int trod2,
    int*index);
void slam rg_64 (long n1, long n2, float *a, long trod1, long
        trod2, long *index);

\section*{PURPOSE}
slam rg w ill create a perm utation listw hich will m erge the elem ents ofA (w hich is com posed of tw o independently sorted sets) into a single setw hich is sorted in ascending order.

\section*{ARGUMENTS}

N 1 (input)
Length of the first sequence to be m erged.

N 2 (input)
Length of the second sequence to be merged.

A (input) On entry, the first 1 elem ents of A contain a list of num bers w hich are sorted in either ascending ordescending order. Likew ise for the final
N 2 elem ents.

TRD 1 (input)
D escribes the stride to be taken through the anray
A for the firstN 1 elem ents.
\(=-1\) subset is sorted in descending order.
= 1 subset is sorted in ascending order.

TRD 2 (input)
D escribes the stride to be taken through the anray
A forthe first N 1 elem ents.
\(=-1\) subset is sorted in descending order.
\(=1\) subset is sorted in ascending order.

INDEX (output)
On exit this amay w illcontain a perm utation such that if \(B(I)=A(\mathbb{N} D E X(I))\) for \(I=1, N 1+N 2\), then B w illbe sorted in ascending order.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
slarz - applies a realelem entary reflector \(H\) to a real M by \(-\mathrm{N} m\) atrix C , from either the leftor the right

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SLARZ (SDE,M ,N,L,V,INCV,TAU,C,LDC,W ORK)}
CHARACTER * 1SDE
INTEGERM,N,L,INCV,LDC
REAL TAU
REALV (*),C (LDC ,*),W ORK (*)
SUBROUTINE SLARZ_64(S\mathbb{DE,M,N,L,V,INCV,TAU,C,LDC,W ORK)}
CHARACTER * 1SDE
INTEGER*8M,N,L,INCV,LDC
REAL TAU
REALV (*),C (LDC,*),WORK (*)

```

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E\) LARZ (S \(\mathbb{D} E, \mathbb{M}], \mathbb{N}], L, V,[\mathbb{N} C V], T A U, C,[L D C],[W O R K])\)
CHARACTER (LEN=1): :SDE
\(\mathbb{N}\) TEGER :: M , N, L, \(\mathbb{N} C V, L D C\)
REAL ::TAU
REAL,D \(\mathbb{M}\) ENSION (:) ::V,W ORK
REAL,D IM ENSION (: : : : : C
SU BROUTINE LARZ_64 (SDE, \(\mathbb{M}], \mathbb{N}], L, V,[\mathbb{N} C V], T A U, C,[L D C],[W O R K])\)

CHARACTER (LEN=1) ::SDE
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{L}, \mathbb{N} C V, L D C\)
REAL ::TAU
REAL,D \(\mathbb{I M} E N S I O N(:):: V\),W ORK
REAL,D IM ENSION (:,:) ::C

\section*{C INTERFACE}
\#include <sunperfh>
void slarz (char side, intm , intn, int l, float *V, int incv, floattau, float * C , int ldc);
void slarz_64 (charside, long m, long n, long l, float *v, long incv, float tau, float * C , long ldc);

\section*{PURPOSE}
slarz applies a realelem entary reflector \(H\) to a realM -by-N \(m\) atrix \(C\), from either the left or the right. \(H\) is represented in the form
\[
H=I-\tan ^{\star} \mathrm{v}^{\star} \mathrm{v}^{\prime}
\]
\(w\) here tau is a realscalar and \(v\) is a realvector.

If tau \(=0\), then \(H\) is taken to be the unitm atrix.
\(H\) is a product of \(k\) elem entary reflectors as retumed by STZRZF.

\section*{ARGUMENTS}

SIDE (input)
\(=L^{\prime}:\) form \(H * C\)
\(=R\) ': form \(C * H\)

M (input) The num ber of row s of the \(m\) atrix \(C\).

N (input) The num ber of colum ns of the m atrix C .

L (input) The num ber ofentries of the vector \(V\) containing the \(m\) eaningful part of the \(H\) ouseholdervectors. If \(S \mathbb{D} E=L ', M>=L>=0\), if \(S \mathbb{D} E=R \prime, N>=L\) \(>=0\) 。

V (input) The vector v in the representation of H as retumed by STZRZF.V is notused ifTA \(U=0\).
\(\mathbb{N} C V\) (input)
The increm entbetw een elem ents ofv. \(\mathbb{N} C V\) <> 0.

TAU (input)
The value tau in the representation of \(H\).
C (input/output)
On entry, the M -by -N m atrix C. On exit, C is overw ritten by the \(m\) atrix \(H\) * \(C\) if \(S \mathbb{D} E=L\) ', or C * H if \(S \mathbb{D} E=R\) '.
LDC (input)
The leading dim ension of the array C.LD C >= \(m a x(1, M)\).

W ORK (w orkspace)
\((\mathbb{N})\) if \(S \mathbb{D} E=L^{\prime}\) or \(\left.M\right)\) if \(S \mathbb{D} E=R^{\prime}\)

\section*{FURTHER DETAILS}

B ased on contributions by
A. Petitet, C om puterScience D ept., U niv . of Tenn., K noxville, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
slarzb - applies a realblock reflectort or its transpose \(\mathrm{H} * * \mathrm{~T}\) to a real distributed M -by -N C from the leftor the right

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SLARZB (S\mathbb{DE,TRANS,D IRECT,STOREV,M,N,K,L,V,LDV,T,}}\mathbf{M},\textrm{L},\textrm{L}
LDT,C,LDC,W ORK,LDW ORK)
CHARACTER * 1SDE,TRANS,D IRECT,STOREV
\mathbb{NTEGER M,N,K,L,LDV,LDT,LDC,LDW ORK}
REALV (LDV,*),T (LDT,*),C (LDC,*),W ORK (LDW ORK,*)
SUBROUT\mathbb{NE SLARZB_64 (SDE,TRANS,D RECT,STOREV,M,N,K,L,V,LDV,}
T,LDT,C,LDC,W ORK,LDW ORK)
CHARACTER * 1S\mathbb{E,TRANS,D IRECT,STOREV}
\mathbb{NTEGER*8M ,N,K,L,LDV,LDT,LDC,LDW ORK}
REALV (LDV,*),T (LDT,*),C (LDC,*),W ORK (LDW ORK,*)

```

\section*{F95 INTERFACE}

SU BROUTINE LARZB (SDE,TRANS,D \(\mathbb{R E C T}, \operatorname{STOREV}, \mathbb{M}], \mathbb{N}], K, L, V,[L D V]\), T, [LDT], C, [LDC], [W ORK], [LDW ORK])

CHARACTER (LEN=1) ::SDE,TRANS,D \(\mathbb{R E C T}, S T O R E V\)
\(\mathbb{N} T E G E R:: M, N, K, L, L D V, L D T, L D C, L D W\) ORK
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::V,T,C,W ORK
SU BROUTINE LARZB_64 (SDE,TRANS,D \(\mathbb{R E C T}, S T O R E V, \mathbb{M}], \mathbb{N}], K, L, V\), [LDV],T, [LDT], C, [LDC], [W ORK], [LDW ORK])

CHARACTER (LEN=1) ::SDE,TRANS,D \(\mathbb{R E C T}\), STO REV
\(\mathbb{N}\) TEGER (8) ::M , N , K, L,LDV,LD T,LDC,LDW ORK
REAL,D IM ENSION (:,:) ::V,T,C,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void slarzb (char side, char trans, chardirect, char storev, int \(m\), int \(n\), int \(k\), int \(l\), float *v, int ldv, float *t, int ldt, float * c , int ldc, int ldw ork);
void slarzb_64 (charside, char trans, char direct, char storev, long m, long \(n\), long \(k\), long ll, float * v , long ldv, float *t, long ldt, float *c, long ldc, long ldw ork);

\section*{PURPOSE}
slarzb applies a realblock reflector H or its transpose \(H * * T\) to a realdistributed \(M\) boy -N C from the leftorthe right.

C unently, only STOREV = R'and D \(\mathbb{R E C T}=\mathrm{B}\) 'are supported.

\section*{ARGUMENTS}

STDE (input)
= L ': apply H orH 'from the Left
= R': apply H orH 'from the Right

TRANS (input)
= N ': apply H N o transpose)
= C': apply H ' (I ranspose)

D \(\mathbb{R E C T}\) (input)
Indicates how \(H\) is form ed from a product of ele-
\(m\) entary reflectors = F':H = H (1) H (2) ...H (k)
(Forw ard, not supported yet)
\(=B^{\prime}: H=H(k) \ldots H(2) H(1)(B a c k w a r d)\)

STOREV (input)
Indicates how the vectors which define the elem en-
tary reflectors are stored:
= C ':Colum nw ise (notsup-
ported yet)
= R : R Row wise

M (input) The num ber of row sof the \(m\) atrix \(C\).

N (input) The num ber of colum ns of the m atrix C .

K (input) The order of the \(m\) atrix \(\mathrm{T} \vDash\) the num ber of elem entary reflectors whose product defines the block reflector).

L (input) The num berof colum ns of the \(m\) atrix \(V\) containing the \(m\) eaningfulpart of the \(H\) ouseholder reflectors. If \(S \mathbb{D} E=L{ }^{\prime}, M>=L>=0\), if \(S \mathbb{D} E=R \prime, N>=L\) \(>=0\).
V (input) IfSTOREV = \(\mathrm{C}^{\prime}, \mathrm{NV}=\mathrm{K}\); if \(S T O R E V=\mathrm{R}, \mathrm{NV}=\mathrm{L}\).

LD V (input)
The leading dim ension of the array V . If \(\mathrm{STOREV}=\) C',LDV >= L; ifSTOREV = R',LDV >=K.

T (input) The triangularK -by K m atrix T in the representation of the block reflector.

LD T (input)
The leading dim ension of the array T.LD \(T>=K\).

C (input/output)
On entry, the \(M\) boy -N m atrix C. On exit, C is overw ritten by \(\mathrm{H} * \mathrm{C}\) orH \({ }^{*} \mathrm{C}\) orC *H orC *H '.

LD C (input)
The leading dim ension of the aray C. LD C >= max (1,M).

W ORK (w orkspace)
dim ension (MAX \(M, N\) ) , K )
LDW ORK (input)
The leading dim ension of the array \(W\) ORK. If \(S \mathbb{D} E\)
\(=\mathrm{L}\) ', LDW ORK >=max ( \(1, \mathrm{~N}\) ); ifS \(\mathbb{D} E=\mathrm{R}\) ',LDW ORK \(>=\max (1, M)\).

\section*{FURTHER DETAILS}

B ased on contributions by
A.Petitet, C om puterScience D ept., U niv . of Tenn., K noxville, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
slarzt-form the triangular factor \(T\) of a real block reflector H of order> \(n\), which is defined as a product ofk elem entary reflectors

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SLARZT D RECT,STOREV,N,K,V,LDV,TAU,T,LDT)}
CHARACTER * 1D RECT,STOREV
INTEGERN,K,LDV,LDT
REALV (LDV,*),TAU (*),T (LDT,*)
SUBROUT\mathbb{NE SLARZT_64D RRECT,STOREV,N,K,V,LDV,TAU,T,LDT)}
CHARACTER * 1D RRECT,STOREV
INTEGER*8N,K,LDV,LDT
REAL V (LDV ,*),TAU (*),T (LDT,*)

```
F95 INTERFACE
    SUBROUT \(\mathbb{N} E\) LARZT (D \(\mathbb{R E C T}, \operatorname{STOREV}, \mathrm{N}, \mathrm{K}, \mathrm{V},[\mathrm{LDV}], T A U, T,[L D T])\)
    CHARACTER (LEN=1) ::D \(\mathbb{R E C T}\),STOREV
    \(\mathbb{N} T E G E R:: N, K, L D V, L D T\)
    REAL,D \(\mathbb{I M}\) ENSION (:) ::TAU
    REAL,D \(\mathbb{M}\) ENSION (:,:)::V,T
    SU BROUTINE LARZT_64 D \(\mathbb{R E C T}, \operatorname{STOREV}, N, K, V,[L D V], T A U, T,[L D T])\)
    CHARACTER (LEN=1) ::D \(\mathbb{R E C T}, S T O R E V\)
    \(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{K}, \mathrm{LDV}\),LDT
    REAL,D \(\mathbb{M}\) ENSION (:) ::TAU

\section*{C INTERFACE}
\#include <sunperfh>
void slarzt(chardirect, charstorev, intn, int \(k\), float \({ }^{*}\) v, int ldv, float *tau, float *t, int ldt);
void slarzt 64 (chardirect, charstorev, long n, long k, float *V, long ldv, float*tau,float*t, long ldt);

\section*{PURPOSE}
slarzt form s the triangular factor \(T\) of a realblock reflector \(H\) of order \(>\mathrm{n}\), which is defined as a product of \(k\) elem entary reflectors.

IfD \(\mathbb{R E C T}=\mathrm{F}^{\prime}, \mathrm{H}=\mathrm{H}(1) \mathrm{H}(2) \ldots \mathrm{H}(\mathrm{k})\) and T is upper triangular;

IfD \(\mathbb{R E C T}=\mathrm{B}^{\prime}, \mathrm{H}=\mathrm{H}(\mathrm{k}) \ldots \mathrm{H}(2) \mathrm{H}(1)\) and T is lower triangular.

IfSTOREV = C', the vector which defines the elem entary reflector \(H\) ( \(i\) ) is stored in the i-th colum \(n\) of the array \(V\), and
\[
H=I-V * T * V^{\prime}
\]

IfSTOREV = R', the vector which defines the elem entary reflectorH (i) is stored in the \(i\)-th row of the array \(V\), and
\[
\mathrm{H}=\mathrm{I}-\mathrm{V}^{\prime} \star \mathrm{T} * \mathrm{~V}
\]

Currently, only STOREV = R'and D \(\mathbb{R E C T}=\mathrm{B}\) 'are supported.

\section*{ARGUMENTS}

D \(\mathbb{R E C T}\) (input)
Specifies the order in which the elem entary
reflectors are multiplied to form the block
reflector:
\(=F\) : H = H (1) H (2) . . H (k) Forw ard, notsup-
ported yet)
\(=B^{\prime}: H=H(k) \ldots H(2) H(1)\) Backw ard)

\section*{STOREV (input)}

Specifies how the vectors w hich define the elem entary reflectors are stored (see also Further
D etails):
= R ': row w ise
N (input) The order of the block reflector \(\mathrm{H} . \mathrm{N}>=0\).
\(K\) (input) The order of the triangular factor \(T \vDash\) the num ber of elem entary reflectors). \(\mathrm{K}>=1\).
\(V\) (input) ( \(L D V, K\) ) if \(S T O R E V=C^{\prime}(L D V, N)\) if \(S T O R E V=R^{\prime}\) Them atrix \(V\). See furtherdetails.

LD V (input)
The leading dim ension of the array V . If \(\operatorname{STOREV}=\) \(C\) ', LDV \(>=\max (1, N)\); ifSTOREV = \(R\) ', LDV \(>=K\).

TAU (input)
TAU (i) must contain the scalar factor of the elem entary reflectort (i).
\(T\) (input) The \(k\) by \(k\) triangular factor \(T\) of the block reflector. If \(\mathrm{D} \mathbb{R E C T}=\mathrm{F}^{\prime}, \mathrm{T}\) is upper triangular; if \(\mathrm{D} \mathbb{R E C T}=\mathrm{B}\) ', T is low er triangular. The restof the aray is notused.

LD T (input)
The leading dim ension of the array T.LD T >=K.

\section*{FURTHER DETAILS}

B ased on contributions by
A. Petitet, C om puter Science D ept., U niv. of Tenn., K noxville, U SA

The shape of the \(m\) atrix \(V\) and the storage of the vectors which define the \(H\) (i) is bestillustrated by the follow ing exam ple w th \(\mathrm{n}=5\) and \(\mathrm{k}=3\). The elem ents equal to 1 are not stored; the comesponding aray elem ents arem odified but restored on exit. The restof the array is notused.

D \(\mathbb{R E C T}=\mathrm{F}^{\prime}\) and STOREV \(=\mathrm{C}^{\prime}: \quad \mathrm{D} \mathbb{R E C T}=\mathrm{F}^{\prime}\) and STOREV = R':
```

                                    V
    ```

```

(v1 v2 v3) (v1 v1 v1 v1 v1 ···..1
)
V = (v1 v2 v3 ) (v2 v2 v2 v2 v2 .

```
```

..1 )
(v1 v2 v3 ) (v3 v3 v3 v3 v3 .
.1 )
(v1 v2 v3 )
. . .
1..
1.
1
D\mathbb{RECT}=\mp@subsup{B}{}{\prime}\mathrm{ 'andSTOREV = C': D PRECT = B' and}
STOREV = R':
1

```

```

    . 1
        (1 . . ..v1 v1 v1 v1 v1 )
        . . }
    v2 v2 v2 )
...
v3 v3 v3 )
•••
(v1 v2 v3 )
V = (v1 v2 v3)
(v1 v2 v3 )

```

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
slasit-the num bers in \(D\) in increasing order (if \(\mathbb{D}=\mathrm{I}\) ) or in decreasing order (if \(\mathbb{D}=D^{\prime}\) )

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SLASRT(\mathbb{D},N,D,\mathbb{NFO)}}\mathbf{N}=()
CHARACTER * 1\mathbb{D}
\mathbb{NTEGERN, INFO}
REALD(*)

```

```

CHARACTER * 1 \mathbb{D}
INTEGER*8N,INFO
REALD (*)

```
F95 INTERFACE
    SUBROUTINE LASRT ( \(\mathbb{D}, \mathbb{N}], D,[\mathbb{N} F O]\) )
    CHARACTER (LEN=1) :: \(\mathbb{D}\)
    \(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
    REAL,D \(\mathbb{M}\) ENSION (:) ::D
    SU BROUTINE LASRT_64 ( \(\mathbb{D}, \mathbb{N}], D,[\mathbb{N} F O])\)
    CHARACTER (LEN=1) :: \(\mathbb{D}\)
    \(\mathbb{N} T E G E R(8):: N, \mathbb{N} F O\)
    REAL,D \(\mathbb{M}\) ENSION (:) ::D
C INTERFACE
    \#include <sunperfh>
void slastt(char id, intn, float *d, int *info);
void slasit 64 (char id, long n, float *d, long *info);

\section*{PURPOSE}
slast the num bers in \(D\) in increasing order (if \(\mathbb{D}=\mathrm{I}\) ) or in decreasing order (if \(\mathbb{D}=D^{\prime}\) ).

U se Q uick Sort, reverting to Insertion sort on arrays of size \(<=20\). D im ension of STA CK \(\lim\) its N to about \(2 * * 32\).

\section*{ARGUMENTS}

ID (input)
= I': sortD in increasing order;
\(=D^{\prime}\) ': sortD in decreasing order.

N (input) The length of the aray D .

D (input/output)
On entry, the array to be sorted. On exit, D has
been sorted into increasing order \((\mathbb{D}(1)<=\ldots<=\)
D (N ) ) or into decreasing order (D (1) >= ... >=
\(D(\mathbb{N})\) ), depending on \(\mathbb{D}\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
slatzm -routine is deprecated and has been replaced by routine SO RM RZ

\section*{SYNOPSIS}

```

CHARACTER * 1 SDE
INTEGERM,N,\mathbb{NCV,LDC}
REAL TAU
REAL V (*),C1 (LD C ,*),C2 (LD C ,*),W ORK (*)
SUBROUT\mathbb{NE SLATZM _64(S\mathbb{DE,M,N,V,INCV,TAU,C1,C2,LDC,W ORK)}}\mathbf{N},\textrm{N},\textrm{N}
CHARACTER * 1SDEE
INTEGER*8M,N,\mathbb{NCV,LDC}
REAL TAU
REALV (*),C1 (LDC ,*),C2 (LD C ,*),W ORK (*)

```

\section*{F95 INTERFACE}

SUBROUTINE LATZM (SDE, \(\mathbb{M}], \mathbb{N}], V,[\mathbb{N} C V], T A U, C 1, C 2,[L D C],[\mathbb{O R K}])\)

CHARACTER (LEN=1) ::SDE
\(\mathbb{N} T E G E R:: M, N, \mathbb{N C V}\),LDC
REAL ::TAU
REAL,D \(\mathbb{I M}\) ENSION (:) ::V,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::C1,C2
SU BROUTINE LATZM_64 (SDE, \(\mathbb{M}], \mathbb{N}], V,[\mathbb{N C V}], T A U, C 1, C 2,[\operatorname{DC}]\), [ W ORK])

CHARACTER (LEN=1)::SDE
\(\mathbb{N}\) TEGER (8) :: M , N , \(\mathbb{N} C V, L D C\)
REAL ::TAU
REAL,D \(\mathbb{I M} E N S I O N(:):: V\),W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::C1,C2

\section*{C INTERFACE}
\#include <sunperfh>
void slatrm (char side, intm, intn, float *v, int incv, float tau, float * c1, float *c2, int ldc);
void slatrm _64 (charside, long m, long n, float *v, long incv, float tau, float * c1, float * c2, long ldc);

\section*{PURPOSE}
slatzm routine is deprecated and has been replaced by routine SO RM RZ.

SLA TZM applies a H ouseholderm atrix generated by STZRQF to a \(m\) atrix.

LetP \(=I-\tan ^{\star} u^{\star} u^{\prime}, u=(1)\),
(v)
where \(v\) is an ( \(m-1\) ) vector if \(S \mathbb{D} E=\mathbb{L}\) ', ora ( \(n-1\) ) vector if \(S \mathbb{D} E=R\).

IfS \(\mathbb{D}\) E equals L ', let
\(C=[C 1] 1\)
[C2]m-1
n
Then C is overw rilten by P *C .

If \(S \mathbb{D}\) E equals R', let
\(C=[C 1, C 2] m\)
1 n-1
Then C is overw ritten by C * P .

\section*{ARGUMENTS}
```

SID E (input)
= L': form P * C
= R': form C * P

```

M (input) The num ber of row s of the m atrix C.

N (input) The num ber of colum ns of the \(m\) atrix \(C\).
\(V\) (input) \((1+\mathbb{M}-1) * a b s(\mathbb{N C V}))\) if \(S \mathbb{D} E=\mathbb{L}^{\prime}(1+\mathbb{N}-\) \(1) * a b s(\mathbb{N} C V))\) if \(S \mathbb{D} E=R\) 'The vectorv in the representation ofP. \(V\) is notused if \(T A U=0\).
\(\mathbb{N} C V\) (input)
The increm entbetw een elem ents ofv. \(\mathbb{N} C V<>0\)

TAU (input)
The value tau in the representation ofP.

C1 (input/output)
\((\mathrm{LDC}, \mathrm{N})\) if \(S \mathbb{D} E=\mathrm{L}^{\prime}(\mathrm{M}, \mathbf{1})\) if \(S \mathbb{D} E=\mathrm{R}^{\prime} \mathrm{On}\) entry, the \(n\)-vector \(C 1\) if \(S \mathbb{D} E=L\) ', orthemvectorC 1 if \(S \mathbb{D} E=R\).

On exit, the first row of \({ }^{*} C\) if \(S \mathbb{D} E=\mathrm{L}\) ', or the first colum \(n\) of \(C\) * \(P\) if \(S I D E=R\) '.

C2 (input/output)
\((L D C, N)\) if \(S \mathbb{D} E=L^{\prime}(L D C, N-1)\) if \(S \mathbb{D} E=R^{\prime}\)
On entry, the ( \(m-1\) ) xnmatrix \(C 2\) if \(S \mathbb{D} E=L\) ', or them \(x(n-1) m\) atrix \(C 2\) if \(S D E=R\) '.

On exit, rows 2 m ofP* C if \(S \mathbb{D} \mathrm{E}=\mathrm{L}\) ', orcolum ns 2 m of \(\mathrm{C} * \mathrm{P}\) if \(S \mathbb{D} E=\mathrm{R}^{\prime}\).

LD C (input)
The leading dim ension of the arrays C 1 and C 2.LD C \(>=(1, \mathrm{M})\).

W ORK (w orkspace)
\((\mathbb{N})\) if \(S \mathbb{D} E=L^{\prime}(M)\) if \(S \mathbb{D} E=R^{\prime}\)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
snm 2 -Retum the Euclidian norm of a vector.

\section*{SYNOPSIS}
```

REALFUNCTION SNRM 2N,X,INCX)
\mathbb{NTEGER N, INCX}
REALX (*)
REAL FUNCTION SNRM 2_64 N,X,\mathbb{NCX)}
INTEGER*8N,\mathbb{NCX}
REALX (*)
F95 INTERFACE
REAL FUNCTION NRM 2(N ],X,[\mathbb{NCX ])}
INTEGER ::N,\mathbb{NCX}
REAL,D IM ENSION (:) ::X
REAL FUNCTION NRM 2_64 (N ],X,[\mathbb{NCX])}
\mathbb{NTEGER (8) ::N,\mathbb{NCX}}\mathbf{N}=\mp@code{N}
REAL,D IM ENSION (:) ::X
C INTERFACE
\#include <sunperfh>
float snmm 2 (intn, float*x, int incx);

```
    floatsnm 2_64 (long n, float *x, long incx);

\section*{PURPOSE}
snm 2 Retum the Euclidian norm of a vector \(x\) where \(x\) is an n -vector.

\section*{ARGUMENTS}

N (input)
O n entry, \(N\) specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.
X (input)
\((1+(n-1) * a b s(\mathbb{N} C X))\). On entry, the increm ented array X m ust contain the vectorx. U nchanged on exit.
\(\mathbb{N} C X\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X\) m ustbe positive. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sopgtr-generate a real orthogonal matrix \(Q\) which is defined as the product ofn-1 elem entary reflectors \(H\) (i) of ordern, as retumed by SSPTRD using packed storage

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SOPGTR(UPLO,N,AP,TAU,Q,LDQ,W ORK,INFO)}
CHARACTER * 1 UPLO
NNTEGERN,LDQ,\mathbb{NFO}
REALAP(*),TAU(*),Q (LDQ ,*),W ORK (*)
SUBROUT\mathbb{NE SOPGTR_64(UPLO,N,AP,TAU,Q,LDQ,W ORK,INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N,LDQ,INFO}
REALAP(*),TAU (*),Q (LDQ ,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE OPGTR (UPLO, \(\mathbb{N}], A P, T A U, Q,[L D Q],[W O R K],[\mathbb{N} F O])\)
CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathrm{LD} Q, \mathbb{N} F O\)
REAL,D IM ENSION (:) ::AP,TAU,W ORK
REAL,D IM ENSION (:,:)::Q

SU BROU T INE OPG TR_64 (UPLO, \(\mathbb{N}], A P, T A U, Q,[L D Q],[\mathbb{N} O R K],[\mathbb{N} F O])\)
CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} Q, \mathbb{N}\) FO
REAL,D \(\mathbb{M}\) ENSION (:) ::AP,TAU,W ORK
REAL,D \(\mathbb{I}\) ENSION (:,:) ::Q

\section*{C INTERFACE}
\#include <sunperfh>
void sopgtr(charuplo, intn, float *ap, float *tau, float
*q, int ldq, int *info);
void sopgtr_64 (charuplo, long n, float *ap, float *tau, float *q, long ldq, long *info);

\section*{PURPOSE}
sopgtrgenerates a realorthogonalm atrix Q which is defined as the product of n-1 elem entary reflectors \(H\) (i) ofordern, as retumed by SSPTRD using packed storage:
if \(U P L O=U ', Q=H(n-1) \ldots H(2) H(1)\),
if \(U P L O=L^{\prime}, Q=H(1) H(2) \ldots H(n-1)\).

\section*{ARGUMENTS}

UPLO (input)
= U ':U pper triangular packed storage used in previous call to SSPTRD; = L ': Low er triangular packed storage used in previous call to SSPTRD .

N (input) The order of the matrix \(\mathrm{Q} . \mathrm{N}>=0\).
AP (input)
The vectors w hich define the elem entary reflectors, as retumed by SSPTRD .

TAU (input)
TAU (i) m ust contain the scalar factor of the elem entary reflectort (i), as retumed by SSPTRD.

Q (output)
The \(N\)-by -N orthogonalm atrix Q .
LD Q (input)
The leading dim ension of the array \(\mathrm{Q} . \mathrm{LD} \mathrm{Q}>=\) \(\max (1, N)\).

W ORK (w orkspace)
dim ension ( \(\mathbb{N}-1\) )
\(\mathbb{N} F O\) (output)
= 0: successfulexit
< 0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sopm tr-overw rite the general real \(M\)-by N matrix C with \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R{ }^{\prime} T R A N S=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUTINE SOPM TR (S\mathbb{DE,UPLO,TRANS,M,N,AP,TAU,C,LDC,W ORK,}
\mathbb{NFO)}
CHARACTER * 1SIDE,UPLO,TRANS
\mathbb{NTEGERM,N,LDC,INFO}
REALAP (*),TAU (*),C (LDC,*),W ORK (*)
SU BROUTINE SOPM TR_64 (SDDE,UPLO,TRANS,M ,N,AP,TAU,C,LDC,W ORK,
\mathbb{NFO)}

```
CHARACTER * 1 SIDE, UPLO, TRANS
\(\mathbb{N}\) TEGER*8 \(\mathrm{M}, \mathrm{N}, \mathrm{LD} \mathrm{C}, \mathbb{N}\) FO
REALAP (*),TAU (*), C (LDC, \(\left.{ }^{*}\right), \mathrm{W} O R K(*)\)

\section*{F95 INTERFACE}

SU BROUTINE OPM TR (SDE, UPLO, [TRANS], \(\mathbb{M}], \mathbb{N}], A P, T A U, C,[L D C]\), \([\mathbb{W} O R K],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::SDE,UPLO,TRANS
\(\mathbb{N} T E G E R:: M, N, L D C, \mathbb{N F O}\)
REAL,D IM ENSION (:) ::AP,TAU,W ORK
REAL,D IM ENSION (:,:) ::C
SU BROUTINE OPM TR_64 (SIDE,UPLO, [TRANS], \(\mathbb{M}], \mathbb{N}], A P, T A U, C,[L D C]\), [ \(\mathrm{W} O \mathrm{RK}\) ], [ \(\mathbb{N} F \mathrm{FO}\) )

CHARACTER (LEN=1) ::SDE,UPLO,TRANS
\(\mathbb{N}\) TEGER (8) :: M , N, LD C , \(\mathbb{N}\) FO
REAL,D \(\mathbb{I M}\) ENSION (:) ::AP,TAU,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::C

\section*{C INTERFACE}
\#include <sunperfh>
void sopm tr (char side, char uplo, char trans, intm, int n, float *ap, float *tau, float *\({ }^{\text {c }}\), intldc, int *info);
void sopm tr_64 (charside, charuple, char trans, long m, long n, float *ap, float *tau, float * c, long ldc, long *info);

\section*{PURPOSE}
sopm troverw rites the general real \(M\) boy \(N \mathrm{~N}\) m atrix \(C\) w ith

where \(Q\) is a real orthogonalm atrix of order nq, w ith nq = m if \(S \mathbb{D} E=L\) 'and \(n q=n\) if \(S \mathbb{D} E=R\) '. Q is defined as the product of nq-1 elem entary reflectors, as retumed by SSP TRD using packed storage:
if \(U P L O=U\) ', \(Q=H(n q-1) \ldots H(2) H(1) ;\)
if \(U P L O=L^{\prime}, Q=H(1) H(2) \ldots H(n q-1)\).

\section*{ARGUMENTS}

SIDE (input)
\(=\mathrm{L}\) ': apply Q or \(\mathrm{Q} * * \mathrm{~T}\) from the Left;
\(=R\) ': apply Q or \(\mathrm{Q} * * \mathrm{~T}\) from the R ight.

UPLO (input)
\(=\mathrm{U}\) ': U ppertriangular packed storage used in previous call to SSPTRD;= L':Low er triangular packed storage used in previous call to SSPTRD .

TRANS (input)
\(=\mathrm{N}: \mathrm{N}\) o transpose, apply Q ;
\(=T\) ': T ranspose, apply \(Q * * T\).

TRANS is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

M (input) The num ber of row s of the \(m\) atrix \(C . M>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\).
AP (input)
\((M *(M+1) / 2)\) if \(S D E=L(\mathbb{N} *(N+1) / 2)\) if \(S D E=\)
\(R\) ' The vectors which define the elem entary
reflectors, as retumed by SSP TRD . A P ism odified
by the routine but restored on exit.
TAU (input)
or \((\mathbb{N}-1)\) if \(S \mathbb{D} E=R^{\prime} T A U(i)\) must contain the scalar factor of the elem entary reflectorH (i), as retumed by SSPTRD.
C (input/output)
On entry, the M by -N m atrix C. On exit, C is overw ritten by Q * C or \(\mathrm{Q} * \mathrm{~T}^{*} \mathrm{C}\) or \(\mathrm{C}^{*} \mathrm{Q} * * \mathrm{~T}\) or \(\mathrm{C}{ }^{\mathrm{Q}} \mathrm{Q}\).

LD C (input)
The leading dim ension of the array C.LD C >= \(\max (1, \mathrm{M})\).

W ORK (w orkspace)
\((\mathbb{N})\) if \(S \mathbb{D} E=L^{\prime}(M)\) if \(S \mathbb{D} E=R^{\prime}\)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an ille-
galvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorg2l-generate an \(m\) by \(n\) realm atrix \(Q w\) th orthonorm al colum ns,

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SORG2L M,N,K,A,LDA,TAU,W ORK, INFO)}
INTEGERM,N,K,LDA, INFO
REALA (LDA,*),TAU (*),W ORK (*)
SUBROUT\mathbb{NE SORG2L_64M,N,K,A,LDA,TAU,W ORK,INFO)}
INTEGER*8M,N,K,LDA, INFO
REALA (LDA,*),TAU (*),W ORK (*)
F95 INTERFACE
SU BROUT\mathbb{NE ORG 2L (\mathbb{M ], N ], [K ],A , [LDA ],TAU, [W ORK ], [NFO ])}}\mathbf{N}\mathrm{ )}
\mathbb{NTEGER ::M,N,K,LDA,}\mathbb{N}FO
REAL,D IM ENSION (:) ::TAU,W ORK
REAL,D IM ENSION (:,:) ::A

```

```

    \mathbb{NTEGER (8)::M,N,K,LDA, NNFO}
    REAL,DIM ENSION (:) ::TAU,W ORK
    REAL,D IM ENSION (:,:) ::A
    C INTERFACE
\#include <sunperfh>
void sorg2l(intm, intn, intk, float*a, int Ida, float

```
void sorg2l_64 (long m, long n, long k, float*a, long lda, float *tau, long *info);

\section*{PURPOSE}
sorg2lL generates an \(m\) by \(n\) realm atrix \(Q w\) th orthonorm al colum ns, which is defined as the lastn colum ns of a product ofk elem entary reflectors of orderm
\[
Q=H(k) \ldots H(2) H(1)
\]
as retumed by SG EQ LF .

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(Q . M>=0\).

N (input) The num ber of colum ns of the matrix \(\mathrm{Q} . \mathrm{M}>=\mathrm{N} \quad>=\) 0.
\(K\) (input) The num ber of elem entary reflectors \(w\) hose product defines the m atrix \(\mathrm{Q} . \mathrm{N}>=\mathrm{K}>=0\).

A (input/output)
On entry, the \((n-k+i)\)-th colum nm ust contain the vector which defines the elem entary reflector H (i), for \(i=1,2, \ldots, k\), as retumed by SGEQ LF in the last k colum ns of its array argum entA. On exit, the \(m\) by \(n m\) atrix \(Q\).

LD A (input)
The first dimension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by SGEQ LF .

W ORK (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N}\) FO (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthas an illegalvahue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorg \(2 r\)-generate \(a n m\) by \(n\) realm atrix \(Q\) with orthonorm al colum ns,

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SORG2RM,N,K,A,LDA,TAU,W ORK, NNFO)}
INTEGERM,N,K,LDA, INFO
REALA (LDA,*),TAU (*),W ORK (*)
SUBROUTINE SORG 2R_64 M,N,K,A,LDA,TAU,W ORK,\mathbb{NFO)}
\mathbb{NTEGER*8 M,N,K,LDA,}\mathbb{N}FO
REALA (LDA,*),TAU (*),W ORK (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE ORG 2R ( $\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{N} O R K],[\mathbb{N} F O])$
$\mathbb{N} T E G E R:: M, N, K, L D A, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::TAU,W ORK
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A
SU BROUTINE ORG 2R_64 ( $\mathbb{M}], \mathbb{N}],[\mathbb{K}], A,[L D A], T A U,[\mathbb{W} O R K],[\mathbb{N} F O])$
$\mathbb{N} T E G E R(8):: M, N, K, L D A, \mathbb{N F O}$
REAL,D IM ENSION (:) ::TAU,W ORK
REAL,D IM ENSION (:,:) ::A

```

\section*{C INTERFACE}
```

\#include < sunperfh>
void sorg2r(intm, intn, intk, float*a, int Ida, float

```
void sorg2r_64 (long m, long n, long k, float*a, long lda, float*tau, long *info);

\section*{PURPOSE}
sorg \(2 r R\) generates an \(m\) by \(n\) realm atrix \(Q w i t h\) orthonorm al colum ns, which is defined as the firstn colum ns of a product of \(k\) elem entary reflectors of orderm
\[
Q=H(1) H(2) \ldots H(k)
\]
as retumed by SG EQ RF .

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(Q . M>=0\).

N (input) The num ber of colum ns of the matrix Q.M >= N >= 0.
\(K\) (input) The num ber of elem entary reflectors \(w\) hose product defines the m atrix \(\mathrm{Q} . \mathrm{N}>=\mathrm{K}>=0\).

A (input/output)
On entry, the \(i\)-th columnm ustcontain the vector which defines the elem entary reflectorH (i), for i
\(=1,2, \ldots, k\), as retumed by SGEQRF in the first \(k\)
colum ns of its array argum entA. On exit, them by \(-\mathrm{n} m\) atrix Q .

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by SGEQRF.

W ORK (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthas an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorgbr-generate one of the realorthogonalm atrioes Q or \(P * * T\) determ ined by SGEBRD when reducing a realm atrix \(A\) to bidiagonal form

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SORGBR(NECT,M,N,K,A,LDA,TAU,W ORK,LW ORK,INFO)}
CHARACTER * 1 VECT
INTEGER M,N,K,LDA,LW ORK,\mathbb{NFO}
REALA (LDA,*),TAU (*),W ORK (*)

```

```

CHARACTER * 1VECT
INTEGER*8M,N,K,LDA,LW ORK,\mathbb{NFO}
REALA (LDA,*),TAU (*),W ORK (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE ORGBR $\operatorname{VECT}, \mathrm{M}, \mathbb{N}], K, A,[L D A], T A U,[W O R K],[L W$ ORK ], [ $\mathbb{N}$ FO ])
CHARACTER (LEN=1)::VECT
$\mathbb{N}$ TEGER ::M,N,K,LDA,LW ORK, $\mathbb{N} F O$
REAL,D IM ENSION (:) ::TAU,W ORK
REAL,D $\mathbb{I}$ ENSION (:,:) ::A
SUBROUTINE ORGBR_64 $N E C T, M, \mathbb{N}], K, A,[L D A], T A U,[W O R K],[L W$ ORK ], [ $\mathbb{N} F O$ ])

```

CHARACTER (LEN=1) ::VECT
\(\mathbb{N} T E G E R(8):: M, N, K, L D A, L W O R K, \mathbb{N} F O\)

REAL,D \(\mathbb{I M} E N S I O N(:):\) TAU ,W ORK
REAL,D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void sorgbr(charvect, intm, intn, intk, float *a, int lda, float *tau, int *info);
void sorgbr_64 (charvect, long m, long \(n\), long \(k\), float *a, long lda, float *tau, long *info);

\section*{PURPOSE}
sorgbrgenerates one of the real orthogonal \(m\) atrices \(Q\) or \(P * * T\) determ ined by SGEBRD when reducing a realm atrix \(A\) to bidiagonal form \(: A=Q * B * P * * T . Q\) and \(P * * T\) are defined as products of elem entary reflectors H (i) orG (i) respectively.

IfVECT \(=Q\) ', \(A\) is assum ed to have been an \(M\) boy \(K m\) atrix, and Q is of orderM :
ifm \(>=k, Q=H(1) H(2) \ldots H(k)\) and \(S O R G B R\) retums the firstn colum ns of \(Q\), where \(m>=n>=k\);
ifm \(<k, Q=H(1) H(2) \ldots H(m-1)\) and \(S O R G B R\) retums \(Q\) as an M -by \(M\) m atrix.

IfVECT \(=P\) ', A is assum ed to have been a K -by -N m atrix, and \(\mathrm{P} * * \mathrm{~T}\) is oforderN:
if \(k<n, P^{* *} T=G(k) \ldots G(2) G(1)\) and \(S O R G B R\) retums the firstm row \(\operatorname{sofP}\) **T, where \(\mathrm{n}>=\mathrm{m}>=\mathrm{k}\); ifk \(>=n, P * * T=G(n-1) \ldots G(2) G(1)\) and \(S O R G B R\) retums \(\mathrm{P} * * \mathrm{~T}\) as an N -by -N m atrix.

\section*{ARGUMENTS}

VECT (input)
Specifies w hether the m atrix \(Q\) orthem atrix \(P * * T\)
is required, as defined in the transform ation
applied by SGEBRD :
= Q ': generate Q ;
\(=P\) : generate \(\mathrm{P} * * \mathrm{~T}\).

M (input) The num ber of row s of the \(m\) atrix \(Q\) orP**T to be retumed. \(\mathrm{M}>=0\).

N (input) The num ber of colum ns of the \(m\) atrix Q or \(\mathrm{P} * * T\) to
be retumed. \(\mathrm{N}>=0\). IfVECT \(=Q \mathrm{~V}^{\prime}, \mathrm{M}>=\mathrm{N}>=\) \(m\) in \((M, K) ;\) ifVECT \(=P^{\prime}, N>=M>=m\) in \((\mathbb{N}, K)\).
\(K\) (input) IfVECT = Q ', the num berof colum ns in the original \(M\) boy \(K m\) atrix reduced by \(S G E B R D\). IfVECT \(=\) \(P\) ', the num ber of row \(s\) in the original \(K\) boy \(N\) m atrix reduced by SG EBRD . \(\mathrm{K}>=0\).

A (input/output)
O n entry, the vectors w hich define the elem entary reflectors, as retumed by SGEBRD. On exit, the M łoy-N m atrix Q orP**T.

LD A (input)
The leading dim ension of the array \(\mathrm{A} . \operatorname{LDA}>=\) \(\max (1, M)\).

TAU (input)
\((m\) in \((\mathbb{M}, K))\) ifVECT = \(Q^{\prime}(m\) in \((\mathbb{N}, K))\) ifVECT \(=P^{\prime}\)
TAU (i) m ustcontain the scalar factor of the ele\(m\) entary reflectorH (i) orG (i), which determ ines Q or \(\mathrm{P} * * \mathrm{~T}\), as retumed by \(\mathrm{SG} E B R D\) in its array argu\(m\) entTAUQ orTAUP.

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the amay W ORK. LW ORK >= \(m\) ax \((1, m\) in \(M, N))\). Foroptim um perform ance LW ORK >= \(m\) in \((M, N) * N B, w h e r e N B\) is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the \(i\) th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorghr-generate a real orthogonal matrix \(Q\) which is defined as the productof \(\mathbb{H}\) I-HO elem entary reflectors of orderN, as retumed by SG EH RD

\section*{SYNOPSIS}


```

REAL A (LDA,*),TAU (*),W ORK (*)

```

```

\mathbb{NTEGER*8N,\mathbb{LO,\mathbb{HI},LDA,LW ORK,INFO}}\mathbf{N}=\mp@code{L}
REALA (LDA,*),TAU (*),W ORK (*)

```
F95 INTERFACE
SUBROUTINE ORGHR ( \(\mathbb{N}], \mathbb{I} O, \mathbb{H} I, A,[L D A], T A U,[W O R K],[L W O R K],[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathbb{L} \mathrm{O}, \mathbb{H} \mathrm{I}\), LDA, LW ORK, \(\mathbb{N} F O\)
REAL,D IM ENSION (:) ::TAU,W ORK
REAL,D IM ENSION (:,:) ::A
SU BROUTINE ORGHR_64 (N ], \(\mathbb{I} O, \mathbb{H} I, A,[L D A], T A U,[\mathbb{O R K}],[L W\) ORK ],
    [ \(\mathbb{N} F O\) ])
\(\mathbb{N} T E G E R(8):: N, \mathbb{L} O, \mathbb{H} I, L D A, L W O R K, \mathbb{N} F O\)
REAL,D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{I M}\) ENSION (: : : : : A

\section*{C INTERFACE}
\#include < sunperfh>
void sorghr(intn, intilo, int ini, float *a, int lda, float *tau, int *info);
void sorghr_64 (long n, long ilo, long ihi, float *a, long lda, float *tau, long *info);

\section*{PURPOSE}
sorghrgenerates a realorthogonalm atrix \(Q\) which is defined as the product of \(\mathbb{H}\) I-IIO elem entary reflectors of order \(N\), as retumed by SG EH RD :
\(Q=H\) (مlo) H ( \(\mathrm{il}+1\) ) . . . H (ini-1).

\section*{ARGUMENTS}

N (input) The order of the m atrix \(\mathrm{Q} . \mathrm{N}>=0\).

ILO (input)
IO 0 and \(\mathbb{H}\) Im usthave the sam e values as in the previous call of SGEHRD.Q is equal to the unit \(m\) atrix except in the subm atrix
Q ( \(\mathrm{N}>0 ;\) HO \(=1\) and \(\mathbb{H} \mathrm{F}=0\), if \(\mathrm{N}=0\).

IH I (input)
See the description of IIO .

A (input/output)
O n entry, the vectors w hich define the elem entary reflectors, as retumed by SGEHRD. On exit, the N boy -N orthogonalm atrix Q .

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflector \(H\) (i), as retumed by SGEHRD.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

The dim ension of the array \(W\) ORK. LW ORK >= \(\mathbb{H} I-\mathbb{H O}\). For optim um perform ance LW ORK \(>=(\mathbb{H} I-\mathbb{H} O)^{*} N B\), where NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim al size of theW ORK ancay, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
<0: if \(\mathbb{N F O}=-\) i, the \(i\)-th argum ent had an illegal value

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorgl2 -generate an \(m\) by \(n\) realm atrix \(Q \mathrm{w}\) th orthonorm al row S,

\section*{SYNOPSIS}

SUBROUTINE SORGL2 \(M, N, K, A, L D A, T A U, W O R K, \mathbb{N} F O\) )
\(\mathbb{N}\) TEGER \(\mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LD} \mathrm{A}, \mathbb{I N} F \mathrm{O}\)
REALA (LDA, *), TAU (*), W ORK (*)

SU BROUTINE SORGL2_64 M,N,K,A,LDA,TAU,WORK, \(\mathbb{N} F O\) )
\(\mathbb{N}\) TEGER*8M,N,K,LDA, \(\mathbb{N} F O\)
REALA (LDA,*),TAU (*),WORK (*)

\section*{F95 INTERFACE}

SU BROUTINE ORGL2 ( \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{N} O R K],[\mathbb{N} F O])\)
\(\mathbb{N} T E G E R:: M, N, K, L D A, \mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A
SU BROUTINE ORGL2_64 ( \(\mathbb{M}], \mathbb{N}],[\mathbb{K}], A,[L D A], T A U,[\mathbb{W} O R K],[\mathbb{N F O}])\)
\(\mathbb{N} T E G E R(8):: M, N, K, L D A, \mathbb{N F O}\)
REAL,D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
REAL,D \(\mathbb{I}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void sorg12 (intm, intn, intk, float *a, int lda, float
*tau, int *info);
void sorg12_64 (long m, long n, long k, float*a, long lda, float *tau, long *info);

\section*{PURPOSE}
sorgl2 generates an \(m\) by \(n\) realm atrix \(Q\) with orthonorm al row \(s\), which is defined as the firstm row s of a product ofk elem entary reflectors of ordern
\[
Q=H(k) \ldots H(2) H(1)
\]
as retumed by SG ELQ F.

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{Q} . \mathrm{M}>=0\).
N (input) The num ber of Colmm ns of the m atrix \(\mathrm{Q} . \mathrm{N}>=\mathrm{M}\).
\(K\) (input) The num ber of elem entary reflectors \(w\) hose product defines the \(m\) atrix \(Q . M>=K>=0\).

A (input/output)
On entry, the \(i\)-th row must contain the vector which defines the elem entary reflectort (i), for i \(=1,2, \ldots, k\), as retumed by SGELQF in the first \(k\) row sof its array argum entA. On exit, the \(m\)-by- \(n\) \(m\) atrix \(Q\).

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the elem entary reflectorH (i), as retumed by SGELQ F.

W ORK (w orkspace)
dim ension M )
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthas an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorglq - generate an \(M\) boy N realm atrix Q with orthonorm al row S,

\section*{SYNOPSIS}

SU BROUTINE SORGLQ \(M, N, K, A, L D A, T A U, W\) ORK,LDW ORK, \(\mathbb{N} F O\) )
\(\mathbb{I N}\) TEGER \(\mathrm{M}, \mathrm{N}, \mathrm{K}, \operatorname{LDA}, \operatorname{LDW} \mathrm{ORK}, \mathbb{N} F O\)
REALA (LDA, *), TAU (*), W ORK (*)

SU BROUTINE SORGLQ_64 \(M, N, K, A, L D A, T A U, W\) ORK,LDW ORK, \(\mathbb{N} F O\) )
\(\mathbb{N}\) TEGER* \(8 \mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathrm{LDW}\) ORK, \(\mathbb{N} F \mathrm{O}\)
REALA (LDA,*),TAU (*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE ORGLQ \(M, \mathbb{N}], \mathbb{K}], A,[L D A], T A U,[W O R K],[L D W O R K],[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER ::M,N,K,LDA,LDW ORK, \(\mathbb{N}\) FO
REAL,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A

SU BROUTINE ORGLQ_64 \(M, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{N} O R K]\), [LDW ORK ], [ \(\mathbb{N}\) FO ])
\(\mathbb{N}\) TEGER (8) ::M,N,K,LDA,LDW ORK, \(\mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{M}\) ENSION (: : : : : A

\section*{C INTERFACE}
\#include < sunperfh>
void sorglq (intm, intn, intk, float *a, int lda, float
*tau, int *info);
void sorglq_64 (long m, long n, long k, float*a, long lda, float *tau, long *info);

\section*{PURPOSE}
sorglq generates an \(M\) by -N realm atrix Q w ith orthonorm al row \(S\), which is defined as the firstM row s of a product of \(K\) elem entary reflectors of orderN
\[
Q=H(k) \ldots H(2) H(1)
\]
as retumed by SGELQF.

\section*{ARGUMENTS}

M (input) The num ber of row sof the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the \(m\) atrix \(Q \cdot N>=M\).
\(K\) (input) The num ber of elem entary reflectors whose product defines the \(m\) atrix \(Q . M>=K>=0\).

A (input/output)
On entry, the i-th row must contain the vector which defines the elem entary reflectorH (i), for i \(=1,2, \ldots, k\), as retumed by SG ELQF in the first k row sof its array argum entA. On exit, the \(M\) by \(-\mathbb{N}\) \(m\) atrix \(Q\).

LDA (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the elem entary reflectorH (i), as retumed by SG ELQF.

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the anay W ORK. LDW ORK >= \(\max (1, M)\). Foroptim um perform ance LDW ORK \(>=M * N B\),
w here N B is the optim al.blocksize.

IfLD W ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA .
\(\mathbb{N}\) FO (output)
= 0: successfiulexit
\(<0:\) if \(\mathbb{I N F O}=-i\), the \(i\) th argum enthas an illegalvałue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorgql-generate an M by-N realm atrix Q w th orthonorm al colum ns,

\section*{SYNOPSIS}

SU BROUTINE SORGQLM,N,K,A,LDA,TAU,WORK,LDWORK, \(\mathbb{N} F O\) )
\(\mathbb{I N}\) TEGER \(\mathrm{M}, \mathrm{N}, \mathrm{K}, \operatorname{LDA}, \operatorname{LDW} \mathrm{ORK}, \mathbb{N} F O\)
REALA (LDA, *), TAU (*), W ORK (*)

SUBROUTINE SORGQL_64 M,N,K,A,LDA,TAU,W ORK,LDW ORK, \(\mathbb{N} F O\) )
\(\mathbb{N}\) TEGER* \(8 \mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathrm{LDW}\) ORK, \(\mathbb{N} F \mathrm{~F}\)
REALA (LDA,*),TAU (*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE ORGQL M, \(\mathbb{N}], \mathbb{K}], A,[L D A], T A U,[W O R K],[L D W O R K],[\mathbb{N} F O])\)
\(\mathbb{N}\) TEGER ::M,N,K,LDA,LDW ORK, \(\mathbb{N}\) FO
REAL,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A

SU BROUTINE ORGQL_64 \(M, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{N} O R K]\), [LDW ORK ], [ \(\mathbb{N}\) FO ])
\(\mathbb{N}\) TEGER (8) ::M,N,K,LDA,LDW ORK, \(\mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{M}\) ENSION (: : : : ::A

\section*{C INTERFACE}
\#include < sunperfh>
void sorgql(intm, intn, intk, float*a, int lda, float
*tau, int *info);
void sorgql_ 64 (long \(m\), long \(n\), long k, float*a, long lda, float *tau, long *info);

\section*{PURPOSE}
sorgqlgenerates an M -by -N realm atrix Q w ith orthonorm al colum ns, which is defined as the lastN colum ns of a product of K elem entary reflectors of orderM
\[
Q=H(k) \ldots H(2) H(1)
\]
as retumed by SGEQ LF .

\section*{ARGUMENTS}

M (input) The num ber of row sof the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the matrix Q.M \(>=\mathrm{N} \quad>=\) 0.
\(K\) (input) The num ber of elem entary reflectors whose product defines the \(m\) atrix \(Q . N>=K>=0\).

A (input/output)
On entry, the \((n-k+i)\)-th colum nm ust contain the vector which defines the elem entary reflector H (i), for \(i=1,2, \ldots, k\), as retumed by SGEQ LF in the last \(k\) colum ns of its anay argum entA. On exit, the \(M\)-by-N \(m\) atrix \(Q\).

LD A (input)
The first dim ension of the array A. LDA >= \(m a x(1, M)\).

TAU (input)
TAU (i) \(m\) ust contain the scalar factor of the elem entary reflectorH (i), as retumed by SG EQ LF.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al
LDW ORK.
LDW ORK (input)
The dim ension of the anay W ORK. LDW ORK >=
\(m\) ax \((1, N)\). Foroptim um perform ance LD \(W\) ORK \(>=N * N B\), where NB is the optim alblocksize.

If LD W ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthas an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorgqr-generate an \(M\)-by-N realm atrix Q with orthonorm al colum ns,

\section*{SYNOPSIS}

SU BROUTINE SORGQRM,N,K,A,LDA,TAU,WORK,LDWORK, \(\mathbb{N} F O\) )
\(\mathbb{I N}\) TEGER \(\mathrm{M}, \mathrm{N}, \mathrm{K}, \operatorname{LDA}, \operatorname{LDW} \mathrm{ORK}, \mathbb{N} F O\)
REALA (LDA, *), TAU (*), W ORK (*)

SU BROUTINE SORGQR_64 M,N,K,A,LDA,TAU,WORK,LDWORK, \(\mathbb{N} F O\) )
\(\mathbb{N}\) TEGER* \(8 \mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathrm{LDW}\) ORK, \(\mathbb{N} F \mathrm{O}\)
REALA (LDA,*),TAU (*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINEORGQRM, \(\mathbb{N}],[K], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N F O}])\)
\(\mathbb{N} T E G E R:: M, N, K, L D A, L D W\) ORK, \(\mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A
SU BROUTINE ORGQR_64 \(\mathbb{M}, \mathbb{N}],[K], A,[L D A], T A U,[W O R K],[L D W O R K]\), [ \(\mathbb{N}\) FO ])
\(\mathbb{N}\) TEGER (8) ::M,N,K,LDA,LDW ORK, \(\mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{I}\) ENSION (: : : : : : A

\section*{C INTERFACE}
\#include < sunperfh>
void sorgqr(intm, intn, intk, float*a, int lda, float
*tau, int*info);
void sorgqr_64 (long m, long n, long k, float*a, long lda, float *tau, long *info);

\section*{PURPOSE}
sorgqrgenerates an M -by -N realm atrix Q w ith orthonorm al colum ns, which is defined as the firstN colum ns of a productof \(K\) elem entary reflectors of orderM
\[
Q=H(1) H(2) \ldots H(k)
\]
as retumed by SG EQ RF .

\section*{ARGUMENTS}

M (input) The num ber of row sof the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the matrix \(\mathrm{Q} . \mathrm{M}>=\mathrm{N} \quad>=\) 0.
\(K\) (input) The num ber of elem entary reflectors whose product defines the \(m\) atrix \(Q . N>=K>=0\).

A (input/output)
On entry, the \(i\)-th columnm ust contain the vector which defines the elem entary reflector \(H\) (i), for i
\(=1,2, \ldots, k\), as retumed by \(S G E Q R F\) in the first \(k\) colum ns of its array argum entA. On exit, the \(M-\) by-N matrix \(Q\).

LD A (input)
The first dim ension of the array A. LDA >= \(m a x(1, M)\).

TAU (input)
TAU (i) \(m\) ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by SG EQRF.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al
LDW ORK.
LDW ORK (input)
The dimension of the amay W ORK. LDW ORK >=
\(m\) ax \((1, N)\). Foroptim um perform ance LD \(W\) ORK \(>=N * N B\), where NB is the optim alblocksize.

If LD W ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthas an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorgn2 -generate an \(m\) by \(n\) realm atrix \(Q\) with orthonorm al row S,

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SORGR2M,N,K,A,LDA,TAU,W ORK, NNFO)}
INTEGERM,N,K,LDA, INFO
REALA (LDA,*),TAU (*),W ORK (*)
SUBROUT\mathbb{NE SORGR2_64M,N,K,A,LDA,TAU,W ORK,INFO)}
INTEGER*8M,N,K,LDA,\mathbb{NFO}
REALA (LDA,*),TAU (*),W ORK (*)
F95 INTERFACE
SUBROUT\mathbb{NE ORGR2 (\mathbb{M ], N ], [K ],A, [LDA ],TAU , [W ORK ], [NFO ])}}\mathbf{N}\mathrm{ )}
\mathbb{NTEGER ::M ,N,K,LDA,NNFO}
REAL,DIM ENSION (:) ::TAU,W ORK
REAL,D IM ENSION (:,:) ::A

```

```

    \mathbb{NTEGER (8)::M,N,K,LDA, NNFO}
    REAL,DIM ENSION (:) ::TAU,W ORK
    REAL,D IM ENSION (:,:) ::A
    C INTERFACE
\#include <sunperfh>
void sorgn2 (intm, intn, intk, float*a, int Ida, float

```
*tau, int *info);
void sorgr2_64 (long m, long n, long k, float *a, long lda, float *tau, long *info);

\section*{PURPOSE}
sorgr2 generates an \(m\) by \(n\) realm atrix \(Q\) with orthonorm al row \(s\), which is defined as the lastm row s of a product ofk elem entary reflectors of ordern
\[
Q=H(1) H(2) \ldots H(k)
\]
as retumed by SGERQF.

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{Q} . \mathrm{M}>=0\).
N (input) The num ber of Colm ns of the m atrix \(\mathrm{Q} . \mathrm{N}>=\mathrm{M}\).
\(K\) (input) The num ber of elem entary reflectors \(w\) hose product defines the m atrix \(\mathrm{Q} . \mathrm{M}>=\mathrm{K}>=0\).

A (input/output)
O \(n\) entry, the ( \(m-k+i\) )-th row \(m\) ustcontain the vector which defines the elem entary reflectorH (i), for \(i=1,2, \ldots, k\), as retumed by \(S G E R Q F\) in the lastk row sof its array argum entA. O n exit, the m by \(n\) matrix \(Q\).

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the elem entary reflectorH (i), as retumed by SG ERQ F.

W ORK (w orkspace)
dim ension M )
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthas an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorgrq - generate an \(M\)-by-N realm atrix \(Q\) with orthonorm al row S,

\section*{SYNOPSIS}

SU BROUTINE SORGRQ \(M, N, K, A, L D A, T A U, W O R K, L D W O R K, \mathbb{N} F O\) )
\(\mathbb{I N}\) TEGER \(\mathrm{M}, \mathrm{N}, \mathrm{K}, \operatorname{LDA}, \operatorname{LDW} \mathrm{ORK}, \mathbb{N} F O\)
REALA (LDA, *), TAU (*), W ORK (*)

SU BROUTINE SORGRQ_64 M,N,K,A,LDA,TAU,W ORK,LDW ORK, \(\mathbb{N} F O\) )
\(\mathbb{N}\) TEGER* \(8 \mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathrm{LDW}\) ORK, \(\mathbb{N} F \mathrm{~F}\)
REALA (LDA,*),TAU (*), W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE ORGRQ \(M, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N} F O])\)
\(\mathbb{N} T E G E R:: M, N, K, L D A, L D W\) ORK, \(\mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A
SU BROUTINE ORGRQ_64 \(\mathbb{M}, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{N} O R K]\), \(L D W\) ORK ], [ \(\mathbb{N}\) FO ])
\(\mathbb{N}\) TEGER (8) ::M,N,K,LDA,LDW ORK, \(\mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL,D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include < sunperfh>
void sorgrq (intm, intn, intk, float *a, int lda, float
*tau, int *info);
void sorgrq_64 (long m, long n, long k, float*a, long lda, float *tau, long *info);

\section*{PURPOSE}
sorgrq generates an M -by -N realm atrix Q w ith orthonorm al row S , w hich is defined as the lastM row s of a product of K elem entary reflectors of orderN
\[
Q=H(1) H(2) \ldots H(k)
\]
as retumed by SGERQF.

\section*{ARGUMENTS}

M (input) The num ber of row sof the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the \(m\) atrix \(Q . N>=M\).
\(K\) (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q . M>=K>=0\).

A (input/output)
O \(n\) entry, the ( \(m-k+i\) )-th row \(m\) ustcontain the vector which defines the elem entary reflectorH (i), fori=1,2,..,k, as retumed by SGERQF in the lastk row sof its array argum entA. O n exit, the M toy-N m atrix Q .

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by SGERQF.

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the amay W ORK. LDW ORK >= \(\max (1, M)\). Foroptim um perform ance LDW ORK \(>=M * N B\),
w here N B is the optim al.blocksize.

IfLD W ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA .
\(\mathbb{N}\) FO (output)
= 0: successfiulexit
\(<0:\) if \(\mathbb{I N F O}=-i\), the \(i\) th argum enthas an illegalvałue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorgtr-generate a real orthogonal matrix Q which is defined as the product ofn-1 elem entary reflectors of order \(N\), as retumed by SSY TRD

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SORGTR (UPLO,N,A,LDA,TAU,W ORK,LW ORK, INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER N,LDA,LW ORK,INFO}
REALA (LDA,*),TAU (*),W ORK (*)
SUBROUT\mathbb{NE SORGTR_64(UPLO,N,A,LDA,TAU,W ORK,LW ORK, INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N,LDA,LW ORK,\mathbb{NFO}}\mathbf{N}\mathrm{ , L}
REALA (LDA,*),TAU (*),W ORK (*)

```
F95 INTERFACE
    SU BROUTINE ORG TR (UPLO, \(\mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L W O R K],[\mathbb{N F O}])\)
    CHARACTER (LEN=1)::UPLO
    \(\mathbb{N}\) TEGER ::N,LDA,LW ORK, \(\mathbb{N}\) FO
    REAL,D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
    REAL,D \(\mathbb{M}\) ENSION (:,:)::A
    SU BROUTINE ORGTR_64 (UPLO, \(\mathbb{N}\) ],A, [LDA ],TAU, [W ORK ], [LW ORK ], [ \(\mathbb{N} F O\) ])
    CHARACTER (LEN=1) ::UPLO
    \(\mathbb{N}\) TEGER (8) :: N, LDA,LW ORK, \(\mathbb{N}\) FO
    REAL,D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
    REAL,D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void sorgtr (charuplo, intn, float *a, int lda, float *tau, int*info);
void sorgtr_64 (char uplo, long \(n\), float*a, long lda, float *tau, long *info);

\section*{PURPOSE}
sorgtrgenerates a realorthogonalm atrix Q which is defined as the productofn-1 elem entary reflectors of order \(N\), as retumed by SSY TRD :
if \(\mathrm{P} P \mathrm{LO}=\mathrm{U}, \mathrm{Q}=\mathrm{H}(\mathrm{n}-1) \ldots \mathrm{H}(2) \mathrm{H}(1)\),
if \(U P L O=L^{\prime}, Q=H(1) H(2) \ldots H(n-1)\).

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ::U ppertriangle of A contains elem entary
reflectors from SSY TRD ; = L ': Low ertriangle of A
contains elem entary reflectors from SSY TRD.
N (input) The order of the m atrix \(\mathrm{Q} . \mathrm{N}>=0\).
A (input/output)
O \(n\) entry, the vectors which define the elem entary reflectors, as retumed by SSY TRD. On exit, the N boy-N orthogonalm atrix Q .

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

TAU (input)
TAU (i) \(m\) ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by SSY TRD.

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, \mathrm{~W}\) ORK (1) retums the optim al
LW ORK.
LW ORK (input)
The dimension of the array W ORK. LW ORK >=
\(\max (1, N-1)\). Foroptim um perform ance LW ORK \(>=\mathbb{N}-\) 1) \({ }^{N} \mathrm{~N}\), where \(N B\) is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N F O}=-\mathrm{i}\), the i -th argum ent had an illegal value

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorm br-VECT = Q',SORM BR overw rites the general real M by \(-\mathrm{N} m\) atrix C w th \(S \mathbb{D} E=\mathrm{L} \mathrm{S}^{\prime} \mathrm{D} E=\mathrm{R}^{\prime} \mathrm{TRANS}=\mathrm{N}^{\prime}\)

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SORM BR NECT,SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,}
W ORK,LW ORK,INFO)
CHARACTER * 1VECT,SIDE,TRANS
\mathbb{NTEGERM,N,K,LDA,LDC,LW ORK,NNFO}
REAL A (LDA,*),TAU (*),C (LD C ,*),W ORK (*)
SUBROUT\mathbb{NE SORM BR_64 NECT,SDDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,}
W ORK,LW ORK,INFO)

```
CHARACTER * 1 VECT, SIDE,TRANS
\(\mathbb{N}\) TEGER*8M,N,K,LDA,LDC,LW ORK, \(\mathbb{N} F O\)


\section*{F95 INTERFACE}

SU BROUTINE ORM BR \(N E C T, S \mathbb{D} E,[T R A N S], \mathbb{M}], \mathbb{N}], K, A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::VECT,SDE,TRANS
\(\mathbb{N} T E G E R:: M, N, K, L D A, L D C, L W O R K, \mathbb{N} F O\)
REAL,D IM ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A, C
SU BROUTINE ORM BR_64 NECT,SDE, [TRANS], \(\mathbb{M}], \mathbb{N}], K, A,[L D A], T A U\), C, [LDC], [W ORK ], [LW ORK], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::VECT,SDE,TRANS
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathrm{LD} \mathrm{C}, \mathrm{LW} \mathrm{ORK}, \mathbb{N}\) FO REAL,D \(\mathbb{I M} E N S I O N\) (:) ::TAU,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A , C

\section*{C INTERFACE}
\#include <sunperfh>
void som br (charvect, char side, chartrans, intm, int n, int \(k\), float *a, int lda, float *tau, float * C , int ldc, int *info);
void sorm br_64 (charvect, charside, char trans, long m, long \(n\), long k, float*a, long lda, float *tau, float * C , long ldc, long *info);

\section*{PURPOSE}
sorm brVECT = Q ', SORM BR overw rites the general realM -by-N m atrix C w ith
```

            \(S \mathbb{D E}=\mathbb{L}^{\prime} \quad S \mathbb{D E}=\mathrm{R}^{\prime} \operatorname{TRANS}=\mathrm{N}^{\prime}:\)
    Q * C C * Q TRANS = T': Q**T *C C *
Q**T

```

IfVECT = P', SO RM BR overw rites the general real M -by-N \(m\) atrix C w th
\[
S \mathbb{D} E=\mathbb{L}^{\prime} \quad S \mathbb{D} E=R^{\prime}
\]

TRANS \(=N^{\prime}: \quad P * C \quad C * P\)
TRANS = T': P**T*C C * P **T
\(H\) ere \(Q\) and \(P * * T\) are the orthogonal \(m\) atrices determ ined by SG EBRD when reducing a realm atrix A to bidiagonal form : A = \(Q * B * P * * T . Q\) and \(P * * T\) are defined asproducts of elem entary reflectors H (i) and G (i) respectively.

Letnq \(=m\) if \(S \mathbb{D} E=L\) 'and \(n q=n\) if \(S \mathbb{D} E=R\) '. Thus nq is the order of the orthogonalm atrix Q orP**T that is applied.

IfV ECT = Q', A is assum ed to have been an \(N Q\) boy \(\mathrm{K} m\) atrix:
ifnq \(>=k, Q=H(1) H(2) \ldots H(k)\);
ifng \(<k, Q=H(1) H(2) \ldots H(n q-1)\).

IfVECT \(=P^{\prime}, \mathrm{A}\) is assum ed to have been a K boy -NQ m atrix:
if \(k<n q, P=G(1) G(2) . . . G(k)\);
ifk \(>=n q, P=G(1) G(2) \ldots G(n q-1)\).

\section*{ARGUMENTS}

VECT (input)
\(=\mathrm{Q}\) ': apply Q orQ \(\mathrm{A}^{*} \mathrm{~T}\);
= P ': apply P orP**T.

SID E (input)
\(=\mathbb{L}\) ': apply \(\mathrm{Q}, \mathrm{Q}\) **T, P orP**T from the Left;
\(=R\) ': apply \(Q, Q * * T, P\) or \(P * * T\) from the \(R\) ight.

TRANS (input)
\(=\mathrm{N}: \mathrm{N}\) o transpose, apply Q orP;
\(=T\) ': Transpose, apply \(\mathrm{Q}^{* *} \mathrm{~T}\) or \(\mathrm{P}^{* *} \mathrm{~T}\).

TRANS is defaulted to \(N\) 'forF \(95 \mathbb{N}\) TERFACE.

M (input) The num ber of row s of the m atrix \(\mathrm{C} . \mathrm{M}>=0\).
N (input) The num ber of colum ns of the \(m\) atrix \(\mathrm{C} . \mathrm{N}>=0\).

K (input) IfVECT = Q', the num ber of colum ns in the original \(m\) atrix reduced by \(S G E B R D\). IfVECT \(=P\) ', the num ber of row \(S\) in the originalm atrix reduced by SGEBRD.K \(>=0\).

A (input) (LDA,min (nq,K)) ifVECT = Q' (LDA,nq) if \(\mathrm{VECT}=\mathrm{P}\) 'The vectors w hich define the elem entary reflectors H (i) and G (i) , w hose products determ ine the \(m\) atrioes \(Q\) and \(P\), as retumed by SG EBRD.

LD A (input)
The leading dim ension of the array A. If VECT = Q', LDA \(>=\max (1, n q)\); if \(V E C T=P^{\prime}, L D A>=\) \(m\) ax \((1, m\) in \((n q, K))\).

TAU (input)
TAU (i) m ustcontain the scalar factor of the ele\(m\) entary reflectorH (i) orG (i) which determ ines Q orP, as retumed by SGEBRD in the array argum ent TAUQ orTAUP.

C (input/output)
On entry, the \(M\) boy -N m atrix C . On exit, C is overw ritten by \(\mathrm{Q} * \mathrm{C}\) or \(\mathrm{Q} * \mathrm{~T} * \mathrm{C}\) or \(\mathrm{C} \mathrm{Q}^{* *} \mathrm{~T}\) or C Q or \(\mathrm{P} * \mathrm{C}\) or \(\mathrm{P} * * \mathrm{~T} * \mathrm{C}\) or C * or \(\mathrm{C} * \mathrm{P} * * \mathrm{~T}\).

LD C (input)
The leading dim ension of the aray C. LD C >= \(\max (1, M)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al

LW ORK .

LW ORK (input)
The dim ension of the array \(W\) ORK. IfSDE \(E L\) ',
LW ORK >= max (1,N); if \(S \mathbb{D} E=R '\) LW ORK >=
max ( \(1, \mathrm{M}\) ). Foroptim um perform ance LW ORK >= N *NB
if \(S \mathbb{D} E=L^{\prime}\), and LW \(O R K>=M * N B\) if \(S \mathbb{D} E=R\) ',
where NB is the optim alblocksize.
IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LW ORK is issued by X ERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorm hr-overw rite the general real M -by N m atrix C w ith \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime} T R A N S=N^{\prime}\)

\section*{SYNOPSIS}

```

    W ORK,LW ORK,INFO)
    CHARACTER * 1SIDE,TRANS

```

```

REAL A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
SUBROUT\mathbb{NE SORM HR_64 (S\mathbb{DE,TRANS,M,N,\mathbb{LO},\mathbb{H}I,A,LDA,TAU,C,}}\mathbf{T},\textrm{T},\textrm{T}
LDC,W ORK,LW ORK,\mathbb{NFO)}
CHARACTER * 1SDDE,TRANS

```

```

REALA (LDA,*),TAU (*),C (LDC,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE ORM HR (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}], \mathbb{H}, \mathbb{H} I, A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [NFO])

CHARACTER (LEN=1) ::SDE,TRANS
\(\mathbb{N} T E G E R:: M, N, \mathbb{L O}, \mathbb{H} I, L D A, L D C, L W O R K, \mathbb{N} F O\)
REAL,D IM ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A, C
SUBROUTINE ORM HR_64 (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}], \mathbb{H}, \mathbb{H} I, A,[L D A], T A U\), C, [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SIDE,TRANS
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathbb{L} \mathrm{O}, \mathbb{H} \mathrm{I}, \mathrm{LD} A, L D C, L W O R K, \mathbb{N} F O\) REAL,D \(\mathbb{I M}\) ENSION (:) ::TAU ,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A , C

\section*{C INTERFACE}
\#include <sunperfh>
void sorm hr(char side, chartrans, intm, int \(n\), int ilo, int ini, float *a, int lda, float *tau, float * C , int ldc, int *info);
void sorm hr_64 (charside, chartrans, long m, long n, long ilo, long ihi, float *a, long lda, float*tau, float * C, long ldc, long *info);

\section*{PURPOSE}
som hroverw rites the general real M toy N matrix C w ith

\(w\) here \(Q\) is a realorthogonalm atrix of ordernq, with nq \(=m\) if \(S \mathbb{D} E=L\) 'and \(n q=n\) if \(S \mathbb{D} E=R\) '. Q is defined as the productof \(\mathbb{H}\) I-ILO elem entary reflectors, as retumed by SG EHRD :
\(Q=H\) (مlo) H (

\section*{ARGUMENTS}

SID E (input)
\(=\mathrm{L}\) ': apply Q or \(\mathrm{Q}^{* *} \mathrm{~T}\) from the Left;
\(=R\) ': apply \(Q\) or \(\mathrm{Q}^{* * T}\) from the R ight.

TRANS (input)
\(=\mathrm{N}\) : N o transpose, apply Q ;
\(=T\) ': T ranspose, apply \(Q^{* *} T\).

TRANS is defaulted to \(N\) 'forF \(95 \mathbb{N}\) TERFACE.

M (input) The num ber of row s of the \(m\) atrix \(C . M>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\).

IIO (input)
IIO and IH Im usthave the sam evalues as in the
previous call of SGEHRD.Q is equal to the unit
\(m\) atrix except in the subm atrix

Q (ilo+1:ihi,ilo+1: : ihi). IfS \(\mathbb{D} E=4\) ', then \(1<=\)
\(\mathbb{H O}<=\mathbb{H} I<=M\), if \(M>0\), and \(\mathbb{H O}=1\) and \(\mathbb{H} I=\)
0 , if \(M=0\); if \(S \mathbb{D} E=R\) ', then \(1<=\mathbb{H O}<=\mathbb{H} I\)
\(<=\mathrm{N}\), if \(\mathrm{N}>0\), and \(\mathbb{H} \mathrm{O}=1\) and \(\mathbb{H} \mathrm{I}=0\), if \(\mathrm{N}=0\).

IH I (input)
See the description of IIO .

A (input) (LDA,M) if \(S \mathbb{D} E=L^{\prime}(\mathbb{L D A} N)\) if \(S \mathbb{D} E=R^{\prime}\) The vectors which define the elem entary reflectors, as retumed by SGEHRD.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, M)\) ifSDE \(=L ; \operatorname{LDA}>=\max (1, N)\) if \(S \mathbb{D} E=\) R'.

TAU (input)
\((M-1)\) if \(S \mathbb{D} E=L^{\prime}(\mathbb{N}-1)\) if \(S \mathbb{D E}=R^{\prime} T A U(i)\) \(m\) ust contain the scalar factor of the elem entary reflectorH (i), as retumed by SG EH RD .

C (input/output)
On entry, the \(M-b y-N\) matrix C. On exit, \(C\) is overw ritten by Q * C or \(\mathrm{Q} * * \mathrm{~T} * \mathrm{C}\) or C * \(\mathrm{Q} * \mathrm{~T}\) or C * Q .

LD C (input)
The leading dim ension of the aray C. LD C >= max (1, M).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al
LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. IfSDE \(=\mathrm{L}\) ', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LW ORK \(>=\) \(\max (1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=L\) ', and LW ORK \(>=M * N B\) if \(S \mathbb{D} E=R\) ', w here NB is the optim alblocksize.

If LW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an ille-
galvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorm lq -overw rite the general real M -by-N m atrix C w ith \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R{ }^{\prime} T R A N S=N^{\prime}\)

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SORM LQ (S\mathbb{DE,TRANS,M ,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{T},\textrm{L},\textrm{L}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
\mathbb{NTEGERM,N,K,LDA,LDC,LW ORK,NNFO}
REAL A (LDA,*),TAU (*),C (LD C ,*),W ORK (*)
SU BROUT\mathbb{NE SORM LQ_64 (SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}
LW ORK,\mathbb{NFO)}

```
CHARACTER * 1 SIDE,TRANS
\(\mathbb{N}\) TEGER*8M,N,K,LDA,LDC,LWORK, \(\mathbb{N} F O\)
REALA (LDA, \()^{*}\) ), TAU (*), C (LDC, \(\left.{ }^{\star}\right), \mathrm{W}\) ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE ORM LQ (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C,[L D C]\), [W ORK], [LW ORK ], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::SDE,TRANS
\(\mathbb{N}\) TEGER :: M , N, K, LDA, LD C, LW ORK, \(\mathbb{N} F O\)
REAL,D IM ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A, C
SU BROUTINE ORM LQ_64 (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C\), [LDC ], [W ORK ], [LW ORK ], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SDE,TRANS
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathrm{LD} \mathrm{C}, \mathrm{LW}\) ORK, \(\mathbb{N}\) FO REAL,D \(\mathbb{M}\) ENSION (:) ::TAU ,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A , C

\section*{C INTERFACE}
\#include <sunperfh>
void sorm lq (char side, chartrans, int m, int \(n\), int \(k\), float *a, int lda, float *tau, float * C , int ldc, int*info);
void sorm lq_64 (charside, chartrans, long m, long n, long k, float *a, long lda, float *tau, float * C , long ldc, long *info);

\section*{PURPOSE}
sorm lq overw rites the general real M boy -N m atrix \(\mathrm{C} w\) ith

where \(Q\) is a realorthogonalm atrix defined as the product ofk elem entary reflectors
\[
Q=H(k) \ldots H(2) H(1)
\]
as retumed by SGELQF.Q is oforderM if \(S \mathbb{D} E=\mathbb{L}\) 'and of orderN ifSDE = R'.

\section*{ARGUMENTS}

SID E (input)
\(=\mathrm{L}\) ': apply Q or \(\mathrm{Q}{ }^{* *} \mathrm{~T}\) from the Left;
\(=R\) ': apply Q or \(\mathrm{Q} * * \mathrm{~T}\) from the R ight.

TRANS (input)
\(=\mathrm{N}\) ': N o transpose, apply Q ;
= T': T ranspose, apply Q ** T .

TRAN \(S\) is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

M (input) The num ber of row s of the \(m\) atrix \(C . M>=0\).

N (input) The num ber of colum ns of the matrix \(\mathrm{C} . \mathrm{N}>=0\).

K (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q\). IfS \(\mathbb{D} E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0\); if \(S \mathbb{D} E=R \prime, N>=K>=0\) 。
 \(i\)-th row m ustcontain the vectorw hich defines the elem entary reflector \(H\) ( \(i\) ), for \(i=1,2, \ldots, k\), as retumed by \(S G E L Q F\) in the firstk row sof its array argum entA. A ism odified by the routine butrestored on exit.

LDA (input)
The leading dim ension of the array A. LDA >= \(\max (1, K)\).

TAU (input)
TAU (i) m ustcontain the scalar factor of the ele\(m\) entary reflectorH (i), as retumed by SG ELQ F.

C (input/output)
On entry, the \(M-b y-N m\) atrix \(C\). On exit, \(C\) is overw rilten by Q * C or \(\mathrm{Q} * * \mathrm{~T} * \mathrm{C}\) or \(\mathrm{C} * \mathrm{Q} * \mathrm{~T}\) or C Q .

LD C (input)
The leading dim ension of the array \(C . \operatorname{LDC}>=\) \(\max (1, M)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the aray W ORK. IfSIDE = L', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LWORK \(>=\) \(m\) ax \((1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=L\) ', and LW ORK \(>=M * N B\) ifS \(\mathbb{D} E=R\) ', w here N B is the optim alblocksize.

If LW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorm ql-overw rite the general real \(M\)-by N matrix C with \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime}\) TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SORM QL (S\mathbb{DE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{M},\textrm{L},\textrm{T}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
\mathbb{NTEGERM,N,K,LDA,LDC,LW ORK,NNFO}
REAL A (LDA,*),TAU (*),C (LD C ,*),W ORK (*)
SUBROUT\mathbb{NE SORM QL_64(SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}
LW ORK,\mathbb{NFO)}

```
CHARACTER * 1 SIDE,TRANS
\(\mathbb{N}\) TEGER*8M,N,K,LDA,LDC,LWORK, \(\mathbb{N} F O\)
REALA (LDA, \()^{*}\) ), TAU (*), C (LDC, \(\left.{ }^{\star}\right), \mathrm{W}\) ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE ORM QL (SDE, [TRANS], M ], \(\mathbb{N}],[K], A,[L D A], T A U, C,[L D C]\), [W ORK], [LW ORK], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::SDE,TRANS
\(\mathbb{N}\) TEGER :: M , N, K, LDA, LD C, LW ORK, \(\mathbb{N} F O\)
REAL,D IM ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A, C
SU BROUTINE ORM QL_64 (SDE, [TRANS], M ], \(\mathbb{N}],[K], A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SDE,TRANS
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathrm{LD} \mathrm{C}, \mathrm{LW}\) ORK, \(\mathbb{N}\) FO REAL,D \(\mathbb{M}\) ENSION (:) ::TAU ,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A , C

\section*{C INTERFACE}
\#include <sunperfh>
void sorm ql(char side, chartrans, int \(m\), int \(n\), int \(k\), float *a, int lda, float *tau, float * c , int ldc, int*info);
void sorm ql_64 (charside, chartrans, long m, long n, long k, float *a, long lda, float *tau, float * c, long ldc, long *info);

\section*{PURPOSE}
sorm qloverw rites the general real M boy -N m atrix \(\mathrm{C} w\) ith

where \(Q\) is a realorthogonalm atrix defined as the product ofk elem entary reflectors
\[
\mathrm{Q}=\mathrm{H}(\mathrm{k}) \ldots \mathrm{H}(2) \mathrm{H}(1)
\]
as retumed by SGEQLF. Q is oforderM if \(S \mathbb{D} E=\mathbb{L}\) 'and of orderN ifSDE = R '.

\section*{ARGUMENTS}

SID E (input)
\(=\mathrm{L}\) ': apply Q or \(\mathrm{Q}{ }^{* *} \mathrm{~T}\) from the Left;
\(=R\) ': apply Q or \(\mathrm{Q} * * \mathrm{~T}\) from the R ight.

TRANS (input)
\(=\mathrm{N}\) ': N o transpose, apply Q ;
= T': T ranspose, apply Q ** T .

TRAN \(S\) is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

M (input) The num ber of row s of the \(m\) atrix \(C . M>=0\).

N (input) The num ber of colum ns of the matrix \(\mathrm{C} . \mathrm{N}>=0\).

K (input) The num ber of elem entary reflectors w hose product defines them atrix Q . IfSID \(\mathrm{E}=\mathrm{L} \mathrm{I}^{\prime}, \mathrm{M}>=\mathrm{K}>=0\); if \(S \mathbb{D} E=R \prime, N>=K>=0\) 。

A (input) The i-th colum \(n\) must contain the vector which defines the elem entary reflector H (i), for \(i=\) \(1,2, \ldots, k\), as retumed by SGEQ LF in the last \(k\) colum ns of its aray argum entA. A ism odified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A . If \(\mathrm{SDE}=\) L', LDA \(>=m a x(1, M)\); if \(S \mathbb{D E}=R \prime\),LDA \(>=\) \(\max (1, N)\).

TAU (input)
TAU (i) \(m\) ust contain the scalar factor of the ele\(m\) entary reflector H (i), as retumed by SGEQ LF .

C (input/output)
On entry, the \(M\) by \(-N\) matrix C. On exit, \(C\) is overw ritten by Q * C or \(\mathrm{Q} * * \mathrm{~T}\) * C or C Q Q * T or C * Q .

LD C (input)
The leading dim ension of the array C.LDC >= \(\mathrm{max}(1, \mathrm{M})\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. IfSDE \(E L\) ', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LW ORK >= \(\max (1, M)\). Foroptim um perform ance LW ORK >= \(N\) *NB if \(S \mathbb{D} E=L '\), and \(L W O R K>=M * N B\) if \(S \mathbb{D} E=R\) ', where NB is the optim alblocksize.

IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK amray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorm qr-overw rite the general real \(M\) by \(-N m\) atrix \(C\) with \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime} T R A N S=N^{\prime}\)

\section*{SYNOPSIS}

```

    LW ORK,\mathbb{NFO)}
    CHARACTER * 1SIDE,TRANS
\mathbb{NTEGERM,N,K,LDA,LDC,LW ORK,NNFO}
REAL A (LDA,*),TAU (*),C (LD C ,*),W ORK (*)
SUBROUT\mathbb{NE SORM QR_64(S\mathbb{DE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{T},\textrm{T},\textrm{T}
LW ORK,\mathbb{NFO)}

```
CHARACTER * 1 SIDE,TRANS
\(\mathbb{N}\) TEGER*8M,N,K,LDA,LDC,LW ORK, \(\mathbb{N} F O\)
REALA (LDA, \()^{*}\) ), TAU (*), C (LDC, \(\left.{ }^{\star}\right), \mathrm{W}\) ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE ORM QR (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C,[L D C]\), \(\mathbb{W}\) ORK ], [LW ORK ], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::SDE,TRANS
\(\mathbb{N}\) TEGER :: M , N, K, LDA, LD C, LW ORK, \(\mathbb{N} F O\)
REAL,D IM ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:)::A,C
SUBROUTINE ORM QR_64 (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C\), [LDC ], [W ORK ], [LW ORK ], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::SDE,TRANS
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathrm{LD} \mathrm{C}, \mathrm{LW}\) ORK, \(\mathbb{N}\) FO REAL,D \(\mathbb{M}\) ENSION (:) ::TAU ,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A , C

\section*{C INTERFACE}
\#include <sunperfh>
void sorm qr(char side, chartrans, int \(m\), int \(n\), int \(k\), float *a, int lda, float *tau, float * c , int ldc, int*info);
void sorm qr_64 (charside, chartrans, long m, long n, long k, float *a, long lda, float *tau, float * C , long ldc, long *info);

\section*{PURPOSE}
sorm qroverw rites the general real M toy -N m atrix C w ith TRANS = \(\mathrm{T}: \quad \mathrm{Q}^{* *} \mathrm{~T} * \mathrm{C} \quad \mathrm{C} * \mathrm{Q}^{\star *} \mathrm{~T}\)
where \(Q\) is a realorthogonalm atrix defined as the product ofk elem entary reflectors
\[
Q=H(1) H(2) \ldots H(k)
\]
as retumed by SGEQRF.Q is oforderM if \(S \mathbb{D} E=\mathbb{L}\) 'and of orderN ifSDE = R'.

\section*{ARGUMENTS}

SID E (input)
\(=\mathbb{L}\) ': apply Q orQ \({ }^{* *} \mathrm{~T}\) from the Left;
\(=R\) ': apply Q or \(\mathrm{Q} * * \mathrm{~T}\) from the R ight.

TRANS (input)
\(=N\) ': N o transpose, apply Q ;
= T': T ranspose, apply Q ** T .

TRAN \(S\) is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{C} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the matrix \(\mathrm{C} . \mathrm{N}>=0\).

K (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q\). IfS \(\mathbb{D} E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0\); if \(S \mathbb{D} E=R \prime, N>=K>=0\) 。

A (input) The i-th colum \(n\) must contain the vector which defines the elem entary reflector \(H\) (i), for \(i=\) \(1,2, \ldots, k\), as retumed by SGEQ RF in the first \(k\) colum ns of its anay argum entA. A ism odified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A . If \(\mathrm{SDE} \mathrm{E}=\) L', LDA >= max (1,M); if \(S \mathbb{D E}=R\) ',LDA \(>=\) \(\max (1, N)\).

TAU (input)
TAU (i) \(m\) ust contain the scalar factor of the ele\(m\) entary reflector \(H\) (i), as retumed by SG EQ RF .

C (input/output)
On entry, the \(M\) by \(-N\) matrix C. On exit, \(C\) is overw ritten by Q * C or \(\mathrm{Q} * \mathrm{~T}\) * C or C * \(\mathrm{Q} * \mathrm{~T}\) or C * Q .

LD C (input)
The leading dim ension of the array C.LDC >= max (1, M).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK ( 1 ) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. IfSDE \(=\mathrm{L}\) ', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LW ORK >= \(\max (1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=L\) ', and LW ORK \(>=M * N B\) if \(S \mathbb{D} E=R\) ', where NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK ancay, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorm rq-overw rite the general real \(M\)-by N matrix C with \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime} T R A N S=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SORM RQ (S\mathbb{DE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{M,}\mathbf{N},\textrm{M}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
\mathbb{NTEGERM,N,K,LDA,LDC,LW ORK,NNFO}
REAL A (LDA,*),TAU (*),C (LD C ,*),W ORK (*)
SUBROUT\mathbb{NE SORMRQ_64(SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}
LW ORK,\mathbb{NFO)}

```
CHARACTER * 1 SIDE,TRANS
\(\mathbb{N}\) TEGER*8M,N,K,LDA,LDC,LWORK, \(\mathbb{N} F O\)
REALA (LDA, \()^{*}\) ), TAU (*), C (LDC, \(\left.{ }^{\star}\right), \mathrm{W}\) ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE ORMRQ (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C,[L D C]\), \(\mathbb{W}\) ORK ], [LW ORK ], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::SDE,TRANS
\(\mathbb{N}\) TEGER :: M , N, K, LDA, LD C, LW ORK, \(\mathbb{N} F O\)
REAL,D IM ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:)::A,C
SU BROUTINE ORMRQ_64 (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C\), [LD C ], [W ORK ], [LW ORK ], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::SDE,TRANS
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathrm{LD} \mathrm{C}, \mathrm{LW}\) ORK, \(\mathbb{N}\) FO REAL,D \(\mathbb{M}\) ENSION (:) ::TAU ,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A , C

\section*{C INTERFACE}
\#include <sunperfh>
void sorm rq (char side, chartrans, int \(m\), int \(n\), int \(k\), float *a, int lda, float *tau, float * c , int ldc, int*info);
void sorm rq_64 (charside, chartrans, long m, long n, long k, float *a, long lda, float *tau, float * c, long ldc, long *info);

\section*{PURPOSE}
sorm rq overw rites the general real \(M\) toy \(-N\) m atrix \(C\) w ith TRANS = T': \(\mathrm{Q}^{* *} \mathrm{~T} * \mathrm{C} \quad \mathrm{C} * \mathrm{Q}^{\star *} \mathrm{~T}\)
where \(Q\) is a realorthogonalm atrix defined as the product ofk elem entary reflectors
\[
Q=H(1) H(2) \ldots H(k)
\]
as retumed by SGERQF.Q is oforderM if \(S \mathbb{D} E=\mathbb{L}\) 'and of orderN ifSDE = R'.

\section*{ARGUMENTS}

SID E (input)
\(=\mathbb{L}\) ': apply Q orQ \({ }^{* *} \mathrm{~T}\) from the Left;
\(=R\) ': apply Q or \(\mathrm{Q} * * \mathrm{~T}\) from the R ight.

TRANS (input)
\(=N\) ': N o transpose, apply Q ;
= T': T ranspose, apply Q ** T .

TRAN \(S\) is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{C} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the matrix \(\mathrm{C} . \mathrm{N}>=0\).

K (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q\). IfS \(\mathbb{D} E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0\); if \(S \mathbb{D} E=R \prime, N>=K>=0\) 。
 \(i\)-th row m ustcontain the vectorw hich defines the elem entary reflector \(H\) ( \(i\) ), for \(i=1,2, \ldots, k\), as retumed by \(S G E R Q F\) in the lastk row \(s\) of its anay argum entA. A is modified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, K)\).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectorH (i), as retumed by SG ERQ F.

C (input/output)
On entry, the \(M\) boy- \(\mathrm{N} m\) atrix C . On exit, C is overw rilten by \(\mathrm{Q} * \mathrm{C}\) or \(\mathrm{Q} * * \mathrm{~T} * \mathrm{C}\) or C * \(\mathrm{Q} * \mathrm{~T}\) or C Q .

LD C (input)
The leading dim ension of the array \(C . \operatorname{LDC}>=\) \(\max (1, M)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the amay W ORK. IfSIDE = L', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LWORK >= \(m\) ax \((1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D E}=L^{\prime}\) ', and LW ORK \(>=M * N B\) ifSDE \(=R\) ', w here N B is the optim alblocksize.

If LW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sorm rz -overw rite the general real M -by N matrix C with \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime} T R A N S=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SORMRZ (S\mathbb{DE,TRANS,M,N,K,L,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{M},\textrm{L},\textrm{L}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SDDE,TRANS
\mathbb{NTEGERM,N,K,L,LDA,LDC,LW ORK, INFO}
REAL A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
SUBROUT\mathbb{NE SORMRZ_64(SIDE,TRANS,M,N,K,L,A,LDA,TAU,C,LDC,}
W ORK,LW ORK,\mathbb{NFO)}

```
CHARACTER * 1 SIDE,TRANS
\(\mathbb{N} T E G E R * 8 M, N, K, L, L D A, L D C, L W O R K, \mathbb{N} F O\)
REALA (LDA, \(\left.{ }^{\star}\right)\),TAU (*), C (LDC \(\left.{ }^{\star}\right), \mathrm{W} O R K\left({ }^{*}\right)\)

\section*{F95 INTERFACE}

SU BROUTINE ORMRZ (SDE,TRANS, \(\mathbb{M}], \mathbb{N}], K, L, A,[L D A], T A U, C,[L D C]\), [W ORK ], [LW ORK ], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::SIDE,TRANS
\(\mathbb{N}\) TEGER ::M,N,K,L,LDA,LDC,LW ORK, \(\mathbb{N}\) FO
REAL,D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A, C
SUBROUTINE ORMRZ_64 (SDE,TRANS, \(\mathbb{M}], \mathbb{N}], K, L, A,[L D A], T A U, C\),
[LDC], [W ORK ], [LW ORK], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1) ::SIDE,TRANS
\(\mathbb{N}\) TEGER (8) :: M , N , K,L,LDA,LDC,LW ORK, \(\mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL,D IM ENSION (:,:) ::A, C

\section*{C INTERFACE}
\#include < sunperfh>
void sorm rz (char side, char trans, intm, intn, intk, int l, float *a, intlda, float*tau, float *c, int ldc, int*info);
void sorm rz_64 (charside, chartrans, long m, long n, long
k, long l, float *a, long lda, float *tau, float
*c, long ldc, long *info);

\section*{PURPOSE}
sorm rz overw rites the general real M -by-N m atrix C with TRANS \(=T: \quad Q * * T C \quad C * Q * T\)
where \(Q\) is a realorthogonalm atrix defined as the product ofk elem entary reflectors
\[
Q=H(1) H(2) \ldots H(k)
\]
as retumed by STZRZF.Q is oforderM if \(S \mathbb{D} E=L\) 'and of orderN if \(S \mathbb{D} E=R\).

\section*{ARGUMENTS}

STDE (input)
\(=\mathrm{L}\) ': apply Q orQ \({ }^{* *}\) T from the Left;
= R ': apply Q orQ **T from the R ight.
TRANS (input)
= N ': N o transpose, apply Q ;
= T': T ranspose, apply Q **T .

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{C} . \mathrm{M}>=0\).
N (input) The num ber of collum ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\).
\(K\) (input) The num ber of elem entary reflectors whose product defines the \(m\) atrix \(Q\). If \(S \mathbb{D} E=L^{\prime}, \mathrm{M}>=K>=0\);
if \(S \mathbb{D} E=R \prime, N>=K>=0\).

L (input) The num ber of colum ns of the \(m\) atrix A containing
the \(m\) eaningfulpart of the \(H\) ouseholder reflectors.
If \(S \mathbb{D} E=L \prime, M>=L>=0\), if \(S \mathbb{D} E=R \prime N>=L\)
\(>=0\) 。

A (input) (LDA, M) if \(S \mathbb{D} E=L \prime\) ( \(L D A, N\) ) if \(S \mathbb{D} E=R^{\prime}\) The \(i\)-th row \(m\) ustcontain the vectorw hich defines the elem entary reflector \(H\) ( \(i\) ), for \(i=1,2, \ldots, k\), as retumed by STZRZF in the lastk row s of its array argum entA. A is modified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array \(\mathrm{A} . \mathrm{LDA}>=\) \(\max (1, K)\).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflector H (i), as retumed by STZRZF.

C (input/output)
On entry, the \(M\) boy -N m atrix C . On exit, C is overw ritten by Q * C or \(\mathrm{Q} * * \mathrm{H} * \mathrm{C}\) or \(\mathrm{C} \mathrm{Q}^{*}{ }^{*} \mathrm{H}\) or C * Q .

LD C (input)
The leading dim ension of the aray C.LDC >= \(\max (1, M)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. If \(S \mathbb{D} E=\mathbb{L}\) ', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R \prime\) LW ORK >= \(m\) ax \((1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=L '\), and LW ORK \(>=M * N B\) if \(S D E=R \prime\), w here N B is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

INFO (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

B ased on contributions by
A. Petitet, C om puter Science D ept., U niv . of Tenn., K noxville, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sorm tr-overw rite the general real \(M\)-by -N matrix \(\mathrm{C} w\) th \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R{ }^{\prime}\) TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUTINE SORM TR (SDE E,UPLO,TRANS,M ,N,A,LDA,TAU,C,LDC,W ORK,
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,UPLO,TRANS
INTEGERM,N,LDA,LDC,LW ORK, INFO
REAL A (LDA,*),TAU (*),C (LD C ,*),W ORK (*)
SU BROUT\mathbb{NE SORM TR_64(SDDE,UPLO,TRANS,M ,N,A,LDA,TAU,C,LDC,}
W ORK,LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,UPLO,TRANS
INTEGER*8M,N,LDA,LDC,LW ORK,\mathbb{N FO}
REALA (LDA,*),TAU (*),C (LDC,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE ORM TR (SDE, UPLO, [TRANS], \(\mathbb{M}], \mathbb{N}], A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SDE,UPLO,TRANS
\(\mathbb{N} T E G E R:: M, N, L D A, L D C, L W O R K, \mathbb{N F O}\)
REAL,D IM ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A, C
SU BROUTINE ORM TR_64 (SDE,UPLO, [TRANS], \(\mathbb{M}], \mathbb{N}], A,[L D A], T A U, C\), [LDC ], [W ORK ], [LW ORK ], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::SDE,UPLO,TRANS
\(\mathbb{N}\) TEGER (8) ::M,N,LDA,LDC,LW ORK, \(\mathbb{N} F O\)
REAL,D \(\mathbb{I}\) ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A, C

\section*{C INTERFACE}
\#include < sunperfh>
void sorm tréchar side, char uple, char trans, intm, int n, float *a, int lda, float *tau, float * \(c\), int ldc, int*info);
void som tr_64 (charside, charuplo, char trans, long m, long n , float *a, long lda, float *tau, float *c, long ldc, long *info);

\section*{PURPOSE}
sorm troverw rites the general real M boy -N matrix C w th TRANS = T': Q**T * C C *Q**T
where \(Q\) is a realorthogonalm atrix of ordernq, \(w\) th \(n q=m\) if \(S \mathbb{D} E=L\) 'and \(n q=n\) if \(S \mathbb{D} E=R '^{\prime} . Q\) is defined as the product of nq-1 elem entary reflectors, as retumed by SSY TRD :
if \(U P L O=U ', Q=H(n q-1) \ldots\) (2) \(\mathrm{H}(1)\);
if \(\mathrm{UPLO}=\mathrm{L}, \mathrm{Q}=\mathrm{H}(1) \mathrm{H}(2) \ldots \mathrm{H}(\mathrm{nq}-1)\).

\section*{ARGUMENTS}

SIDE (input)
\(=\mathrm{L}\) ': apply Q orQ \({ }^{* *}\) T from the Left;
\(=R\) ': apply Q orQ \({ }^{* *} \mathrm{~T}\) from the R ight.
UPLO (input)
= U :: U ppertriangle of A contains elem entary
reflectors from SSY TRD ; = L ': Low ertriangle of A
contains elem entary reflectors from SSY TRD.
TRANS (input)
\(=\mathrm{N}\) ': N o transpose, apply Q ;
\(=T\) ': Transpose, apply \(Q * * T\).
TRANS is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

M (input) The num ber of row s of the \(m\) atrix \(C . M>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\).
A (input) ( \(L D A, M\) ) if \(S \mathbb{D} E=L^{\prime}(L D A, N)\) if \(S \mathbb{D} E=R^{\prime}\) The vectors w hich define the elem entary reflectors, as retumed by SSY TRD.

LD A (input)
The leading dim ension of the anay A. LDA >= \(\max (1, M)\) ifS \(\mathbb{D} E=L ; L D A>=m a x(1, N)\) if \(S \mathbb{D} E=\) R.

TAU (input)
\((M-1)\) ifSTDE \(=\mathbb{L}(N-1)\) if \(S \mathbb{D E}=R^{\prime} T A U\) (i) \(m\) ust contain the scalar factor of the elem entary reflectorH (i), as retumed by SSY TRD .

C (input/output)
On entry, the \(M\) boy -N m atrix C . On exit, C is overw ritten by \(Q * C\) or \(Q * * T * C\) or \(C * Q * *\) or \(C * Q\).

LD C (input)
The leading dim ension of the aray C. LD C >= \(m\) ax (1, M).

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. IfSDE \(=\mathrm{L}\) ', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LW ORK >= \(m a x(1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=L '\), and \(L W O R K>=M * N B\) if \(S \mathbb{D} E=R\) ', w here NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N N}\) FO = -i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
spbcon -estim ate the reciprocalof the condition num ber (in the 1 -norm ) of a real sym metric positive definite band \(m\) atrix using the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(\mathrm{A}=\) L*L**T com puted by SPBTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SPBCON (UPLO,N,KD,A,LDA,ANORM,RCOND,W ORK,W ORK 2,}
\mathbb{NFO)}
CHARACTER * 1 UPLO
INTEGERN,KD,LDA,\mathbb{NFO}
\mathbb{NTEGER W ORK2 (*)}
REAL ANORM,RCOND
REALA (LDA,*),W ORK (*)
SUBROUT\mathbb{NE SPBCON_64 (UPLO,N,KD,A,LDA,ANORM,RCOND,W ORK,}
W ORK2, \mathbb{NFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N,KD,LDA,INFO}
\mathbb{NTEGER*8W ORK2 (*)}
REAL ANORM,RCOND
REALA (LDA,*),W ORK (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE PBCON $\mathbb{U} P L O, \mathbb{N}], K D, A,[L D A], A N O R M, R C O N D,[W O R K]$, [W ORK2], [ $\mathbb{N} F O$ ])
CHARACTER (LEN=1)::UPLO
$\mathbb{N} T E G E R:: N, K D, L D A, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I}$ ENSION (:) ::W ORK 2

```

REAL ::ANORM,RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::W ORK
REAL,D \(\mathbb{M}\) ENSION (: : : : : A

SU BROUTINE PBCON_64 (UPLO, \(\mathbb{N}], K D, A,[L D A], A N O R M, R C O N D,[W O R K]\), [ W ORK2], \([\mathbb{N} \mathrm{FO}])\)

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{KD}, \mathrm{LD} A, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{I M} \operatorname{ENSION}\) (:) ::W ORK2
REAL ::ANORM,RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void spbcon (charuple, intn, int kd, float *a, int lda, floatanorm, float*rcond, int*info);
void spbcon_64 (charuplo, long n, long kd, float *a, long lda, float anorm , float *roond, long *info);

\section*{PURPOSE}
spbcon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a real symm etric positive definite band m atrix using the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(\mathrm{A}=\) L*L**T com puted by SPB TRF .

A \(n\) estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND = 1 / (ANORM * norm (inv (A))).

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': U pper triangular factor stored in A ;
= L ': Low er triangular factor stored in A .

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

K D (input)
The num ber of superdiagonals of the \(m\) atrix \(A\) if \(\mathrm{UPLO}=\mathrm{U}\) ', or the num berof subdiagonals ifUPLO \(=\mathrm{L}^{\prime} . \mathrm{KD}>=0\) 。

A (input) The triangular factorU or \(L\) from the Cholesky
factorization \(A=U * * T * U\) orA \(=\mathrm{L} * \mathrm{~L} * * T\) of the band
\(m\) atrix \(A\), stored in the first \(K D+1\) row \(s\) of the array. The \(j\) th colum n ofU orL is stored in the \(j\) th colum \(n\) of the array A as follow s: if UPLO
\(=U \prime A(k d+1+i-j)=U(i, j)\) for \(\max (1, j\)
\(\mathrm{kd})<=\dot{i}<=\dot{j}\) ifUPLO \(=\mathrm{L}\) ', \(A(1+i-j)=L(i, j)\)
for \(\dot{j}=i<=m\) in \((n, j+k d)\).

LD A (input)
The leading dim ension of the aray A. LDA >= K D +1 。
ANORM (input)
The 1-norm (or infinity-norm ) of the symm etric band \(m\) atrix A.

\section*{RCOND (output)}

The reciprocal of the condition num ber of the \(m\) atrix \(A\), com puted as RCOND \(=1 /(A N O R M \star A \mathbb{N V N M ) \text { , }}\) \(w\) here \(A \mathbb{N} V N M\) is an estim ate of the 1 -norm of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension ( \(3 * N\) )

W ORK2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the i-th argum enthad an ille-
galvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
spbequ -com pute row and colum n scalings intended to equilibrate a sym \(m\) etric positive definite band \(m\) atrix \(A\) and reduce its condition num ber (w ith respect to the tw o-norm )

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SPBEQU(UPLO,N,KD,A,LDA,SCALE,SCOND,AMAX,INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGERN,KD,LDA,NNFO}
REAL SCOND,AMAX
REALA (LDA,*),SCALE (*)
SU BROUT\mathbb{NE SPBEQU_64 (UPLO ,N ,KD ,A ,LDA, SCALE,SCOND,AM AX,}
INFO)
CHARACTER * 1 UPLO
INTEGER*8N,KD,LDA,INFO
REALSCOND,AMAX
REAL A (LDA,*),SCALE (*)

```

\section*{F95 INTERFACE}

SU BROUTINE PBEQU (UPLO, \(\mathbb{N}], K D, A,[L D A], S C A L E, S C O N D, A M A X\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N} T E G E R:: N, K D, L D A, \mathbb{N} F O\)
REAL ::SCOND,AMAX
REAL,D \(\mathbb{M}\) ENSION (:) ::SCALE
REAL,D \(\mathbb{I}\) ENSION (:,:) ::A
SU BROUTINE PBEQU_64 (UPLO, \(\mathbb{N}]\) ],KD,A, [LDA],SCALE, SCOND,AMAX,
[ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{KD}, \mathrm{LD} \mathrm{A}, \mathbb{N} F O\)
REAL :: SCOND,AMAX
REAL,D \(\mathbb{I M} E N S I O N\) (:) ::SCALE
REAL,D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void spbequ (charuplo, intn, intkd, float *a, int lda, float *scale, float *scond, float *am ax, int *info);
void spbequ_64 (charuplo, long n, long kd, float *a, long lda, float *scale, float *scond, float *am ax, long *info);

\section*{PURPOSE}
spbequ com putes row and colum n scalings intended to equilibrate a sym \(m\) etric positive definite band \(m\) atrix \(A\) and reduce its condition num ber ( \(w\) ith respect to the tw o-norm ) . S contains the scale factors, \(S(i)=1 /\) sqrt (A ( \(i, i)\) ), chosen so that the scaled matrix \(B \quad w\) ith elem ents \(B(i, j)=\) \(S(i) \star A(i, 7) * S(i)\) has ones on the diagonal. This choige of \(S\) puts the condition num berofB w ithin a factor \(N\) of the sm allest possible condition num ber over allpossible diagonalscalings.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': Upper triangular ofA is stored;
\(=\mathbb{L}\) ': Low ertriangularofA is stored.

N (input) The order of them atrix A. \(\mathrm{N}>=0\).

K D (input)
The num ber of superdiagonals of the \(m\) atrix \(A\) if \(\mathrm{UPLO}=\mathrm{U}\) ', or the num ber of subdiagonals ifUPLO \(=\mathbb{L} . \mathrm{KD}>=0\) 。

A (input) The upper or low er triangle of the sym \(m\) etric band \(m\) atrix \(A\), stored in the firstK \(D+1\) row s of the anay. The \(j\) th colum \(n\) ofA is stored in the \(j\) th colum \(n\) of the anray A as follow s : if UPLO = U',

A \((k d+1+i-j, j)=A(i, 7)\) for \(\max (1, j k d)<=i<=j\) if UPLO \(=L^{\prime}, A(1+i-j)=A(i, 7)\) for \(\dot{j}=i<=m\) in \((n, \dot{j}+k d)\).

LD A (input)
The leading dim ension of the array A. LDA >= KD+1.

SCALE (output)
If \(\mathbb{N} F O=0, S C A L E\) contains the scale factors for A.

SCOND (output)
If \(\mathbb{N} F O=0, S C A\) LE contains the ratio of the sm allest SCA LE (i) to the largestSCA LE (i). If SCOND \(>=0.1\) and AMAX is neither too large nor too sm all, it is notw orth scaling by SCA LE .

AM AX (output)
A bsolute value of largestm atrix elem ent. IfA M A X
is very close to overflow orvery close to under-
flow , the m atrix should be scaled.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvahue.
\(>0\) : if \(\mathbb{N F O}=\) i, the \(i\)-th diagonal elem ent is nonpositive.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
spbrifs - im prove the com puted solution to a system of linear equations when the coefficientm atrix is sym \(m\) etric positive definite and banded, and provides emrorbounds and backw ard errorestim ates for the solution

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SPBRFS (UPLO,N,KD,NRHS,A,LDA,AF,LDAF,B,LDB,X,}
LD X,FERR,BERR,W ORK,W ORK 2, INFO)
CHARACTER * 1 UPLO
INTEGERN,KD,NRHS,LDA,LDAF,LDB,LDX,INFO
INTEGER W ORK2 (*)
REAL A (LDA,*), AF (LDAF,*), B (LDB ,*), X (LDX ,*), FERR (*),
BERR (*),W ORK (*)
SU BROUTINE SPBRFS_64 (UPLO,N,KD,NRHS,A,LDA,AF,LDAF,B,LDB,
X,LDX,FERR,BERR,W ORK,W ORK 2, \mathbb{NFO)}
CHARACTER * 1 UPLO
INTEGER*8N,KD,NRHS,LDA,LDAF,LDB,LDX,INFO
INTEGER*8 W ORK 2 (*)
REAL A (LDA,*), AF (LDAF,*), B (LDB ,*), X (LDX,*), FERR (*),
BERR (*),W ORK (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE PBRFS (UPLO, $\mathbb{N}], K D, \mathbb{N R H S}], A,[L D A], A F,[L D A F], B$, [LD B], X, [LDX ],FERR,BERR, [W ORK], [W ORK 2], [ $\mathbb{N F O}]$ )
CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R:: N, K D, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I}$ ENSION (:) ::W ORK2

```

REAL,D \(\mathbb{I M} E N S I O N(:):: F E R R, B E R R, W\) ORK
REAL,D \(\operatorname{IM}\) ENSION (:,:) ::A,AF,B,X
SUBROUTINE PBRFS_64 (UPLO, \(\mathbb{N}], K D,[N R H S], A,[L D A], A F,[L D A F]\), \(B,[L D B], X,[L D X], F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N} F O])\)

CHARACTER (LEN=1) :: UPLO
\(\mathbb{N}\) TEGER (8) :: N , KD, NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{M}\) ENSION (:) ::W ORK2
REAL,D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK
REAL,D \(\mathbb{I M}\) ENSION (: : : : : A, AF, B, X

\section*{C INTERFACE}
\#include <sunperfh>
void spbrfs (char uplo, intn, intkd, int nins, float *a, int lda, float*af, int ldaf, float*b, int ldb, float *x, int ldx, float * ferr, float *berr, int *info);
void spbrfs_64 (charuplo, long n, long kd, long nrhs, float
*a, long lda, float *af, long ldaf, float *b, long ldlb, float *x, long ldx, float * ferr, float *berr, long *info);

\section*{PURPOSE}
spbrfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is sym \(m\) etric positive definite and banded, and provides emorbounds and backw ard error estim ates forthe solution.

\section*{ARGUMENTS}

UPLO (input)
= U : : U pper triangle ofA is stored;
= L': Low ertriangle ofA is stored.
N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).
KD (input)
The num ber of superdiagonals of the \(m\) atrix \(A\) if UPLO \(=\mathrm{U}\) ', or the num berof subdiagonals if UPLO \(=\mathbb{L}^{\prime} . \mathrm{KD}>=0\) 。

NRHS (input)
The num ber of righthand sides, i.e., the num ber
of collm ns of the m atrices B and X. NRH S \(>=0\).

A (input) The upper or low er triangle of the sym m etric band \(m\) atrix \(A\), stored in the firstKD +1 row s of the anay. The \(j\) th colum \(n\) ofA is stored in the \(j\) th colum \(n\) of the anray A as follow \(s\) : if UPLO \(=\mathrm{U}\) ', \(A(k d+1+i-j, j)=A(i, j)\) for \(m a x(1, j k d)<=i<=j\) if \(\mathrm{UPLO}=\mathrm{L}\) ', \(A(1+i-j, \bar{j})=A(i, j)\) for \(j<=i<=m\) in \((n, j+k d)\).

LDA (input)
The leading dim ension of the anay A. LDA >= K D +1.
AF (input)
The triangular factor \(U\) or \(L\) from the Cholesky factorization \(A=U * * T * U\) orA \(=\mathrm{L} * \mathrm{~L} * * T\) of the band \(m\) atrix A as computed by SPBTRF, in the same storage form atas A (see A).

LDAF (input)
The leading dim ension of the array AF. LDAF >= K D +1 .
\(B\) (input) The righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by SPB TRS. On exit, the im proved solution \(m\) atrix X .

LD X (input)
The leading dim ension of the array X. LD X >= \(\max (1, N)\).

\section*{FERR (output)}

The estim ated forw ard enorbound for each solution vectorX ( 7 ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(i), F E R R(i)\) is an estim ated upperbound forthe \(m\) agnitude of the largest ele\(m\) entin ( \(X(\mathcal{J})-X\) TRUE) divided by the \(m\) agninude of the largestelem entin X ( 7 ). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slightoverestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each
solution vectorX (i) (ie., the sm allest relative
change in any elem entofA orB thatm akes X ( 7 ) an exactsolution).

W ORK (w orkspace)
dim ension ( \(3 * N\) )
W ORK 2 (w orkspace) dim ension \(\mathbb{N}\) )
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N} F O=-i\), the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
spbstf - com pute a splitCholesky factorization of a real
sym \(m\) etric posilive definite band \(m\) atrix A

\section*{SYNOPSIS}
```

SUBROUTINE SPBSTF (UPLO,N,KD,AB,LDAB,\mathbb{NFO)}
CHARACTER * 1 UPLO
INTEGERN,KD,LDAB,INFO
REALAB (LDAB,*)
SU BROUT\mathbb{NE SPBSTF_64(UPLO,N,KD,AB,LDAB,INFO)}
CHARACTER * 1 UPLO
INTEGER*8N,KD,LDAB,INFO
REALAB (LDAB,*)
F95 INTERFACE
SUBROUT\mathbb{NE PBSTF (UPLO, N ],KD,AB,[LDAB], [NNO ])}
CHARACTER (LEN=1) ::UPLO
\mathbb{NTEGER ::N,KD,LDAB,INFO}
REAL,DIM ENSION (:,:)::AB
SU BROUT\mathbb{NE PBSTF_64 (UPLO , N ],KD ,AB, [LDAB ], [N FO ])}
CHARACTER (LEN=1) ::UPLO

```

```

REAL,D IM ENSION (:,:) ::AB

```
void spbstf(charuplo, intn, intkd, float*ab, int ldab, int*info);
void spbstf_ 64 (charuplo, long n, long kd, float *ab, long ldab, long *info);

\section*{PURPOSE}
spbstf com putes a splitC holesky factorization of a real sym m etric positive definite band \(m\) atrix A.

This routine is designed to be used in conjunction with SSBG ST .
The factorization has the form \(A=S * * T * S\) where \(S\) is a band \(m\) atrix of the sam e bandw idth as A and the follow ing structure:
\[
\begin{array}{r}
S=\left(\begin{array}{ll}
U & ) \\
(M \quad L)
\end{array}, ~\right.
\end{array}
\]
w here U is upper triangular of orderm \(=(\mathrm{n}+\mathrm{kd}) / 2\), and L is low er triangular of ordern-m .

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': U ppertriangle ofA is stored;
\(=\mathrm{L}\) ': Low er triangle ofA is stored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

KD (input)
The num ber of superdiagonals of the \(m\) atrix \(A\) if
\(\mathrm{UPLO}=\mathrm{U}\) ', orthe num ber of subdiagonals ifU PLO
\(=\mathbb{L} . \mathrm{KD}>=0\) 。

A B (input/output)
O n entry, the upper or low er triangle of the sym \(m\) etric band \(m\) atrix \(A\), stored in the first kd+1 row s of the amay. The \(j\) th colum n of A is stored in the \(j\) th colum \(n\) of the array A B as follow s: if \(\mathrm{UPLO}=\mathrm{U}^{\prime}, \mathrm{AB}(\mathrm{kd}+1+i-j, j)=A(i, j)\) for \(\mathrm{max}(1, j\) \(\mathrm{kd})<=i<=\dot{j}\) ifU PLO \(=L^{\prime}, \mathrm{AB}(1+i-j, j)=A(i, j)\) for \(j<=i<=m\) in \((n, j+k d)\).

On exit, if \(\mathbb{N F O}=0\), the factors from the split Cholesky factorization \(A=S * * T * S\). See Further D etails.

LDAB (input)
The leading dim ension of the array AB. LD A B >= K D +1 .
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue \(>0:\) if \(\mathbb{N} F O=\) i, the factorization could not be com pleted, because the updated elem enta (i,i) w as negative; the \(m\) atrix \(A\) is notposilive definite.

\section*{FURTHER DETAILS}

The band storage schem e is illustrated by the follow ing exam ple, w hen \(N=7, K D=2\) :
\begin{tabular}{|c|c|}
\hline \multicolumn{2}{|l|}{\(\mathrm{S}=(\mathrm{s} 11 \mathrm{~s} 12 \mathrm{~s} 13\)} \\
\hline ( & s22 s23 s24 ) \\
\hline ( & s33 s34 ) \\
\hline ( & s44 ) \\
\hline ( & s53 s54 s55 \\
\hline ( & s64 s65 s66 ) \\
\hline ( & s75 s76 s77) \\
\hline
\end{tabular}

If \(\mathrm{U} P \mathrm{O}=\mathrm{U}\) ', the amay A B holds:
on entry: on exit:
* * a13 a24 a35 a46 a57 * * s13 s24 s53
s64 s75
* a12 a23 a34 a45 a56 a67 * s12 s23 s34 s54
s65 s76 a11 a22 a33 a44 a55 a66 a77 s11 s22 s33
s44 s55 s66 s77

IfU PLO = L', the array AB holds:
on entry: on exit:
```

a11 a22 a33 a44 a55 a66 a77 s11 s22 s33 s44 s55
s66 s77 a21 a32 a43 a54 a65 a76 * s12 s23 s34
s54 s65 s76 * a31 a42 a53 a64 a64 * * s13
s24 s53 s64 s75 * *

```

A may elem entsm arked * are notused by the routine.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
spbsv - com pute the solution to a real system of linear equations A * X = B,

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SPBSV (UPLO,N ,ND IA G ,NRHS,A ,LDA ,B,LDB, IN FO )}
CHARACTER * 1 UPLO
INTEGERN,ND IA G,NRHS,LDA,LDB, IN FO
REALA (LDA,*),B(LDB,*)
SU BROUT\mathbb{NE SPBSV_64(UPLO,N,ND IA G ,NRHS,A,LDA,B,LDB, IN FO )}
CHARACTER * 1 UPLO
INTEGER*8N,NDIAG,NRHS,LDA,LDB, INFO
REAL A (LDA,*),B(LDB,*)

```

\section*{F95 INTERFACE}

SU BROUTINE PBSV (UPLO, \(\mathbb{N}], N D \mathbb{I A} G, \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N} T E G E R:: N, N D \mathbb{A} G, N R H S, L D A, L D B, \mathbb{N} F O\)
REAL,D \(\mathbb{I}\) ENSION (: : : : : A, B
SU BROUTINE PBSV_64 (UPLO, \(\mathbb{N}], N D \mathbb{I} G, \mathbb{N} R H S], A,[L D A], B,[L D B]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N} T E G E R(8):: N, N D \mathbb{I} G, N R H S, L D A, L D B, \mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include <sunperfh>
void spbsv (char uplo, intn, intndiag, intnins, float *a, int lda, float *b, int ldlo, int *info);
void spbss_64 (charuplo, long n, long ndiag, long nihs, float *a, long lda, float *b, long ldb, long *info);

\section*{PURPOSE}
spbsv com putes the solution to a realsystem of linear equations
\(A * X=B, w h e r e A\) is an \(N\) boy \(N\) sym m etric positive defintie band \(m\) atrix and X and B are N -by-N R H S m atrices. The Cholesky decom position is used to factorA as
\(A=U * * T * U\), if \(U P L O=U\) ', or
\(A=L \star L \star * T\), if \(\mathrm{UPLO}=\mathrm{L}\) ',
\(w\) here \(U\) is an uppertriangularband \(m\) atrix, and \(L\) is a low er triangular band \(m\) atrix, w ith the sam e num ber of superdiagonals or subdiagonals as A. The factored form of A is then used to solve the system ofequations A * X \(=\mathrm{B}\).

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': Uppertriangle of A is stored;
\(=\mathrm{L}\) ': Low ertriangle ofA is stored.

N (input) The num ber of linear equations, ie., the order of them atrix \(A . N>=0\).

ND IA G (input)
The num ber of superdiagonals of the \(m\) atrix \(A\) if \(\mathrm{UPLO}=\mathrm{U}\) ', or the num ber of subdiagonals ifU PLO
\(=\mathbb{L}\) '. NDIAG > \(=0\) 。

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of the m atrix B. NRHS \(>=0\).

A (input/output)
O n entry, the upper or low er triangle of the sym \(m\) etric band \(m\) atrix \(A\), stored in the firstN D IA \(G+1\) row s of the array. The jth colum n of A is stored in the \(j\) th colum \(n\) of the array \(A\) as follow \(s\) : if
 \(\max (1, j \mathrm{jND} \mathrm{IAG})<=i<=j\) if \(\mathrm{UPLO}=\mathrm{L}^{\prime}, \mathrm{A}(1+i-j)\)
\(=A(i, j)\) for \(j<i<=m\) in \((\mathbb{N}, j+N D\) IA G ). See below for furtherdetails.

On exit, if \(\mathbb{N} F O=0\), the triangular factor \(U\) orL
from the Cholesky factorization \(A=U * * T * U\) or \(A=\)
\(\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\) of the band m atrix A , in the sam e storage form atas A.

LD A (input)
The leading dim ension of the aray A. LD A >= N D IA G +1.
B (input/output)
On entry, the N -by-NRHS righthand side m atrix B. On exit, if \(\mathbb{N F O}=0\), the N boy -N RH S solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array \(\mathrm{B} . \operatorname{LD} \mathrm{B}>=\) \(\max (1, N)\).

IN FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvałue \(>0:\) if \(\mathbb{N F O}=i\), the leading \(m\) inoroforder iof \(A\) is notpositive definite, so the factorization could notbe com pleted, and the solution has not been com puted.

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, w hen \(N=6, N D I A G=2\), and \(U P L O=U ':\)

On entry: On exit:
* * a13 a24 a35 a46 * * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55 u66

Sim ilarly, ifU PLO = 'L 'the form atofA is as follow s:

On entry: On exit:

a31 a42 a53 a64 * * 131142153164 * *

A may elem entsm arked * are notused by the routine.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
spbsvx - use the C holesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\) to com pute the solution to a realsystem of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SPBSVX (FACT,UPLO,N,NDIAG,NRHS,A,LDA,AF,LDAF,}
EQUED,S,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK2,
\mathbb{NFO)}

```
CHARACTER * 1 FACT, UPLO, EQUED
\(\mathbb{N}\) TEGERN,ND \(\operatorname{IA} G, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N}\) TEGERWORK2 (*)
REALRCOND

\(\operatorname{FERR}\) (*), \(\operatorname{BERR}\) ( \(^{\star}\) ), \(\mathrm{W} O \operatorname{OR}\left({ }^{*}\right)\)
SU BROUTINE SPBSVX_64 EACT, UPLO,N,ND IAG,NRHS,A,LDA,AF,LDAF,
    EQUED,S,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK2,
    \(\mathbb{N} F O\) )
CHARACTER * 1 FACT, UPLO, EQUED
\(\mathbb{N} T E G E R * 8 N, N D \mathbb{A} G, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{I N T E G E R} * 8 \mathrm{~W}\) ORK 2 ( \({ }^{*}\) )
REAL RCOND

\(\operatorname{FERR}\) ( \(\left.^{*}\right), \operatorname{BERR}\) ( \(\left.^{*}\right), \mathrm{W} O R K(*)\)

\section*{F95 INTERFACE}

SU BROUTINE PBSVX (FACT,UPLO, \(\mathbb{N}], N D \mathbb{I} G, \mathbb{N R H S}], A,[L D A], A F,[L D A F]\), EQUED, S, B, [LDB],X, [LDX],RCOND,FERR,BERR, [W ORK],
[W ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER ( \(L E N=1\) ) : :FACT, UPLO, EQUED
\(\mathbb{N} T E G E R:: N, N D \mathbb{A} G, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) ::W ORK2
REAL: RCOND
REAL,D \(\mathbb{I M} E N S I O N(:):: S, F E R R, B E R R, W O R K\)
REAL,D \(\mathbb{I}\) ENSION (: : : : : A , AF, B , X

SU BROUTINE PBSVX_64 (FACT, UPLO, \(\mathbb{N}], N D I A G, ~ \mathbb{N} R H S], A,[L D A], A F\), [LDAF], EQUED, S, B, [LDB],X, [LDX],RCOND,FERR,BERR, \([\mathbb{W} O R K],[\mathbb{W} O R K 2],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::FACT, UPLO, EQUED
\(\mathbb{N}\) TEGER (8) :: N, ND IAG,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{M}\) ENSION (:) ::WORK2
REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) :: S, FERR,BERR,W ORK
REAL,D \(\mathbb{I M}\) ENSION (: : : : : : A , AF, B , X

\section*{C INTERFACE}
\#include <sunperfh>
void spbsvx (char fact, charuplo, int n, int ndiag, int nhs, float *a, int lda, float *af, int ldaf, char equed, float *s, float *b, int ldb, float *x, int ldx, float *roond, float *ferr, float *berr, int *info);
void spbsvx_64 (char fact, charuplo, long n, long ndiag, long nihs, float *a, long lda, float*af, long ldaf, charequed, float *s, float *b, long ldb, float *x, long ldx, float*roond, float*ferr, float *berr, long *info);

\section*{PURPOSE}
spbsvx uses the C holesky factorization \(A=U * * T * U\) or \(A=\) L*L**T to com pute the solution to a realsystem of linear equations
\(A * X=B\), where \(A\) is an \(N\) boy \(-N\) sym m etric positive definite band \(m\) atrix and \(X\) and \(B\) are \(N\) boy-N RH S m atrices.

Emorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT \(=\) E', real scaling factors are computed to
equilibrate
the system :
\(\operatorname{diag}(S) * A * \operatorname{diag}(S) * \operatorname{inv}(\operatorname{diag}(S)) * X=\operatorname{diag}(S) * B\)
W hether or not the system w illbe equilibrated depends on the scaling of the m atrix A, but if equilibration is used, A is
overw ritten by diag \((S) \star A * d i a g(S)\) and \(B\) by diag \((S) * B\).
2. IfFACT = N 'or E ', the Cholesky decom position is used to
factor them atrix A (afterequilibration ifFACT = E) as
\(A=U * * T * U\), ifUPLO \(=U\) ', or
\(\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\), if \(\mathrm{UPLO}=\mathrm{L}\) ',
\(w\) here \(U\) is an upper triangularband \(m\) atrix, and \(L\) is a
low er
triangularband \(m\) atrix.
3. If the leading i-by-iprincipal m inor is not positive definite,
then the routine retums w ith \(\mathbb{N} F O=\) i. O therw ise, the factored
form ofA is used to estim ate the condition num ber of the \(m\) atrix
A. If the reciprocal of the condition num ber is less than \(m\) achine precision, \(\mathbb{N} F O=N+1\) is retumed as a w aming, but the routine
stillgoes on to solve for X and com pute errorbounds as described below .
4.The system of equations is solved for \(X\) using the factored form of A.
5. Herative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for it.
6. Ifequilibration w as used, the \(m\) atrix \(X\) is prem ultiplied by diag (S) so that it solves the original system before equilibration.

\section*{ARGUMENTS}

\section*{FACT (input)}

Specifies w hether ornot the factored form of the \(m\) atrix A is supplied on entry, and ifnot, w hether the m atrix A should be equilibrated before it is factored. = F : On entry, AF contains the factored form ofA. IfEQUED = \(Y\) ', them atrix A has been equilibrated \(w\) ith scaling factors given by \(S\). A and AF w illnotbe m odified. \(=\mathrm{N}\) ': Them atrix A w illbe copied to A F and factored.
= E : : The matrix A will be equilibrated if necessary, then copied to A F and factored.

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.
N (input) The num ber of linearequations, i.e., the order of the matrix A. \(\mathrm{N}>=0\).
ND IA G (input)
The num ber of superdiagonals of the matrix A if UPLO = U',orthe num berof subdiagonals ifU PLO
= L'. ND IAG >= 0 .

NRHS (input)
The num ber of right-hand sides, i.e., the num ber of collm ns of the matrices B and X. NRHS >= 0 .

A (input/output)
On entry, the upper or low er triangle of the sym \(m\) etric band \(m\) atrix \(A\), stored in the firstND IA G +1 row s of the array, except ifFACT = F'and EQUED \(=Y\) ', then A m ust contain the equilibrated \(m\) atrix diag \((S){ }^{*} A\) *diag \((S)\). The jth colum n of A is stored in the \(j\) th colum \(n\) of the array \(A\) as follow \(s\) : if UPLO = U', A NDIAG+1+i-ji) =A (i, \(\boldsymbol{j}\) ) for \(\max (1, j\) ND IAG \()<=i<=j\) if UPLO \(=\mathrm{L}\) ', A ( \(1+i-j, j)\) \(=A(i, j)\) for \(j=i<=m\) in \((\mathbb{N}, \dot{j}+N D \mathbb{I A} G)\). See below for furtherdetails.

Onexit, ifFACT = E' and EQUED = Y', A is overw rilten by diag (S)*A *diag (S).

LDA (input)
The leading dim ension of the array A. LDA >= ND IA G +1.

AF (input/output)
If FACT = F ', then \(A F\) is an inputargum entand on entry contains the triangular factorU or \(L\) from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(\mathrm{A}=\)
\(L * L * * T\) of the band \(m\) atrix \(A\), in the sam e storage form atas A (see A). IfEQUED \(=Y\) ', then \(A F\) is the factored form of the equilibrated \(m\) atrix A.

IfFACT \(=N\) ', then AF is an output argum ent and on exit retums the triangular factorU orL from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(\mathrm{A}=\) L*L**T。

IfFACT = E', then AF is an output argum ent and on exitretums the triangular factor \(U\) or \(L\) from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\) of the equilibrated \(m\) atrix \(A\) (see the description of for the form of the equilibrated m atrix).
LD AF (input)
The leading dim ension of the array AF. LDAF >= \(N D I A G+1\).

EQUED (input)
Specifies the form of equilibration thatw as done.
\(=\mathrm{N}\) : N o equilibration (alw ays true iffACT = N 7 。
\(=Y^{\prime}:\) Equilibration w as done, i.e., A has been replaced by diag \((\mathrm{S})\) * A * diag \((\mathrm{S})\). EQUED is an inputargum entifFACT = F'; otherw ise, it is an outputargum ent.

S (input/output)
The scale factors forA; notaccessed if EQUED = \(N^{\prime} . S\) is an inputargum entifFACT = \(F^{\prime}\); otherw ise, S is an outputargum ent. IfFACT \(=\mathrm{F}^{\prime}\) and EQUED \(=Y\) ', each elem entof m ustbe posilive.

B (input/output)
O n entry, the N Hoy-NRH S righthand side m atrix B.
On exit, if EQUED = \(N^{\prime}\), \(B\) is notm odified; if EQUED \(=Y\) ', \(B\) is overw ritten by diag \((S) * B\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{I N F O}=\mathrm{N}+1\), the N -by-NRHS solution
\(m\) atrix \(X\) to the original system of equations.
\(N\) ote that if EQUED \(=Y\) ', A and \(B\) arem odified on exit, and the solution to the equilibrated system is inv (diag (S ) ) *X .

\section*{LD X (input)}

The leading dim ension of the anay X . LD X >= \(\max (1, N)\).

\section*{RCOND (output)}

The estim ate of the reciprocal condition num ber of the \(m\) atrix \(A\) after equilibration (if done). If RCOND is less than them achine precision (in particular, ifRCOND \(=0\) ), the \(m\) atrix is singular to w orking precision. This condition is indicated by a retum code of \(\mathbb{N} F O>0\).

FERR (output)
The estim ated forw ard emorbound for each solution vector \(X(\mathcal{)}\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution comesponding to \(X(\mathcal{O})\), FERR ( \()\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(\mathrm{X}(\mathcal{i})-\mathrm{X}\) TRU E ) divided by the m agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each
solution vectorX (j) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension ( \(3 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{N N}\) FO = -i, the i-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N} F O=i\), and \(i\) is
<= N : the leading \(m\) inoroforderiof \(A\) is not positive definite, so the factorization could not be com pleted, and the solution has not been com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1\) : U is nonsingular, butRCOND is less than machine precision, \(m\) eaning that the \(m\) atrix is singular to \(w\) orking precision. N evertheless, the solution and error bounds are com puted because there are a num ber of situations w here the com puted solution can bem ore accurate than the value ofRCOND w ould

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, when \(N=6, N D \mathbb{I A G}=2\), and UPLO \(=U\) ':

Tw o-dim ensional storage of the sym \(m\) etric \(m\) atrix A : all al2 a13 a22 a23 a24
a33 a34 a35
a44 a45 a46
a55 a56
(aijong (ä̈)) a66
\(B\) and storage of the upper triangle ofA :
* * a13 a24 a35 a46
* a12 a23 a34 a45 a56
a11 a22 a33 a44 a55 a66

Sim ilarly, if U PLO = L'the form atofA is as follow s:
a11 a22 a33 a44 a55 a66
a21 a32 a43 a54 a65 *
a31 a42 a53 a64 * *
A ray elem entsm arked * are notused by the routine.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
spbtf2 - com pute the C holesky factorization of a real sym m etric positive definite band \(m\) atrix \(A\)

\section*{SYNOPSIS}
```

SUBROUTINE SPBTF2(UPLO,N,KD,AB,LDAB,\mathbb{NFO)}
CHARACTER * 1 UPLO
INTEGERN,KD,LDAB,INFO
REALAB (LDAB,*)
SU BROUT\mathbb{NE SPBTF2_64(UPLO,N,KD,AB,LDAB,INFO)}
CHARACTER * 1 UPLO
INTEGER*8N,KD,LDAB,INFO
REALAB (LDAB,*)
F95 INTERFACE
SU BROUT\mathbb{NE PBTF2 (UPLO, N ],KD,AB,[LDAB], [NFO ])}
CHARACTER (LEN=1) ::UPLO
\mathbb{NTEGER ::N,KD,LDAB,INFO}
REAL,DIM ENSION (:,:)::AB
SU BROUT\mathbb{NE PBTF2_64 (UPLO, N ],KD ,AB, [LDAB ], [NNO ])}
CHARACTER (LEN=1) ::UPLO

```

```

    REAL,D IM ENSION (:,:) ::AB
    ```
void spbtf2 (charuplo, intn, intkd, float *ab, int ldab, int*info);
void spbtf2_64 (charuplo, long n, long kd, float *ab, long ldab, long *info);

\section*{PURPOSE}
spbtf2 com putes the C holesky factorization of a real sym \(m\) etric positive definite band \(m\) atrix A.

The factorization has the form
```

A = U'* U , ifUPLO = U',or
A = L * L', ifUPLO = L',

```
where \(U\) is an uppertriangularm atrix, \(U\) 'is the transpose of \(U\), and \(L\) is low er triangular.

This is the unblocked version of the algorithm, calling Level2 BLAS.

\section*{ARGUMENTS}

\section*{UPLO (input)}

Specifies w hether the upper or low er triangular
part of the sym m etricm atrix A is stored:
= U ': U pper triangular
= L': Low er triangular

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

KD (input)
The num ber of super-diagonals of the m atrix A if UPLO \(=\mathrm{U}\) ', or the num ber of sub-diagonals if UPLO \(=\mathbb{L} . \mathrm{KD}>=0\) 。

A B (input/output)
O n entry, the upper or low er triangle of the sym \(m\) etric band \(m\) atrix \(A\), stored in the firstK \(D+1\) row s of the array. The \(j\) th colum n ofA is stored in the \(j\) th colum n of the array A B as follow s: if \(\mathrm{UPLO}=\mathrm{U}\) ', AB \((k d+1+i-j, j)=A(i, j)\) for \(\max (1, j\) \(\mathrm{kd})<=i<=\dot{j}\) ifUPLO \(=L ', A B(1+i-j, j)=A(i, j)\) for \(j<=i<=m\) in \((n, j+k d)\).

On exit, if \(\mathbb{N} F O=0\), the triangular factor \(U\) orL
from the Cholesky factorization \(\mathrm{A}=\mathrm{U}\) * U or \(\mathrm{A}=\) L * L ' of the band m atrix A , in the sam e storage form atas A.

LDAB (input)
The leading dim ension of the array AB. LD AB >= KD+1.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-k\), the \(k\)-th argum enthad an illegalvalue \(>0:\) if \(\mathbb{N} F O=k\), the leading \(m\) inoroforderk is notpositive definite, and the factorization could notbe com pleted.

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, when \(\mathrm{N}=6, \mathrm{KD}=2\), and \(\mathrm{U} P L O=\mathrm{U}\) ':

On entry: On exit:
* * a13 a24 a35 a46 * * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55 u66

Sim ilarly, if UPLO = L'the form atofA is as follow s:
On entry: On exit:

166
a21 a32 a43 a54 a65 * \(121 \quad 132143154165\)
*
a31 a42 a53 a64 * * 131142153164 *

A ray elem entsm arked * are not used by the routine.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
spbtrf-com pute the C holesky factorization of a real sym \(m\) etric positive definite band \(m\) atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SPBTRF(UPLO,N,KD,A,LDA, INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGERN,KD,LDA,}\mathbb{NFO}
REALA (LDA,*)
SU BROUT\mathbb{NE SPBTRF_64(UPLO,N,KD,A ,LDA, INFO)}
CHARACTER * 1 UPLO
INTEGER*8N,KD,LDA,INFO
REALA (LDA,*)
F95 INTERFACE
SUBROUT\mathbb{NE PBTRF (UPLO, N ],KD ,A,[LDA ], [NNFO])}
CHARACTER (LEN=1) ::UPLO
INTEGER ::N,KD,LDA,}\mathbb{N}F
REAL,D IM ENSION (:,:) ::A
SU BROUT\mathbb{NE PBTRF_64 (UPLO, N ],KD ,A, [LDA ], [\mathbb{NFO ])}}\mathbf{(})=
CHARACTER (LEN=1)::UPLO
\mathbb{NTEGER (8) ::N,KD,LDA,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REAL,D IM ENSION (:r:) ::A
void spbtrf(charuplo, intn, intkd, float *a, int lda, int *info);
void spbtrf_64 (char uplo, long n, long kd, float *a, long lda, long *info);

## PURPOSE

spbtrf com putes the C holesky factorization of a real sym $m$ etric positive definite band $m$ atrix A.

The factorization has the form

$$
\begin{aligned}
& A=U * * T * U, \text { if } \mathrm{UPLO}=\mathrm{U}^{\prime} \text {, or } \\
& \mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}, \text { if } \mathrm{UPLO}=\mathrm{L}^{\prime}
\end{aligned}
$$

$w$ here $U$ is an upper triangularm atrix and $L$ is low er triangular.

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': Uppertriangle of $A$ is stored;
$=1 \mathrm{~L}$ ': Low er triangle of A is stored.

N (input) The order of them atrix $A . N>=0$.

KD (input)
The num berof superdiagonals of the $m$ atrix $A$ if
$\mathrm{UPLO}=\mathrm{U}$ ', or the num ber of subdiagonals ifU PLO
$=L^{\prime} . K D>=0$ 。

A (input/output)
O n entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstKD +1 row s of the array. The $j$ th colum n of A is stored in the $j$ th colum $n$ of the anay $A$ as follow $s$ : if $\mathrm{UPLO}=U ', A(k d+1+i-j, j)=A(i, j)$ for $m a x(1, j$ $\mathrm{kd})<=i<=j$ if $\mathrm{UPLO}={ }^{\prime}$ ', $A(1+i-j, j)=A(i, j)$ for $\dot{j}=i<=m$ in $(n, j+k d)$.

On exit, if $\mathbb{N F O}=0$, the triangular factor $U$ orL from the Cholesky factorization $A=U * * T * U$ or $A=$ $\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ of the band m atrix A , in the sam e storage form atas A.

The leading dim ension of the array A. LDA >= K D +1 .
$\mathbb{N F O}$ (output)
= 0 : successfiulexit
<0: if $\mathbb{N}$ FO = -i, the i-th argum ent had an illegalvalue
$>0:$ if $\mathbb{N} F O=i$, the leading $m$ inoroforder is notpositive definite, and the factorization could notbe com pleted.

## FURTHER DETAILS

The band storage scheme is illustrated by the follow ing exam ple, when $N=6, K D=2$, and $U P L O=U:$
On entry: On exit:

```
    * a13 a24 a35 a46 * * u13 u24 u35
u46
    * a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
    a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55
u66
```

Sim ilarly, if UPLO = L 'the form atofA is as follow s:
On entry: Onexit:
a11 a22 a33 a44 a55 a66 $111 \quad 122 \quad 133144 \quad 155$
166
a21 a32 a43 a54 a65 * $121 \quad 132143154165$
*
a31 a42 a53 a64 * * 131142153164 *
*

A rray elem entsm arked * are notused by the routine.
C ontributed by
PeterM ayes and G inseppe Radicati, $\mathbb{B M}$ EC SEC , Rom e, M arch 23,1989

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

spbtrs-solve a system of linearequations $A * X=B$ w th a sym $m$ etric positive definite band $m$ atrix $A$ using the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ orA $=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ com puted by SPBTRF

## SYNOPSIS

```
SUBROUT\mathbb{NE SPBTRS (UPLO,N,KD,NRHS,A,LDA,B,LDB,NNFO)}
CHARACTER * 1 UPLO
INTEGERN,KD,NRHS,LDA,LDB,INFO
REAL A (LDA,*),B (LDB,*)
SUBROUT\mathbb{NE SPBTRS_64(UPLO,N,KD,NRHS,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO
NNTEGER*8N,KD,NRHS,LDA,LDB,\mathbb{NFO}
REALA (LDA,*),B (LDB,*)
```

F95 INTERFACE
SU BROUTINE PBTRS (UPLO, $\mathbb{N}], K D, \mathbb{N R H S}], A,[L D A], B,[L D B],[\mathbb{N F O}])$
CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER ::N,KD,NRHS,LDA,LDB, $\mathbb{N}$ FO
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A,B
SU BROUTINE PBTRS_64 (UPLO, $\mathbb{N}], K D, \mathbb{N} R H S], A,[L D A], B,[L D B]$,
[ $\mathbb{N} F O$ ])
CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER (8) ::N, KD,NRHS,LDA, LD B, $\mathbb{N} F O$
REAL,D $\mathbb{I M}$ ENSION (: : : : ::A, B

## C INTERFACE

\#include <sunperfh>
void spbtrs (charuplo, intn, intkd, int nrhs, float *a, int lda, float *b, int ldlo, int *info);
void spbtrs_64 (charuplo, long n, long kd, long nihs, float
*a, long lda, float *b, long ldb, long *info);

## PURPOSE

spbters solves a system of linearequations $A * X=B$ with a sym $m$ etric positive definite band $m$ atrix A using the C holesky factorization $A=U * * T * U$ orA $=L * L * * T$ com puted by SPBTRF .

## ARGUMENTS

UPLO (input)
$=U$ ': Upper triangular factorstored in A;
$=\mathrm{L}$ ': Low ertriangular factorstored in A.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

KD (input)
The num ber of superdiagonals of the $m$ atrix $A$ if
$\mathrm{UPLO}=\mathrm{U}$ ', orthe num ber of subdiagonals ifU PLO
$=\mathbb{L}^{\prime} . \mathrm{KD}>=0$ 。

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the m atrix B. NRH S $>=0$.

A (input) The triangular factor $U$ or $L$ from the Cholesky
factorization $A=U * * T * U$ or $A=L * L * * T$ of the band $m$ atrix A, stored in the first KD +1 row $s$ of the array. The $j$ th colum $n$ of $U$ orL is stored in the $j$ th colum n of the amay A as follow s: if UPLO $=U \prime, A(k d+1+i-j)=U(i, j)$ for $m a x(1, j$
$\mathrm{kd})<=i<=\dot{j}$ ifUPLO $=L \prime$ ' A $(1+i-j, j)=L(i, j)$
for $\dot{j}=i<=m$ in $(n, j+k d)$.

LD A (input)
The leading dim ension of the array A. LD A >= K D +1.

B (input/output)
O $n$ entry, the right hand side m atrix B. On exit,
the solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the aray B. LD B $>=$ $\max (1, N)$.

INFO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

spocon -estim ate the reciprocalof the condition num ber (in the 1 -norm ) of a realsym $m$ etric positive definite $m$ atrix using the C holesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ com puted by SPO TRF

## SYNOPSIS

```
SUBROUT\mathbb{NE SPOCON(UPLO,N,A,LDA,ANORM,RCOND,W ORK,W ORK2,\mathbb{NFO)}}\mathbf{N}\mathrm{ (N,N}
```

CHARACTER * 1 UPLO
$\mathbb{N}$ TEGER $N, L D A, \mathbb{N} F O$
$\mathbb{N}$ TEGERWORK2(*)
REAL ANORM,RCOND
REALA (LDA,*),WORK (*)
SU BROUTINE SPOCON_64 (UPLO,N,A,LDA,ANORM,RCOND,WORK,W ORK2,
$\mathbb{N} F O$ )
CHARACTER * 1 UPLO
$\mathbb{N}$ TEGER*8 $N, L D A, \mathbb{N} F O$
$\mathbb{N}$ TEGER*8WORK2 (*)
REALANORM,RCOND
REALA (LDA,*), WORK (*)

## F95 INTERFACE

SUBROUTINE POCON (UPLO, $\mathbb{N}], A,[L D A], A N O R M, R C O N D,[W O R K],[W$ ORK2], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER ::N,LDA, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) ::W ORK2
REAL ::ANORM,RCOND

SU BROUTINEPOCON_64 (UPLO, $\mathbb{N}], A,[L D A], A N O R M, R C O N D,[W O R K],[W O R K 2]$, [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1) :: UPLO
$\mathbb{N}$ TEGER (8) :: N , LDA, $\mathbb{N}$ FO
$\mathbb{N}$ TEGER (8), D $\mathbb{M}$ ENSION (:) ::WORK2
REAL ::ANORM,RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::W ORK
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A

## C INTERFACE

\#include < sunperfh>
void spocon (charuple, int n, float *a, int lda, float anorm, float *rcond, int*info);
void spocon_64 (charuplo, long n, float *a, long lda, float anorm , float *rcond, long *info);

## PURPOSE

spocon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a realsym m etric positive definite $m$ atrix using the C holesky factorization $A=U * * T * U$ or $A=L * L * * T$ com puted by SPO TRF .

A $n$ estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND = 1/(ANORM * norm (inv (A))).

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': Upper triangle of A is stored;
$=\mathrm{L}$ ': Low er triangle of $A$ is stored.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input) The triangular factor $U$ or $L$ from the Cholesky factorization $A=U * * T * U$ orA $=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$, as com puted by SPO TRF.

LDA (input)
The leading dim ension of the aray A. LDA >= $\max (1, N)$.

ANORM (input)
The 1-norm (or infinity-norm ) of the symmetric $m$ atrix A.

## RCOND (output)

The reciprocal of the condition number of the $m$ atrix $A$, com puted as RCOND $=1$ ( $A N O R M * A \mathbb{N} V N M$ ), where $A \mathbb{N} V N M$ is an estim ate of the 1 -nom of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension ( $3 * \mathrm{~N}$ )
W ORK 2 (w orkspace)
dim ension (N)
$\mathbb{I N F O}$ (output)
= 0 : successfiulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

spoequ - com pute row and colum n scalings intended to equilibrate a sym $m$ etric positive definite $m$ atrix $A$ and reduce its condition num ber (w ith respect to the tw o-norm )

## SYNOPSIS

```
SUBROUT\mathbb{NE SPOEQU N,A,LDA,SCALE,SCOND,AMAX,INFO)}
```

$\mathbb{N} T E G E R N, L D A, \mathbb{N} F O$
REAL SCOND,AMAX
REAL A (LDA,*), SCALE (*)
SUBROUTINE SPOEQU_64 $\mathbb{N}, A, L D A, S C A L E, S C O N D, A M A X, \mathbb{N} F O)$
$\mathbb{N}$ TEGER*8N,LDA, $\mathbb{N} F O$
REAL SCOND,AMAX
REALA (LDA,*), SCALE (*)

## F95 INTERFACE

SUBROUTINE POEQU ( $\mathbb{N}$ ],A, [LDA ],SCALE, SCOND ,AMAX, [ $\mathbb{N F O}$ ])
$\mathbb{N} T E G E R:: N, L D A, \mathbb{N} F O$
REAL ::SCOND,AMAX
REAL,D IM ENSION (:) ::SCALE
REAL,D $\mathbb{M}$ ENSION (:,:) ::A

SU BROUTINE POEQU_64 ( $\mathbb{N}$ ],A, [LDA ],SCALE,SCOND,AMAX, [ $\mathbb{N} F O$ ])
$\mathbb{N}$ TEGER (8) :: N, LDA, $\mathbb{N} F O$
REAL ::SCOND,AMAX
REAL,D IM ENSION (:) ::SCALE
REAL,D IM ENSION (:,:) ::A

## C INTERFACE

\#include <sunperfh>
void spoequ (intn, float *a, int lda, float *scale, float
*scond, float *am ax, int *info);
void spoequ_64 (long n, float *a, long lda, float *scale, float *scond, float *am ax, long *info);

## PURPOSE

spoequ com putes row and colum n scalings intended to equilibrate a sym $m$ etric positive definite $m$ atrix $A$ and reduce its condition num ber (w ith respect to the tw o-nom ). S contains the scale factors, $S(i)=1 /$ squt $(A)(i, i))$, chosen so that the scaled $m$ atrix B w ith elem ents $B(i, j)=S(i) \star A(i, j) * S(i)$ has ones on the diagonal. This choice of $S$ puts the condition num berofB w ithin a factor $N$ of the sm allestpossible condition num ber overallpossible diagonal scalings.

## ARGUMENTS

N (input) The order of the matrix A. $\mathrm{N}>=0$.
A (input) The $\mathrm{N}-$ by -N sym $m$ etric positive definite $m$ atrix whose scaling factors are to be com puted. O nly the diagonalelem ents ofA are referenced.

LD A (input)
The leading dim ension of the anay A. LDA >= $\max (1, N)$.

SCA LE (output)
If $\mathbb{N} F O=0$, SCA LE contains the scale factors for A.

SCOND (output)
If $\mathbb{N} F O=0$, SCA LE contains the ratio of the sm allest SCALE (i) to the largestSCA LE (i). IfSCOND $>=0.1$ and AMAX is neither too large nor too sm all, it is notw orth scaling by SCA LE .

AM AX (output)
A bsolute value of largestm atrix elem ent. If A M A X is very close to overflow orvery close to underflow , the m atrix should be scaled.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N F O}=$ i, the ith diagonal elem ent is nonpositive.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sporfs - im prove the com puted solution to a system of linear equations when the coefficientm atrix is sym $m$ etric positive definite,

## SYNOPSIS

```
SUBROUT\mathbb{NE SPORFS (UPLO,N,NRHS,A,LDA,AF,LDAF,B,LDB,X,LDX,}
    FERR,BERR,W ORK,W ORK2,INFO)
CHARACTER * 1 UPLO
\mathbb{NTEGER N,NRHS,LDA,LDAF,LDB,LDX,}\mathbb{N}FO
INTEGER W ORK2 (*)
REAL A (LDA ,*), AF (LDAF,*), B (LDB,*), X (LDX ,*), FERR (*),
BERR (*),W ORK (*)
SUBROUT\mathbb{NE SPORFS_64 (UPLO,N,NRHS,A,LDA,AF,LDAF,B,LDB,X,LDX,}
    FERR,BERR,W ORK,W ORK2, INFO)
CHARACTER * 1 UPLO
INTEGER*8 N,NRHS,LDA,LDAF,LDB,LDX,NNFO
\mathbb{NTEGER*8 W ORK2 (*)}
REAL A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX ,*), FERR (*),
BERR (*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE PORFS (UPLO, $\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F], B,[L D B]$, X, [LDX],FERR,BERR, [W ORK], [W ORK 2], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: W$ ORK2
REAL,D $\mathbb{I}$ ENSION (:) ::FERR,BERR,W ORK

SU BROUTINE PORFS_64 (UPLO, $\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F], B,[L D B]$, $\mathrm{X},[\mathrm{LD} \mathrm{X}], \mathrm{FERR}, \mathrm{BERR},[\mathrm{W} O R K],[W O R K 2],[\mathbb{N} F O])$

CHARACTER (LEN=1) ::UPLO
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} A, L D A F, L D B, L D X, \mathbb{N} F O$
$\mathbb{N}$ TEGER (8), D $\mathbb{M}$ ENSION (:) ::WORK2
REAL,D $\mathbb{M}$ ENSION (:) ::FERR,BERR,W ORK
REAL,D $\mathbb{I}$ ENSION (:,:) ::A,AF,B,X

## C INTERFACE

\#include <sunperfh>
void sporfs (char uplo, intn, intnrhs, float *a, int lda, float *af, int ldaf, float *b, int ldlb, float *x, int ldx, float * ferr, float *berr, int*info);
void sporfs_64 (charuplo, long n, long nrhs, float*a, long lda, float *af, long ldaf, float*b, long ldb, float *x, long ldx, float * ferr, float *berr, long *info);

## PURPOSE

sporfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is sym $m$ etric positive definite, and provides errorbounds and backw ard emoresti$m$ ates for the solution.

## ARGUMENTS

## UPLO (input)

$=\mathrm{U}$ ': U pper triangle ofA is stored;
$=\mathbb{L}$ ': Low ertriangle of $A$ is stored.

N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atriges B and X. NRH S $>=0$.

A (input) The symm etricm atrix A. If $\mathrm{APLO}=\mathrm{U}$ ', the leading N -by -N uppertriangularpart of $A$ contains the upper triangularpart of the $m$ atrix $A$, and the strictly low ertriangularpartofA is notreferenced. IfUPLO = L', the leading N boy-N lower triangularpart ofA contains the low er triangular
part of the m atrix A, and the strictly upper triangularpart of A is not referenced.

LD A (input)
The leading dim ension of the array A. LD A >= $\max (1, N)$.

AF (input)
The triangular factor $U$ or $L$ from the Cholesky factorization $A=U * * T * U$ or $A=L * L * * T$, as com puted by SPO TRF .
LDAF (input)
The leading dim ension of the array AF. LDAF >= $\max (1, N)$.
$B$ (input) The righthand side m atrix B .

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.

X (input/output)
O n entry, the solution $m$ atrix $X$, as com puted by SPO TRS. On exit, the im proved solution $m$ atrix X .

LD X (input)
The leading dim ension of the aray X. LDX >= $\max (1, N)$.

## FERR (output)

The estim ated forw ard emorbound for each solution vectorX ( 7 ) (the $j$ th colum $n$ of the solution $m$ atrix $X$ ). If XTRUE is the true solution corresponding to $X(\lambda), F E R R(i)$ is an estim ated upperbound for the m agnitude of the largest ele$m$ entin ( $\mathrm{X}(\mathcal{j})-\mathrm{X}$ TRUE) divided by the $m$ agnitude of the largestelem entin X ( $\mathrm{j}^{\prime}$ ) . The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slightoverestim ate of the true error.

## BERR (output)

The com ponentw ise relative backw ard error of each
solution vectorX (i) (ie., the sm allest relative change in any elem entofA orB thatm akes X (i) an exactsolution).

W ORK (w orkspace)
dim ension ( $3 * N$ )

W ORK 2 (w orkspace)
dim ension $\mathbb{N}$ )
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N F O}=-i$, the $i$-th argum ent had an illegal value

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sposv -com pute the solution to a real system of linear equations $A * X=B$,

## SYNOPSIS

```
SUBROUTINE SPOSV (UPLO,N,NRHS,A,LDA,B,LDB,INFO)
CHARACTER * 1 UPLO
INTEGERN,NRHS,LDA,LDB,INFO
REALA (LDA,*),B (LDB,*)
SUBROUTINE SPOSV_64 (UPLO,N,NRHS,A,LDA,B,LDB,INFO)
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N,NRHS,LDA,LDB,INFO}
REALA (LDA,*),B (LDB,*)
```

F95 INTERFACE
SU BROUTINEPOSV (UPLO, $\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N F O}])$
CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:,:) ::A, B
SUBROUTINEPOSV_64 (UPLO, $\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N F O}])$
CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D B, \mathbb{N} F O$
REAL,D IM ENSION (: :: : ::A, B
C INTERFACE
\#include <sunperfh>
void sposv (char uplo, intn, intnrhs, float *a, int lda, float*b, int lalb, int *info);
void sposv_64 (charuplo, long n, long nrhs, float *a, long lda, float *b, long ldlo, long *info);

## PURPOSE

sposv com putes the solution to a realsystem of linearequations
$A * X=B$, where $A$ is an $N$ boy $-N$ sym m etric positive defintie $m$ atrix and $X$ and $B$ are $N$ boy $N$ R H S $m$ atrices.

The Cholesky decom position is used to factorA as
$A=U * * T * U$, if $U P L O=U$ ', or
$A=L * L \star * T$, if $\mathrm{UPLO}=\mathrm{L}$ ',
$w$ here $U$ is an uppertriangularm atrix and $L$ is a low er triangular $m$ atrix. The factored form of $A$ is then used to solve the system ofequations $A * X=B$.

## ARGUMENTS

UPLO (input)
$=U$ ': U pper triangle of $A$ is stored;
$=\mathbb{L}$ ': Low ertriangle of $A$ is stored.

N (input) The num ber of linearequations, i.e., the order of the matrix A. $N>=0$.

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of the m atrix B. NRHS $>=0$.

A (input/output)
O n entry, the sym m etric m atrix A. If UPLO = U', the leading N -oy N uppertriangularpart of $A$ contains the upper triangular part of the $m$ atrix $A$, and the strictly low er triangularpart of A is not referenced. If UPLO $=\mathrm{L}$ ', the leading N -by -N low er triangularpart of A contains the low er triangularpart of the m atrix A, and the strictly upper triangularpart of A is notreferenced.

On exit, if $\mathbb{N} F O=0$, the factor $U$ orL from the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$.

LD A (input)
The leading dim ension of the anay A. LD A >= $\max (1, N)$.

B (input/output)
On entry, the N -by-NRHS righthand side matrix B.
On exi, if $\mathbb{N} F O=0$, the $N$ by $N$ RH S solution
matrix $X$.

LD B (input)
The leading dim ension of the aray B. LD B >= $\max (1, N)$.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{I N}$ FO = -i, the i-th argum enthad an illegalvalue
$>0$ : if $\mathbb{N F O}=\mathrm{i}$, the leading m inoroforderiof
A is notpositive definite, so the factorization could not.be com pleted, and the solution has not been com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sposvx - use the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=$ $\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ to com pute the solution to a realsystem of linear equations $A * X=B$,

## SYNOPSIS

```
SUBROUT\mathbb{NE SPOSVX (FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,EQUED,}
    S,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1FACT,UPLO,EQUED
\mathbb{NTEGER N,NRHS,LDA,LDAF,LDB,LDX,}\mathbb{N}FO
\mathbb{NTEGERWORK2(*)}
REALRCOND
REAL A (LDA,*), AF (LDAF,*), S (*), B (LDB,*), X (LDX,*),
FERR (*),BERR (*),W ORK (*)
SU BROUT\mathbb{NE SPOSVX_64(FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,EQUED,}
    S,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK2,INFO)
CHARACTER * 1 FACT,UPLO,EQUED
INTEGER*8N,NRHS,LDA,LDAF,LDB,LDX,}\mathbb{N}F
INTEGER*8W ORK2 (*)
REALRCOND
REAL A (LDA,*), AF (LDAF,*), S (*), B (LDB,*), X (LDX,*),
FERR (*),BERR (*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE POSVX $\mathbb{E A C T}, \mathrm{UPLO}, \mathbb{N}], \mathbb{N R H S}], A,[L D A], A F,[L D A F]$, EQUED, $S, B$, [LDB], $X,[L D X], R C O N D, F E R R, B E R R,[W O R K]$, [W ORK 2], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::FACT,UPLO,EQUED
$\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: W$ ORK2
REAL ::RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::S,FERR,BERR,W ORK
REAL,D $\mathbb{I}$ ENSION (:,:) ::A,AF,B,X

SU BROUTINE POSVX_64 (FACT, UPLO, $\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F]$, EQUED, $S, B,[L D B], X,[L D X], R C O N D, F E R R, B E R R,[W O R K]$, [W ORK2], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::FACT,UPLO, EQUED
$\mathbb{N} \operatorname{TEGER}$ (8) :: N , NRHS,LDA,LDAF,LDB,LDX, $\mathbb{N}$ FO
$\mathbb{N}$ TEGER (8), D $\mathbb{I M} \operatorname{ENSION}(:):$ W ORK2
REAL ::RCOND
REAL,D $\mathbb{I}$ ENSION (:) :: S,FERR,BERR,W ORK
REAL,D $\mathbb{I}$ ENSION (:,:) ::A,AF,B,X

## C INTERFACE

\#include <sunperfh>
void sposvx (char fact, charuplo, intn, intnrhs, float *a, int lda, float *af, intllaf, charequed, float *s, float*b, int ldb, float*x, int ldx, float *rcond, float * ferr, float *berr, int*info);
void sposvx_64 (char fact, char uplo, long n, long nrhs, float *a, long lda, float*af, long ldaf, char equed, float*s, float*b, long ldb, float *x, long ldx, float *roond, float *ferr, float *berr, long *info);

## PURPOSE

sposvx uses the C holesky factorization $A=U * * T * U$ or $A=$ $\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ to com pute the solution to a realsystem of linear equations
$A * X=B, w h e r e A$ is an $N$ boy $-N$ sym m etric positive defintie $m$ atrix and $X$ and $B$ are N boy-N R H S m atrices.

E morbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:

1. IfFACT = E', real scaling factors are com puted to equilibrate
the system :
diag $(\mathrm{S})$ * A * diag $(\mathrm{S})$ * inv $(\operatorname{diag}(\mathrm{S}))$ * X $=\operatorname{diag}(\mathrm{S})$ * B
W hether or not the system w illbe equilibrated depends on
the
scaling of them atrix A , but if equilibration is used, A is
overw rilten by diag $(S) \star A * \operatorname{diag}(S)$ and $B$ by diag $(S) * B$.
2. IfFACT $=\mathrm{N}$ 'or E', the Cholesky decom position is used to
factor the $m$ atrix A (afterequilibration ifFACT $=\mathrm{E}$ )
as
$A=U * * T * U$, if $U P L O=U$ ', or
$A=L * L * * T$, if $U P L O=L^{\prime}$,
where $U$ is an uppertriangularm atrix and $L$ is a low er triangular
$m$ atrix.
3. If the leading i-by-iprincipal m inor is not positive definite, then the routine retums w ith $\mathbb{N} F O=i$. O therw ise, the factored
form ofA is used to estim ate the condition num ber of the $m$ atrix
A. If the reciprocal of the condition num ber is less than $m$ achine
precision, $\mathbb{N F F O}=\mathrm{N}+1$ is retumed as a w aming, but the routine
still goes on to solve for X and com pute errorbounds as described below .
4.The system of equations is solved for $X$ using the factored form
of A.
4. Herative refinem ent is applied to im prove the com puted solution
$m$ atrix and calculate error bounds and backw ard error estim ates
for it.
5. Ifequilibration w as used, the $m$ atrix $X$ is prem ultiplied by diag (S) so that it solves the original system before equilibration.

## ARGUMENTS

FACT (input)
Specifies w hether ornot the factored form of the $m$ atrix A is supplied on entry, and if not, whether the m atrix A should be equilibrated before it is
factored. = F ': On entry, AF contains the factored form of A. IfEQUED $=Y$ ', the matrix A has been equilibrated $w$ ith scaling factors given by $S$. A and $\mathrm{A} F \mathrm{w}$ illnotbe m odified. $=\mathrm{N}$ ': The m atrix
A w illlbe copied to A F and factored.
= E': The matrix A will be equilibrated if necessary, then copied to AF and factored.

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The num ber of linearequations, ie., the order of them atrix $A . N>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the $m$ atrices $B$ and $X$. NRHS $>=0$.
A (input/output)
O n entry, the sym m etric $m$ atrix A, except ifFA C T = $\mathrm{F}^{\prime}$ and EQUED = Y ', then A mustcontain the equilibrated $m$ atrix diag $(S) \star A$ *diag $(S)$. If UPLO =
U', the leading N -by -N uppertriangular partofA contains the upper triangularpartof the $m$ atrix A, and the strictly low er triangular partofA is not referenced. IfU PLO = L' ', the leading N -by -N low er triangularpart of A contains the low er triangularpart of the matrix A, and the strictly upper triangular partofA is not referenced. A is notm odified ifFACT = F'or N', or ifFACT = E'and EQUED = N 'on exit.

Onexit, ifFACT = E' and EQUED = $\mathrm{Y}^{\prime}, \mathrm{A}$ is overw rilten by diag $(S) * A * d i a g(S)$.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, \mathbb{N})$.

AF (input/output)
If $F A C T=F$ ', then $A F$ is an inputargum ent and on entry contains the triangular factorU or $L$ from the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=$ $\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$, in the sam e storage form atas A. IfEQUED ne. $N$ ', then $A F$ is the factored form of the equilibrated $m$ atrix diag $(S) * A$ *diag $(S)$.

If FA C T = N ', then AF is an output argum ent and on exit retums the triangular factor $U$ orL from the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=$
$\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ of the originalm atrix A .

If FAC $\mathrm{T}=\mathrm{E}$ ', then AF is an output argum ent and on exit retums the triangular factorU orL from the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=$ $\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ of the equilibrated m atrix A (see the description of $A$ for the form of the equilibrated $m$ atrix).

LDAF (input)
The leading dim ension of the array AF. LDAF >= $\max (1, N)$.

## EQUED (input)

Specifies the form of equilibration thatw as done. = N ': N o equilibration (alw ays true iffA C T = N ${ }^{2}$.
$=Y^{\prime}:$ Equilibration $w$ as done, i.e., A has been
replaced by diag (S) * A * diag (S). EQUED is an inputargum ent if $F A C T=F$ '; otherw ise, it is an output argum ent.
$S$ (input/output)
The scale factors forA ; not accessed if EQUED = $\mathrm{N}^{\prime} . \mathrm{S}$ is an inputargum entifFACT=F'; otherw ise, S is an outputargum ent. IfFACT = $\mathrm{F}^{\prime}$ and EQUED = Y', each elem entofs m ustibe positive.

B (input/output)
On entry, the $\mathrm{N}-\mathrm{by}-\mathrm{NRH} \mathrm{S}$ righthand side m atrix B . On extr, if EQUED $=N$ ', $B$ is notm odified; if EQUED = $Y$ ', $B$ is overw ritten by diag ( S ) * B .

LD B (input)
The leading din ension of the aray $B$. LD B >= $\max (1, N)$.

X (output)
If $\mathbb{N} F O=0$ or $\mathbb{N} F O=N+1$, the $N$ by N RH $S$ solution
$m$ atrix $X$ to the original system of equations.
$N$ ote that ifEQUED $=Y$ ', $A$ and $B$ are $m$ odified on exit, and the solution to the equilibrated system is inv (diag (S) ) *X .

LD X (input)
The leading dim ension of the array X . LD X >= $\max (1, N)$.

RCOND (output)
The estim ate of the reciprocal condition num ber of
the $m$ atrix $A$ afterequilibration (if done). If RCOND is less than the $m$ achine precision (in particular, ifRCOND $=0$ ), the $m$ atrix is singular to w orking precision. This condition is indicated by a retum code of $\mathbb{N}$ FO >0.

FERR (output)
The estim ated forw ard errorbound for each solution vectorX ( $\mathcal{H}$ ) (the $j$ th column of the solution $m$ atrix X). If XTRUE is the true solution comesponding to $X(\mathcal{O})$, FERR ( $)$ is an estim ated upperbound for the $m$ agnitude of the largest ele$m$ ent in ( $\mathrm{X}(\mathcal{j})-\mathrm{X}$ TRUE) divided by the m agnitude of the largestelem ent in $X(\mathcal{H})$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slightoverestim ate of the true error.

## BERR (output)

The com ponentw ise relative backw ard error of each
solution vectorX (i) (ie., the sm allest relative change in any elem entofA orB thatm akes X ( $j$ ) an exactsolution).

W ORK (w orkspace)
dim ension ( $3 \star$ N )
W ORK 2 (w orkspace)
dim ension $\mathbb{N}$ )
$\mathbb{I N F O}$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{I N F O}=-$ i, the $i$-th argum enthad an illegalvalue
> 0 : if $\mathbb{N F O}=\mathrm{i}$, and i is
<= N : the leading $m$ inor oforderiof $A$ is not positive definite, so the factorization could not be com pleted, and the solution has not been com puted. RCOND $=0$ is retumed. $=\mathrm{N}+1: \mathrm{U}$ is nonsingular, butRCOND is less than machine precision, $m$ eaning that the $m$ atrix is singular to $w$ orking precision. N evertheless, the solution and error bounds are com puted because there are a num ber of situations w here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

spotf2 -com pute the C holesky factorization of a real sym $m$ etric positive definite $m$ atrix $A$

## SYNOPSIS

```
SUBROUT\mathbb{NE SPOTF2(UPLO,N,A,LDA, INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER N,LDA,}\mathbb{NFO}
REALA (LDA,*)
```



```
CHARACTER * 1 UPLO
INTEGER*8N,LDA, INFO
REALA (LDA,*)
F95 INTERFACE
    SUBROUT\mathbb{NE POTF2 (UPLO, NN,A,[LDA ], [NFO])}
    CHARACTER (LEN=1) ::UPLO
    \mathbb{NTEGER::N,LDA,}\mathbb{NFO}
    REAL,DIM ENSION (:,:) ::A
    SUBROUT\mathbb{NE POTF2_64 (UPLO, N ],A, [LDA ], [NFO ])}
    CHARACTER (LEN=1) ::UPLO
    \mathbb{NTEGER (8) ::N,LDA,}\mathbb{NFO}
    REAL,DIM ENSION (:,:) ::A
C INTERFACE
    #include <sunperfh>
```

void spotf2 (charuple, intn, float *a, int lda, int *info);
void spotf2_64 (charuplo, long n, float*a, long lda, long *info);

## PURPOSE

spotff com putes the C holesky factorization of a real sym $m$ etric positive definite $m$ atrix A .

The factorization has the form
$A=U^{\prime} \star U$, if $U P L O=U '$, or
$A=L * L \prime$, if $\mathrm{L} P \mathrm{LO}=\mathrm{L}$ ',
$w$ here $U$ is an upper triangular $m$ atrix and $L$ is lower triangular.

This is the unblocked version of the algorithm, calling Level2 BLAS.

## ARGUMENTS

UPLO (input)
Specifies w hether the upper or low er triangular
part of the symm etric m atrix A is stored. $=U$ ':
U pper triangular
= L': Low er triangular

N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
O $n$ entry, the sym $m$ etric $m$ atrix $A$. If $U P L O=U '$, the leading $n$ by $n$ uppertriangularpartofA contains the upper triangular part of the $m$ atrix $A$, and the strictly low er triangularpart of A is not referenced. If $\mathrm{UPLO}=\mathrm{L}$ ', the leading n by n low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of $A$ is notreferenced.

On exit, if $\mathbb{N F O}=0$, the factor $U$ orL from the Cholesky factorization $A=U$ * U orA $=\mathrm{L} \star \mathrm{L}$ '.

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, N)$.

## = 0: successfulexit

$<0:$ if $\mathbb{N}$ FO $=-\mathrm{k}$, the k -th argum enthad an illegalvalue
$>0:$ if $\mathbb{N} F O=k$, the leading $m$ inor of order $k$ is notpositive definite, and the factorization could not.be com pleted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

spotrf-com pute the C holesky factorization of a real sym $m$ etric positive definite $m$ atrix $A$

## SYNOPSIS

```
SUBROUTINE SPOTRF(UPLO,N,A,LDA, INFO)
```

CHARACTER * 1 UPLO
$\mathbb{N} T E G E R N, L D A, \mathbb{N} F O$
REALA (LDA,*)
SUBROUTINE SPOTRF_64 (UPLO,N,A,LDA, $\mathbb{N} F O$ )
CHARACTER * 1 UPLO
$\mathbb{N} T E G E R * 8 N, L D A, \mathbb{N} F O$
REALA (LDA,*)
F95 INTERFACE
SU BROUTINE POTRF (UPLO, $\mathbb{N}], A,[L D A],[\mathbb{N F O}])$
CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R:: N, L D A, \mathbb{N F O}$
REAL,D $\mathbb{M}$ ENSION (:,:) ::A
SU BROUTINE POTRF_64 (UPLO, $\mathbb{N}], A,[L D A],[\mathbb{N} F O])$
CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER (8) :: $\mathrm{N}, \mathrm{LD} A, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:,:) ::A
C INTERFACE
\#include <sunperfh>
void spotrf(charuplo, intn, float *a, int lda, int *info);
void spotrf_64 (charuplo, long n, float*a, long lda, long *info);

## PURPOSE

spotrf com putes the C holesky factorization of a real sym $m$ etric positive definite m atrix A .

The factorization has the form
$A=U * * T * U$, if $U P L O=U '$, or
$A=L * L * * T$, if $\mathrm{UPLO}=\mathrm{L} \prime$ ',
$w$ here $U$ is an upper triangular $m$ atrix and $L$ is lower triangular.

This is the block version of the algorithm , calling Level 3 BLAS.

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': U pper triangle of A is stored;
= $\mathbb{L}$ ': Low ertriangle of is stored.

N (input) The order of them atrix A. N $>=0$.

A (input/output)
O n entry, the sym m etric matrix A. If UPLO = U ', the leading N -by -N uppertriangularpartofA contains the upper triangularpart of the $m$ atrix $A$, and the strictly low er triangularpart of $A$ is not referenced. If UPLO = 'L', the leading N -by N low er triangularpart of $A$ contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of A is notreferenced.

On exit, if $\mathbb{N F O}=0$, the factor $U$ orL from the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$.

LDA (input)
The leading dim ension of the array A. LDA >= max (1,N).

INFO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N F O}=-$ i, the $i$-th argum enthad an illegalvahue
> 0 : if $\mathbb{N} F O=i$, the leading $m$ inoroforder is notpositive definite, and the factorization could notbe com pleted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

spotri-com pute the inverse of a real sym $m$ etric positive definite $m$ atrix $A$ using the Cholesky factorization $A=$ $\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ orA $=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ com puted by SPO TRF

## SYNOPSIS

```
SUBROUT\mathbb{NE SPOTRI(UPLO,N,A,LDA, INFO)}
CHARACTER * 1 UPLO
NNTEGER N,LDA, INFO
REALA (LDA,*)
SUBROUT\mathbb{NE SPOTRI_64(UPLO,N,A,LDA, INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N,LDA,INFO}
REALA (LDA,*)
F95 INTERFACE
    SU BROUT\mathbb{NE POTRI(UPLO, N ],A , [LDA ], [NNO ])}
    CHARACTER (LEN=1) ::UPLO
    \mathbb{NTEGER ::N,LDA, INFO}
    REAL,D IM ENSION (:,:) ::A
    SU BROUT\mathbb{NE POTRI_64 (UPLO, N ],A, [LDA ], [NNO ])}
    CHARACTER (LEN=1) ::UPLO
    \mathbb{NTEGER (8) ::N,LDA,}\mathbb{NFO}
    REAL,D IM ENSION (:,:) ::A
\#include <sunperfh>
void spotri(char uplo, intn, float *a, int lda, int *info);
void spotri_64 (charuple, long n, float *a, long lda, long *info);

\section*{PURPOSE}
spotricom putes the inverse of a real symm etric positive definite \(m\) atrix \(A\) using the Cholesky factorization \(A=\) \(\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\) com puted by SPO TRF .

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}:\) : Upper triangle of A is stored;
\(=\mathbb{L}\) ': Low er triangle of A is stored.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the triangular factor \(U\) or \(L\) from the Cholesky factorization \(A=U * * T * U\) or \(A=L * L * * T\), as com puted by SPO TRF. On exit, the upper or low er triangle of the (sym metric) inverse of A, overw riting the input factorU orL.

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, N)\).

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-\) i, the i-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the \((i, i)\) elem entof the factor
U orL is zero, and the inverse could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
spotrs-solve a system of linearequations \(A * X=B\) w th a sym \(m\) etric positive definite \(m\) atrix \(A\) using the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\) com puted by SPO TRF

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SPOTRS (UPLO,N,NRHS,A,LDA,B,LD B, INFO)}
CHARACTER * 1 UPLO
INTEGERN,NRHS,LDA,LDB,INFO
REALA (LDA,*),B (LDB,*)
SUBROUT\mathbb{NE SPOTRS_64(UPLO,N,NRHS,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO
INTEGER*8N,NRHS,LDA,LDB,\mathbb{NFO}
REALA (LDA,*),B (LDB,*)

```
F95 INTERFACE
    SU BROUTINE POTRS (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N F O}])\)
    CHARACTER (LEN=1)::UPLO
    \(\mathbb{N}\) TEGER ::N,NRHS,LDA,LDB, \(\mathbb{N}\) FO
    REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A,B
    SU BROUTINE POTRS_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N F O}])\)
    CHARACTER (LEN=1) ::UPLO
    \(\mathbb{N}\) TEGER (8) ::N,NRHS,LDA,LDB, \(\mathbb{N}\) FO
    REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A,B
void spotrs (charuple, intn, intnins, float *a, int lda, float *b, int ldb, int *info);
void spotrs_64 (charuplo, long n, long nins, float*a, long lda, float *b, long lolb, long *info);

\section*{PURPOSE}
spoters solves a system of linearequations \(A * X=B\) with a sym \(m\) etric positive definite \(m\) atrix \(A\) using the Cholesky factorization \(A=U * * T * U\) or \(A=L * L * * T\) com puted by SPOTRF.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': Uppertriangle of \(A\) is stored;
\(=\mathbb{L}\) ': Low ertriangle of \(A\) is stored.

N (input) The order of them atrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the m atrix B. NRH S \(>=0\).

A (input) The triangular factorU or \(L\) from the Cholesky
factorization \(A=U * * T * U\) orA \(=\mathrm{L} * \mathrm{~L} * * T\), as com puted by SPO TRF.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

B (input/output)
On entry, the right hand side m atrix B. On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sppcon -estim ate the reciprocal of the condition num ber (in the 1 -norm ) of a realsym \(m\) etric positive definite packed \(m\) atrix using the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(\mathrm{A}=\) L *L**T com puted by SPPTRF

\section*{SYNOPSIS}

CHARACTER * 1 UPLO
\(\mathbb{N}\) TEGER \(N, \mathbb{I N F O}\)
\(\mathbb{N}\) TEGER W ORK 2 (*)
REAL ANORM,RCOND
REALA (*),W ORK (*)
SUBROUTINE SPPCON_64 (UPLO,N,A,ANORM,RCOND,WORK,WORK2, \(\mathbb{N} F O\) )
CHARACTER * 1 UPLO
\(\mathbb{N}\) TEGER*8 \(\mathrm{N}, \mathbb{I N}\) FO
\(\mathbb{N}\) TEGER * 8 W ORK 2 (*)
REAL ANORM,RCOND
REALA (*), W ORK (*)

\section*{F95 INTERFACE}

SUBROUTINE PPCON (UPLO,N,A,ANORM,RCOND, [WORK], [W ORK2], [ \(\mathbb{N} F O]\) )
CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) ::W ORK 2
REAL ::ANORM,RCOND
REAL,D \(\mathbb{I M}\) ENSION (:) ::A,W ORK

CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER (8) ::N, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8),D \(\mathbb{I M}\) ENSION (:) ::W ORK 2
REAL ::ANORM,RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::A,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void sppcon (charuplo, intn, float *a, float anorm , float *rcond, int*info);
void sppcon_64 (charuplo, long n, float *a, float anorm , float *roond, long *info);

\section*{PURPOSE}
sppcon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a realsym \(m\) etric positive definite packed \(m\) atrix using the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) or \(\mathrm{A}=\) L*L**T com puted by SPPTRF.

A n estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND = 1 / (ANORM * norm (inv (A))).

\section*{ARGUMENTS}

UPLO (input)
= U : U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrix A. N >=0.

A (input) The triangular factorU or L from the Cholesky
factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\), packed colum nw ise in a linearanay. The jth colum \(n\) of U or L is stored in the array A as follow s: if UPLO = U',A \((i+(j-1) * j 2)=U(i, 7)\) for \(1<=i<=j\) if UPLO = L', A (i+ ( \(j-1)^{*}(2 n-j / 2)=L(i, 7)\) for j=i<=n.

ANORM (input)
The 1-norm (or infinity-nom ) of the symmetric \(m\) atrix A.

\section*{RCOND (output)}

The reciprocal of the condition num ber of the
\(m\) atrix \(A\), com puted as RCOND = 1/(ANORM *A \(\mathbb{N} V N M)\),
where \(A \mathbb{N} V N M\) is an estim ate of the 1 -nom of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension ( \(3 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an ille-
galvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
sppequ - com pute row and colum n scalings intended to equilibrate a symm etric positive definite \(m\) atrix \(A\) in packed storage and reduce its condition num ber (w ith respect to the tw o-norm )

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SPPEQU (UPLO,N,A,SCALE,SCOND,AMAX,INFO)}

```
CHARACTER * 1 UPLO
\(\mathbb{N}\) TEGER \(N, \mathbb{N} F O\)
REAL SCOND,AMAX
REALA (*), SCALE (*)
SU BROUTINE SPPEQU_64(UPLO,N,A,SCALE,SCOND,AMAX, \(\mathbb{N} F O\) )
CHARACTER * 1 UPLO
\(\mathbb{N} T E G E R * 8 N, \mathbb{N} F O\)
REAL SCOND,AMAX
REALA (*), SCALE (*)

\section*{F95 INTERFACE}

SU BROUTINE PPEQU (UPLO, \(\mathbb{N}\) ],A, SCALE, SCOND,AMAX, [ \(\mathbb{N} F O\) ])
CHARACTER (LEN=1)::UPLO
\(\mathbb{N}\) TEGER :: N, \(\mathbb{N} F O\)
REAL ::SCOND,AMAX
REAL,D \(\mathbb{M}\) ENSION (:) ::A, SCALE

SU BROUTINE PPEQU_64 (UPLO, \(\mathbb{N}], A, S C A L E, S C O N D, A M A X,[\mathbb{N} F O\) ])
CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R(8):: N, \mathbb{N} F O\)
REAL ::SCOND,AMAX
REAL,D \(\mathbb{I M} E N S I O N(:):: A, S C A L E\)

\section*{C INTERFACE}
\#include <sunperfh>
void sppequ (charuplo, intn, float *a, float *scale, float
*scond, float *am ax, int *info);
void sppequ_64 (charuplo, long n, float *a, float *scale, float *scond, float *am ax, long *info);

\section*{PURPOSE}
sppequ com putes row and colum n scalings intended to equilibrate a symm etric positive definite \(m\) atrix \(A\) in packed storage and reduce its condition num ber (w ith respect to the tw o-norm). \(S\) contains the scale factors, \(S(i)=1 /\) sqit \((A\) ( \(i, i)\) ), chosen so that the scaled \(m\) atrix \(B\) w ith elem ents \(B(i, j)=S(i) * A(i, j) * S(i)\) has ones on the diagonal. This choige ofS puts the condition num ber of B w ithin a factor N of the sm allestpossible condition num berover all possible diagonal scalings.

\section*{ARGUMENTS}
```

UPLO (input)

```
\(=\mathrm{U}\) : : U ppertriangle of A is stored;
= L': Low er triangle ofA is stored.
N (input) The order of the m atrix A. \(\mathrm{N}>=0\).

A (input) The upper or low er triangle of the sym m etric \(m\) atrix A, packed colum nw ise in a linear array. The \(j\) th column of A is stored in the array A as follows: if UPLO = U',A (i+ \((j-1) * j 2)=A(i, 7)\) for \(1<=i<=\dot{j}\) ifUPLO \(=\mathrm{L}\) ', A ( \(\left.i+(j-1)^{*}(2 n-j) / 2\right)\) \(=A(i, j)\) for \(j=i<=n\).

SCALE (output)
If \(\mathbb{N} F O=0, S C A L E\) contains the scale factors for A.

SCOND (output)
If \(\mathbb{N} F O=0, S C A L E\) contains the ratio of the sm allest SCA LE (i) to the largestSCA LE (i). IfSCOND
>= 0.1 and AMAX is neither too large nor too sm all, it is notw orth scaling by SCA LE .

AMAX (output)
A bsolute value of largestm atrix elem ent. IfA M A X
is very close to overflow orvery close to underflow , the m atrix should be scaled.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the i-th argum enthad an illegalvalue
> 0 : if \(\mathbb{N F O}=\) i, the \(i\)-th diagonal elem ent is nonpositive.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
spprfs - im prove the com puted solution to a system of linear equations when the coefficientm atrix is sym \(m\) etric positive definite and packed, and provides emrorbounds and backw ard error estim ates for the solution

\section*{SYNOPSIS}
```

SU BROUTINE SPPRFS (UPLO,N,NRHS,A,AF,B,LDB,X,LDX,FERR,BERR,
W ORK,W ORK2, NNFO)
CHARACTER * 1 UPLO
\mathbb{N TEGER N,NRHS,LDB,LDX, IN FO}
INTEGER W ORK2 (*)
REALA (*), AF (*), B (LDB **), X (LDX **), FERR (*), BERR (*),
WORK (*)
SU BROUT\mathbb{NE SPPRFS_64 (UPLO,N,NRHS,A,AF,B,LDB,X,LDX ,FERR,}
BERR,W ORK,W ORK2, NNFO)
CHARACTER * 1 UPLO
INTEGER*8N,NRHS,LDB,LDX, IN FO
INTEGER*8 W ORK2 (*)
REALA (*), AF (*), B (LDB ,*), X (LDX **), FERR (*), BERR (*),
W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE PPRFS (UPLO,N, NRHS],A,AF,B, [LDB],X, [LDX],FERR, BERR, [W ORK], [W ORK2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION (:) ::W ORK2
```

REAL,DIM ENSION (:) ::A,AF,FERR,BERR,W ORK

```
REAL,D \(\mathbb{M}\) ENSION (:,:) ::B,X

SU BROUT \(\mathbb{N} E\) PPRFS_64 (UPLO , N, \(\mathbb{N} R H S], A, A F, B,[L D B], X,[L D X], F E R R\), BERR, [WORK], [WORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) :: UPLO
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER (8), D \(\mathbb{M}\) ENSION (:) ::W ORK2
REAL,D \(\mathbb{I}\) ENSION (:) ::A,AF,FERR,BERR,WORK
REAL,D \(\mathbb{M}\) ENSION (: :) ::B,X

\section*{C INTERFACE}
\#include < sunperfh>
void spprfs (char uplo, intn, intnrhs, float *a, float *af, float *b, int ldb, float *x, int ldx, float *ferr, float *berr, int*info);
void spprfs_64 (charuplo, long n, long nihs, float *a, float *af, float *b, long ldb, float *x, long ldx, float * ferr, float *berr, long *info);

\section*{PURPOSE}
spprfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is sym \(m\) etric positive definite and packed, and provides emrorbounds and backw ard enror estim ates for the solution.

\section*{ARGUMENTS}

\section*{UPLO (input)}
\(=\mathrm{U}\) : : U pper triangle ofA is stored;
= L ': Low ertriangle ofA is stored.
\(N\) (input) The order of the matrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber
of colum ns of the m atrices B and X. NRHS >=0.
A (input) The upper or low er triangle of the sym \(m\) etric \(m\) atrix A, packed colum nw ise in a linear array. The jth column of A is stored in the array A as follow s: if UPLO \(=U{ }^{\prime}\), A \((i+(j-1) * j 2)=A(i, 7)\) for \(1<=i<=j\) ifUPLO \(=L\) ', A ( \(i+(j-1)^{*}(2 n-j / 2)\)
\(=A(i, 7)\) for \(j=i<=n\).

\section*{AF (input)}

The triangular factor \(U\) or \(L\) from the Cholesky factorization \(A=U * * T * U\) or \(A=L * L * *\), as com puted by SPPTRF /CPPTRF, packed colum nw ise in a linear aray in the sam e form at as A (see A).
\(B\) (input) The righthand side \(m\) atrix \(B\).
LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, N)\).
\(X\) (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by SPPTRS. On exit, the im proved solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the aray X . LD X >= \(\max (1, \mathbb{N})\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X()\) (the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{O}), \operatorname{FERR}(\underset{)}{(1)}\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(\mathrm{X}(\mathcal{j})-\mathrm{XTRUE}\) ) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true enror.

\section*{BERR (output)}

The com ponentw ise relative backw ard emor of each
solution vector \(X(\mathcal{j})\) (i.e., the sm allest relative
change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dím ension ( \(3 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfinlexit
<0: if \(\mathbb{I N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
sppss -com pute the solution to a real system of linear equations A * X = B,

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SPPSV (UPLO,N,NRHS,A,B,LDB,INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER N,NRHS,LDB,INFO}
REALA (*),B (LDB,*)
SU BROUT\mathbb{NE SPPSV_64 (UPLO,N,NRHS,A,B,LDB,INFO)}
CHARACTER * 1 UPLO
INTEGER*8N,NRHS,LDB,INFO
REALA (*),B (LDB,*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE PPSV (UPLO,N, NRHS],A,B, [LDB], [ $\mathbb{N} F O$ ])
CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER :: N, NRHS,LDB, $\mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::A
REAL,D $\mathbb{I M}$ ENSION (: : : : : B
SU BROUTINE PPSV_64 (UPLO,N, $\mathbb{N} R H S], A, B,[L D B],[\mathbb{N F O}])$
CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER (8) :: N,NRHS,LDB, $\mathbb{N} F O$
REAL,D IM ENSION (:) ::A
REAL,D $\mathbb{I}$ ENSION (:,:) ::B

```

\section*{C INTERFACE}
\#include <sunperfh>
void sppsv (char uplo, intn, intnrhs, float *a, float *b, intldb, int *info);
void sppsv_64 (charuplo, long n, long nrhs, float *a, float *b, long ldb, long *info);

\section*{PURPOSE}
sppsv com putes the solution to a real system of linearequations
\(A * X=B\), where \(A\) is an \(N\) boy \(-N\) sym \(m\) etric positive defintie \(m\) atrix stored in packed form at and \(X\) and \(B\) are \(N\) by-NRH S \(m\) atrices.

The Cholesky decom position is used to factorA as
\[
A=U * * T * U, \text { if } U P L O=U ' \text {,or }
\]
\[
\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}, \text { if } \mathrm{U} P \mathrm{LO}=\mathrm{L}^{\prime},
\]
\(w\) here \(U\) is an upper triangularm atrix and \(L\) is a low er triangular \(m\) atrix. The factored form of \(A\) is then used to solve the system ofequations \(A \mathrm{X}=\mathrm{B}\).

\section*{ARGUMENTS}

UPLO (input)
= U ': U pper triangle ofA is stored;
= L': Low er triangle ofA is stored.
N (input) The num ber of linearequations, i.e., the order of the \(m\) atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colm ns of them atrix B. NRHS \(>=0\).

A (input/output)
O \(n\) entry, the upper or low ertriangle of the sym \(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linear array. The \(j\) th colum \(n\) of \(A\) is stored in the array A as follows: if UPLO \(=U^{\prime}, A(i+(j\)
 \((j-1) *(2 n-j / 2)=A(i, 7)\) for \(\dot{j}=i<=n\). See below for further details.

On exit, if \(\mathbb{N F O}=0\), the factor \(U\) orL from the Cholesky factorization \(A=U * * T * U\) or \(A=L * L * * T\), in the sam e storage form at as A.

B (input/output)
On entry, the N -by-NRHS righthand side m atrix B. On ex弌, if \(\mathbb{N F O}=0\), the N boy \(-\mathrm{NRH} S\) solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the anay \(\mathrm{B} . \operatorname{LDB}>=\) \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the \(i-\) th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N}\) FO \(=i\), the leading \(m\) inoroforder iof \(A\) is notposilive definite, so the factorization could notbe com pleted, and the solution has not been com puted.

\section*{FURTHER DETAILS}

The packed storage schem e is illustrated by the follow ing exam ple when \(N=4, \mathrm{UPLO}=\mathrm{U}\) ':

Tw o-dim ensional storage of the sym \(m\) etric \(m\) atrix A :
```

al1 a12 al3 a14
a22 a23 a24
a33 a34 (aij= con\g (aï))
a44

```

Packed storage of the upper triangle ofA:
\(A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]\)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
```

sppsvx - use the C holesky factorization A = U **T *U or A =
L*L**T to com pute the solution to a realsystem of linear
equations A * X = B,

```

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE SPPSVX (FACT,UPLO,N,NRHS,A,AF,EQUED,S,B,LDB,}
X,LDX,RCOND,FERR,BERR,W ORK,W ORK2,INFO)
CHARACTER * 1FACT,UPLO,EQUED
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
INTEGER W ORK2 (*)
REALRCOND
REAL A (*), AF (*), S (*), B (LDB,*), X (LDX,*), FERR (*),
BERR (*),W ORK (*)
SU BROUT\mathbb{NE SPPSVX_64 (FACT,UPLO,N,NRHS,A,AF,EQUED,S,B,}
LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK 2, INFO)
CHARACTER * 1FACT,UPLO,EQUED
INTEGER*8N,NRHS,LDB,LDX,INFO
INTEGER*8 W ORK2 (*)
REALRCOND
REAL A (*), AF (*), S (*), B (LDB,*), X (LDX,*), FERR (*),
BERR (*),W ORK (*)

```

F95 INTERFACE
SU BROUTINE PPSVX (FACT,UPLO, \(\mathbb{N}], \mathbb{N} R H S], A, A F, E Q U E D, S, B\), [LDB],X, [LDX],RCOND ,FERR, BERR, [W ORK], [W ORK 2], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::FACT,UPLO,EQUED
\(\mathbb{N}\) TEGER ::N,NRHS,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: W\) ORK2
REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::A,AF,S,FERR,BERR,W ORK
REAL,D \(\mathbb{M}\) ENSION (: : : : : B , X

SUBROUTINE PPSVX_64 (FACT, UPLO, \(\mathbb{N}], \mathbb{N} R H S], A, A F, E Q U E D, S, B\), \([\lfloor D B], X,[\llbracket D X], R C O N D, F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::FACT,UPLO, EQUED
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):\) W ORK 2
REAL ::RCOND
REAL,D IM ENSION (:) ::A, AF, S, FERR, BERR,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::B,X

\section*{C INTERFACE}
\#include <sunperfh>
void sppsvx (char fact, charuplo, intn, intnrhs, float*a, float *af, char equed, float*s, float*b, int ldlb, float * \(x\), int ldx, float *roond, float * ferr, float*berr, int*info);
void sppsvx_64 (char fact, char uplo, long n, long nihs, float *a, float*af, charequed, float *s, float *b, long ldb, float *x, long ldx, float *rcond, float *ferr, float *berr, long *info);

\section*{PURPOSE}
sppsvx uses the C holesky factorization \(A=U * * T * U\) or \(A=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\) to com pute the solution to a realsystem of linear equations
\(A * X=B\), where \(A\) is an \(N\) boy \(-N\) sym m etric positive defintem atrix stored in packed form atand \(X\) and \(B\) are \(N\) foy-N RH S m atrices.

E rrorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT = E', real scaling factors are computed to equiliorate
the system :
\(\operatorname{diag}(S) * A * \operatorname{diag}(S) * \operatorname{inv}(\operatorname{diag}(S)) * X=\operatorname{diag}(S) * B\)
W hether or not the system w illbe equilibrated depends on the
scaling of the m atrix A , but ifequilibration is used, A
overw rilten by diag \((\mathrm{S}) \star A\) *diag \((\mathrm{S})\) and B by diag \((\mathrm{S}) \star\) B .
2. IfFACT = N 'or E', the Cholesky decom position is used to
factor them atrix A (afterequilibration ifFACT =E)
as
\[
\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}, \text { if } \mathrm{UPLO}=\mathrm{U} \text { ', or }
\]
\(\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\), ifUPLO = L ',
w here U is an upper triangularm atrix and L is a low er triangular
\(m\) atrix.
3. If the leading i-by-iprincipal m inor is not positive definite,
then the routine retums \(w\) ith \(\mathbb{N F O}=\) i. O therw ise, the factored
form ofA is used to estim ate the condition num ber of the \(m\) atrix
A. If the reciprocal of the condition num ber is less than \(m\) achine
precision, \(\mathbb{N} F O=N+1\) is retumed as a w aming, but the routine
still goes on to solve forX and com pute errorbounds as described below .
4. The system ofequations is solved forX using the factored form
of A.
5. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for it.
6. If equilibration \(w\) as used, the \(m\) atrix \(X\) is prem ultiplied by
diag (S) so that it solves the original system before equilibration.

\section*{ARGUMENTS}

FACT (input)
Specifies w hether or not the factored form of the \(m\) atrix \(A\) is supplied on entry, and if not, whether them atrix A should be equilibrated before it is factored. = F': On entry, AF contains the fac-
tored form ofA. IfEQUED \(=Y\) ', them atrix \(A\) has been equilibrated \(w\) ith scaling factors given by \(S\). A and AF w illnotbe m odified. \(=\mathrm{N}\) ': Them atrix A w illbe copied to A F and factored.
\(=\mathrm{E}\) : The matrix A w ill be equilibrated if necessary, then copied to AF and factored.

UPLO (input)
= U : U ppertriangle ofA is stored;
= L': Low er triangle ofA is stored.

N (input) The num ber of linear equations, i.e., the order of them atrix A. \(N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrices B and X. NRHS >=0.
A (input/output)
O \(n\) entry, the upper or low er triangle of the sym \(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linear array, exceptifFACT = F'and EQUED = \(Y^{\prime}\) ', then A \(m\) ust contain the equilibrated \(m\) atrix diag \((S) * A * \operatorname{diag}(S)\). The \(j\) th column ofA is stored in the array A as follow s: if UPLO = U', A (i+ \((j-1) * j 2)=A(i, 7)\) for \(1<=i<=j\) if \(U P L O=L \prime\), \(A(i+(j-1) *(2 n-j) / 2)=A(i, j)\) for \(j=i<=n\). See below for further details. A is not \(m\) odified if FACT = F' or \(\mathrm{N}^{\prime}\), orifFACT = E'andEQUED = N 'on exit.

On exit, ifFACT = E' and EQUED = \(\mathrm{Y}^{\prime}\), A is overw ritten by diag (S)*A *diag (S).

AF (input/output)
\((\mathbb{N} *(N+1) / 2)\) IfFACT \(=F\) ', then \(A F\) is an input argum ent and on entry contains the triangular factor \(U\) orL from the Cholesky factorization \(\mathrm{A}=\) \(U{ }^{*} U\) or \(A=L * L\) ', in the sam e storage form atas \(A\). IfEQUED ne. \(N\) ', then \(A F\) is the factored form of the equilibrated \(m\) atrix \(A\).

IfFACT = N ', then AF is an output argum ent and on exit retums the triangular factorU orL from the Cholesky factorization \(\mathrm{A}=\mathrm{U}\) * U orA \(=\mathrm{L} \star \mathrm{L}\) 'of the originalm atrix A.

IfFACT=E', then AF is an output argum ent and on exit retums the triangular factor \(U\) orL from the Cholesky factorization \(\mathrm{A}=\mathrm{U}\) * U orA \(=\mathrm{L}\) * L 'of the equilibrated \(m\) atrix A (see the description of

A for the form of the equilibrated \(m\) atrix).

EQUED (input)
Specifies the form of equilibration thatw as done.
\(=N\) ': No equilibration (alw ays true ifFA CT = N \({ }^{2}\).
\(=Y\) ': Equilibration \(w\) as done, i.e., A has been replaced by diag (S) * A * diag (S). EQUED is an input argum ent ifFACT = F '; otherw ise, it is an outputargum ent.

S (input/output)
The scale factors forA ; not accessed if EQUED = \(\mathrm{N}^{\prime} . \mathrm{S}\) is an inputargum ent ifFACT = F '; otherw ise, S is an outputargum ent. IfFACT \(=\mathrm{F}^{\prime}\) and
EQUED = Y', each elem entofS m ustbe positive.
B (input/output)
On entry, the N -by-N RH S righthand side m atrix B .
On exit, if EQUED = \(N\) ', \(B\) is notm odified; if
EQUED \(=Y\) ', \(B\) is overw ritten by diag \((S) * B\).

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=N+1\), the \(N\)-by-NRHS solution
\(m\) atrix \(X\) to the original system of equations.
\(N\) ote that if EQUED = \(Y\) ', A and B arem odified on exit, and the solution to the equilibrated system is inv (diag \((S)) * X\).

LD X (input)
The leading dim ension of the array X . LD X >= \(\max (1, N)\).

\section*{RCOND (output)}

The estim ate of the reciprocal condition num ber of the matrix A afterequilibration (if done). If
RCOND is less than them achine precision (in particular, ifRCOND \(=0\) ), the \(m\) atrix is singular to w orking precision. This condition is indicated by a retum code of \(\mathbb{N} \mathrm{FO}>0\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X(\neg)\) the \(j\) th colum \(n\) of the solution matrix X). If XTRUE is the true solution comesponding to \(X(\mathcal{O}), \operatorname{FERR}(\mathcal{O})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele-
\(m\) ent in \((X(\mathcal{O})\) X TRUE \()\) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector \(X(\mathcal{)}\) ) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).
W ORK (w orkspace)
dím ension ( \(3 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N N}\) FO = -i, the i-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N} F O=i\), and \(i\) is
\(<=\mathrm{N}\) : the leading m inor oforderiof A is not positive definite, so the factorization could not be com pleted, and the solution has not been com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1: \mathrm{U}\) is nonsingular, butRCOND is less than machine precision, \(m\) eaning that the \(m\) atrix is singularto working precision. Nevertheless, the solution and error bounds are com puted because there are a num berof siluations w here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{FURTHER DETAILS}

The packed storage schem e is illustrated by the follow ing exam plewhen \(\mathrm{N}=4, \mathrm{UPLO}=\mathrm{U}\) ':

Tw o-dim ensional storage of the sym \(m\) etric \(m\) atrix A :
```

al1 a12 al3 a14
a22 a23 a24
a33 a34 (aij= conjg (aji))

```
            a44

Packed storage of the upper triangle ofA :
\(A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]\)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
spptrf-com pute the C holesky factorization of a real sym \(m\) etric positive definite \(m\) atrix A stored in packed form at

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SPPTRF (UPLO,N,A,INFO)}
CHARACTER * 1 UPLO
INTEGER N, INFO
REALA (*)
SU BROUT\mathbb{NE SPPTRF_64(UPLO,N,A,INFO)}
CHARACTER * 1 UPLO
INTEGER*8 N, INFO
REALA (*)
F95 INTERFACE

```

```

CHARACTER (LEN=1)::UPLO
\mathbb{NTEGER ::N,\mathbb{NFO}}\mathbf{N}=
REAL,D IM ENSION (:) ::A
SUBROUTINE PPTRF_64 (UPLO,N,A,[\mathbb{NFO ])}
CHARACTER (LEN=1)::UPLO
\mathbb{NTEGER (8)::N,\mathbb{NFO}}\mathbf{N}=\mp@code{A}
REAL,DIM ENSION (:) ::A

```
void spptrf(charuplo, intn, float *a, int *info);
void spptrf_64 (char uplo, long n, float *a, long *info);

\section*{PURPOSE}
spptrf com putes the C holesky factorization of a real sym \(m\) etric positive definite \(m\) atrix A stored in packed form at.

The factorization has the form
\(A=U * * T * U\), if \(U P L O=U '\) 'or
\(A=L * L * * T\), if \(\mathrm{UPLO}=\mathrm{L}^{\prime}\),
\(w\) here \(U\) is an upper triangularm atrix and \(L\) is low er triangular.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': Upper triangle ofA is stored;
= L': Low er triangle of A is stored.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the upper or low ertriangle of the sym \(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linear amray. The jth colum n of \(A\) is stored in the array \(A\) as follow s: if UPLO \(=U '\) 'A (i+ (j \(1)^{\star} j 2\) ) \(=A(i, j)\) for \(1<=i<=j\) ifUPLO \(=\mathbb{L}\) ', A ( \(i+\) \((j-1) *(2 n-j / 2)=A(i, j)\) for \(j=i<=n\). See below for further details.

On exit, if \(\mathbb{N} F O=0\), the triangular factor \(U\) orL from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) orA \(=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\), in the sam e storage form atas A.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvałue
\(>0:\) if \(\mathbb{N F O}=i\), the leading \(m\) inoroforder is notpositive definite, and the factorization could notbe com pleted.

\section*{FURTHER DETAILS}

The packed storage schem e is illustrated by the follow ing exam ple w hen \(N=4, \mathrm{UPLO}=\mathrm{U}\) ':

Tw o-dim ensionalstorage of the sym \(m\) etric \(m\) atrix A :
```

a11 a12 a13 a14
a22 a23 a24
a33 a34 (aij= a\ddot{i})
a44

```

Packed storage of the upper triangle ofA :
\[
A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]
\]

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
spptri-com pute the inverse of a real symm etric positive definite \(m\) atrix \(A\) using the Cholesky factorization \(A=\) \(\mathrm{U} * * \mathrm{~T} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}\) com puted by SPPTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE SPPTRI(UPLO,N,A, NNFO)}
CHARACTER * 1 UPLO
INTEGER N,\mathbb{NFO}
REALA (*)
SU BROUT\mathbb{NE SPPTRI_64 (UPLO,N,A , IN FO )}
CHARACTER * 1 UPLO
INTEGER*8N,\mathbb{NFO}
REALA (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE PPTRI(UPLO , N, A , [ $\mathbb{N F O}$ ])
CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER ::N, $\mathbb{N} F O$
REAL,D IM ENSION (:) ::A
SU BROUTINE PPTRI_64 (UPLO,N,A, [NFO ])
CHARACTER (LEN=1)::UPLO
$\mathbb{N}$ TEGER ( 8 ) :: N, $\mathbb{N} F O$
REAL,D IM ENSION (:) ::A

```
void spptri(char upl0, intn, float*a, int *info);
```

void spptri_64 (charuplo, long n, float *a, long *info);

## PURPOSE

spptricom putes the inverse of a real symm etric positive definite $m$ atrix $A$ using the Cholesky factorization $A=$ $\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ orA $=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ com puted by SPPTRF.

## ARGUMENTS

UPLO (input)
$=\mathrm{U}:$ : Upper triangular factor is stored in A;
$=L^{\prime}:$ Low er triangular factor is stored in A.

N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
On entry, the triangular factor $U$ or $L$ from the Cholesky factorization $A=U * * T * U$ or $A=L * L * * T$, packed colum nw ise as a linear anray. The jth colum $n$ ofU orL is stored in the amay A as fol low s: if UPLO $=U$ ', A $(i+(j 1) \star j 2)=U(i, j)$ for $1<=i<=j$ if UPLO $=L$ ', A $(i+(j 1) *(2 n-j) / 2)$
$=L(i, j)$ for $\dot{j}=i<=n$.

O n exit, the upper or low er triangle of the (sym $m$ etric) inverse of $A$, overw riting the input factor U orL.
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N F O}=-i$, the i-th argum enthad an illegalvałue
$>0:$ if $\mathbb{N} F O=i$, the ( $(1, i)$ elem entof the factor
U orL is zero, and the inverse could notbe com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

spptrs - solve a system of linearequations A *X = B w th a sym $m$ etric positive definite $m$ atrix $A$ in packed storage using the Cholesky factorization $\mathrm{A}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ orA $=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ computed by SPPTRF

## SYNOPSIS

```
SUBROUTINE SPPTRS(UPLO,N,NRHS,A,B,LDB,\mathbb{NFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER N,NRHS,LDB,INFO}
REALA (*),B (LDB,*)
SU BROUT\mathbb{NE SPPTRS_64 (UPLO,N,NRHS,A,B,LD B,NNFO)}
CHARACTER * 1UPLO
INTEGER*8N,NRHS,LDB, INFO
REALA (*),B (LDB,*)
```


## F95 INTERFACE

```
SU BROUTINE PPTRS (UPLO,N, NRHS],A,B,[LDB],[NFO])
CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER ::N,NRHS,LDB, \(\mathbb{N}\) FO
REAL,D \(\mathbb{I}\) ENSION (:) ::A
REAL,D \(\mathbb{I}\) ENSION (: : : : : B
SU BROUTINE PPTRS_64 (UPLO ,N, \(\mathbb{N} R \mathrm{R}\) S], A, B, [LDB], [ \(\mathbb{N} F \mathrm{FO}\) ])
CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER ( 8 ) :: \(\mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} \mathrm{B}, \mathbb{N} F \mathrm{O}\)
REAL,D \(\mathbb{M}\) ENSION (:) ::A
```

REAL,D IM ENSION (:,: : : B

## C INTERFACE

\#include <sunperfh>
void spptrs (charuplo, intn, intnins, float *a, float *b, int ldld, int *info);
void spptrs_64 (charuplo, long n, long nrhs, float *a, float *b, long ldb, long *info) ;

## PURPOSE

spptrs solves a system of linearequations A *X $=\mathrm{B}$ w ith a sym $m$ etric posilive definite $m$ atrix A in packed storage using the Cholesky factorization $A=U * * T * U$ or $A=L * L * * T$ com puted by SPPTRF.

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': U pper triangle ofA is stored;
$=\mathbb{L}$ ': Low ertriangle of A is stored.

N (input) The order of the matrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the m atrix B. NRHS $>=0$.

A (input) The triangular factor $U$ or L from the Cholesky
factorization $A=U * * T * U$ orA $=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$, packed
colum nw ise in a linearanay. The jth colum n of
U or L is stored in the array A as follow s: if $\mathrm{UPLO}=\mathrm{U}, \mathrm{A}(i+(j-1) \star j 2)=\mathrm{U}(i, j)$ for $1<=\dot{i}<=\dot{j}$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{A}\left(i+(j-1)^{*}(2 n-j) / 2\right)=\mathrm{L}(i, j)$ for $j=i<=n$ 。

B (input/output)
O $n$ entry, the right hand side $m$ atrix $B$. On exit, the solution $m$ atrix X .

LD B (input)
The leading dim ension of the array B . LD B >= $\max (1, N)$.
= 0: successfulexit
< 0: if $\mathbb{N}$ FO $=-$ i, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

sptcon - oom pute the reciprocal of the condition num ber (in the 1 -norm ) of a realsym $m$ etric positive definite tridiagonalm atrix using the factorization $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ or $\mathrm{A}=$ $\mathrm{U} * * \mathrm{~T} * \mathrm{D} * \mathrm{U}$ com puted by SPTTRF

## SYNOPSIS

SU BROUTINE SPTCON $\mathbb{N}, D \mathbb{I} G, O F F D, A N O R M, R C O N D, W O R K, \mathbb{N} F O$ )
$\mathbb{N}$ TEGERN, $\mathbb{N} F O$
REALANORM,RCOND
REALDIAG (*),OFFD (*),W ORK (*)
SUBROUTINE SPTCON_64 $\mathbb{N}, D \mathbb{A} G, O F F D, A N O R M, R C O N D, W O R K, \mathbb{N} F O$ )
$\mathbb{N}$ TEGER*8 $\mathrm{N}, \mathbb{I N F O}$
REAL ANORM, RCOND
REALDIAG (*), OFFD (*),WORK (*)

## F95 INTERFACE

SUBROUTINE PTCON ( $\mathbb{N}], D \mathbb{I} G, O F F D, A N O R M, R C O N D,[W O R K],[\mathbb{N} F O])$
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
REAL ::ANORM,RCOND
REAL,D $\mathbb{I M}$ ENSION (:) ::D IA G,OFFD,W ORK
SU BROUTINE PTCON_64 ( $\mathbb{N}], D \mathbb{I} G, O F F D, A N O R M, R C O N D,[W O R K],[\mathbb{N} F O])$
$\mathbb{N}$ TEGER (8) ::N, $\mathbb{N} F O$
REAL ::ANORM,RCOND
REAL,D $\mathbb{I}$ ENSION (:) ::D IA G,OFFD ,W ORK

## C INTERFACE

\#include <sunperfh>
void sptcon (intn, float*diag, float *offd, float anorm, float *roond, int*info);
void sptcon_64 (long n, float *diag, float *offd, float anorm, float *rcond, long *info);

## PURPOSE

sptcon com putes the reciprocal of the condition num ber (in the 1 -norm ) of a realsym $m$ etric positive definite tridiagonalm atrix using the factorization $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ or $\mathrm{A}=$ $\mathrm{U} * * \mathrm{~T} * \mathrm{D}$ * U com puted by SPTTRF.
Norm (inv (A)) is com puted by a direct method, and the reciprocal of the condition num ber is com puted as

$$
\operatorname{RCOND}=1 /(\operatorname{ANORM} * \operatorname{nom}(\operatorname{inv}(A))) .
$$

## ARGUMENTS

N (input) The order of the matrix $A . N>=0$.

D IA G (input)
Then diagonalelem ents of the diagonal $m$ atrix D IA G from the factorization of A, as com puted by SPTTRF.

OFFD (input)
The ( $n-1$ ) off-diagonalelem ents of the unit bidiagonal factorU orl from the factorization ofA, as com puted by SPTTRF .

## ANORM (input)

The 1-norm of the originalm atrix A.

## RCOND (output)

The reciprocal of the condition number of the $m$ atrix $A$, com puted asRCOND = 1/(ANORM *A $\mathbb{N} V N M)$, where $A \mathbb{N} V N M$ is the 1 -nom of inv ( $A$ ) com puted in this routine.

W ORK (w orkspace)
dim ension $(\mathbb{N})$
$=0$ : successfulexit
< 0: if $\mathbb{N}$ FO = -i, the i-th argum enthad an illegalvalue

## FURTHER DETAILS

Them ethod used is described in N icholas J . H igham, "E fficient A lgorithm s for C om puting the C ondition N um berofa TridiagonalM atrix", SIA M J. Sci.Stat. C om put., V ol. 7, No. 1, January 1986.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

spteqr - com pute alleigenvalues and, optionally, eigenvectors of a sym $m$ etric positive definite tridiagonalm atrix by first factoring the $m$ atrix using SPTTRF, and then calling SBD SQ R to com pute the singular values of the bidiagonal factor

## SYNOPSIS

```
SUBROUT\mathbb{NE SPTEQR (COMPZ,N,D,E,Z,LDZ,W ORK,INFO)}
CHARACTER * 1 COMPZ
INTEGERN,LDZ,\mathbb{NFO}
REALD (*),E (*),Z (LD Z ,*),W ORK (*)
SUBROUT\mathbb{NE SPTEQR_64(COMPZ,N,D,E,Z,LDZ,W ORK,INFO)}
CHARACTER * 1 COMPZ
INTEGER*8N,LD Z,INFO
REALD (*),E (*),Z (LD Z,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE PTEQR (COMPZ, $\mathbb{N}], D, E, Z,[L D Z],[W O R K],[\mathbb{N} F O])$

CHARACTER (LEN=1) ::COM PZ
$\mathbb{N} T E G E R:: N, L D Z, \mathbb{N} F O$
REAL,D $\mathbb{I}$ ENSION (:) ::D ,E,W ORK
REAL,D IM ENSION (:,:) ::Z
SU BROUTINE PTEQR_64 (COMPZ, $\mathbb{N}], D, E, Z,[L D Z],[W O R K],[\mathbb{N} F O])$
CHARACTER (LEN=1) ::COM PZ
$\mathbb{N}$ TEGER (8) :: N, LD Z, $\mathbb{N} F O$

REAL,D $\mathbb{M} E N S I O N(:):$,, E, W ORK
REAL,D $\mathbb{I}$ ENSION (:,:) ::Z

## C INTERFACE

\#include <sunperfh>
void spteqr(char com pz, intn, float *d, float *e, float *z, int ldz, int*info);
void spteqr_64 (charcom pz, long n, float *d, float *e, float *z, long ldz, long *info);

## PURPOSE

spteqr computes all eigenvalues and, optionally, eigenvectors of a sym $m$ etric positive definite tridiagonal $m$ atrix by first factoring the $m$ atrix using SPTTRF, and then calling SBD SQ R to com pute the singularvalues of the bidiagonal factor.

This routine com putes the eigenvalues of the positive definite tridiagonal $m$ atrix to high relative accuracy. This $m$ eans that if the eigenvalues range overm any orders ofm agnitude in size, then the sm alleigenvalues and corresponding eigenvectorsw illbe com puted m ore accurately than, for exam ple, w th the standard Q R m ethod.

The eigenvectors of a fullorband sym $m$ etric posilive defintie $m$ atrix can also be found ifSSY TRD, SSPTRD, orSSBTRD has been used to reduce this $m$ atrix to tridiagonal form . (The reduction to tridiagonal form, how ever, $m$ ay preclude the possibility of obtaining high relative accuracy in the sm all eigenvalues of the originalm atrix, ifthese eigenvalues range overm any orders ofm agnitude.)

## ARGUMENTS

COMPZ (input)
$=\mathrm{N}^{\prime}:$ C om pute eigenvalues only .
$=\mathrm{V}:$ : Com pute eigenvectors of originalsym m etric
$m$ atrix also. A reay $Z$ contains the orthogonal $m$ atrix used to reduce the originalm atrix to tridiagonal form . = $I ':$ C om pute eigenvectors of tridiagonalm atrix also.

N (input) The order of the m atrix. $\mathrm{N}>=0$.

D (input/output)
O n entry, the n diagonalelem ents of the tridiagonalm atrix. On norm alexit, $D$ contains the eigenvalues, in descending order.

E (input/output)
O $n$ entry, the ( $n-1$ ) subdiagonal elem ents of the tridiagonal matrix. On exit, E hasbeen destroyed.

Z (input) On entry, if $\mathrm{COMPZ}=\mathrm{V}$ ', the orthogonal m atrix used in the reduction to tridiagonal form. On exit, if $C O M P Z=V$ ', the orthonorm aleigenvectors of the original sym $m$ etric $m$ atrix; if $C O M P Z=I$ ', the orthonorm aleigenvectors of the tridiagonal $m$ atrix. If $\mathbb{N} F O>0$ on exit, $Z$ contains the eigenvectors associated with only the stored eigenvalues. If $C O M P Z=N$ ', then $Z$ is not referenced.

LD Z (input)
The leading din ension of the array $\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1$, and ifCOM PZ $=V$ 'or $I$ ', $L D Z>=m a x(1, N)$.

W ORK (w orkspace)
dim ension (4*N )
$\mathbb{N} F O$ (output)
= 0: successfulexit.
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum enthad an illegalvahue.
> 0: if $\mathbb{N} F O=i$, and $i$ is: <= $N$ the Cholesky factorization of the $m$ atrix could notbe perform ed because the $i$-th principalm inorw as not positive definite. > N the SVD algorithm failed to converge; if $\mathbb{N} F O=\mathrm{N}+$ i, ioff-diagonal elem ents of the bidiagonal factor did not converge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sptrifs - im prove the com puted solution to a system of linear equations $w$ hen the coefficientm atrix is sym $m$ etric positive definite and tridiagonal, and provides error bounds and backw ard errorestim ates for the solution

## SYNOPSIS

```
SUBROUT\mathbb{NE SPTRFS N,NRHS,D IAG,OFFD,D IAGF,OFFDF,B,LDB,X,LDX,}
    FERR,BERR,W ORK,INFO)
\mathbb{NTEGERN,NRHS,LDB,LDX,}\mathbb{NFO}
REAL DIAG (*),OFFD (*), DIAGF (*), OFFDF (*), B (LDB,*),
X (LDX,*),FERR (*),BERR (*),WORK (*)
SU BROUTINE SPTRFS_64 N,NRHS,DIAG,OFFD,D IAGF,OFFDF,B,LDB,X,
    LDX,FERR,BERR,WORK,INFO)
\mathbb{NTEGER*8N,NRHS,LDB,LDX, INFO}
REAL DIAG (*),OFFD (*), DIAGF (*), OFFDF (*), B (LDB,*),
X (LDX,*),FERR (*),BERR (*),W ORK (*)
```


## F95 INTERFACE

SU BROUT $\mathbb{N} E \operatorname{PTRFS}(\mathbb{N}], \mathbb{N} R H S], D \mathbb{I A}$, OFFD, D $\mathbb{A} G F, O F F D F, B,[L D B], X$, [LDX],FERR,BERR, [W ORK], [NFO])
$\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N} F O$
REAL,D $\mathbb{I M} E N S I O N(:):: D \mathbb{A} G, O F F D, D \mathbb{A} G F, O F F D F, F E R R, B E R R$,
W ORK
REAL,D IM ENSION (: : : : : B , X
SU BROUTINEPTRFS_64 (N), NRHS],D IA G,OFFD,D IA GF,OFFDF,B,[LDB], $\mathrm{X},[\operatorname{LDX}], F E R R, B E R R,[\mathbb{W} O R K],[\mathbb{N} F O])$
$\mathbb{N}$ TEGER (8) :: N , NRHS,LD B, LD X , $\mathbb{N}$ FO
REAL, D $\mathbb{M} E N S I O N$ (:) ::D $\mathbb{A} G, O F F D, D I A G F, O F F D F, F E R R, B E R R$, W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) ::B,X

## C INTERFACE

\#include <sunperfh>
void sptrfs (intn, intnrhs, float *diag, float * offd, float *diagf, float *offdf, float *b, int ldb, float *x, int ldx, float * ferr, float *berr, int*info);
void sptrifs_64 (long n, long nihs, float *diag, float *offf, float *diagf, float *offdf, float *b, long ldb, float *x, long ldx, float * ferr, float *berr, long *info);

## PURPOSE

sptrfs im proves the com puted solution to a system of linear equations $w$ hen the coefficientm atrix is sym $m$ etric positive definite and tridiagonal, and provides error bounds and backw ard errorestim ates for the solution.

## ARGUMENTS

N (input) The order of them atrix A. $\mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRH S $>=0$.

D IA G (input)
The $n$ diagonalelem ents of the tridiagonal $m$ atrix
A.

OFFD (input)
The ( $n-1$ ) subdiagonalelem ents of the tridiagonal $m$ atrix $A$.

D IA GF (input)
The $n$ diagonalelem ents of the diagonal $m$ atrix
D IA G from the factorization com puted by SPTTRF .

OFFDF (input)
The ( $n-1$ ) subdiagonalelem ents of the unitbidiag-
onal factor $L$ from the factorization com puted by SPTTRF.
$B$ (input) The righthand side $m$ atrix $B$.
LD B (input)
The leading dim ension of the anay $B$. LD B >= $\max (1, N)$.

X (input/output)
On entry, the solution $m$ atrix $X$, as com puted by SPTTRS. On exit, the im proved solution $m$ atrix $X$.

LD X (input)
The leading dim ension of the anay X . LD X >= $\max (1, \mathbb{N})$.

FERR (output)
The forw ard errorbound foreach solution vector $X(j)$ (the $j$ th collm $n$ of the solution $m$ atrix $X$ ). If $X T R U E$ is the true solution corresponding to $X(\mathcal{j}), \operatorname{FERR}(\mathcal{J})$ is an estim ated upperbound for the $m$ agnitude of the largestelem ent in (X ( $)$-X TRU E) divided by the $m$ agnitude of the largestelem ent in $\mathrm{X}(\mathrm{J})$.

BERR (output)
The com ponentw ise relative backw ard error of each
solution vectorX (i) (i.e., the sm allest relative change in any elem entofA orB thatm akesX ( $\mathcal{j}$ ) an exactsolution).

W ORK (w orkspace)
dim ension $\left(2{ }^{*} \mathrm{~N}\right)$
$\mathbb{N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sptsv - com pute the solution to a real system of linear equations $A * X=B$, where $A$ is an $N$ by $N$ symm etric positive definite tridiagonalm atrix, and X and B are N -by-NRHS m atrices.

## SYNOPSIS

```
SUBROUT\mathbb{NE SPTSV N,NRHS,D IAG,SUB,B,LDB,INFO)}
```

$\mathbb{N}$ TEGER N,NRHS,LDB, $\mathbb{N} F O$
REALDIAG (*), SUB (*), B (LDB,*)
SUBROUTINESPTSV_64 $\mathbb{N}, N R H S, D \mathbb{A} G, S U B, B, L D B, \mathbb{N} F O)$
$\mathbb{N}$ TEGER*8N,NRHS,LDB, $\mathbb{N} F O$
REALDIAG (*), SUB (*), B (LDB,$\left.^{\star}\right)$

## F95 INTERFACE

SU BROUTINE PTSV ( $\mathbb{N}], \mathbb{N} R H S], D \mathbb{A} G, S U B, B,[L D B],[\mathbb{N} F O])$
$\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O$
REAL,D $\mathbb{I M} E N S I O N(:):$ : $\mathbb{A} G, S U B$
REAL,D $\mathbb{I}$ ENSION (:,:) ::B
SUBROUTINEPTSV_64 ( $\mathbb{N}], \mathbb{N} R H S], D \mathbb{A} G, S U B, B,[L D B],[\mathbb{N} F O])$
$\mathbb{N}$ TEGER (8) ::N,NRHS,LDB, $\mathbb{N}$ FO
REAL,D $\mathbb{I M} E N S I O N(:):: D \mathbb{I A}, S U B$
REAL,D $\mathbb{M}$ ENSION (: : : : : B

## C INTERFACE

\#include < sunperfh>
void sptsv (intn, intnrhs, float*diag, float *sub, float *b, int ldb, int *info);
void sptsv_64 (long n, long nrhs, float *diag, float *sub, float *b, long ldb, long *info);

## PURPOSE

sptsv com putes the solution to a realsystem of linear equations $A * X=B$, where $A$ is an $N$ boy $N$ symm etric positive definite tridiagonalm atrix, and $X$ and $B$ are $N$ boy-NRHS $m$ atrices.
$A$ is factored as $A=L * D * L * * T$, and the factored form of $A$ is then used to solve the system of equations.

## ARGUMENTS

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS $>=0$.

D IA G (input/output)
O n entry, the n diagonalelem ents of the tridiagonalm atrix A. On exit, the $n$ diagonalelem ents of the diagonalm atrix D IA G from the factorization $A$ $=\mathrm{L} * \mathrm{D} \mathbb{I} \mathrm{G} * \mathrm{~L} * * \mathrm{~T}$.

SU B (input/output)
O $n$ entry, the $(n-1)$ subdiagonal elem ents of the tridiagonalm atrix A. On exit, the ( $\mathrm{n}-1$ ) subdiagonalelem ents of the unit.bidiagonal factor $L$ from the L *D IA G *L**T factorization ofA. (SU B can also be regarded as the superdiagonal of the unit.bidiagonal factor $U$ from the $U * * T * D I A G * U$ factorization ofA.)

B (input/output)
O n entry, the N boy-N RH S righthand side m atrix B. On exit, if $\mathbb{N F O}=0$, the N by N RH S solution m atrix $X$.

LD B (input)
The leading dim ension of the array B. LD B >=
$\max (1, N)$.
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the i-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N F O}=i$, the leading $m$ inoroforder $i$ is not positive definite, and the solution has not
been com puted. The factorization has not been com pleted unless $i=N$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sptsvx - use the factorization $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ to com pute the solution to a realsystem of linearequations $A * X=B$, where $A$ is an $N$ by $-N$ symm etric positive definite tridiagonal $m$ atrix and $X$ and $B$ are $N$-by $-N$ R H S $m$ atrices

## SYNOPSIS

```
SUBROUT\mathbb{NE SPTSVX (FACT,N,NRHS,DIAG,SUB,D IAGF,SUBF,B,LDB,X,}
    LDX,RCOND,FERR,BERR,W ORK,INFO)
CHARACTER * 1FACT
\mathbb{NTEGERN,NRHS,LDB,LDX,}\mathbb{NFO}
REALRCOND
REALDIAG (*),SUB (*),D IA GF (*),SUBF (*),B (LDB,*),X (LDX,*),
FERR (*),BERR (*),W ORK (*)
SU BROUTINE SPTSVX_64(FACT,N,NRHS,DIAG,SUB,D IAGF,SUBF,B,LDB,
    X,LDX,RCOND,FERR,BERR,W ORK,\mathbb{NFO)}
CHARACTER * 1FACT
\mathbb{N}TEGER*8N,NRHS,LDB,LDX,INFO
REALRCOND
REALD IA G (*),SUB (*),D IA GF (*),SU BF (*),B (LD B ,*),X (LDX ,*),
FERR (*),BERR (*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINEPTSVX $\mathbb{E A C T}, \mathbb{N}], \mathbb{N} R H S], D \mathbb{A} G, S U B, D \mathbb{A} G F, S U B F, B,[L D B]$, $\mathrm{X},[\operatorname{LD} \mathrm{X}], R C O N D, F E R R, B E R R,[W O R K],[\mathbb{N} F O])$

CHARACTER (LEN=1)::FACT
$\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N} F O$
REAL ::RCOND

REAL, D $\mathbb{M} E N S I O N(:):$ D $\mathbb{A} G, S U B, D \mathbb{A} G F, S U B F, F E R R, B E R R$,
W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) ::B , X

SU BROUTINE PTSVX_64 (FACT, $\mathbb{N}], \mathbb{N R H S}], D \mathbb{I A G}, \operatorname{SUB}, D \mathbb{A} G F, S U B F, B$,


CHARACTER (LEN=1) ::FACT
$\mathbb{N}$ TEGER (8) :: N , NRHS,LDB,LDX, $\mathbb{N}$ FO
REAL ::RCOND
REAL, D $\mathbb{I M} E N S I O N$ (:) :: D IAG, SUB, DIAGF, SUBF, FERR, BERR,
W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) ::B,X

## C INTERFACE

\#include <sunperfh>
void sptsvx (char fact, intn, intnrhs, float *diag, float
*sub, float *diagf, float *subf, float*b, int
ldlb, float *x, int ldx, float *roond, float * ferr, float*berr, int*info);
void sptsvx_64 (charfact, long n, long nrhs, float *diag, float *sub, float*diagf, float*subf, float*b, long ldlb, float *x, long ldx, float *rcond, float * ferr, float *berr, long *info);

## PURPOSE

sptsvx uses the factorization $A=L * D * L * * T$ to com pute the solution to a realsystem of linearequations $A * X=B$, where A is an N boy N symm etric positive definite tridiagonal $m$ atrix and X and B are N Hy N R H S m atrioes.

Emorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:

1. IfFACT $=N$ ', them atrix $A$ is factored as $A=L * D * L * * T$, where L
is a unit low erbidiagonalm atrix and $D$ is diagonal. The factorization can also be regarded as having the form
$A=U * * T * D * U$.
2. If the leading iboy-iprincipal $m$ inor is not positive definite,
then the routine retums w ith $\mathbb{N F O}=$ i. O therw ise, the factored
form ofA is used to estim ate the condition num ber of the
m atrix
A. If the reciprocal of the condition num ber is less than $m$ achine
precision, $\mathbb{N}$ FO $=\mathrm{N}+1$ is retumed as a w aming, but the routine
still goes on to solve for X and com pute emorbounds as described below .
3.The system ofequations is solved forX using the factored form of A.
3. Iterative refinem ent is applied to im prove the com puted solution
$m$ atrix and calculate error bounds and backw ard error estim ates
for it.

## ARGUMENTS

FACT (input)
Specifies w hether ornot the factored form of $A$ has been supplied on entry. = F': On entry, D IA GF and SUBF contain the factored form of A.
D IA G , SUB, D IA G F , and SUBF w illnotbe m odified.
$=N$ ': Them atrix A w illbe copied to D IA GF and SUBF and factored.

N (input) The order of them atrix $A . N>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber
of colum ns of the $m$ atriges $B$ and X. NRHS $>=0$.

D IA G (input)
The $n$ diagonalelem ents of the tridiagonal matrix A.

SU B (input)
The ( $n-1$ ) subdiagonalelem ents of the tridiagonal $m$ atrix A.

D IA GF (input/output)
IfFACT = $\mathrm{F}^{\prime}$, then $D \mathrm{IAGF}$ is an inputargum entand on entry contains the $n$ diagonalelem ents of the diagonalm atrix D IA G from the $L$ *D IA G *L **T factorization ofA. IfFACT $=N$ ', then D IA GF is an outputargum entand on exitcontains the $n$ diagonal
elem ents of the diagonal $m$ atrix $D \mathbb{I A}$ from the $\mathrm{L} * \mathrm{D}$ IA $G * \mathrm{~L} * *$ T factorization ofA.

## SU BF (input/output)

IfFACT $=\mathrm{F}^{\prime}$, then SUBF is an inputargum ent and on entry contains the ( $n-1$ ) subdiagonalelem ents of the unit bidiagonal factor $L$ from the $\mathrm{L} * \mathrm{D} \mathbb{I A}{ }^{*} \mathrm{~L} * * \mathrm{~T}$ factorization of A . If $\mathrm{FACT}=\mathrm{N}^{\prime}$, then SUBF is an outputargum entand on exit contains the ( $n-1$ ) subdiagonalelem ents of the unit bidiagonal factorL from the $L * D \operatorname{IAG} * L * * T$ factorization ofA.
$B$ (input) The N by -N RH S righthand side $m$ atrix $B$.
LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.

X (output)
If $\mathbb{N} F O=0$ of $\mathbb{N} F O=N+1$, the N -by-NRHS solution $m$ atrix $X$.

LD X (input)
The leading dim ension of the amay X . LD X >= $\max (1, N)$.

## RCOND (output)

The reciprocalcondition num berof the $m$ atrix $A$. If RCOND is less than the $m$ achine precision (in particular, ifRCOND $=0$ ), the $m$ atrix is singular to working precision. This condition is indicated by a retum code of $\mathbb{N}$ FO $>0$.

FERR (output)
The forw ard emorbound foreach solution vector $X$ (j) (the $j$ th colum $n$ of the solution $m$ atrix $X$ ). IfXTRUE is the true solution comesponding to $X(\mathcal{j})$, FERR $(\mathcal{j})$ is an estim ated upperbound for the $m$ agnitude of the largestelem ent in (X ( $)$-X TRU E) divided by the $m$ agnitude of the largestelem ent in $\mathrm{X}(\underset{)}{ }$.

BERR (output)
The com ponentw ise relative backw ard emorof each solution vector $X(\mathcal{j})$ (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( $\mathcal{j}$ ) an exactsolution).

W ORK (w orkspace)
dim ension $(2 * N)$
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N F O}=-$ i, the i-th argum enthad an illegalvalue
$>0$ : if $\mathbb{N} F O=i$, and $i$ is
$<=N$ : the leading $m$ inor oforderiof $A$ is not positive definite, so the factorization could not
be com pleted, and the solution has not been com puted. RCOND $=0$ is retumed. $=\mathrm{N}+1: \mathrm{U}$ is nonsingular, butRCOND is less than machine precision, $m$ eaning that the $m$ atrix is singular to w orking precision. Nevertheless, the solution and error bounds are com puted because there are a num ber of situations w here the com puted solution can bem ore accurate than the value ofRC OND w ould suggest.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

spttrf-com pute the L *D *L 'factorization of a real sym m etric positive definite tridiagonalm atrix A

## SYNOPSIS

SUBROUTINE SPTTRF $\mathbb{N}, ~ D \mathbb{I A G}, ~ O F F D, ~ \mathbb{N} F O$ )
$\mathbb{N}$ TEGER $N, \mathbb{N} F O$
REALDIAG (*), OFFD ( ${ }^{*}$ )
SU BROUTINE SPTTRF_64 $\mathbb{N}, D \mathbb{A} G, O F F D, \mathbb{N} F O$ )
$\mathbb{N}$ TEGER*8N, $\mathbb{N} F O$
REALDIAG (*), OFFD (*)

## F95 INTERFACE

SU BROUTINE PTTRF ( $\mathbb{N}], D \mathbb{I A} G, O F F D,[\mathbb{N} F O])$
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
REAL,D IM ENSION (:) ::D IA G,OFFD
SU BROUTINE PTTRF_64 ( $\mathbb{N}$ ],D $\mathbb{I A} G, O F F D,[\mathbb{N} F O]$ )
$\mathbb{N}$ TEGER (8) ::N, $\mathbb{N} F O$
REAL,D IM ENSION (:) ::D IA G,OFFD

## C INTERFACE

\#include <sunperfh>
void spttrf(intn, float *diag, float *offd, int *info);
void spttrf_64 (long n, float *diag, float *offd, long

## PURPOSE

spttrf com putes the L *D *L 'factorization of a real sym m etric positive definite tridiagonalm atrix A. The factorization $m$ ay also be regarded as having the form $A=U * D * U$.

## ARGUMENTS

N (input) The order of the m atrix A. $\mathrm{N}>=0$.
D IA G (input/output)
On entry, the $n$ diagonalelem ents of the tridiagonalm atrix A. On exit, the $n$ diagonalelem ents of the diagonalm atrix D IA G from the $L * D \mathbb{I A} G * L^{\prime}$ factorization of A.

OFFD (input/output)
O $n$ entry, the $(n-1)$ subdiagonal elem ents of the tridiagonalm atrix A. On exit, the ( $n-1$ ) subdiagonalelem ents of the unit.bidiagonal factorL from the L*D IA G *L' factorization ofA. OFFD can also be regarded as the superdiagonal of the unitbidiagonal factor $U$ from the $U$ *D IA G *U factorization ofA.
$\mathbb{N}$ FO (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N} F O=-\mathrm{k}$, the k -th argum enthad an illegalvalue
$>0$ : if $\mathbb{N} F O=k$, the leading $m$ inoroforderk is notpositive definite; if $k<N$, the factorization could notbe com pleted, while if $\mathrm{k}=\mathrm{N}$, the factorization w as com pleted, butD $\mathbb{I A} G \mathbb{N})=0$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sptters - solve a tridiagonalsystem of the form A * X = B using the $L * D * L$ 'factorization of A com puted by SPTTRF

## SYNOPSIS

SU BROUTINE SPTTRS $\mathbb{N}, N R H S, D \mathbb{I A}, ~ O F F D, B, L D B, \mathbb{N F O}$ )
$\mathbb{N}$ TEGER N,NRHS,LDB, $\mathbb{N} F O$
REALDIAG (*), OFFD (*), B (LDB,$^{*}$ )
SUBROUTINE SPTTRS_64 $\mathbb{N}, N R H S, D \mathbb{A} G, O F F D, B, L D B, \mathbb{N} F O)$
$\mathbb{N} T E G E R * 8 N, N R H S, L D B, \mathbb{N} F O$
REALDIAG (*),OFFD (*), B (LDB,*)

## F95 INTERFACE

SU BROUTINE PTTRS ( $\mathbb{N}], \mathbb{N} R H S], D \mathbb{A} G, O F F D, B,[L D B],[\mathbb{N} F O])$
$\mathbb{N}$ TEGER :: N,NRHS,LDB, $\mathbb{N}$ FO
REAL,D $\mathbb{I}$ ENSION (:) ::D IA G,OFFD
REAL,D $\mathbb{M}$ ENSION (:,:) ::B
SUBROUTINEPTTRS_64 ( $\mathbb{N}], \mathbb{N} R H S], D \mathbb{I A} G, O F F D, B,[L D B],[\mathbb{N} F O])$
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} B, \mathbb{N} F \mathrm{O}$
REAL,D $\mathbb{I M} E N S I O N(:):: D \mathbb{I A G}, O F F D$
REAL,D $\mathbb{M}$ ENSION (:,:) ::B

## C INTERFACE

\#include <sunperfh>
void spttres(intn, intnrhs, float *diag, float *offf, float
void spttrs_64 (long n, long nihs, float *diag, float *offd, float *b, long ldb, long *info);

## PURPOSE

sptters solves a tridiagonal system of the form
A * $\mathrm{X}=\mathrm{B}$ using the L *D *L 'factorization of A com puted by SPTTRF. $D$ is a diagonalm atrix specified in the vectorD, $L$ is a unitbidiagonalm atrix $w$ hose subdiagonal is specified in the vector $E$, and $X$ and $B$ are $N$ by NRH S $m$ atrices.

## ARGUMENTS

N (input) The order of the tridiagonalm atrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS $>=0$.

D IA G (input)
Then diagonalelem ents of the diagonal $m$ atrix
D IA G from the $L * D \mathbb{I A}$ * $L$ 'factorization of $A$.

OFFD (input/output)
The ( $n-1$ ) subdiagonalelem ents of the unitbidiagonal factor L from the $L$ *D IA G *L 'factorization of A. OFFD can also be regarded as the superdiagonal of the unitbidiagonal factor $U$ from the factorization $\mathrm{A}=\mathrm{U}$ *D IA A * .

B (input/output)
On entry, the righthand side vectors B for the system of linearequations. On exit, the solution vectors, $X$.

LD B (input)
The leading dim ension of the array $B$. LD B >= $\max (1, N)$.
$\mathbb{N} F O$ (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N} F O=-\mathrm{k}$, the k -th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sptts2 -solve a tridiagonalsystem of the form A * X = B using the $L * D * L$ 'factorization of $A$ com puted by SPTTRF

## SYNOPSIS

SUBROUTINE SPTTS $2 \mathbb{N}, N R H S, D, E, B, L D B)$
$\mathbb{I N} T E G E R N, N R H S, L D B$
REALD (*), E (*), B (LDB,*)

SU BROUTINE SPTTS2_64 $\mathbb{N}, N R H S, D, E, B, L D B)$
$\mathbb{N}$ TEGER*8N,NRHS,LDB
REALD (*), E (*), B (LDB,*)

## F95 INTERFACE

SU BROUTINE SPTTS2 $\mathbb{N}, N R H S, D, E, B, L D B)$
$\mathbb{N} T E G E R:: N$,NRHS,LDB
REAL,D IM ENSION (:) :: D , E
REAL,D $\mathbb{M}$ ENSION (:,:) ::B
SU BROUTINE SPTTS2_64 $\mathbb{N}, N R H S, D, E, B, L D B)$
$\mathbb{N T E G E R}(8):: N, N R H S, L D B$
REAL,D IM ENSION (:) ::D,E
REAL,D $\mathbb{M}$ ENSION (:,:) ::B

## C INTERFACE

\#include <sunperfh>
void sptts2 (intn, intnrhs, float *d, float*e, float *b,
intlab);
void sptts2_64 (long n, long nrhs, float*d, float *e, float *b, long ldb);

## PURPOSE

sptts2 solves a tridiagonal system of the form
A * $\mathrm{X}=\mathrm{B}$ using the $\mathrm{L} * \mathrm{D}$ *L 'factorization of A com puted by SPTTRF. D is a diagonalm atrix specified in the vectorD, $L$ is a unitbidiagonalm atrix w hose subdiagonal is specified in the vectorE, and $X$ and $B$ are $N$ by N RH S $m$ atrioes.

## ARGUMENTS

N (input) The order of the tridiagonalm atrix $\mathrm{A} . \mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS $>=0$.
$D$ (input) The $n$ diagonalelem ents of the diagonal $m$ atrix $D$ from the $L * D * L$ 'factorization ofA.

E (input) The ( $n-1$ ) subdiagonalelem ents of the unitbidiagonal factor $L$ from the $L * D * L$ 'factorization of A. E can also be regarded as the superdiagonal of the unit bidiagonal factor $U$ from the factorization $A$ $=U * D * U$.

B (input/output)
On entry, the righthand side vectors $B$ for the system of linear equations. On exit, the solution vectors, $X$.

LD B (input)
The leading dim ension of the array $B$. LD B $>=$ $\max (1, \mathbb{N})$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

srot-A pply a G iven s rotation constructed by SRO TG .

## SYNOPSIS



```
\mathbb{NTEGER N, INCX,INCY}
REALC,S
REALX (*),Y (*)
SU BROUT\mathbb{NE SROT_64 N,X,INCX,Y,INCY,C,S)}
INTEGER*8N,\mathbb{NCX,INCY}
REALC,S
REALX (*),Y (*)
```

F95 INTERFACE
SU BROUTINE ROT ( $\mathbb{N}$ ], X, $[\mathbb{N} C X], Y,[\mathbb{N} C Y], C, S)$
$\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y$
REAL ::C,S
REAL,D IM ENSION (:) :: X,Y
SU BROUTINEROT_64 (N ],X, [ $\mathbb{N} C X], Y,[\mathbb{N} C Y], C, S)$
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{N} C X, \mathbb{N} C Y$
REAL ::C,S
REAL,D IM ENSION (:) :: X,Y

## C INTERFACE

\#include <sunperfh>
void srot(intn, float * $x$, int incx, float * $y$, int incy, float c, float.s);
void srot 64 (long $n$, float *x, long incx, float *y, long incy, float c, floats);

## PURPOSE

srotA pply a G iven 5 rotation constructed by SRO TG .

## ARGUMENTS

$N$ (input)
On entry, $N$ specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.

X (input/output)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C X))$. Before entry, the increm ented amay $X$ must contain the vectorx.
U nchanged on exit.
$\mathbb{N} C X$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of $\mathrm{X} . \mathbb{N} C X$ m ustnotbe zero. U nchanged on exit.

Y (input/output)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))$. On entry, the increm ented array $Y$ m ust contain the vectory. On exit, $Y$ is overw ritten by the updated vectory.
$\mathbb{N} C Y$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $Y . \mathbb{N} C Y$ m ustnot.be zero. U nchanged on exit.

C (input) On entry, the C rotation value constructed by SRO TG. U nchanged on exit.
$S$ (input) On entry, the $S$ rotation value constructed by SRO TG. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
srotg -C onstruct a G iven S plane rotation
```


## SYNOPSIS

$$
\text { SU BROUTINE SROTG }(A, B, C, S)
$$

REALA,B,C,S
SUBROUTINE SROTG_64(A,B,C,S)
REALA,B,C,S
F95 INTERFACE
SUBROUTINEROTG (A, B , C, S)
REAL ::A,B,C,S
SU BROUTINEROTG_64(A,B,C,S)

REAL ::A,B,C,S
C INTERFACE
\#include <sunperfh>
void srotg (float *a, float *b, float *C, float*s);
void srotg_64 (float *a, float *b, float *c, float *s);

## PURPOSE

srotg C onstruct a G iven splane rotation that will annihilate an elem entof a vector.

## ARGUMENTS

A (input/output)
O n entry, A contains the entry in the firstvector that comesponds to the elem ent to be annihilated in the second vector. On exit, contains the nonzero elem ent of the rotated vector.

B (input/output)
On entry, $B$ contains the entry to be annihilated in the second vector. On exit, contains eithers or $1 / C$ depending on which of the inputvalues ofA and $B$ is larger.

C (output)
On exit, $C$ and $S$ are the elem ents of the rotation $m$ atrix thatw illlibe applied to anninilate $B$.

S (output)
See the description of C .

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS


## NAME

sroti- A pply an indexed $G$ ivens rotation.

## SYNOPSIS

```
SUBROUT\mathbb{NE SROTINZ,X,NNDX,Y,C,S)}
INTEGER NZ
INTEGER \mathbb{NDX (*)}
REALC,S
REALX (*),Y (*)
SUBROUTINE SROTI_64NZ,X,\mathbb{NDX,Y,C,S)}
INTEGER*8NZ
INTEGER*8 \mathbb{NDX (*)}
REALC,S
REALX (*),Y (*)
F95 IN TERFACE
SUBROUTINEROTI(NZ],X,\mathbb{NDX,Y,C,S)}
\mathbb{NTEGER ::NZ}
\mathbb{NTEGER,D IM ENSION (:)::\mathbb{NDX}}\mathbf{N}={
REAL::C,S
REAL,D IM ENSION (:) ::X,Y
```



```
INTEGER (8)::N Z
\mathbb{NTEGER (8),D IM ENSION (:) ::\mathbb{NDX}}\mathbf{N}=\mp@code{N}
REAL::C,S
REAL,D IM ENSION (:) ::X,Y
```

SRO T I -A pplies a G ivens rotation to a sparse vector $x$ stored in com pressed form and anothervectory in full storage form

```
do \(i=1, n\)
    tem \(p=-s^{*} x(i)+c^{*} y(\) indx (i) \()\)
    \(x\) (i) \(=c^{*} x(i)+s^{*} y(\) ind \((i))\)
    \(y(\) indx (i) \()=\) tem \(p\)
enddo
```


## ARGUMENTS

NZ (input) - $\mathbb{N}$ TEGER
$N$ um ber of elem ents in the com pressed form .
U nchanged on exit.
$X$ (input)
V ector containing the values of the com pressed form .
$\mathbb{N} D \mathrm{X}$ (input) - $\mathbb{N}$ TEGER
$V$ ector containing the indices of the com pressed
form. It is assum ed that the elem ents in $\mathbb{N} D \mathrm{X}$ are
distinctand greater than zero. U nchanged on exit.
Y (input/output)
V ectoron inputw hich contains the vectorY in full
storage form. On exit, only the elem ents
corresponding to the indices in $\mathbb{N}$ D X have been
m odified.

C (input)
Scalardefining the G ivens rotation
$S$ (input)
Scalardefining the G ivens rotation

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

srotm -A pply a G entlem an Sm odified G iven's rotation constructed by SRO TM G .

## SYNOPSIS



```
INTEGERN,\mathbb{NCX,\mathbb{NCY}}\mathbf{N}=\mp@code{N}
REALX (*),Y (*),PARAM (*)
SUBROUT\mathbb{NE SROTM _64 N,X,NNCX,Y,INCY,PARAM)}
INTEGER*8N,\mathbb{NCX,INCY}
REALX (*),Y (*),PARAM (*)
F95 INTERFACE
```



```
    \mathbb{NTEGER::N,\mathbb{NCX,}\mathbb{NCY}}\mathbf{T}\mathrm{ \}
    REAL,D IM ENSION (:) ::X,Y,PARAM
```




```
    REAL,D IM ENSION (:) ::X,Y,PARAM
```


## C INTERFACE

```
\#include <sunperfh>
void srotm (intn, float *x, int incx, float *y, int incy, float *param );
```

void srotm _64 (long $n$, float *x, long incx, float *y, long incy, float *param );

## PURPOSE

srotm A pply a G iven 5 rotation constructed by SRO TM G .

## ARGUMENTS

N (input)
O n entry, $N$ specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.
X (input/output)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C X))$. On entry, the increm ented array $X \mathrm{~m}$ ustcontain the vectorx. On exit, $X$ is overw rilten by the updated vector $x$.
$\mathbb{N} C X$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of $\mathrm{X} . \mathbb{N} C X$ m ustnotbe zero. U nchanged on exit.

Y (input/output)
( $1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C Y)$ ). On entry, the increm ented array $Y$ mustcontain the vectory. On exit, $Y$ is overw rilten by the updated vectory.
$\mathbb{N} C Y$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of Y. $\mathbb{N} C Y$ m ustnotbe zero. U nchanged on exit.

PARAM (input)
On entry, the rotation values constructed by SRO TM G. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

srotm g -C onstruct a Gentlem an's m odified G iven's plane rotation

## SYNOPSIS

SU BROUTINE SROTM G (D1,D 2, B1, B2, PARAM)
REAL D 1, D 2, B1,B2
REAL PARAM ( ${ }^{*}$ )

SU BROUTINE SROTM G_64 (D 1,D 2,B1,B2,PARAM)
REAL D 1, D 2, B1,B2
REAL PARAM ( ${ }^{*}$ )

## F95 INTERFACE

SU BROUTINE ROTM G (D1,D 2,B1,B2,PARAM)
REAL ::D 1, D 2, B1,B2
REAL,D IM ENSION (:) ::PARAM
SU BROUTINEROTM G_64D 1,D 2,B1,B2,PARAM)

REAL ::D 1,D 2,B1,B2
REAL,D $\mathbb{I}$ ENSION (:) ::PARAM

## C INTERFACE

\#include <sunperfh>
void srotm $g$ (float d1, floatd2, floatb1, float b2, float *param );
void srotm g_64 (floatd1, floatd2, floatb1, float.b2, float
*param );

## PURPOSE

srotm g C onstruct G entlem an Sm odified a G iven S plane rotation thatw illannihilate an elem entof a vector.

## ARGUMENTS

D 1 (input/output)
On entry, the first diagonal entry in the $H$ $m$ atrix. On exit, changed to reflect the effectof the transform ation.
D 2 (input/output)
On entry, the second diagonal entry in the $H$ $m$ atrix. On exit, changed to reflect the effect of the transform ation.

B1 (input/output)
O $n$ entry, the firstelem ent of the vector to which the $H$ matrix is applied. On exit, changed to reflect the effect of the transform ation.

B2 (input)
O $n$ entry, the second elem ent of the vector to which the H m atrix is applied. U nchanged on exit.

PARAM (output)
On exit, PARAM (1) describes the form of the rotation matrix $H$, and PARAM (2.5) contain the $H$ $m$ atrix.

IfPARAM (1) = -2 then $H=I$ and no elem ents of PARAM arem odified.

IfPARAM ( 1 ) = -1 then PARAM (2) $=$ h11, PARAM ( 3 ) = h21, PARAM (4) = h12, and PARAM (5) $=\mathrm{h} 22$.

IfPARAM $(1)=0$ then h11 = h22 = 1 , PARAM $(3)=$ $h 21$, and PARAM (4) $=$ h12.

IfPARAM $(1)=1$ then $\mathrm{h} 12=1, \mathrm{~h} 21=-1, \operatorname{PARAM}(2)=$ $h 11$, and PARAM (5) $=$ h22.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssbev - com pute all the eigenvalues and, optionally, eigenvectors of a realsym $m$ etric band $m$ atrix $A$

## SYNOPSIS

```
SUBROUT\mathbb{NE SSBEV (OOBZ,UPLO,N,KD,A,LDA,W ,Z,LD Z,W ORK, INFO)}
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGERN,KD,LDA,LD Z,INFO}
REALA (LDA,*),W (*),Z (LD Z,*),W ORK (*)
SUBROUTINE SSBEV_64 (JOBZ,UPLO,N,KD,A,LDA,W ,Z,LD Z,W ORK,
        \mathbb{NFO)}
CHARACTER * 1 OOBZ,UPLO
INTEGER*8N,KD,LDA,LD Z,INFO
REALA (LDA,*),W (*),Z (LD Z,*),W ORK (*)
```


## F95 INTERFACE

```
SU BROUTINE SBEV (JOBZ,UPLO, \(\mathbb{N}], K D, A,[L D A], W, Z,[L D Z],[W O R K]\), [ \(\mathbb{N}\) FO ])
```

CHARACTER (LEN=1)::JOBZ,UPLO
$\mathbb{N}$ TEGER ::N,KD,LDA,LD Z, $\mathbb{N} F O$
REAL,D IM ENSION (:) ::W ,W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) ::A, Z
SU BROUTINE SBEV_64 (JOBZ,UPLO, $\mathbb{N}], K D, A,[L D A], W, Z,[L D Z]$, [ $\mathrm{W} O \mathrm{RK}$ ], [ $\mathbb{N} F \mathrm{O}]$ )

CHARACTER (LEN=1):: JOBZ, UPLO
$\mathbb{N} T E G E R(8):: N, K D, L D A, L D Z, \mathbb{N F O}$

REAL,D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL,D IM ENSION (:,:) ::A , Z

## C INTERFACE

\#include <sunperfh>
void ssbev (char jंjbz, charuplo, intn, int kd, float *a, int lda, float * w , float *z, int ldz, int *info);
void ssbev_64 (char jobz, char uplo, long n, long kd, float
*a, long lda, float *w, float *z, long ldz, long *info);

## PURPOSE

ssbev com putes all the eigenvalues and, optionally, eigenvectors of a realsym $m$ etric band $m$ atrix $A$.

## ARGUMENTS

JO B Z (input)
$=\mathrm{N}$ ': C om pute eigenvalues only;
$=\mathrm{V}$ ': C om pute eigenvalues and eigenvectors.

UPLO (input)
$=\mathrm{U}$ ': Uppertriangle of A is stored;
= L ': Low er triangle ofA is stored.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

KD (input)
The num ber of superdiagonals of the $m$ atrix $A$ if $\mathrm{UPLO}=\mathrm{U}$ ', orthe num ber of subdiagonals ifUPLO
$=\mathbb{L} . \mathrm{KD}>=0$.

A (input/output)
O n entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstKD +1 row s of the array. The $j$ th colum n of A is stored in the $j$ th colum $n$ of the amay $A$ as follow $s$ : if $\mathrm{UPLO}=\mathrm{U}$ ', $\mathrm{A}(k d+1+i-j, j)=A(i, j)$ for $m a x(1, j$ $\mathrm{kd})<=i<=j$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{A}(1+i-j, j)=A(i, j)$ for $\dot{j}=i<=m$ in $(n, j+k d)$.

On exit, A is overw rilten by values generated during the reduction to tridiagonal form. IfUPLO = U ', the first superdiagonal and the diagonal of
the tridiagonal m atrix T are retumed in row SKD and $K D+1$ ofA, and if $U P L O=L$ ', the diagonaland first subdiagonal of T are retumed in the first tw o row sofA.

LD A (input)
The leading dim ension of the array A. LD A >= KD + 1.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.
$Z$ (input) If $J O B Z=V$ ', then if $\mathbb{N F O}=0, Z$ contains the orthonom aleigenvectors of the $m$ atrix $A, w$ ith the i-th colum $n$ of $Z$ holding the eigenvector associated w th $W$ (i). If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced.

LD $Z$ (input)
The leading dm ension of the array Z . LD $\mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD $Z>=\mathrm{max}(1, \mathrm{~N})$.

W ORK (w orkspace)
dim ension M A X ( $1,3 * \mathrm{~N}-2)$ )
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the i-th argum ent had an illegalvalue
> 0 : if $\mathbb{N} F O=$ i, the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssbevd - com pute all the eigenvalues and, optionally, eigenvectors of a realsym $m$ etric band $m$ atrix A

## SYNOPSIS

```
SUBROUT\mathbb{NE SSBEVD (OBZ,UPLO,N,KD,AB,LDAB,W,Z,LD Z,W ORK,}
    LW ORK,IN ORK,LIN ORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGER N,KD,LDAB,LD Z,LW ORK,LIN ORK,INFO}
INTEGER IN ORK (*)
REALAB (LDAB,*),W (*),Z (LD Z ,`),W ORK (*)
SU BROUT\mathbb{NE SSBEVD_64 (JOBZ,UPLO,N,KD,AB,LDAB,W ,Z,LD Z,W ORK,}
    LW ORK,IN ORK,LIN ORK,\mathbb{NFO)}
```

CHARACTER * 1 JOBZ, UPLO
$\mathbb{N}$ TEGER*8N,KD,LDAB,LD Z,LW ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I N}$ ORK (*)
REALAB (LDAB,*), W (*), Z (LD Z , $\left.{ }^{\star}\right), \mathrm{W} O R K\left({ }^{*}\right)$

## F95 INTERFACE

SU BROUTINE SBEVD (JOBZ, UPLO, $\mathbb{N}], K D, A B,[L D A B], W, Z,[L D Z],[W O R K]$, [LW ORK ], [ $\mathbb{W}$ ORK ], [LIN ORK ], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1)::JOBZ, UPLO
$\mathbb{N} T E G E R:: N, K D, L D A B, L D Z, L W O R K, L \mathbb{I N} O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{I N}$ ORK
REAL,D $\mathbb{I}$ ENSION (:) ::W,W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) ::AB,Z
SU BROUTINE SBEVD_64 (JOBZ,UPLO, $\mathbb{N}], K D, A B,[L D A B], W, Z,[L D Z]$,
[W ORK ], [LW ORK ], [IW ORK], [LIN ORK], [NFO])

CHARACTER (LEN=1)::JOBZ,UPLO
$\mathbb{N}$ TEGER (8) ::N,KD,LDAB,LDZ,LW ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I W}$ ORK
REAL,D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL,D $\mathbb{M}$ ENSION (:,:)::AB,Z

## C INTERFACE

\#include <sunperfh>
void ssbevd (char jobz, charuplo, intn, int kd, float *ab, int ldab, float *w , float *z, int ldz, int *info);
void ssbevd_64 (char j̀bz, charuplo, long n, long kd, float *ab, long ldab, float * ${ }_{\mathrm{w}}$, float * z , long ldz , long *info);

## PURPOSE

ssbevd com putes all the eigenvalues and, optionally, eigenvectors of a realsym $m$ etric band $m$ atrix $A$. If eigenvectors are desired, it uses a divide and conquer algorithm .

The divide and conquer algorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on $m$ achines w ith a guard digit in add/subtract, or on those binary $m$ achines $w$ thout guard digits w hich subtract like the $C$ ray X -M P , C ray Y M P , C ray C-90, orC ray-2. It could conceivably fail on hexadecim al or decim al $m$ achines $w$ thout guard digits, butw e know of none.

## ARGUMENTS

JOBZ (input)
= N ': C om pute eigenvalues only;
$=\mathrm{V}$ : C om pute eigenvalues and eigenvectors.

UPLO (input)
$=\mathrm{U}$ : U ppertriangle ofA is stored;
= LL': Low ertriangle ofA is stored.
N (input) The order of the matrix $\mathrm{A} . \mathrm{N}>=0$.

KD (input)
The num ber of superdiagonals of the $m$ atrix A if
UPLO $=\mathrm{U}$ ', or the num berof subdiagonals if UPLO
= L'. KD >= 0 。

AB (input/output)
O n entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstK $D+1$ row s of the array. The $j$ th colum n of A is stored in the $j$ th colum $n$ of the array $A B$ as follow $s$ : if $\mathrm{UPLO}=\mathrm{U}, \mathrm{AB}(\mathrm{kd}+1+i-j, j)=\mathrm{A}(i, j)$ for $\mathrm{max}(1, j$ $\mathrm{kd})<=\mathrm{i}<=\dot{j}$ ifUPLO $=\mathrm{L}, \mathrm{AB}(1+i-j, j)=A(i, j)$ for $j=i<=m$ in $(n, j+k d)$.

On exit, AB is overw rilten by values generated during the reduction to tridiagonal form. IfU PLO = U', the first superdiagonaland the diagonalof the tridiagonal m atrix T are retumed in row $\mathrm{SK} D$ and KD +1 of $A B$, and ifUPLO $=~ L '$ ', the diagonal and first subdiagonal of T are retumed in the first tw o row sofA B.

LDAB (input)
The leading dim ension of the array A B. LD A B >=KD +1 .

W (output)
If $\mathbb{N F} F=0$, the eigenvalues in ascending order.

Z (output)
If $\mathrm{JOBZ}=\mathrm{V}$ ', then if $\mathbb{N} F O=0, \mathrm{Z}$ contains the orthonorm aleigenvectors of the $m$ atrix $A, w$ th the i-th colum $n$ of $Z$ holding the eigenvector associated w th $W$ (i). If $\mathrm{OOBZ}=\mathrm{N}$ ', then Z is not referenced.

LD $Z$ (input)
The leading dim ension of the array $Z . L D Z>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD $Z>=\max (1, N)$.

W ORK (w orkspace)
dim ension (LW ORK)On exit, if $\mathbb{N F O}=0, W$ ORK (1)
retums the optim alLW ORK.
LW ORK (input)
The dim ension of the array W ORK. If $\mathrm{N}<=1$, LW ORK mustibe at least1. If 0 BZ $=\mathrm{N}$ 'and $\mathrm{N}>$
2, LW ORK m ustbe at least $2{ }^{*} \mathrm{~N}$. If $\mathrm{JO} \mathrm{BZ}=\mathrm{V}$ 'and
$\mathrm{N}>2$,LW ORK mustbe at least ( $1+5 \star \mathrm{~N}+2{ }^{\star} \mathrm{N} * * 2$
).

IfLW O RK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IN ORK (w orkspace/output)
On exit, if $\mathbb{N} F O=0, \mathbb{I N}$ ORK (1) retums the optim al LIV ORK.
LIV ORK (input)
The dim ension of the array $L \mathbb{I N} O R K$. If $\operatorname{JOBZ}=\mathrm{N}^{\prime}$ or $\mathrm{N}<=1$, LIN ORK mustbe at least1. If $\operatorname{JOBZ}=$ $V$ 'and $N>2$, LIW ORK mustbe at least $3+5{ }^{*} \mathrm{~N}$.

If LIV ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the $\mathbb{I W}$ ORK array, retums this value as the first entry of the $\mathbb{I V}$ ORK amay, and no errorm essage related to LIIN ORK is issued by X ERBLA.
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0$ : if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue
> 0 : if $\mathbb{N} F O=$ i, the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate
tridiagonal form did not converge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssbevx - com pute selected eigenvalues and, optionally, eigenvectors of a realsym $m$ etric band $m$ atrix A

## SYNOPSIS

```
SUBROUTINE SSBEVX (OBZ,RANGE,UPLO,N,KD,A,LDA,Q,LDQ,VL,
```



```
CHARACTER * 1 JOBZ,RANGE,UPLO
\mathbb{NTEGERN,KD,LDA,LDQ,\mathbb{I},\mathbb{U},NFOUND,LD Z,\mathbb{NFO}}\mathbf{N},\mp@code{L}
\mathbb{NTEGER IN ORK2 (*),\mathbb{FA LI (*)}}\mathbf{(})
REALVL,VU,ABTOL
REALA (LDA ,*),Q (LDQ ,*),W (*),Z (LD Z ,*),W ORK (*)
SU BROUT\mathbb{NE SSBEVX_64(0)BZ,RANGE,UPLO,N,KD,A,LDA,Q,LDQ,VL,}
```



```
CHARACTER * 1 JOBZ,RANGE,UPLO
\mathbb{NTEGER*8N,KD,LDA,LDQ,\mathbb{L},\mathbb{U},NFOUND,LD Z,\mathbb{NFO}}\mathbf{N},\mp@code{L}
NNTEGER*8 IN ORK2 (*), \mathbb{FA [H (*)}
REALVL,VU,ABTOL
REALA (LDA,*),Q (LDQ & *),W (*),Z (LD Z,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE SBEVX ( $\operatorname{OBB} \mathrm{Z}, \mathrm{RANGE}, \mathrm{UPLO}, \mathbb{N}], K D, A,[L D A], Q,[L D Q]$, VL,VU, $\mathbb{L}, \mathbb{U}, A B T O L, N F O U N D, W, Z,[L D Z],[W$ ORK ], [IW ORK2], $\mathbb{F A} \mathbb{L},[\mathbb{N F O}])$

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
$\mathbb{N}$ TEGER ::N,KD,LDA,LDQ, $\mathbb{L}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I}$ ENSION (:) :: $\mathbb{I W}$ ORK2, $\mathbb{F} A \mathbb{I}$
REAL ::VL,VU,ABTOL

REAL,D $\mathbb{I M} E N S I O N(:):: W, W O R K$
REAL,D $M$ ENSION (:,:) ::A, Q, Z

SU BROUTINE SBEVX_64 (JOBZ,RANGE,UPLO, $\mathbb{N}], K D, A,[L D A], Q,[L D Q]$, $\mathrm{VL}, \mathrm{VU}, \mathbb{I}, \mathbb{I U}, \mathrm{ABTOL}, \mathrm{NFOUND}, \mathrm{W}, \mathrm{Z},[\mathrm{LD} \mathrm{Z}],[\mathrm{W} O R K],[\mathbb{I} \mathrm{O} O \mathrm{R} 2]$, $\mathbb{F A} \mathbb{I}$, [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
$\mathbb{N}$ TEGER (8) :: $N, K D, L D A, L D Q, \mathbb{L}, \mathbb{Z}, N F O U N D, L D Z, \mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} E N S I O N(:):: \mathbb{I N} O R K 2, \mathbb{F} A \mathbb{I}$
REAL ::VL,VU,ABTOL
REAL,D $\mathbb{I}$ ENSION (:) ::W ,W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) ::A, Q, Z

## C INTERFACE

\#include <sunperfh>
void ssbevx (char jंbz, char range, charuple, intn, intkd, float *a, int lda, float *q, intldq, float vl, float vu, int il, int iu, float abtol, int *nfound, float *W , float * z , int $1 d z$, int *ifail, int*info);
void ssbevx_64 (char jंbz, char range, char uplo, long n, long kd, float *a, long lda, float *q, long ldq, float vl, float vu, long il, long iu, floatabtol, long *nfound, float *w, float *z, long ldz, long *ifail, long *info);

## PURPOSE

ssbevx com putes selected eigenvalues and, optionally, eigenvectors of a realsym $m$ etric band $m$ atrix A. Eigenvahues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}$ ': C om pute eigenvalues only;
$=V$ ': C om pute eigenvalues and eigenvectors.
RANGE (input)
= A ': alleigenvalues w ill.be found;
= V ':alleigenvalues in the half-open interval
( $\mathrm{NL}, \mathrm{VU}]$ w ill be found; = I': the $\mathbb{I}$-th through
IU -th eigenvaluesw illbe found.

UPLO (input)
= U ': U ppertriangle ofA is stored;
= LL': Low ertriangle ofA is stored.
N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.
KD (input)
The num ber of superdiagonals of the $m$ atrix $A$ if UPLO = U',orthe num berof subdiagonals ifU PLO $=\mathbb{L} \cdot \mathrm{KD}>=0$.

A (input/output)
On entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstKD +1 row s of the amay. The $j$ th colum $n$ of $A$ is stored in the $j$ th colum n of the amay $A$ as follow $s$ : if $\mathrm{UPLO}=\mathrm{U}^{\prime}, \mathrm{A}(\mathrm{kd}+1+\mathrm{i}-j, j)=A(i, j)$ for $\mathrm{max}(1, j$ $\mathrm{kd})<=i<=j$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{A}(1+i-j, j)=A(i, j)$ for $j<=i<=m$ in $(n, j+k d)$.

On exit, A is overw rilten by values generated during the reduction to tridiagonal form. If $\mathrm{PLO}=$ U ', the first superdiagonal and the diagonal of the tridiagonal m atrix T are retumed in row SKD and KD +1 of , and if $\mathrm{PLO}=\mathrm{L}$ ', the diagonal and first subdiagonal of T are retumed in the first two row sofA.

LD A (input)
The leading dim ension of the array A. LD A >=KD + 1.

Q (output)
If $\mathrm{JOBZ}=\mathrm{V}$ ', the $\mathrm{N}-$ by N orthogonal m atrix used in the reduction to tridiagonal form. If $\mathrm{JO} \mathrm{BZ}=$ N ', the array Q is not referenced.

LD Q (input)
The leading dim ension of the array $Q$. If $J 0 B Z=$ V ', then LD $Q>=\max (1, N)$.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU. N ot referenced ifRANGE = A 'or I'.

VU (input)
Se the description of V L .
II (input)

If RA N GE=I', the indiges (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{H}<=\mathbb{Z}<=N$, if $N>0$; $\mathbb{H}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N otreferenced ifRANGE $=$ A 'or V'.

IU (input)
See the description of II.

ABTOL (input)
The absolute emortolerance forthe eigenvalues.
A $n$ approxim ate eigenvalue is accepted as converged w hen itis determ ined to lie in an interval [a,b]
of w idth less than orequal to
$A B T O L+E P S * \max (a|, b|)$,
where EPS is them achine precision. If ABTOL is less than or equal to zero, then EPS*|I|will.be used in its place, w here $\mathrm{F} \mid$ is the 1 -norm of the tridiagonal m atrix obtained by reducing A to tridiagonalform .

E igenvalues w illbe com puted m ostaccurately when ABTOL is set to tw ice the underflow threshold 2*SLAMCH (S ), notzero. If this routine retums w ith $\mathbb{N}$ FO $>0$, indicating that som e eigenvectors did not converge, try setting ABTOL to 2*SLAM CH (S ).

See "C om puting Sm allSingularV alues of B idiagonal M atrices $w$ ith G uaranteed H igh Relative A ccuracy," by D em m eland K ahan, LAPA CK W orking N ote \#3.

NFOUND (output)
The total num ber of eigenvalues found. $0<=$
NFOUND <= N. IfRANGE = A', NFOUND = N, and if RANGE $=$ 'I', NFOUND $=\mathbb{I}-\mathbb{L}+1$ 。

W (output)
The first NFOUND elem ents contain the selected eigenvalues in ascending order.
$Z$ (input) If $\mathrm{OB} Z=V^{\prime}$, then if $\mathbb{N} F O=0$, the first $N F O U N D$ colum ns of $Z$ contain the orthonorm aleigenvectors of the matrix A corresponding to the selected eigenvalues, w ith the $i-$ th colum n of $Z$ holding the eigenvector associated w ith $W$ (i). If an eigenvector fails to converge, then that colum n of Z contains the latest approxim ation to the eigenvector, and the index of the eigenvector is retumed in

ㅍAII. If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced.
$N$ ote: the user must ensure that at least
$m$ ax ( $1, N F O U N D$ ) colum ns are supplied in the array $Z$;
if RANGE $=V$ ', the exactvalue ofNFOUND is not know $n$ in advance and an upperbound $m$ ustbe used.
LD Z (input)
The leading dim ension of the array Z. LD Z $>=1$, and if $\mathrm{OBB}=\mathrm{V}^{\prime}$, LD Z > = $\mathrm{max}(1, N)$.

W ORK (w orkspace)
dim ension $(7 \star \mathrm{~N})$

IW ORK 2 (w orkspace)

FAII (output)
If $\mathrm{OBZ}=\mathrm{V}^{\prime}$ ', then if $\mathbb{N F O}=0$, the first NFOUND
elements of $\mathbb{F A} I L$ are zero. If $\mathbb{N F O}>0$, then
IFA II contains the indices of the eigenvectors
that failed to converge. If $\mathrm{OBZ}=\mathrm{N}$ ', then
$\mathbb{F} A \mathbb{I}$ is notreferenced.
$\mathbb{N}$ FO (output)
= 0: successfulexit.
$<0$ : if $\mathbb{N} F O=-i$, the i-th argum enthad an illegalvalue.
$>0:$ if $\mathbb{N} F O=i$, then ieigenvectors failed to converge. Their indiges are stored in array正AII.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssogst-reduce a realsym $m$ etric-definite banded generalized eigenproblem $A * x=\operatorname{lam}$ bda*B*x to standard form $C * y=$ lam bda*y,

## SYNOPSIS

```
SUBROUT\mathbb{NE SSBGST NECT,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,X,LDX,}
    W ORK,INFO)
CHARACTER * 1 VECT,UPLO
\mathbb{NTEGER N,KA,KB,LDAB,LDBB,LDX,}\mathbb{N}FO
REALAB (LDAB,*),BB (LDBB,*),X (LDX,*),W ORK (*)
SU BROUT\mathbb{NE SSBGST_64 NECT,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,X,}
    LDX,W ORK,\mathbb{NFO)}
```

CHARACTER * 1 VECT, UPLO
$\mathbb{N} T E G E R * 8 N, K A, K B, L D A B, L D B B, L D X, \mathbb{N} F O$
REALAB (LDAB,*), BB (LDBB,*), X (LDX,*),W ORK (*)

## F95 INTERFACE

SU BROUTINE SBGST $N E C T, U P L O, \mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B], X$, [LDX], [W ORK], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) ::VECT,UPLO
$\mathbb{N} T E G E R:: N, K A, K B, L D A B, L D B B, L D X, \mathbb{N F O}$
REAL,D $\mathbb{M}$ ENSION (:) ::W ORK
REAL,D IM ENSION (: : : : : AB, BB, X

SU BROUTINE SBGST_64 NECT,UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B]$, $\mathrm{X},[\mathrm{LD} \mathrm{X}],[\mathrm{W}$ ORK ], [ $\mathbb{N} F \mathrm{O}])$

CHARACTER (LEN=1) ::VECT,UPLO
$\mathbb{N}$ TEGER (8) ::N, KA, KB,LDAB,LDBB,LDX, $\mathbb{N} F O$
REAL,D $\mathbb{I M}$ ENSION (:) ::W ORK
REAL,D $\mathbb{M}$ ENSION (: : : : : AB, BB, X

## C INTERFACE

\#include <sunperfh>
void ssbogst(charvect, charuplo, intn, int ka, int kb, float *ab, int ldab, float *bb, int ldbb, float * $x$, int ldx $x$, int *info);
void ssbgst_64 (charvect, charuplo, long n, long ka, long kb, float *ab, long ldab, float *bb, long ldbb, float *x, long ldx, long *info);

## PURPOSE

ssogst reduces a real sym $m$ etric-definite banded generalized eigenproblem $A * x=\operatorname{lam}$ bda*B*x to standard form $C * y=$ lam bda*y, such thatC has the sam e bandw idth asA .

B m usthave been previously factorized as $S * * T$ *S by SPBSTF, using a split Cholesky factorization. A is overw ritten by C $=\mathrm{X} * * \mathrm{~T} * \mathrm{~A} * \mathrm{X}$, where $\mathrm{X}=\mathrm{S} * *(-1) * \mathrm{Q}$ and Q is an orthogonal $m$ atrix chosen to preserve the bandw idth ofA.

## ARGUMENTS

## VECT (input)

$=N$ ': do not form the transform ation $m$ atrix $X$;
$=\mathrm{V}$ ': form X .

UPLO (input)
= U ': U pper triangle ofA is stored;
= L': Low er triangle ofA is stored.

N (input) The order of the m atrices A and $\mathrm{B} . \mathrm{N}>=0$.

KA (input)
The num ber of superdiagonals of the $m$ atrix A if UPLO $=\mathrm{U}$ ', ort the num berof subdiagonals if UPLO $=\mathbb{L}^{\prime} . \mathrm{KA}>=0$.

KB (input)
The num ber of superdiagonals of the $m$ atrix $B$ if
U PLO = U',orthe num berof subdiagonals ifU PLO
$=L^{\prime} . K A>=K B>=0$.

AB (input/output)
O $n$ entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstka+1 row s of the amay. The $j$ th colum $n$ of $A$ is stored in the $j$ th colum $n$ of the amay $A B$ as follows: if $\mathrm{UPLO}=\mathrm{U}, \mathrm{AB}(k a+1+i-j)=A(i, j)$ for $m a x(1, j$ $\mathrm{ka})<=\mathrm{i}<=j$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{AB}(1+i-j, j)=A(i, j)$ for $j<=i<=m$ in $(n, j+k a)$.

On exit, the transform ed matrix $X * * T * A * X$, stored in the sam e form at as $A$.

LD AB (input)
The leading dim ension of the array $A B$. LD AB >= KA+1.

BB (input)
The banded factors from the split Cholesky factorization ofB, as retumed by SPBSTF, stored in the first $K B+1$ row sof the array.

LD BB (input)
The leading dim ension of the array BB. LD BB >= K B+1.

X (output)
IfVECT = $V$ ', the $n-b y-n m$ atrix $X$. If VECT $=$ N ', the array X is not referenced.

LD X (input)
The leading dim ension of the aray X . LD X >= $\max (1, N)$ if VECT $=V$ ';LD X >= 1 otherw ise.

W ORK (w orkspace)
dim ension $(2 * N)$
$\mathbb{N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{N}$ FO = -i, the i-th argum ent had an illegalvahue.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssogv -com pute all the eigenvalues, and optionally, the eigenvectors of a real generalized sym $m$ etric-definite banded eigenproblem, of the form $A * x=(l a m . b d a) * B * x$

## SYNOPSIS

```
SU BROUT\mathbb{NE SSBGV (OOBZ,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,W,Z,}
    LD Z,W ORK,\mathbb{NFO)}
```

CHARACTER * 1 JOBZ, UPLO
$\mathbb{N}$ TEGER $N, K A, K B, L D A B, L D B B, L D Z, \mathbb{N} F O$
REALAB (LDAB, $\left.{ }^{\star}\right)$, BB ( $\left.\mathbb{L} D B B, \star\right), W(\star), Z(L D Z, \star), W O R K(*)$
SU BROUTINE SSBGV_64 (JOBZ,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,W,Z,
LD $\mathrm{Z}, \mathrm{W}$ ORK, $\mathbb{N} F \mathrm{O}$ )
CHARACTER * 1 JOBZ, UPLO
$\mathbb{N} T E G E R * 8 N, K A, K B, L D A B, L D B B, L D Z, \mathbb{N} F O$
REALAB (LDAB,*),BB(LDBB,*), W ( $\left.{ }^{\star}\right), \mathrm{Z}(\mathrm{LD} Z, \star), \mathrm{W} O R K(*)$

## F95 INTERFACE

SUBROUTINE SBGV (OBZ,UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B], W$, Z, [LD Z], [W ORK], [NFO])

CHARACTER (LEN=1):: JOBZ, UPLO
$\mathbb{N} T E G E R:: N, K A, K B, L D A B, L D B B, L D Z, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL,D $\mathbb{M}$ ENSION (: : : : : AB, BB, Z

SU BROUTINE SBGV_64 (JOBZ, UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B]$, W , Z, [LD Z], [W ORK ], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1)::JOBZ,UPLO
$\mathbb{N}$ TEGER (8) ::N, KA, KB,LDAB,LDBB,LD Z , $\mathbb{N}$ FO
REAL,D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) ::AB,BB,Z

## C INTERFACE

\#include <sunperfh>
void ssogv (char jंbz, charuplo, int n, int ka, int kb, float *ab, int ldab, float *bb, int ldbb, float * ${ }_{\mathrm{w}}$, float *z, int ldz, int *info);
void ssbgv_64 (char jobz, charuplo, long n, long ka, long kb, float *ab, long ldab, float *bb, long ldbb, float *w, float *z, long ldz, long *info);

## PURPOSE

ssogv com putes all the eigenvalues, and optionally, the eigenvectors of a realgeneralized sym $m$ etric-definite banded eigenproblem, of the form $A$ *x $=(\operatorname{lam} . b d a) * B * x$. H ere $A$ and $B$ are assum ed to be sym $m$ etric and banded, and $B$ is also positive definite.

## ARGUMENTS

$J 0 \mathrm{BZ}$ (input)
= N ': C om pute eigenvalues only;
$=\mathrm{V}$ : C om pute eigenvalues and eigenvectors.

UPLO (input)
= U ': U ppertriangles of $A$ and $B$ are stored;
= L': Low ertriangles of A and B are stored.

N (input) The order of the m atrices A and $\mathrm{B} . \mathrm{N}>=0$.

KA (input)
The num ber of superdiagonals of the $m$ atrix A if UPLO $=U$ ', orthe num berof subdiagonals if UPLO $=\mathrm{L} \cdot \mathrm{KA}>=0$.

K B (input)
The num ber of superdiagonals of the $m$ atrix $B$ if UPLO = U',orthe num berof subdiagonals ifU PLO = L' $\cdot \mathrm{KB}>=0$.

A B (input/output)

O $n$ entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstka+1 row s of the array. The $j$ th colum n of $A$ is stored in the $j$ th colum $n$ of the array AB as follow s: if UPLO = U', AB $(k a+1+i-j)=A(i, j)$ for max $(1, j$ $\mathrm{ka})<=i<=j ;$ ifUPLO $=\mathrm{L}, \mathrm{AB}(1+i-j)=A(i, j)$ for $\dot{j}=i<=m$ in $(n, j+k a)$.

On exit, the contents of $A$ are destroyed.

LDAB (input)
The leading dim ension of the array AB. LDAB >= KA+1.
BB (input/output)
On entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $B$, stored in the first kb+1 row sof the array. The $j$ th colum $n$ of $B$ is stored in the $j$ th colum $n$ of the amay BB as follow s: if UPLO $=U$ ', BB $(k b+1+i-j, j)=B(i, j)$ for $m a x(1, j$ $\mathrm{kb})<=\mathrm{i}<=\dot{j}$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{BB}(1+i-j, j)=B(i, 7)$ for $\dot{j}=\mathrm{i}<=\mathrm{m}$ in $(n, j+k b)$.

On exit, the factors from the splitCholesky factorization $B=S * * T * S$, as retumed by SPBSTF .

LD BB (input)
The leading dim ension of the array BB. LD BB >= K B+1.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.
$Z$ (input) If $J O B Z=V '$, then if $\mathbb{N} F O=0, Z$ contains the $m$ atrix $Z$ ofeigenvectors, $w$ ith the $i$-th colum n of Z holding the eigenvector associated with W (i). The eigenvectors are norm alized so that $\mathrm{Z} * * \mathrm{~T} * \mathrm{~B} * \mathrm{Z}=$ I. If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced.

LD Z (input)
The leading din ension of the amray $\mathrm{Z} . \operatorname{LD} \mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD Z >=N.

W ORK (w orkspace)
dim ension ( $3 \star \mathrm{~N}$ )
$\mathbb{N} F O$ (output)
= 0 : successfulexit
< 0: if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an illegalvalue
$>0$ : if $\mathbb{N F O}=\mathrm{i}$, and iis:
<= N : the algorithm failed to converge: i offdiagonal elem ents of an interm ediate tridiagonal form did not converge to zero; > N : if $\mathbb{N} F O=N$ +i , for $1<=\mathrm{i}<=\mathrm{N}$, then SPBSTF retumed $\mathbb{N} F O=i: B$ is not positive definite. The factorization ofB could notbe com pleted and no eigenvahues oreigenvectors w ere com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

ssogvd - com pute all the eigenvalues, and optionally, the eigenvectors of a real generalized sym $m$ etric-definite banded eigenproblem, of the form $A * x=(l a m . b d a) * B * x$

## SYNOPSIS

```
SUBROUT\mathbb{NE SSBGVD(OBZ,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,W,Z,}
```



```
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGER N,KA,KB,LDAB,LDBB,LD Z,LW ORK,LIN ORK, INFO}
INTEGER IN ORK (*)
REALAB (LDAB,*),BB (LDBB,*),W (*),Z (LD Z,*),W ORK (*)
SU BROUT\mathbb{NE SSBGVD_64(JOBZ,UPLO ,N,KA,KB,AB,LDAB,BB,LDBB,W ,Z,}
```



CHARACTER * 1 JOBZ, UPLO
$\mathbb{N} T E G E R * 8 N, K A, K B, L D A B, L D B B, L D Z, L W O R K, L \mathbb{I N} O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I N}$ ORK ( ${ }^{*}$ )
REALAB (LDAB,*), BB (LDBB,*),W (*), Z (LDZ,*),W ORK (*)

## F95 INTERFACE

SUBROUTINE SBGVD (OBZ, UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B], W$, Z, [LD Z], [W ORK ], [LW ORK], [IW ORK ], [LIN ORK], [NFO])

CHARACTER (LEN=1): : JOBZ, UPLO
$\mathbb{N} T E G E R:: N, K A, K B, L D A B, L D B B, L D Z, L W O R K, L \mathbb{I N} O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{I N}$ ORK
REAL,D $\mathbb{I M}$ ENSION (:) ::W,W ORK
REAL,D IM ENSION (:,:) ::AB,BB,Z

SU BROUTINE SBGVD_64 (OOBZ, UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B]$, W, Z, [LD Z], [W ORK ], [LW ORK], [ $\mathbb{W}$ ORK ], [LINORK], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) :: JOBZ,UPLO
$\mathbb{N} T E G E R(8):: N, K A, K B, L D A B, L D B B, L D Z, L W O R K, L \mathbb{N} O R K$, $\mathbb{N}$ FO
$\mathbb{N}$ TEGER (8),D $\mathbb{I M}$ ENSION (:) :: IN ORK
REAL,D $\mathbb{I}$ ENSION (:) ::W ,W ORK
REAL,D $\mathbb{M}$ ENSION (: : : : ::AB,BB,Z

## C INTERFACE

\#include <sunperfh>
void ssogvd (char jobz, charuplo, intn, int ka, int kb, float *ab, int ldab, float *bb, int ldbb, float *W , float * $z$, int ldz, int *info);
void ssbgvd_64 (char jobz, char uplo, long n, long ka, long kb, float *ab, long ldab, float *bb, long ldbb, float *W , float * z , long ldz, long *info);

## PURPOSE

ssogvd com putes all the eigenvalues, and optionally, the eigenvectors of a realgeneralized sym $m$ etric-definite banded eigenproblem, of the form $A * x=(\operatorname{lam} b d a) * B * x$. Here $A$ and $B$ are assum ed to be sym $m$ etric and banded, and $B$ is also positive definite. If eigenvectors are desired, it uses a divide and conquer algorithm .

The divide and conqueralgorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on $m$ achines w ith a guard digit in add/subtract, or on those binary $m$ achines $w$ thout guard digits w hich subtract like the $C$ ray $X-M P, C$ ray $Y \neq M P, C$ ray $C-90$, or C ray-2. It could conceivably fail on hexadecim al or decim al machines w ithout guard digits, butw e know of none.

## ARGUMENTS

```
JOBZ (input)
\(=\mathrm{N}^{\prime}:\) C om pute eigenvalues only;
\(=\mathrm{V}\) : C om pute eigenvalues and eigenvectors.
```

UPLO (input)
$=\mathrm{U}$ : : U pper triangles ofA and B are stored;
$=L^{\prime}:$ Low ertriangles of $A$ and $B$ are stored.

N (input) The order of the matriges $A$ and $B . N>=0$.

KA (input)
The num ber of superdiagonals of the $m$ atrix $A$ if $\mathrm{UPLO}=\mathrm{U}$ ', orthe num berof subdiagonals ifUPLO $=\mathbb{L}$ '. KA > $=0$ 。

K B (input)
The num ber of superdiagonals of the $m$ atrix $B$ if $\mathrm{UPLO}=\mathrm{U}$ ', orthe num ber of subdiagonals ifU PLO $=\mathbb{L}$ 'KB >=0.

A B (input/output)
O n entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstka+1 row s of the anray. The $j$ th colum n of A is stored in the $j$ th colum $n$ of the array A B as follow s: if $\mathrm{UPLO}=\mathrm{U}, \mathrm{AB}(\mathrm{ka}+1+i-j)=A(i, j)$ for $m a x(1, j$ $\mathrm{ka})<=\mathrm{i}<=j$ if $\mathrm{UPLO}=\mathrm{L}$ ', AB $(1+i-j, j)=A(i, j)$ for $\dot{j}=i<=m$ in $(n, j+k a)$.

O n exit, the contents ofA B are destroyed.

LDAB (input)
The leading dim ension of the array AB. LDAB >= K A +1 .

BB (input/output)
O n entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $B$, stored in the first $k b+1$ row s of the amay. The $j$ th colum n of $B$ is stored in the $j$ th colum $n$ of the array $B B$ as follow $s$ : if $\mathrm{UPLO}=\mathrm{U}$ ', $\mathrm{BB}(\mathrm{ka}+1+i-j)=\mathrm{B}(i, j)$ for $\max (1, j$ $\mathrm{kb})<=\dot{i}<=\dot{j}$ ifU PLO $=\mathrm{L}, \mathrm{BB}(1+i-j, j=\mathrm{j}(i, 7)$ for $j<=i<=m$ in $(n, j+k b)$.

On exit, the factors from the split Cholesky factorization $\mathrm{B}=\mathrm{S} * * \mathrm{~T} * \mathrm{~S}$, as retumed by SPBSTF .

LD BB (input)
The leading dim ension of the array BB. LD BB $>=$ K B+1.

W (output)
If $\mathbb{N}$ FO $=0$, the eigenvalues in ascending order.

Z (input) If $\mathrm{OB} \mathrm{B}=\mathrm{V}$ ', then if $\mathbb{N} F O=0, \mathrm{Z}$ contains the $m$ atrix $Z$ of eigenvectors, $w$ ith the $i$-th colum n of
$Z$ holding the eigenvector associated $w$ ith $W$ (i). The eigenvectors are norm alized so $\mathrm{Z} * * \mathrm{~T} * \mathrm{~B} * \mathrm{Z}=\mathrm{I}$. If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced.

LD $Z$ (input)
The leading dm ension of the array Z . LD $\mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD Z $>=\mathrm{max}(1, \mathrm{~N})$.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array W ORK. If $\mathrm{N}<=1$, LW ORK >= 1. If JOBZ $=N$ 'and $N>1$,LW ORK $>=$ 3*N. If $\operatorname{OOBZ}=\mathrm{V}$ 'and $\mathrm{N}>1$,LW ORK >=1 + 5*N + $2 * N * * 2$ 。

If LW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IV ORK (w orkspace/output)
On exit, if LIN ORK > 0, $\mathbb{I N}$ ORK (1) retums the optim all IV ORK.

LIV ORK (input)
The dim ension of the array $\mathbb{I N} O R K$. If $0 \mathrm{OBZ}=\mathrm{N}^{\prime}$ orN <=1,L $\mathbb{I N} O R K>=1$. If $J O B Z=V$ 'and $N>1$, LIN ORK >= $3+5 * N$.

If LIV ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the $\mathbb{I W}$ ORK array, retums this value as the first entry of the $\mathbb{I W}$ ORK array, and no errorm essage related to $L \mathbb{I N} O R K$ is issued by X ERBLA.
$\mathbb{I N F O}$ (output)
= 0 : successfinlexit
<0: if $\mathbb{N N}$ FO = -i, the i-th argum ent had an illegalvalue
> 0 : if $\mathbb{N F O}=\mathrm{i}$, and is:
$<=\mathrm{N}$ : the algorithm failed to converge: i offdiagonal elem ents of an interm ediate tridiagonal form did not converge to zero; > N : if $\mathbb{N} F O=\mathrm{N}$ $+i$, for $1<=i<=N$, then SPBSTF
retumed $\mathbb{N} F O=i: B$ is not positive definite. The factorization ofB could notbe com pleted and
no eigenvalues oreigenvectors w ere com puted.

## FURTHER DETAILS

B ased on contributions by
$M$ ark Fahey,D epartm entofM athem atics, U niv. ofK entucky, U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

ssogvx - com pute selected eigenvalues, and optionally, eigenvectors of a real generalized sym $m$ etric-definite banded eigenproblem, of the form $A * x=(l a m ~ b d a) * B * x$

## SYNOPSIS

```
SUBROUT\mathbb{NE SSBGVX (OBBZ,RANGE,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,}
```



```
    \mathbb{NFO)}
```

CHARACTER * 1 JOBZ,RANGE,UPLO
$\mathbb{N}$ TEGER N,KA,KB,LDAB,LDBB,LDQ, $\mathbb{L}, \mathbb{U}, M, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R \mathbb{I} \operatorname{ORK}\left(^{*}\right)$, $\mathbb{F A} \mathbb{I}$ (*)
REALVL,VU,ABSTOL
REAL AB (LDAB,*), BB (LDBB,*), Q (LDQ,*), W (*), Z (LDZ,*),
W ORK (*)
SU BROUTINE SSBGVX_64 (JOBZ,RANGE,UPLO,N,KA,KB,AB,LDAB,BB,
LDBB, $\mathrm{Q}, \mathrm{LD} Q, V L, V U, \mathbb{I}, \mathbb{U}, A B S T O L, M, W, Z, L D Z, W$ ORK, IW ORK,
$\mathbb{F A} \mathbb{L}, \mathbb{N} F O$ )
CHARACTER * 1 JOBZ,RANGE,UPLO
$\mathbb{N}$ TEGER*8N,KA,KB,LDAB,LDBB,LDQ, $\mathbb{L}, \mathbb{U}, M, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I N}$ ORK $(*), \mathbb{F A} \mathbb{L}(*)$
REALVL,VU,ABSTOL
$\operatorname{REALAB}\left(L D A B,^{\star}\right), B B(L D B B, \star), Q\left(\mathbb{L D} Q,^{\star}\right), W\left({ }^{*}\right), Z(L D Z, \star)$,
W ORK (*)

## F95 INTERFACE

SU BROUTINE SBGVX (OBZ,RANGE,UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B$,

[ $\mathbb{N}$ ORK $], \mathbb{F A} \mathbb{I},[\mathbb{N} F O])$

CHARACTER (LEN=1) :: JOBZ,RANGE, UPLO
$\mathbb{N} T E G E R:: N, K A, K B, L D A B, L D B B, L D Q, \Pi, \mathbb{U}, M, L D Z, \mathbb{N} F O$
$\mathbb{N}$ TEGER,D $\mathbb{M} E N S I O N(:):: \mathbb{I} O R K, \mathbb{F} A \mathbb{L}$
REAL ::VL,VU,ABSTOL
REAL,D $\mathbb{I M}$ ENSION (:) ::W ,W ORK
REAL,D $\mathbb{I M}$ ENSION (:,:) ::AB,BB,Q,Z

SU BROUTINE SBGVX_64 (OBZ,RANGE,UPLO, $\mathbb{N}], K A, K B, A B,[L D A B], B B$, $[[\mathrm{D} B \mathrm{~B}], \mathrm{Q},[\mathrm{LD} Q], \mathrm{VL}, \mathrm{VU}, \mathbb{Z}, \mathbb{U}, \mathrm{ABSTOL}, \mathrm{M}, \mathrm{W}, \mathrm{Z},[\mathrm{LD} \mathrm{Z}],[\mathrm{W} O R K]$,
[ $\mathbb{I N}$ ORK], $\mathbb{F A} \mathbb{I},[\mathbb{N} F O]$ )

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{KA}, \mathrm{KB}, \mathrm{LD} A B, L D B B, L D Q, \mathbb{L}, \mathbb{U}, M, L D Z$,
$\mathbb{N} \mathrm{FO}$
$\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N} O R K, \mathbb{F A} \mathbb{I}$
REAL ::VL,VU,ABSTOL
REAL,D $\mathbb{I M}$ ENSION (:) ::W ,W ORK
REAL,D $\mathbb{I}$ ENSION (:,:) ::AB,BB, Q, Z

## C INTERFACE

\#include <sunperfh>
void ssbgvx (char j̀jbz, char range, char uple, intn, int ka, int kb, float *ab, int ldab, float *bb, int ldbb, float *q, int ldq, floatvl, floatvu, int il, int iu, float abstol, int *m, float * w , float * z , int ldz, int*ifail, int*info);
void ssogvx_64 (char jobz, char range, char uplo, long n, long ka, long kb, float *ab, long ldab, float *bb, long ldbb, float *q, long ldq, floatvl, floatvu, long il, long iu, float abstol, long *m , float *W , float *z, long ldz, long *ifail, long *info);

## PURPOSE

ssbgvx com putes selected eigenvalues, and optionally, eigenvectors of a real generalized symm etric-definite banded eigenproblem, of the form $A{ }^{*} x=(l a m . b d a){ }^{\star} B{ }^{*} x . H$ ere $A$ and $B$ are assum ed to be sym $m$ etric and banded, and $B$ is also positive definite. Eigenvahues and eigenvectors can be selected by specifying either alleigenvalues, a range of values or a range of indices for the desired eigenvalues.

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}$ ': C om pute eigenvahues only;
$=\mathrm{V}$ : C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvalues willbe found.
= V ': alleigenvalues in the half-open interval $\mathrm{NL}, \mathrm{VU}]$ will be found. = I ': the L -th through $\mathbb{I U}$-th eigenvaluesw illlbe found.

UPLO (input)
$=\mathrm{U}$ ': U ppertriangles of $A$ and $B$ are stored;
= L': Low ertriangles of A and B are stored.

N (input) The order of the matrioes A and B. $\mathrm{N}>=0$.
KA (input)
The num ber of superdiagonals of the $m$ atrix $A$ if UPLO $=U$ ', or the num berof subdiagonals ifUPLO
$=\mathrm{L} \cdot \mathrm{KA}>=0$.

KB (input)
The num ber of superdiagonals of the $m$ atrix $B$ if UPLO $=\mathrm{U}$ ', or the num berof subdiagonals if P PLO $=\mathrm{L}^{\prime} . \mathrm{KB}>=0$.

AB (input/output)
O $n$ entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstka+1 row sof the amay. The $j$ th colum $n$ of $A$ is stored in the jth colum n of the anray AB as follows: if $\mathrm{UPLO}=\mathrm{U}$ ', AB $(k a+1+i-j)=A(i, j)$ for $m a x(1, j$ $\mathrm{ka})<=i<=j ;$ ifUPLO $=\mathrm{L} \prime, \mathrm{AB}(1+i-j, j)=A(i, 7)$ for $j=i<=m$ in $(n, j+k a)$.

On exit, the contents of $A B$ are destroyed.
LDAB (input)
The leading dim ension of the array A B . LD A B >= KA+1.

BB (input/output)
O $n$ entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $B$, stored in the first kb +1 row sof the array. The $j$ th colum n of $B$ is stored in the $j$ th colum $n$ of the array $B B$ as follow s: if $\mathrm{UPLO}=\mathrm{U}, \mathrm{BB}(k a+1+i-j)=\mathrm{B}(i, 7)$ for $\max (1, j$ $\mathrm{kb})<=\mathrm{i}<=\dot{j}$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{BB}(1+i-j, j)=\mathrm{B}(i, 7)$ for $j=i<=m$ in $(n, j+k b)$.

On exit, the factors from the split Cholesky factorization $B=S * * T * S$, as retumed by SPBSTF .

LD BB (input)
The leading dim ension of the array BB. LD BB >= K B+1.

Q (output)
If $J O B Z=V$ ', the $n-b y-n$ matrix used in the reduction of $A *=\left(l a m\right.$ bda) ${ }^{\mathrm{B}} \mathrm{K}_{\mathrm{x}} \mathrm{x}$ to standard form, i.e. $C{ }^{*} \mathrm{X}=\left(\mathrm{lam}\right.$ bda) ${ }^{\star} \mathrm{X}$, and consequently C to tridiagonal form. If JOBZ $=N$ ', the amay $Q$ is not referenced.

LD Q (input)
The leading dim ension of the array $Q$. If $J 0 B Z=$ $N^{\prime}, L D Q>=1$. If $J O B Z=V{ }^{\prime}, L D Q>=m a x(1, N)$.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE=A 'or I'.

VU (input)
See the description of V L .
II (input)
IfRA N G E= 'I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{Z}<=\mathbb{U}<=N$, if $N>0$; $\mathbb{H}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N ot referenced ifRANGE $=$ A'or V'.

IU (input)
See the description of $\mathbb{I L}$.
ABSTOL (input)
The absolute error tolerance for the eigenvalues.
A $n$ approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b]
of w idth less than orequal to
ABSTOL + EPS * $\max (|,||$,$) ,$
where EPS is the m achine precision. IfA BSTOL is less than or equal to zero, then EPS* $\mid$ |w illbe used in its place, where $F \mid$ is the 1 -norm of the tridiagonal $m$ atrix obtained by reducing $A$ to tridiagonal form .

E igenvalues w illle com puted m ostaccurately when ABSTOL is set to tw ice the underflow threshold 2*SLAM CH (S ), not zero. If this routine retums w th $\mathbb{N} F O>0$, indicating that som e eigenvectors did not converge, try setting ABSTO L to $2 \star$ SLAM CH (S ).

M (output)
The total num ber ofeigenvalues found. $0<=\mathrm{M}$ <= N . IfRANGE $=\mathrm{A}^{\prime}, \mathrm{M}=\mathrm{N}$, and ifRANGE $=\mathrm{I}^{\prime}, \mathrm{M}=$ $\mathbb{U}-\mathbb{L}+1$.

W (output)
If $\mathbb{N}$ FO $=0$, the eigenvalues in ascending order.
$Z$ (input) If $J O B Z=V '$, then if $\mathbb{N} F O=0, Z$ contains the $m$ atrix $Z$ ofeigenvectors, $w$ ith the $i$-th colum $n$ of $Z$ holding the eigenvector associated $w$ ith $W$ (i). The eigenvectors are norm alized so $\mathrm{Z} * * \mathrm{~T} * \mathrm{~B} * \mathrm{Z}=\mathrm{I}$. If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced.

LD Z (input)
The leading dm ension of the array Z . LD $\mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD $Z>=\mathrm{max}(1, \mathrm{~N})$.

W ORK (w orkspace)
dim ension ( $7 *_{\mathrm{N}}$ )

IV ORK (w orkspace/output)
dim ension ( $5 * \mathrm{~N}$ )
FAII (output)
If $\mathrm{JOBZ}=\mathrm{V}$ ', then if $\mathbb{N F O}=0$, the firstM ele$m$ ents of $\mathbb{F A} \mathbb{I}$ are zero. If $\mathbb{N} F O>0$, then $\mathbb{F} A \mathbb{I}$ contains the indioes of the eigenvalues that failed to converge. If $\mathrm{JOBZ}=\mathrm{N}$ ', then $\mathbb{F A} \mathbb{I}$ is not referenced.
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum ent had an illegalvalue
<= N : if $\mathbb{N} F O=i$, then ieigenvectors failed to converge. Their indioes are stored in $\mathbb{F A} \mathbb{I} .>N$
: SPBSTF retumed an emorcode; ie., if $\mathbb{N} F O=N$
$+i$, for $1<=i<=N$, then the leading $m$ inor of orderiofB is notpositive definite. The factorization of B could not be com pleted and no eigenvalues or eigenvectors w ere com puted.

## FURTHER DETAILS

## B ased on contributions by

M ark Fahey, D epartm entofM athem atics, U nìv. of K entucky, USA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssom $v$-perform them atrix-vector operation $y:=a l p h a * A * x$

+ beta*y


## SYNOPSIS

```
SUBROUTINE SSBMV (UPLO,N,K,ALPHA,A,LDA,X, INCX,BETA,Y,
    INCY)
CHARACTER * 1UPLO
\mathbb{NTEGERN,K,LDA,INCX,}\mathbb{N}CY
REAL ALPHA,BETA
REALA (LDA,*),X (*),Y (*)
SU BROUT\mathbb{NE SSBM V_64 (UPLO ,N,K,ALPHA ,A ,LDA,X,INCX ,BETA,Y,}
        \mathbb{NCY)}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N,K,LDA, INCX,}\mathbb{N}CY
REALALPHA,BETA
REAL A (LDA,*),X (*),Y (*)
```


## F95 INTERFACE

```
SU BROUTINE SBM V (UPLO, \(\mathbb{N}], K, A L P H A, A,[L D A], X,[\mathbb{N C X}], B E T A\), Y, [ \(\mathbb{N} C Y])\)
CHARACTER (LEN=1) ::UPLO
\(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathbb{N} C X, \mathbb{N C Y}\)
REAL ::ALPHA,BETA
REAL,D IM ENSION (:) :: X,Y
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A
SU BROUTINE SBM V_64 (UPLO, \(\mathbb{N}], K, A L P H A, A,[L D A], X,[\mathbb{N} C X]\),
```

CHARACTER (LEN=1) ::UPLO
$\mathbb{N} \operatorname{TEGER}$ (8) :: $N, K, L D A, \mathbb{N} C X, \mathbb{N} C Y$
REAL ::ALPHA,BETA
REAL,D $\mathbb{I M}$ ENSION (:) :: X,Y
REAL,D $\mathbb{I}$ ENSION (:,:) ::A

## C INTERFACE

\#include <sunperfh>
void ssbm v (charuple, intn, intk, floatalpha, float *a, int lda, float *x, int incx, floatbeta, float *y, intincy);
void ssbom v_64 (charuplo, long n, long k, floatalpha, float
*a, long lda, float *x, long incx, floatbeta, float *y, long incy);

## PURPOSE

ssbom v perform s the $m$ atrix-vector operation $y:=a l p h a * A * x+$ beta* $y$, w here alpha and beta are scalars, $x$ and $y$ are $n$ ele$m$ entvectors and $A$ is an $n$ by $n$ sym $m$ etric band $m$ atrix, $w$ ith k superdiagonals.

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the band $m$ atrix $A$ is being supplied as follow s:
$\mathrm{UPLO}=\mathrm{U}$ 'or L ' The upper triangularpart of $A$ is being supplied.

UPLO = L'or I'' The low er triangularpart of A is being supplied.

U nchanged on exit.

N (input)
O n entry, $N$ specifies the order of the $m$ atrix A. $\mathrm{N}>=0$. U nchanged on exit.

K (input)
On entry, $K$ specifies the number of superdiagonals of them atrix A.K $>=0$. U nchanged on
exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
Before entry w ith UPLO = U 'or G ', the leading ( $k+1$ ) by $n$ part of the array A $m$ ust contain the upper triangular band part of the symmetric $m$ atrix, supplied colum $n$ by colum $n$, $w$ th the leading diagonal of the $m$ atrix in row ( $k+1$ ) of the array, the first super-diagonalstarting atposition 2 in row $k$, and so on. The top left $k$ by $k$ triangle of the anray $A$ is not referenced. The follow ing program segm entw illtransfer the upper triangular part of a sym $m$ etric band $m$ atrix from conventional fullm atrix storage to band storage:

$$
\begin{aligned}
& \text { DO } 20, J=1, N \\
& \text { M }=\mathrm{K}+1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{M} A X(1, \mathrm{~J}-\mathrm{K}), \mathrm{J} \\
& \text { A }(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \text { CONTINUE } \\
& 20 \text { CONTINUE }
\end{aligned}
$$

Before entry w ith UPLO = L 'or 1', the leading ( $\mathrm{k}+1$ ) by n partof the array A m ustcontain the low er triangular band part of the symmetric $m$ atrix, supplied colum $n$ by colum $n$, $w$ th the leading diagonalof them atrix in row 1 of the amay, the first sub-diagonalstarting atposition 1 in row 2 , and so on. The bottom right k by $k$ triangle of the array $A$ is not referenced. The follow ing program segm entw illtransfer the low ertriangular part of a sym $m$ etric band $m$ atrix from conventional fullm atrix storage to band storage:

$$
\begin{aligned}
& \text { DO } 20, J=1, N \\
& \mathrm{M}=1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{J}, \mathrm{M} \mathbb{N}(\mathrm{~N}, \mathrm{~J}+\mathrm{K}) \\
& \mathrm{A}(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \text { CONTINUE } \\
& 20 \mathrm{CONTINUE}
\end{aligned}
$$

U nchanged on exit.

LDA (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD A $>=$ (
$\mathrm{k}+1$ ). Unchanged on exit.
X (input)
$(1+(n-1) * a b s(\mathbb{N} C X))$. Before entry, the increm ented array $X$ must contain the vectorx.
U nchanged on exit.
$\mathbb{N C X}$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents ofX. $\mathbb{N} C X<>0$. U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. U nchanged on exit.

Y (input/output)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))$. Before entry, the increm ented array $Y$ m ustcontain the vectory. On exit, $Y$ is overw ritten by the updated vectory.
$\mathbb{N C Y}$ (input)
O n entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $Y$. $\mathbb{N C Y}$ <> 0 . Unchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

ssbtrd-reduce a real sym $m$ etric band $m$ atrix $A$ to sym $m$ etric tridiagonal form T by an orthogonal sim ilarity transform ation

## SYNOPSIS

```
SUBROUT\mathbb{NE SSBTRD NECT,UPLO,N,KD,AB,LDAB,D,E,Q,LDQ,W ORK,}
    INFO)
CHARACTER * 1 VECT,UPLO
NNTEGER N,KD,LDAB,LDQ,NNFO
REAL AB (LDAB,*),D (*),E (*),Q (LDQ ,*),W ORK (*)
SUBROUTINE SSBTRD_64NECT,UPLO,N,KD,AB,LDAB,D,E,Q,LDQ,W ORK,
    \mathbb{NFO )}
CHARACTER * 1 VECT,UPLO
\mathbb{NTEGER*8N,KD,LDAB,LDQ,NNFO}
REALAB (LDAB,*),D (*),E (*),Q (LDQ , *),W ORK (*)
```


## F95 INTERFACE

```
SU BROUTINE SBTRD (NECT, UPLO, \(\mathbb{N}], K D, A B,[L D A B], D, E, Q,[L D Q]\), [ W ORK], [ \(\mathbb{N} F O\) ])
CHARACTER (LEN=1)::VECT,UPLO
\(\mathbb{N} T E G E R:: N, K D, L D A B, L D Q, \mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::D , E,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::AB, Q
SU BROUTINE SBTRD_64NECT, UPLO, \(\mathbb{N}], K D, A B,[L D A B], D, E, Q,[L D Q]\), [ W ORK], [ \(\mathbb{N} F \mathrm{O}\) ])
```

CHARACTER ( $\llcorner E N=1$ ) : : VECT, UPLO
$\mathbb{N} \operatorname{TEGER}(8):: N, K D, L D A B, L D Q, \mathbb{N} F O$
REAL,D $\mathbb{I}$ ENSION (:) :: D , E, W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) ::AB, Q

## C INTERFACE

\#include <sunperfh>
void ssbtrd (charvect, charuplo, intn, intkd, float *ab, int ldab, float*d, float*e, float * $q$, int ldq, int *info);
void ssbtrd_64 (charvect, charuplo, long n, long kd, float *ab, long ldab, float *d, float *e, float *q, long ldq, long *info);

## PURPOSE

ssbtrd reduces a real sym $m$ etric band $m$ atrix $A$ to sym $m$ etric tridiagonal form T by an orthogonalsim ilarity transform ation: $\mathrm{Q} * * \mathrm{~T} * A * Q=T$.

## ARGUMENTS

VECT (input)
$=\mathrm{N}$ : do notform $Q$;
$=\mathrm{V}$ : form Q ;
$=U$ : update a $m$ atrix $X$, by form ing $X * Q$.

UPLO (input)
$=\mathrm{U}$ : U ppertriangle ofA is stored;
$=\mathbb{L}:$ Low ertriangle ofA is stored.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

KD (input)
The num ber of superdiagonals of the $m$ atrix $A$ if
$\mathrm{UPLO}=\mathrm{U}$ ', or the num ber of subdiagonals if $\mathrm{U} P L O$
$=\mathbb{L}^{\prime} . \mathrm{KD}>=0$ 。

A B (input/output)
O $n$ entry, the upper or low er triangle of the sym $m$ etric band $m$ atrix $A$, stored in the firstK $D+1$ row s of the array. The $j$ th colum n of A is stored in the $j$ th colum n of the array AB as follow s: if $\mathrm{UPLO}=\mathrm{U}, \mathrm{AB}(\mathrm{kd}+1+i-j, j)=A(i, j)$ for $\mathrm{max}(1, j$
$\mathrm{kd})<=i<=\dot{j}$ ifUPLO $=\mathrm{L}, \mathrm{AB}(1+i-j)=A(i, 7)$ for $j=i<=m$ in $(n, j+k d)$. O $n$ exit, the diagonalele$m$ ents of AB are overw ritten by the diagonalele$m$ ents of the tridiagonalm atrix $T$; if $K D>0$, the elem ents on the first superdiagonal (if UPLO = U ) or the first subdiagonal (ifU PLO = L ) are overw ritten by the off-diagonalelem ents of $T$; the restofA $B$ is overw ritten by values generated during the reduction.

LD AB (input)
The leading dim ension of the array AB. LD AB >= K D +1 .

D (output)
The diagonalelem ents of the tridiagonalm atrix T .
E (output)
The off-diagonal elem ents of the tridiagonal $m$ atrix $T: E(i)=T(i, i+1)$ if $U P L O=U ; E(i)=$ $T(i+1, i)$ if $\mathrm{PLO}=\mathrm{L}^{\prime}$.

Q (input/output)
On entry, ifVECT $=U$ ', then Q must contain an N by $-\mathrm{N} m$ atrix X ; if $\mathrm{VECT}=\mathrm{N}$ 'or $V$ ', then $Q$ need notbe set.

On exit: if $\mathrm{VECT}=\mathrm{V}$ ', Q contains the N -by -N orthogonalm atrix $Q$; if VECT $=U$ ', $Q$ contains the product $\mathrm{X} * \mathrm{Q}$; ifVECT $=\mathrm{N}$ ', the array Q is not referenced.

LDQ (input)
The leading dim ension of the array $\mathrm{Q} . \mathrm{LD} \mathrm{Q}>=1$, and $L D Q>=N$ ifVECT $=V$ 'or $U '$.

W ORK (w orkspace)
dim ension (N)
$\mathbb{I N F O}$ (output)
= 0 : successfulexit
< $0:$ if $\mathbb{N} F O=-i$, the $i$-th argum enthad an illegalvalue

## FURTHER DETAILS

M odified by Linda K aufm an, BellLabs.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
sscal-C om pute y := alpha * y
```


## SYNOPSIS

```
SUBROUT\mathbb{NE SSCAL N,ALPHA,Y, INCY)}
\mathbb{NTEGER N,\mathbb{NCY}}\mathbf{}\mathrm{ (1)}
REALALPHA
REALY(*)
SU BROUT\mathbb{NE SSCAL_64 N,ALPHA,Y,INCY)}
INTEGER*8N,\mathbb{NCY}
REAL ALPHA
REALY(*)
```

F95 INTERFACE
SU BROUTINE SCAL ( $\mathbb{N}$ ],ALPHA, $Y$, [ $\mathbb{N C Y}]$ )
$\mathbb{N} T E G E R:: N, \mathbb{N C Y}$
REAL ::ALPHA
REAL,D $\mathbb{I M}$ ENSION (:) ::Y
SU BROUTINE SCAL_64 (N ],ALPHA, Y, [ $\mathbb{N} C Y$ ])
$\mathbb{N} T E G E R(8):: N, \mathbb{N} C Y$
REAL ::ALPHA
REAL,D $\mathbb{M}$ ENSION (:) ::Y

## C INTERFACE

\#include < sunperfh>
void sscal(intn, float alpha, float *y, int incy);
void sscal_64 (long n, float alpha, float *y, long incy);

## PURPOSE

sscalC om pute $y:=$ alpha * $y$ w here alpha is a scalar and $y$ is an $n$-vector.

## ARGUMENTS

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N must be at least one for the subroutine to have any visible effect. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

Y (input/output)
( $1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C Y)$ ). On entry, the increm ented amay $Y$ m ust contain the vectory. On exit, $Y$ is overw rilten by the updated vectory.
$\mathbb{N C Y}$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of Y. $\mathbb{N} C Y$ m ustnotbe zero. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS


## NAME

ssctr-Scatters elem ents from $x$ into $y$.

## SYNOPSIS

SUBROUTINE SSCTR $\mathbb{N} Z, X, \mathbb{N} D X, Y)$
$\operatorname{REALX}(\star), Y(*)$
$\mathbb{N}$ TEGER NZ
$\mathbb{N}$ TEGER $\mathbb{I N D X ( * )}$
SUBROUTINE SSCTR_64 $\mathbb{N} Z, X, \mathbb{N} D X, Y)$
REALX (*), $\mathrm{Y}^{(*)}$
$\mathbb{N} T E G E R * 8 N Z$
$\mathbb{N}$ TEGER*8 $\mathbb{I N D X ( * )}$
F95 $\mathbb{I N}$ TERFACE
SUBROUTINE SCTR (NZ],X, $\mathbb{N} D X, Y$ )
REAL,D IM ENSION (:) :: X,Y
$\mathbb{N}$ TEGER ::NZ
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{N D X}$
SUBROUTINE SCTR_64 ( $\mathbb{N} Z], X, \mathbb{N} D X, Y$ )
REAL,D IM ENSION (:) :: X,Y
$\mathbb{N} T E G E R(8):: N Z$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{I} \operatorname{ENSION}(:):: \mathbb{N D X}$

## PURPOSE

in fullstorage form .
do $i=1, n$
$y($ indx (i) $)=x(i)$
enddo

## ARGUMENTS

N Z (input) - $\mathbb{N}$ TEGER
N um ber of elem ents in the com pressed form .
U nchanged on exit.

X (input)
V ector containing the values to be scattered from com pressed form into fill storage form. U nchanged on exit.
$\mathbb{N} D X$ (input) $-\mathbb{N} T E G E R$
$V$ ector containing the indiges of the com pressed form. It is assum ed that the elem ents in $\mathbb{N} D \mathrm{X}$ are distinctand greater than zero. U nchanged on exit.

Y (output)
V ectorw hose elem ents specified by indx have been set to the corresponding entries ofx. Only the elem ents corresponding to the indices in indx have been m odified.

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

sskym m -Skyline form atm atrix-m atrix m ultiply

## SYNOPSIS

```
SUBROUT\mathbb{NE SSKYMM(TRANSA,M,N,K,ALPHA,DESCRA,}
* VAL,PNTR,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER TRANSA,M,N,K,DESCRA (5),
* LDB,LDC,LWORK
\mathbb{NTEGER PNTR (*),}
REAL ALPHA,BETA
REAL VAL(NNZ),B(LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NE SSKYM M _64(TRAN SA ,M ,N,K,ALPHA,DESCRA,}
* VAL,PNTR,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,M,N,K,DESCRA (5),}
* LDB,LDC,LW ORK
INTEGER*8 PNTR (*),
REAL ALPHA,BETA
REAL VAL(NNZ),B(LDB,*),C(LDC,*),WORK(LWORK)
w here NN Z = PN TR (K +1)PNTR (1) (upper triangular)
    NN Z = PN TR (M +1)PN TR (1) (low er triangular)
    PN TR () size = (K +1) (uppertriangular)
    PN TR () size = (M+1) (low er triangular)
```


## F95 INTERFACE

SUBROUTINESKYMM (TRANSA, M, $\mathbb{N}], K, A L P H A, D E S C R A, V A L$,

* PNTR, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
$\mathbb{N}$ TEGER TRANSA, M, K
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: DESCRA, PNTR

REAL ALPHA,BETA
REAL,D $\mathbb{I M}$ ENSION (:) :: VAL
REAL,D $\mathbb{I}$ ENSION (:, :) :: B , C

SUBROUTINE SKYM M_64 (TRANSA , M , $\mathbb{N}], K, A L P H A, D E S C R A, V A L$,

* PNTR, B, [LDB],BETA, C , [LDC], [W ORK], [LW ORK])
$\mathbb{N}$ TEGER*8 TRANSA, M, K
$\mathbb{N} T E G E R * 8, D \mathbb{M}$ ENSION (:) :: DESCRA, PNTR
REAL ALPHA,BETA
REAL,D $\mathbb{I}$ ENSION (:) :: VAL
REAL,D $\mathbb{I}$ ENSION (:, :) :: B , C


## DESCRIPTION

## C <-alpha op (A ) B + beta C

where A LPHA and BETA are scalar, $C$ and $B$ are dense $m$ atrices, $A$ is a m atrix represented in skyline form at and op (A) is one of
$o p(A)=A$ or $o p(A)=A^{\prime}$ or op $(A)=\operatorname{conjg}\left(A^{\prime}\right)$.
( 'indicates m atrix transpose)

## ARGUMENTS

TRA NSA Indicates how to operate $w$ ith the sparse $m$ atrix
0 : operate $w$ ith $m$ atrix
1 : operate $w$ th transpose $m$ atrix
2 : operate $w$ ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M $\quad N$ um ber of row $s$ in matrix A

N $\quad N$ um ber of colum ns in m atrix $C$

K $\quad \mathrm{N}$ um berof colum ns in matrix A

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger aray
D ESCRA (1) m atrix structure
0 : general $\mathbb{N O T}$ SUPPORTED )
1 : symm etric ( $A=A$ )
2 : Herm itian ( $\mathrm{A}=\mathrm{CONJG}(\mathrm{A})$ )
3 :Triangular
4 : Skew (A nti)-Symm etric ( $A=-A$ )
5 :D iagonal
6 : Skew Herm itian ( $A=-C O N J(A)$ )

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 :non-unit
1 : unit
DESCRA (4) A ray base $\mathbb{N} O T \mathbb{M}$ PLEM ENTED)
0 :C C++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? $\mathbb{N} O T \mathbb{I}$ PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () array contain the nonzeros ofA in skyline profile form. Row -oriented ifD ESCRA (2) = 1 (low er triangular), colum n oriented ifD ESCRA (2) = 2 (upper triangular).
PNTR 0) integer anay of length $M+1$ (low ertriangular) or $\mathrm{K}+1$ (upper triangular) such thatPN TR (I) PN TR (1)+1 points to the location in VAL of the firstelem entof the skyline profile in row (colum n) I.

B 0 rectangular array w ith first dim ension LD B.
LD B leading dim ension of $B$
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension of $C$

W ORK ( scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is notreferenced in the currentversion.

## SEE ALSO

N IST FO RTRAN Sparse B las U ser's G uide available at: http://m ath nistgov/n csd/Staff/k Rem ington/Aspoblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

## NOTES/BUGS

The SK Y data structure is not supported for a generalm atrix structure (DESCRA (1)=0).

A lso not supported:

1. low er triangularm atrix $A$ of size $m$ by $n$ where $m>n$
2. uppertriangularm atrix $A$ of size $m$ by $n$ where $m<n$

## Contents

- NAME
- SYNOPSIS


## - F95 INTERFACE

- DESCRIPTION
- ARGUMENTS
- SEE ALSO


## NAME

sskysm - Skyline form at triangular solve

## SYNOPSIS

```
SUBROUT\mathbb{NE SSKYSM(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,}
* VAL,PNTR,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,M,N,UNITD,DESCRA (5),}
* LDB,LDC,LW ORK
\mathbb{NTEGER PNTR(*),}
REAL ALPHA,BETA
REAL DV M),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NE SSKY SM _64(TRANSA,M ,N ,UNITD,DV,A LPHA,DESCRA,}
* VAL,PNTR,
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,M,N,UNITD,DESCRA (5),}
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 PNTR (*),}
REAL ALPHA,BETA
REAL DV M),VAL (NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
where NN Z = PN TR M +1)-PN TR (1) (uppertriangular)
    NNZ = PNTR (K +1)-PNTR (1) (low ertriangular)
    PN TR 0) size = M +1) (uppertriangular)
    PNTR 0 size = (K+1) (low ertriangular)
```


## F95 INTERFACE

SUBROUTINE SKYSM (TRANSA, M, $\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L$, * PNTR, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
$\mathbb{N} T E G E R \quad$ TRANSA, M,UNITD
$\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:)::$ DESCRA, PNTR

REAL ALPHA,BETA
REAL,D $\mathbb{M}$ ENSION (:) :: VAL,DV
REAL,D $\mathbb{I M}$ ENSION (: :) :: B,C


* VAL, PNTR, B, [LDB],BETA,C, [LDC], [W ORK], [LW ORK])
$\mathbb{N} T E G E R * 8$ TRANSA, M, UNITD
$\mathbb{N} T E G E R * 8, D \mathbb{M}$ ENSION (:) :: DESCRA, PNTR
REAL ALPHA,BETA
REAL,D $\mathbb{I M} E N S I O N$ (:) :: VAL, DV
REAL,D $\mathbb{I}$ ENSION (:, :) :: B , C


## DESCRIPTION

$C<-A L P H A \quad O p(A) B+B E T A C \quad C<-A L P H A D O P(A) B+B E T A C$ $C<-A L P H A \operatorname{Op}(A) D B+B E T A C$ where A LPHA and BETA are scalar, $C$ and $B$ are $m$ by $n$ dense $m$ atrices, $D$ is a diagonalscaling $m$ atrix, $A$ is a unit, ornon-unit, upper or low er triangularm atrix represented in skyline form at and $o p(A)$ is one of
$o p(A)=\operatorname{inv}(A)$ or op $(A)=\operatorname{inv}(A)$ or op $(A)=\operatorname{inv}\left(\operatorname{con} \dot{g}\left(A^{\prime}\right)\right)$. (inv denotesm atrix inverse, 'indicatesm atrix transpose)

## ARGUMENTS

TRAN SA Indicates how to operate $w$ th the sparse $m$ atrix 0 : operate $w$ ith $m$ atrix
1 : operate w th transpose $m$ atrix
2 : operate $w$ ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M $\quad \mathrm{N}$ um berof row s in $m$ atrix $A$

N $\quad$ Num berof colum ns in matrix C

UN ITD Type of scaling:
1 : Identily matrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum $n$ scaling)
4 :A utom atic row or colum $n$ scaling (see section N O TES for further details)

DV () A rray of length M containing the diagonalentries of the scaling diagonalm atrix D .

```
D ESCRA () D escriptor argum ent. Fi̇ve elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric (A=A)
2:H erm itian (A = CON JG (A ))
3:Triangular
4 : Skew (A nti)-Symm etric (A=-A )
5 :D iagonal
6:Skew Herm itian (A= CON JG (A ) )
N ote: For the routine, D ESCRA (1)=3 is only supported.
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 :non-unit
1 :unit
DESCRA (4) A nay base \(\mathbb{N} O T \mathbb{M}\) PLEM ENTED)
\(0: C / C++\) com patible
1 :Fortran com patible
DESCRA (5) repeated indices? (NOT \(\mathbb{M}\) PLEM ENTED)
0 : unknown
1 : no repeated indices
```

VAL () array contain the nonzeros ofA in skyline profile form .
Row -oriented ifD ESCRA (2) = 1 (low er triangular), colum n oriented ifD ESCRA (2) $=2$ (upper triangular) .

PN TR () integer array of length $M+1$ (low ertriangular) or
$\mathrm{K}+1$ (upper triangular) such thatPN TR (I)-PN TR (1)+1 points to the location in VAL of the firstelem ent of the skyline profile in row (colum n) I.

B 0 rectangular anay w ith first dim ension LD B .

LD B leading dim ension ofB

BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch amay of length LW ORK.
On exit, ifLW ORK $=-1, W$ ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK array. LW ORK should be at leastM .

Forgood perform ance, LW O RK should generally be larger.

For optim um perform ance on $m$ ultiple processors, LW ORK $>=M$ *N _CPU S where N_CPUS is the m axim um num berof processors available to the program .

If LW ORK $=0$, the routine is to allocate $w$ orkspace needed.

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

## SEE ALSO

N IST FO RTRA N Sparse B las U ser'S G uide available at:
http://m ath nist.gov/m cso/Staff/K Rem ington/Espblas/
"D ocum ent forthe B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlio .org/utk/papers/sparse _ps

## NOTES /BUGS

1.A lso notsupported:
a. low er triangularm atrix A ofsizem by n wherem $>n$
b. upper triangularm atrix $A$ of size $m$ by $n$ where $m<n$
2. N o test for singularity ornear-singularity is included in this routine. Such tests m ust.be perform ed before calling this routine.
3. If U N ITD $=4$, the routine scales the row s of $A$ if $D E S C R A(2)=1$ and the colum ns ofA if $D E S C R A(2)=2$ such that their 2 -norm s are one. The scaling $m$ ay im prove the accuracy of the com puted solution. C orresponding entries of V A L are changed only in this particular case. O n retum D V m atrix stored as a vector contains the diagonalm atrix by w hich the row $s$ (colum ns) have been scaled. U N ITD = 2 if $D E S C R A(2)=1$ and UN ITD $=3$ if $D E S C R A(2)=2$ should be used for the next calls to the routine $w$ ith overw rilten $V A L$ and $D V$.

WORK $(1)=0$ on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here i is the row (colum n) num berw hich 2 -norm is exactly zero.
4. If $D E S C R A(3)=1$ and $U \mathrm{~N}$ ITD $<4$, the unit diagonalelem ents
$m$ ightorm ightnotbe referenced in the SK Y representation of a sparse m atrix. They are not used anyw ay in these cases. ButifUN ITD = 4, the unit diagonalelem ents M U ST be referenced in the SK Y representation.
5.The routine can be applied for solving triangular system $s$ w hen the upper or low er triangle of the general sparse $m$ atrix $A$ is used. H ow ever DESCRA (1) m ustbe equal to 3 in this case.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sspcon -estim ate the reciprocal of the condition num ber (in the 1 -norm ) of a realsymm etric packed $m$ atrix $A$ using the factorization $A=U * D * U * * T$ or $A=L * D * L * * T$ com puted by SSPTRF

## SYNOPSIS

```
SUBROUT\mathbb{NE SSPCON (UPLO,N,AP,\mathbb{PIVOT,ANORM,RCOND,W ORK,IN ORK2,}}\mathbf{N},\mp@code{N}
    \mathbb{NFO)}
```

CHARACTER * 1 UPLO
$\mathbb{N} T E G E R N, \mathbb{N} F O$
$\mathbb{N}$ TEGER $\mathbb{P I V O T}$ (*), $\mathbb{I N}$ ORK 2 (*)
REAL ANORM,RCOND
REALAP ( ${ }^{*}$ ), WORK ( ${ }^{*}$ )
SU BROUTINE SSPCON_64 (UPLO,N,AP, $\mathbb{P} \mathbb{I V O T}, A N O R M, R C O N D, W$ ORK, IV ORK2,
$\mathbb{N} F O$ )
CHARACTER * 1 UPLO
$\mathbb{N}$ TEGER*8N, $\mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{I V O T}\left({ }^{*}\right), \mathbb{I N} O R K 2(*)$
REAL ANORM,RCOND
REALAP (*), W ORK (*)

## F95 INTERFACE

 [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1)::UPLO
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V O T}, \mathbb{I N}$ ORK2

SU BROUTINE SPCON_64 (UPLO,N,AP, $\mathbb{P} \mathbb{I} O$ OT,ANORM,RCOND, $\mathbb{W} O R K]$, [IW ORK2], [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1) :: UPLO
$\mathbb{N}$ TEGER (8) :: N, $\mathbb{N}$ FO
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T, \mathbb{I} W$ ORK 2
REAL ::ANORM,RCOND
REAL,D $\mathbb{I M} E N S I O N(:):: A P, W$ ORK

## C INTERFACE

\#include < sunperfh>
void sspcon (charuplo, intn, float *ap, int *ipívot, float anorm, float*rcond, int*info);
void sspcon_64 (char uplo, long n, float *ap, long *ípívot, float anorm, float * rcond, long *info);

## PURPOSE

sspcon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a realsym $m$ etric packed $m$ atrix A using the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ com puted by SSPTRF.

A $n$ estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND $=1 /$ ANORM * norm (inv (A))).

## ARGUMENTS

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
$=\mathrm{U}$ ': U pper triangular, form is $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$;
= L': Low ertriangular, form is $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$.

N (input) The order of the matrix A. $\mathrm{N}>=0$.
AP (input)
Realarray, dim ension $(\mathbb{N} *+1) / 2$ ) The block diagonal $m$ atrix $D$ and the multipliers used to obtain the factorU orL as com puted by SSPTRF, stored as a packed triangularm atrix.

Integer anray, dim ension $\mathbb{N}$ ) D etails of the inter-
changes and the block structure ofD as determ ined by SSPTRF.

## ANORM (input)

The 1-norm of the originalm atrix A.

## RCOND (output)

The reciprocal of the condition num ber of the $m$ atrix $A$, com puted as RCOND $=1 /(A N O R M * A \mathbb{N} V N M)$, where $A \mathbb{N} V N M$ is an estim ate of the 1 -nom of inv (A) com puted in this routine.
W ORK (w orkspace)
Realamay, dim ension ( $2 \star \mathrm{~N}$ )
IV ORK 2 (w orkspace)
Integer array, dim ension ( $2 \star \mathrm{~N}$ )
$\mathbb{N F O}$ (output)
$=0$ : successfulexit
< 0: if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sspev - com pute all the eigenvalues and, optionally, eigenvectors of a realsym $m$ etric $m$ atrix $A$ in packed storage

## SYNOPSIS

```
SUBROUT\mathbb{NE SSPEV (OBZZ,UPLO,N,AP,W ,Z,LD Z,W ORK, INFO)}
CHARACTER * 1 JOBZ,UPLO
INTEGERN,LDZ,\mathbb{NFO}
REALAP (*),W (*),Z (LD Z ,*),W ORK (*)
SUBROUT\mathbb{NE SSPEV_64(JOBZ,UPLO,N,AP,W,Z,LD Z,W ORK,INFO)}
CHARACTER * 1 JOBZ,UPLO
INTEGER*8N,LDZ,INFO
REALAP (*),W (*),Z (LD Z ,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE SPEV (JOBZ, UPLO ,N,AP, $\mathrm{W}, \mathrm{Z},[\operatorname{LD} Z],[\mathbb{W} O R K],[\mathbb{N} F O])$
CHARACTER (LEN=1): : JOBZ,UPLO
$\mathbb{N} T E G E R:: N, L D Z, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::AP,W,W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) :: Z
SU BROUTINE SPEV_64 (JOBZ, UPLO,N,AP,W,Z,[LD Z], [W ORK], [NFO ])
CHARACTER (LEN=1): : JOBZ,UPLO
$\mathbb{N}$ TEGER (8) :: $\mathrm{N}, \mathrm{LD} \mathrm{Z}, \mathbb{N}$ FO
REAL,D $\mathbb{M}$ ENSION (:) ::AP,W,W ORK
REAL,D IM ENSION (:,:) ::Z

## C INTERFACE

\#include <sunperfh>
void sspev (char jobz, charuplo, intn, float *ap, float *w , float * $z$, int ld $z$, int *info);
void sspev_64 (char j̣bz, charuplo, long n, float *ap, float *W , float * z, long ldz, long *info);

## PURPOSE

sspev com putes all the eigenvalues and, optionally, eigenvectors of a realsym $m$ etric $m$ atrix $A$ in packed storage.

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}^{\prime}:$ C om pute eigenvalues only;
$=\mathrm{V}$ ': C om pute eigenvalues and eigenvectors.

UPLO (input)
$=\mathrm{U}$ ': U pper triangle ofA is stored;
$=\mathbb{L}$ ': Low ertriangle ofA is stored.

N (input) The order of them atrix A. $\mathrm{N}>=0$.

A P (input/output)
Realarray, dim ension $(\mathbb{N} *(N+1) / 2)$ On entry, the upper or low er triangle of the sym $m$ etric $m$ atrix $A$, packed colum nw ise in a linear amray. The jth colum $n$ of A is stored in the array AP as follow s: ifUPLO $=U$ ', AP $(i+(j-1) \star j 2)=A(i, j)$ for $1<=\dot{i}=j$ if $\mathrm{UPLO}=\mathrm{L}^{\prime}, \mathrm{AP}\left(i+(j-1)^{\star}\left(2{ }^{\star} \mathrm{n}-j\right) / 2\right)=$ A $(i, j)$ for $j<i<=n$.

On exit, AP is overw ritten by values generated during the reduction to tridiagonal form . IfU PLO $=\mathrm{U}$ ', the diagonal and first superdiagonal of the tridiagonal $m$ atrix $T$ overw rite the comesponding elem ents of $A$, and if UPLO = L', the diagonal and first subdiagonal of $T$ overw rite the corresponding elem ents ofA.

W (output)
Realaray, dim ension $(\mathbb{N})$ If $\mathbb{N F O}=0$, the eigenvalues in ascending order.

Z (output)
Realaray, dim ension (LD Z , N) If $\mathrm{JO} \mathrm{BZ}=\mathrm{V}$ ', then
if $\mathbb{N} F O=0, Z$ contains the orthonorm aleigenvectors of them atrix $A, w$ ith the $i$-th column of $Z$ holding the eigenvector associated w th W (i). If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced.

LD Z (input)
The leading dim ension of the array $\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD $Z>=\mathrm{max}(1, \mathrm{~N})$.

W ORK (w orkspace)
Realarray, dim ension ( $3 * N$ )
$\mathbb{N F O}$ (output)
= 0: successfulexit.
$<0:$ if $\mathbb{N N F O}=-i$, the i-th argum enthad an illegalvalue.
> 0 : if $\mathbb{N} F O=$ i, the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sspevd - com pute all the eigenvalues and, optionally, eigenvectors of a realsym $m$ etric $m$ atrix A in packed storage

## SYNOPSIS

```
SUBROUT\mathbb{NE SSPEVD (OBZ,UPLO,N,AP,W ,Z,LD Z,W ORK,LW ORK,\mathbb{N ORK,}}\mathbf{~},\textrm{L}
    LIN ORK,\mathbb{NFO )}
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGER N,LD Z,LW ORK,LIN ORK,INFO}
INTEGER IN ORK (*)
REALAP (*),W (*),Z (LD Z ,*),W ORK (*)
SU BROUT\mathbb{NE SSPEVD_64 (JOBZ,UPLO,N,AP,W ,Z,LD Z,W ORK,LW ORK,}
    IN ORK,LIN ORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGER*8N,LD Z,LW ORK,LIN ORK,INFO}
INTEGER*8 IN ORK (*)
REALAP (*),W (*),Z (LD Z ,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE SPEVD (JOBZ, UPLO ,N,AP, W, Z, [LD Z], [W ORK ], [LW ORK ], [ $\mathbb{I N}$ ORK ], [LIN ORK ], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1): : JOBZ, UPLO
$\mathbb{N}$ TEGER ::N,LD Z,LW ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{I N}$ ORK
REAL,D $\mathbb{M}$ ENSION (:) ::AP,W,W ORK
REAL,D $\mathbb{I}$ ENSION (:,:) ::Z
SU BROUTINE SPEVD_64 (JOBZ, UPLO ,N,AP, W, Z, [LD Z], $\mathbb{W}$ ORK ], [LW ORK ],
[ $\mathbb{I N}$ ORK], [LIN ORK ], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1):: OBZ, UPLO
$\mathbb{N}$ TEGER (8) :: $N$, LD Z, LW ORK, LIW ORK, $\mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}$ (8), D $\mathbb{M} \operatorname{ENSION(:)::\mathbb {IW}ORK}$
REAL,D $\mathbb{I M} E N S I O N(:):: A P, W, W O R K$
REAL,D $\mathbb{I}$ ENSION (:,:) ::Z

## C INTERFACE

\#include <sunperfh>
void sspevd (char jobz, charuple, intn, float *ap, float
*W, float * $z$, int ldz, int *info);
void sspevd_64 (char j.bz, char uplo, long n, float *ap, float *W , float *z, long ldz, long *info);

## PURPOSE

sspevd com putes all the eigenvalues and, optionally, eigenvectors of a realsymm etric $m$ atrix $A$ in packed storage. If eigenvectors are desired, ituses a divide and conqueralgorithm .

The divide and conquer algorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on m achines w ith a guard digit in add/subtract, or on those binary $m$ achines $w$ thout guard digits $w$ hich subtract like the $C$ ray
 fail on hexadecim al or decim al machines $w$ ithout guard digits, butw e know of none.

## ARGUMENTS

JO B Z (input)
$=\mathrm{N}$ ': C om pute eigenvalues only;
$=V^{\prime}:$ C om pute eigenvalues and eigenvectors.

UPLO (input)
$=\mathrm{U}$ ': U pper triangle ofA is stored;
$=\mathbb{L}$ ': Low ertriangle ofA is stored.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

A P (input/output)
Realamay, dim ension $(\mathbb{N} *(\mathbb{N}+1) / 2$ ) On entry, the upper or low er triangle of the sym $m$ etric $m$ atrix $A$, packed colum nw ise in a linear array. The jth
collum $n$ ofA is stored in the array AP as follow S : ifUPLO $=U{ }^{\prime}, A P(i+(j-1) * j 2)=A(i, j)$ for
$1<=i<=j$ ifUPLO $=\mathrm{L}$ ', AP $\left(i+(j-1)^{\star}(2 \star n-j) / 2\right)=$ A $(i, j)$ for $j=i<=n$.

On exit, AP is overw rilten by values generated during the reduction to tridiagonal form. If UPLO $=U$ ', the diagonal and first superdiagonal of the tridiagonal $m$ atrix $T$ overw rite the corresponding elem ents ofA, and if U PLO = L', the diagonal and first subdiagonal of $T$ overw rite the comesponding elem ents ofA.
W (output)
Realarray, dim ension $\mathbb{N}$ ) If $\mathbb{N} F O=0$, the eigenvalues in ascending order.
$Z$ (input) Realarray, dim ension ( $\mathrm{LD} \mathrm{Z}, \mathrm{N}$ ) If $\mathrm{JOB} \mathrm{Z}=\mathrm{V}$ ', then if $\mathbb{N} F O=0, Z$ contains the orthonorm aleigenvectors of the $m$ atrix $A, w$ ith the $i$-th column of $Z$ holding the eigenvector associated w ith W (i). If JO BZ $=N$ ', then $Z$ is not referenced.

LD $Z$ (input)
The leading din ension of the array $\mathrm{Z} . \operatorname{LD} \mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD Z $>=\mathrm{max}(1, N)$.

W ORK (w orkspace)
Realarray, dim ension ( $L W$ ORK) On exit, if $\mathbb{N F O}=$ $0, W$ ORK (1) retums the optim allW ORK .

LW ORK (input)
The dim ension of the array W ORK. If $\mathrm{N}<=1$, LW ORK must be at least1. If $\mathrm{JOBZ}=\mathrm{N}$ 'and $\mathrm{N}>$ $1, \mathrm{LW} O R K \mathrm{~m}$ ust.be at least $2 * \mathrm{~N}$. If $\mathrm{JO} \mathrm{BZ}=\mathrm{V}^{\prime}$ and $\mathrm{N}>1$, LW ORK mustbe at least $1+6 * \mathrm{~N}+\mathrm{N} * * 2$.

If LW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

IV ORK (w orkspace/output)
Integer array, dim ension (LIN ORK)On exit, if $\mathbb{I N} F O$ $=0, \mathbb{I N}$ ORK (1) retums the optim alL IV ORK.

LIN ORK (input)
The dim ension of the array IN ORK. If $\mathrm{OOBZ}=\mathrm{N}^{\prime}$ or $\mathrm{N}<=1, \mathrm{~L} \mathbb{I N}$ ORK mustbe at least1. If JOBZ $=$ $V$ 'and $N>1, L \mathbb{I N} O R K$ mustbe at least $3+5 * N$.

If LIV ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim al size of
the $\mathbb{I V} O R K$ array, retums this value as the first
entry of the $\mathbb{I W}$ ORK array, and no errorm essage
related to LIN ORK is issued by X ERBLA.
$\mathbb{N F O}$ (output)
$=0$ : successfulexit
<0: if $\mathbb{N}$ FO $=-i$, the $i$-th argum ent had an illegalvahue.
>0: if $\mathbb{N} F O=$ i, the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sspevx - com pute selected eigenvalues and, optionally, eigenvectors of a realsym $m$ etric $m$ atrix A in packed storage

## SYNOPSIS

```
SUBROUTINE SSPEVX (OBBZ,RANGE,UPLO,N,AP,VL,VU,\mathbb{I},\mathbb{U},ABTOL,
    NFOUND,W,Z,LD Z,WORK,INORK2,\mathbb{FA}\mathbb{L},\mathbb{N}FO)
CHARACTER * 1 JOBZ,RANGE,UPLO
\mathbb{NTEGERN,}\mathbb{I},\mathbb{U},NFOUND,LDZ,\mathbb{NFO}
\mathbb{NTEGER IN ORK2 (*),\mathbb{FA [H (*)}}\mathbf{(})
REALVL,VU,ABTOL
REALAP (*),W (*),Z (LD Z ,*),W ORK (*)
SUBROUTINE SSPEVX_64(JOBZ,RANGE,UPLO,N,AP,VL,VU, II,IU,ABTOL,
    NFOUND,W,Z,LD Z,WORK,INORK2,\mathbb{FA}\mathbb{L},\mathbb{N}FO)
CHARACTER * 1 JOBZ,RANGE,UPLO
\mathbb{NTEGER*8N,\mathbb{N},\mathbb{U},NFOUND,LD Z,INFO}
NNTEGER*8 \mathbb{IN ORK2 (*), \mathbb{FA LI (*)}}\mathbf{(*)}
REALVL,VU,ABTOL
REALAP(*),W (*),Z (LD Z,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE SPEVX (JOBZ,RANGE,UPLO,N,AP,VL,VU, $\mathbb{I}, \mathbb{U}, A B T O L$, $\mathbb{N} F O U N D], W, Z,[L D Z],[W O R K],[\mathbb{W} O R K 2], \mathbb{F} A \mathbb{I},[\mathbb{N} F O])$

CHARACTER (LEN=1):: JOBZ,RANGE,UPLO
$\mathbb{N} T E G E R:: N, \mathbb{I}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: $\mathbb{I W}$ ORK2, $\mathbb{F A} \mathbb{I}$
REAL ::VL,VU,ABTOL
REAL,D $\mathbb{M}$ ENSION (:) ::AP,W,W ORK

SU BROUTINE SPEVX_64 (OBZ,RANGE,UPLO,N,AP,VL,VU, $\mathbb{Z}, \mathbb{Z}, A B T O L$, $[\mathbb{N} F O U N D], W, Z,[L D Z],[W O R K],[\mathbb{W} O R K 2], \mathbb{F} A \mathbb{H},[\mathbb{N} F O])$

CHARACTER ( $L E N=1$ ) : : JOBZ, RANGE, UPLO
$\mathbb{N}$ TEGER (8) :: $N$, $\mathbb{H}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O$
$\mathbb{N}$ TEGER (8), D $\mathbb{M} E N S I O N(:):: \mathbb{I N} O R K 2, \mathbb{F} A \mathbb{I}$
REAL ::VL,VU,ABTOL
REAL,D $\mathbb{M} E N S I O N$ (:) ::AP,W,W ORK
REAL,D $\mathbb{I M}$ ENSION (:,:) ::Z

## C INTERFACE

\#include <sunperfh>
void sspevx (char jobz, char range, charuplo, int n, float *ap, float vl, float vu, intil, intin, float abtol, int *nfound, float * w , float * z , int ldz, int*ifail, int*info);
void sspevx_64 (char jंbz, char range, char uplo, long n, float *ap, floatvl, floatvu, long il, long iu, floatabtol, long *nfound, float *w, float *z, long ldz, long *ifail, long *info);

## PURPOSE

sspevx com putes selected eigenvalues and, optionally, eigenvectors of a real symm etric $m$ atrix A in packed storage. E igenvalues/vectors can be selected by specifying either a range of vahues or range of indices for the desired eigenvalues.

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}$ : C om pute eigenvahues only;
$=\mathrm{V}$ : C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvalues w illbe found;
$=\mathrm{V}$ : alleigenvalues in the half-open interval
(VL, VU] w ill be found; = $\mathrm{I}^{\prime}$ : the $I \mathrm{I}$-th through
$\mathbb{I U}$-th eigenvalues w ill be found.

UPLO (input)
$=\mathrm{U}$ ': Upper triangle ofA is stored;
$=\mathbb{L} ':$ Low ertriangle ofA is stored.

N (input) The order of the m atrix A. $\mathrm{N}>=0$.
AP (input/output)
Realarray, dim ension $(\mathbb{N} *(N+1) / 2)$ On entry, the upper or low er triangle of the sym $m$ etric $m$ atrix $A$, packed colum nw ise in a linear array. The jth colum $n$ ofA is stored in the array AP as follow s : ifUPLO = U',AP $(i+(j-1) * j 2)=A(i, j)$ for
 A $(i, 7)$ for $j=i<=n$.

On exit, AP is overw rilten by values generated during the reduction to tridiagonal form. If PLO = U ', the diagonaland first superdiagonal of the tridiagonal m atrix T overw rite the comesponding elem ents ofA, and if UPLO = L', the diagonal and first subdiagonal of $T$ overw rite the corresponding elem ents ofA .

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE=A'or I'.

VU (input)
Se the description of V L .

II (input)
IfRA NGE= 'I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{L}<=\mathbb{U}<=N$, if $N>0$; $\mathbb{L}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N otreferenced ifRANGE= A 'or V'.

IU (input)
See the description of II.

ABTOL (input)
The absolute error tolerance for the eigenvalues. A $n$ approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b] of w idth less than orequal to

ABTOL + EPS * max ( $\mathfrak{a}|\mathrm{b}|)$,
where EPS is them achine precision. If ABTOL is less than or equalto zero, then EPS* $\mid$ | w illibe used in its place, where $T$ is the 1 -nom of the tridiagonalm atrix obtained by reducing AP to tri-
diagonal form .
E igenvahues w illbe com puted m ostaccurately when ABTOL is set to tw ice the underflow threshold $2 \star$ SLAM CH (S ), notzero. If this routine retums w ith $\mathbb{N}$ FO $>0$, indicating that som e eigenvectors did not converge, try setting ABTOL to $2 *$ SLAM CH (S ).

See "C om puting Sm allSingularV alues ofB idiagonal M atrices w ith G uaranteed H igh Relative A ccuracy," by Demmeland $K$ ahan, LA PA CK W orking $N$ ote \#3.

## NFOUND (output)

The total num ber of eigenvalues found. $0<=$ NFOUND <= N. IfRANGE = A', NFOUND = N, and if RANGE $=$ 'I',NFOUND $=\mathbb{U}-\mathbb{L}+1$.

W (output)
Real array, dim ension $\mathbb{N}$ ) If $\mathbb{N F O}=0$, the selected eigenvalues in ascending order.

Z (output)
Realarray, dim ension (LD $Z, \max (1, M))$ If $\operatorname{OBZ}=$ V ', then if $\mathbb{N} F O=0$, the firstNFOUND colum ns of $Z$ contain the orthonorm al eigenvectors of the $m$ atrix A comesponding to the selected eigenvalues, w th the ith column of $Z$ holding the eigenvector associated w ith W (i). If an eigenvector fails to converge, then that colum $n$ of $Z$ contains the latest approxim ation to the eigenvector, and the index of the eigenvector is retumed in $\mathbb{F A} \mathbb{I}$. If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced. $N$ ote: the user must ensure that at least m ax (1,NFOUND) colum ns are supplied in the anay $Z$; ifRANGE = V', the exact value of FOUND is not know $n$ in advance and an upperbound $m$ ustbe used.

LD $Z$ (input)
The leading $d i m$ ension of the array $Z$. LD $Z>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD Z >= $\mathrm{max}(1, N)$.

W ORK (w orkspace)
Realarray, dim ension ( $8 * \mathrm{~N}$ )
$\mathbb{I N}$ ORK 2 (w orkspace)
Integer array, dim ension ( $5 * \mathrm{~N}$ )
FA $\mathbb{I}$ (output)
Integer array, dim ension $\mathbb{N}$ ) If JOB Z = $V$ ', then if $\mathbb{N} F O=0$, the firstNFOUND elem ents of $\mathbb{F A} \mathbb{I}$
are zero. If $\mathbb{N} F O>0$, then $\mathbb{F A} I I$ contains the
indices of the eigenvectors that failed to converge. If $J 0 B Z=N$ ', then $\mathbb{F A} \mathbb{I}$ is not referenced.
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0$ : if $\mathbb{N F O}=-i$, the $i$-th argum ent had an illegal value
$>0$ : if $\mathbb{N F O}=\mathrm{i}$, then ieigenvectors failed to converge. Their indioes are stored in array FAI.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sspgst-reduce a real sym $m$ etric-definite generalized eigenproblem to standard form, using packed storage

## SYNOPSIS

```
SUBROUT\mathbb{NE SSPGST(TTYPE,UPLO,N,AP,BP,INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER ITYPE,N,INFO}
REALAP (*),BP (*)
SU BROUT\mathbb{NE SSPGST_64(TTYPE,UPLO,N,AP,BP,INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8 ITYPE,N,INFO}
REALAP(*),BP(*)
```

F95 INTERFACE
SU BROUTINE SPGST (TTYPE,UPLO,N,AP,BP, [ $\mathbb{N} F O]$ )
CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, \mathbb{N F O}$
REAL,D $\mathbb{M}$ ENSION (:) ::AP,BP
SU BROUTINE SPGST_64 (TTYPE, UPLO ,N,AP,BP, [iNFO])
CHARACTER (LEN=1) ::UPLO
$\mathbb{N} \operatorname{TEGER}(8):: \mathbb{T Y P E}, \mathrm{N}, \mathbb{I N F O}$
REAL,D $\mathbb{M}$ ENSION (:) ::AP,BP
C INTERFACE
\#include <sunperfh>
void sspgst(int itype, charuplo, intn, float *ap, float *bp, int *info);
void sspgst 64 (long itype, charuplo, long n, float *ap, float*bp, long *info);

## PURPOSE

sspgst reduces a realsym m etric-definite generalized eigenproblem to standard form, using packed storage.

If ITY PE $=1$, the problem is $A * x=$ lam bda*B ${ }^{*} \mathrm{X}$, and $A$ is overw ritten by inv $(U * * T) * A * \operatorname{inv}(U)$ or inv ( $(\mathrm{A})$ *A *inv ( $(\mathrm{L} * * \mathrm{~T})$
If ITYPE $=2$ or 3 , the problem is $A * B * x=$ lam bda* x or $B * A * x=\operatorname{lam}$ bda* $x$, and $A$ is overw rilten by $U * A * U * * T$ or L**T*A*L。

B m usthave been previously factorized as U ** T * U or $\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ by SPPTRF.

## ARGUMENTS

ITYPE (input)
$=1:$ compute $\quad \operatorname{inv}(\mathrm{U} * * T) \star A * \operatorname{inv}(\mathrm{U})$ or
$\operatorname{inv}(\amalg) \star A$ *inv $(\amalg * * T)$;
$=2$ or 3 : com pute $\mathrm{U} * \mathrm{~A} * \mathrm{U} * * \mathrm{~T}$ orL ${ }^{* *} \mathrm{~T} * \mathrm{~A} * \mathrm{~L}$.

UPLO (input)
$=U^{\prime}$ : Uppertriangle of $A$ is stored and $B$ is factored as $\mathrm{U} * * \mathrm{~T} * \mathrm{U} ;=\mathrm{L}:$ : Low er triangle of A stored and B is factored as $\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$.

N (input) The order of the $m$ atriges $A$ and $B . N>=0$.

AP (input/output)
Realanay, dim ension $\mathbb{N}^{*}(\mathbb{N}+1) / 2$ ) On entry, the upper or low er triangle of the sym $m$ etric $m$ atrix $A$, packed colum nw ise in a linear array. The jth colum $n$ ofA is stored in the array AP as follow s: if $U P L O=U ', A P(i+(j 1) \star j 2)=A(i, j)$ for $1<=\dot{<}=\dot{j}$ if $\mathrm{UPLO}=\mathrm{L}^{\prime}, \mathrm{AP}(i+(j 1) *(2 n-j) / 2)=$ A $(i, j)$ for $j<=i<=n$.

On exit, if $\mathbb{N F O}=0$, the transform ed $m$ atrix, stored in the sam e form atas A.
$B P$ (input)
Realarray, dim ension $(\mathbb{N} * \mathbb{N}+1) / 2)$ The triangular factor from the Cholesky factorization of $B$, stored in the sam e form at as A, as retumed by SPPTRF.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sspgv -com pute all the eigenvalues and, optionally, the eigenvectors of a realgeneralized sym $m$ etric-definite eigenproblem, of the form A *x=(lam bda) ${ }^{B}{ }^{*} x, A * B x=(\operatorname{lam} . b d a) * x$, or B *A *x= (lam bda) *x

## SYNOPSIS

```
SU BROUT\mathbb{NE SSPGV (TTYPE,NOBZ,UPLO,N,AP,BP,W ,Z,LD Z,W ORK, INFO)}
CHARACTER * 1 JOBZ,UPLO
INTEGER ITYPE,N,LDZ,INFO
REALAP (*),BP (*),W (*),Z (LD Z,*),W ORK (*)
SU BROUTINE SSPGV_64 (TTYPE,NOBZ,UPLO,N ,AP,BP,W ,Z,LD Z,W ORK,
        \mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGER*8 ITYPE,N,LD Z, INFO}
REALAP (*),BP (*),W (*),Z (LD Z ,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE SPGV (TTYPE, JOBZ, UPLO,N,AP,BP,W,Z,[LD Z], [W ORK ], [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1): : JOBZ,UPLO
$\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D Z, \mathbb{N F O}$
REAL,D $\mathbb{I M}$ ENSION (:) ::AP,BP,W,W ORK
REAL,D $\mathbb{M}$ ENSION (:,:)::Z

SU BROUTINE SPGV_64 (TTYPE, JOBZ, UPLO ,N,AP,BP,W,Z, [LD Z], [W ORK ], [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1): : OBZ Z UPLO
$\mathbb{N} T E G E R(8):: \operatorname{ITY} P E, N, L D Z, \mathbb{N} F O$
REAL,D $\mathbb{I}$ ENSION (:) ::AP,BP,W ,W ORK
REAL,D IM ENSION (:,:) ::Z

## C INTERFACE

\#include <sunperfh>
void sspgv (int itype, char jobz, char uplo, int n, float
*ap, float *bp, float *W , float * z , int ldz, int
*info);
void sspgv_64 (long itype, char jobz, char uplo, long n,
float *ap, float *bp, float*w, float*z, long
ldz, long *info);

## PURPOSE

sspgv com putes all the eigenvalues and, optionally, the eigenvectors of a realgeneralized sym $m$ etric-definite eigenproblem, of the form $A * x=(\operatorname{lam} . b d a) * B * x, A * B x=\left(\operatorname{lam}\right.$ bda) ${ }^{*} x$, or $B *_{A} *_{x}=\left(l a m\right.$ bda) ${ }^{*} x$. H ere $A$ and $B$ are assum ed to be sym $m$ etric, stored in packed form at, and B is also positive definite.

## ARGUMENTS

ITYPE (input)
Specifies the problem type to be solved:
$=1: A{ }^{*} \mathrm{X}=\left(\operatorname{lam}\right.$ bda) ${ }^{\star} \mathrm{B}{ }^{*} \mathrm{X}$
$=2: A * B *_{X}=\left(\operatorname{lam}\right.$ bda) ${ }^{*} \mathrm{X}$
$=3: \mathrm{B}^{\star} \mathrm{A}{ }^{\star} \mathrm{x}=\left(\operatorname{lam}\right.$ bda)${ }^{\star} \mathrm{x}$

JOBZ (input)
$=\mathrm{N}^{\prime}:$ C om pute eigenvalues only;
$=\mathrm{V}$ ': C om pute eigenvalues and eigenvectors.

UPLO (input)
$=\mathrm{U}$ : U pper triangles of $A$ and $B$ are stored;
$=\mathbb{L}$ ': Low ertriangles of $A$ and $B$ are stored.

N (input) The order of the matriges A and $\mathrm{B} . \mathrm{N}>=0$.

A P (input/output)
Realarray, dim ension $(\mathbb{N} *(N+1) / 2)$ On entry, the
upper or low er triangle of the sym $m$ etric $m$ atrix $A$,
packed colum nw ise in a linear anray. The jth
colum $n$ of A is stored in the array AP as follow s:
if $\mathrm{UPLO}=U \mathrm{U}^{\prime}, \mathrm{AP}(i+(j-1) \star j 2)=A(i, j)$ for $1<=i<=j$ ifUPLO $=L '$ AP $(i+(j 1) *(2 * n-j) / 2)=$ A $(i, j)$ for $j<i<=n$.

On exit, the contents of AP are destroyed.

BP (input/output)
Realarray, dim ension $(\mathbb{N} *(N+1) / 2)$ On entry, the upper or low er triangle of the sym $m$ etric $m$ atrix $B$, packed colum nw ise in a linear array. The jth colum $n$ of $B$ is stored in the array BP as follow s: ifUPLO $=U{ }^{\prime}$, BP $(i+(j 1) * j 2)=B(i, 7$ for $1<=i<=j$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{BP}(i+(j 1) *(2 * n-j) / 2)=$ B $(i, j)$ for $j<i<=n$.

On exit, the triangular factor $U$ or L from the Cholesky factorization $\mathrm{B}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{B}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$, in the sam e storage form at as B.

W (output)
Realarray, dim ension $\mathbb{N}$ ) If $\mathbb{N F O}=0$, the eigenvalues in ascending order.

Z (output)
Realarray, dim ension (LD Z, N) If $\mathrm{OB} \mathrm{B}=\mathrm{V}$ ', then if $\mathbb{N} F O=0, Z$ contains them atrix $Z$ of eigenvectors. The eigenvectors are norm alized as follow s: if ITYPE $=1$ or $2, Z * * T * B * Z=I$; if $T T Y P E=3$, $Z * * T * \operatorname{inv}(B) * Z=I$. If $O B Z=N$ ', then $Z$ is not referenced.

LD Z (input)
The leading dim ension of the amay $\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1$, and if $\mathrm{OBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{m}$ ax $(1, N)$.

W ORK (w orkspace)
Realamay, dim ension ( $3 * N$ )
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$ th argum ent had an illegalvalue
> 0: SPPTRF orSSPEV retumed an errorcode:
$<=\mathrm{N}:$ if $\mathbb{N} F O=i, S S P E V$ failed to converge; $i$ offf-diagonalelem ents of an interm ediate tridiagonal form did notconverge to zero. $>\mathrm{N}$ : if $\mathbb{N} F O$ $=n+i$, for $1<=i<=n$, then the leading $m$ inor oforderiofB is not posilive definite. The factorization of $B$ could notbe com pleted and no eigenvalues or eigenvectors w ere com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

sspgvd - com pute all the eigenvalues, and optionally, the eigenvectors of a realgeneralized sym $m$ etric-definite eigenproblem, of the form A *x=(lam bda)*B *x, A *B x=(lam bda) ${ }^{*}$, or B ${ }^{\mathrm{A}} \mathrm{A}_{\mathrm{X}}=\left(\operatorname{lam}\right.$ bda) ${ }^{\mathrm{X}} \mathrm{X}$

## SYNOPSIS

```
SU BROUT\mathbb{NE SSPGVD (TTYPE,NOBZ,UPLO,N,AP,BP,W ,Z,LDZ,W ORK,}
    LW ORK,\mathbb{IN ORK,LIN ORK,INFO)}
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGER ITYPE,N,LDZ,LW ORK,LIN ORK,\mathbb{NFO}}\mathbf{N}\mathrm{ (IN}
INTEGER 㱐ORK (*)
REALAP (*),BP (*),W (*),Z (LD Z ,*),W ORK (*)
SU BROUT\mathbb{NE SSPGVD_64 (TTYPE,NOBZ,UPLO,N,AP,BP,W ,Z,LD Z,W ORK,}
        LW ORK,\mathbb{IN ORK,L\mathbb{IN ORK,INFO)}}\mathbf{N}\mathrm{ )}
```

CHARACTER * 1 JOBZ, UPLO
$\mathbb{N} T E G E R * 8 \mathbb{I T} Y P E, N, L D Z, L W O R K, L \mathbb{N} O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I N}$ ORK (*)
REALAP (*), BP (*), W (*), Z (LD Z, *), W ORK (*)

## F95 INTERFACE

SUBROUTINE SPGVD (TTYPE, JOBZ,UPLO,N,AP,BP,W,Z, [LDZ], [W ORK], [LW ORK], [IW ORK], [LIN ORK], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1): : JOBZ, UPLO
$\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D Z, L W$ ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{I N}$ ORK
REAL,D $\mathbb{I M}$ ENSION (:) ::AP,BP,W ,W ORK

SU BROU T INE SPGVD_64 (ITYPE, JOBZ, UPLO ,N,AP,BP, W, Z, [LD Z], $[\mathbb{W} O R K],[L W O R K],[\mathbb{N}$ ORK $],[L \mathbb{I} W O R K],[\mathbb{N} F O])$

CHARACTER (LEN=1) :: JOBZ, UPLO
$\mathbb{I N} T E G E R(8):: \mathbb{I T Y} P E, N, L D Z, L W O R K, L \mathbb{N} O R K, \mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}$ (8),D $\mathbb{M}$ ENSION (:) :: $\mathbb{I W}$ ORK
REAL,D $\mathbb{I M} E N S I O N(:):: A P, B P, W, W O R K$
REAL,D IM ENSION (:,:) ::Z

## C INTERFACE

\#include <sunperfh>
void sspgvd (int itype, char jंbz, charuplo, int n, float
*ap, float *bp, float *w, float *z, int ldz, int *info);
void sspgvd_64 (long itype, char j̀jbz, char uplo, long n, float *ap, float *bp, float*w, float*z, long ldz, long *info);

## PURPOSE

sspgvd com putes all the eigenvahues, and optionally, the eigenvectors of a realgeneralized sym m etric-definite eigenproblem, of the form $A * x=\left(\operatorname{lam}\right.$ bda) ${ }^{*} \mathrm{~B}{ }^{*} \mathrm{x}, \mathrm{A} * \mathrm{~B} \mathrm{x}=(\operatorname{lam} \mathrm{bda}){ }^{*} \mathrm{x}$, or $B * A * x=\left(l a m\right.$ bda) ${ }^{*} x$. H ere A and B are assum ed to be sym $m$ etric, stored in packed form at, and B is also positive definite. Ifeigenvectors are desired, ituses a divide and conquer algorithm .

The divide and conquer algorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on $m$ achines w ith a guard digit in add/subtract, or on those binary $m$ achines $w$ ithout guard digits w hich subtract like the $C$ ray $\mathrm{X}-\mathrm{M} P$, C ray $Y \mathrm{M} P$, C ray $\mathrm{C}-90$, or C ray- 2 . It could conceivably fail on hexadecim al or decim al machines $w$ thout guard digits, butw e know of none.

## ARGUMENTS

ITYPE (input)
Specifies the problem type to be solved:
$=1: A{ }^{*} \mathrm{x}=(\operatorname{lam} . \mathrm{bda}){ }^{\star} \mathrm{B}{ }^{*} \mathrm{x}$
$=2: A * B *_{X}=\left(\operatorname{lam}\right.$ bda) ${ }^{*} \mathrm{X}$
$=3: B{ }^{*} A{ }^{*} \mathrm{X}=\left(\operatorname{lam}\right.$ bda)${ }^{\star} \mathrm{x}$

JOBZ (input)
= N ': C om pute eigenvalues only;
$=\mathrm{V}:$ : C om pute eigenvahues and eigenvectors.

UPLO (input)
$=\mathrm{U}$ ': U pper triangles of $A$ and $B$ are stored;
$=\mathrm{L}$ ': Low ertriangles of $A$ and $B$ are stored.
N (input) The order of the m atrices A and $\mathrm{B} . \mathrm{N}>=0$.
AP (input/output)
Realarray, dim ension $(\mathbb{N} *(\mathbb{N}+1) / 2$ ) On entry, the upper or low er triangle of the sym $m$ etric $m$ atrix $A$, packed colum nw ise in a linear array. The $j$ th colum $n$ ofA is stored in the array AP as follow s: ifUPLO $=U$ ', AP $(i+(j-1) \star j 2)=A(i, j)$ for
 A $(i, j)$ for $j<=i<=n$.

On exit, the contents ofAP are destroyed.
BP (input/output)
Realarray, dim ension $\mathbb{N}^{*}(\mathbb{N}+1) / 2$ ) On entry, the upper or low er triangle of the sym $m$ etric $m$ atrix $B$, packed colum nw ise in a linear array. The jth colum $n$ of $B$ is stored in the array BP as follow $s$ : if UPLO $=U$ ', BP $(i+(j-1) * j 2)=B(i, j)$ for $1<=i<=j$ ifUPLO $=\mathrm{L}, \mathrm{BP}\left(i+(j 1)^{\star}\left(2{ }^{\star} \mathrm{n}-\mathrm{j}\right) / 2\right)=$ B $(i, 7)$ for $j=i<=n$.

On exit, the triangular factor $U$ or $L$ from the Cholesky factorization $B=U * * T * U$ or $B=L * L * *$, in the sam e storage form atas $B$.

W (output)
Realaray, dim ension $\mathbb{N}$ ) If $\mathbb{N} F O=0$, the eigenvalues in ascending order.

Z (output)
Realarray, dim ension ( $\mathrm{LD} Z, \mathrm{~N}$ ) If $\mathrm{JO} \mathrm{BZ}=\mathrm{V}$ ', then if $\mathbb{N} F O=0, Z$ contains them atrix $Z$ of eigenvectors. The eigenvectors are norm alized as follow $s$ : if $I T Y P E=1$ or $2, Z * * T * B * Z=I$; if $I T Y P E=3$, $Z * * T * \operatorname{inv}(B) * Z=I$. If $J O B Z=N$ ', then $Z$ is not referenced.

LD Z (input)
The leading dim ension of the array $Z$. LD $Z>=1$, and if $\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{max}(1, N)$.

W ORK (w orkspace/output)
Realarray, dim ension (LW ORK) On exit, if $\mathbb{N} F O=$ $0, W$ ORK (1) retums the optim allW ORK.

LW ORK (input)
The dim ension of the array $W$ ORK. If $N<=1$, LW $O R K>=1$. If $J O B Z=N$ 'and $N>1$,LW $O R K>=$ 2*N. If $\operatorname{JOBZ}=\mathrm{V}$ 'and $\mathrm{N}>1$,LW ORK >=1 + 6*N + $2 * N * * 2$ 。

If LW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IN ORK (w orkspace/output)
Integer array, dim ension ( $\llcorner\mathbb{I N}$ ORK) On exit, if $\mathbb{I N}$ FO $=0, \mathbb{I V}$ ORK (1) retums the optim all IV ORK.

LIV ORK (input)
The dim ension of the array $\mathbb{I N} O R K$. If $J O B Z=N^{\prime}$ or $N<=1, L \mathbb{I W} O R K>=1$. If $J O B Z=V$ 'and $N>1$, LIN ORK >= $3+5{ }^{*} \mathrm{~N}$.

If $L \mathbb{I V} O R K=-1$, then a w orkspace query is assum ed; the routine only calculates the optim al size of the $\mathbb{I V}$ ORK array, retums this value as the first entry of the $\mathbb{W}$ ORK anay, and no errorm essage related to $L \mathbb{I N} O R K$ is issued by X ERBLA.
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
< 0: if $\mathbb{N}$ FO = -i, the i-th argum ent had an illegalvalue
> 0: SPPTRF orSSPEVD retumed an errorcode: $<=\mathrm{N}:$ if $\mathbb{N} F O=i, S S P E V D$ failed to converge; i off-diagonalelem ents of an interm ediate tridiagonal form did notconverge to zero; $>\mathrm{N}$ : if $\mathbb{N F O}$ $=N+i$, for $1<=i<=N$, then the leading $m$ inor oforderiofB is not positive definite. The factorization of B could notbe com pleted and no eigenvalues or eigenvectors w ere com puted.

## FURTHER DETAILS

B ased on contributions by
M ark Fahey,D epartm entofM athem atics, U niv. of K entucky, USA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

sspgvx - com pute selected eigenvalues, and optionally, eigenvectors of a realgeneralized sym $m$ etric-definite eigenproblem, of the form $A * x=(\operatorname{lam} . b d a) * B * x, A * B x=(l a m ~ b d a) * x$, or B ${ }^{\mathrm{A}} \mathrm{A}_{\mathrm{X}}=\left(\operatorname{lam}\right.$ boda) ${ }^{*} \mathrm{X}$

## SYNOPSIS

```
SU BROUT\mathbb{NE SSPGVX (TTYPE,JOBZ,RANGE,UPLO,N,AP,BP,VL,VU,IL,}
    \mathbb{U},ABSTOL,M,W ,Z,LDZ,W ORK,IN ORK,\mathbb{FA}\mathbb{I},\mathbb{N}FO)
```

CHARACTER * 1 JOBZ,RANGE,UPLO
$\mathbb{N} T E G E R \mathbb{I T} Y P E, N, \mathbb{I}, \mathbb{U}, M, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R \mathbb{I}$ ORK (*), $\mathbb{F A} \mathbb{I}$ (*)
REALVL,VU,ABSTOL
REALAP (*), BP (*) , W (*), Z (LD Z , *) , W ORK (*)
SU BROUTINE SSPGVX_64 (TTYPE, JOBZ,RANGE, UPLO,N,AP,BP,VL,VU, $\mathbb{I}$,
$\mathbb{U}, \mathrm{ABSTOL}, \mathrm{M}, \mathrm{W}, \mathrm{Z}, \mathrm{LD} \mathrm{Z}, \mathrm{W} O R K, \mathbb{N} O R K, \mathbb{F} A \mathbb{L}, \mathbb{N} F O)$
CHARACTER * 1 JOBZ,RANGE,UPLO
$\mathbb{N}$ TEGER*8 $\mathbb{I T} Y$ PE, $N, \mathbb{L}, \mathbb{U}, \mathrm{M}, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I N}$ ORK (*), $\mathbb{F A} \mathbb{L}(*)$
REALVL,VU,ABSTOL
REALAP (*), BP (*) , W (*) , Z (LD Z , *) , W ORK (*)

## F95 INTERFACE

SU BROUT $\mathbb{N} E \operatorname{SPGVX(TTYPE,~JOBZ,RANGE,UPLO,N,AP,BP,VL,VU,~\amalg ,~}$ $\mathbb{U}, A B S T O L, M, W, Z,[L D Z],[W O R K],[\mathbb{W} O R K], \mathbb{F A} \mathbb{I},[\mathbb{N} F O])$

CHARACTER (LEN=1):: DBZ,RANGE,UPLO
$\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, \mathbb{L}, \mathbb{U}, M, L D Z, \mathbb{N} F O$
$\mathbb{N}$ TEGER,D $\mathbb{M}$ ENSION (:) :: $\mathbb{I N} O R K, \mathbb{F} A \mathbb{L}$ REAL ::VL,VU,ABSTOL
REAL,D $\mathbb{M} E N S I O N$ (:) ::AP,BP,W ,W ORK
REAL,D IM ENSION (:,:) ::Z

SU BROUTINE SPGVX_64 (TTYPE, JOBZ,RANGE, UPLO, N, AP, BP, VL, VU, $\mathbb{I}, \mathbb{U}, A B S T O L, M, W, Z,[L D Z],[W O R K],[\mathbb{W} O R K], \mathbb{F} A \mathbb{L},[\mathbb{N} F O])$

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
$\mathbb{N}$ TEGER (8) :: ITYPE, $N, \mathbb{I}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, \mathbb{N}$ FO
$\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I}$ ORK , $\mathbb{F} A \mathbb{I}$
REAL ::VL,VU,ABSTOL
REAL,D $\mathbb{I}$ ENSION (:) ::AP,BP,W ,W ORK
REAL,D IM ENSION (:,:) ::Z

## C INTERFACE

\#include <sunperfh>
void sspgvx (int itype, char jobz, char range, charuplo, int
n, float *ap, float *bp, floatvl, floatvu, int il, intin, float abstol, int *m, float *W , float * $z$, int ldz, int *ifail, int *info);
void sspgvx_64 (long itype, char jobz, char range, charuplo, long n, float *ap, float *bop, floatvl, floatvu, long il, long iu, float abstol, long *m , float *w , float *z, long ldz, long *ifail, long *info);

## PURPOSE

sspgvx com putes selected eigenvahues, and optionally, eigenvectors of a realgeneralized sym $m$ etric-definite eigenproblem, of the form $A * x=(l a m ~ b d a) * B * x, A * B x=(l a m ~ b d a) * x$, or $B{ }^{*} A{ }^{*} \mathrm{X}=(\operatorname{lam} \operatorname{bda}){ }^{\star} \mathrm{X}$. H ere A and B are assum ed to be sym $m$ etric, stored in packed storage, and $B$ is also positive definite. E igenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

## ARGUMENTS

## ITYPE (input)

Specifies the problem type to be solved:
$=1: A{ }^{*} \mathrm{X}=\left(\mathrm{lam}\right.$ bda) ${ }^{\star} \mathrm{B}{ }^{*} \mathrm{X}$
$=2: A * B *_{x}=(\operatorname{lam} . b d a){ }^{*} \mathrm{x}$
$=3: B * A{ }^{*} \mathrm{X}=(\operatorname{lam} \mathrm{bda}){ }^{\star} \mathrm{x}$

JO B Z (input)
$=\mathrm{N}$ ': C om pute eigenvalues only;
$=\mathrm{V}: \mathrm{C}$ om pute eigenvalues and eigenvectors.
RANGE (input)
= A : alleigenvalues $w$ illbe found.
= V ':alleigenvalues in the half-open interval
( L , VU ] will be found. = $I^{\prime}$ : the $\mathbb{I}$-th through $\mathbb{I U}$-th eigenvaluesw illlbe found.

UPLO (input)
$=\mathrm{U}$ :: U ppertriangle of $A$ and $B$ are stored;
= L': Low ertriangle of A and B are stored.
$N$ (input) The orderof the $m$ atrix pencil $(A, B) . N>=0$.
AP (input/output)
Realarray, dim ension $\mathbb{N}^{*}(\mathbb{N}+1) / 2$ ) On entry, the upper or low er triangle of the sym $m$ etric $m$ atrix $A$, packed colum nw ise in a linear array. The jth colum $n$ ofA is stored in the array AP as follow $s$ : ifUPLO $=U{ }^{\prime}, A P(i+(j-1) * j 2)=A(i, j)$ for $1<=i<=j$ ifUPLO $=\mathrm{L}$ ', AP $\left(i+(j-1)^{\star}(2 \star \mathrm{n}-\bar{j}) / 2\right)=$ A $(i, 1)$ for $j=i<=n$.

On exit, the contents of AP are destroyed.

BP (input/output)
Realarray, dim ension $\mathbb{N} *(\mathbb{N}+1) / 2$ ) On entry, the upper or low ertriangle of the sym $m$ etric $m$ atrix $B$, packed colum nw ise in a linear array. The jth colum $n$ of is stored in the array $B P$ as follow $s$ : if UPLO $=U$ ', BP $(i+(j-1) * j 2)=B(i, j)$ for $1<=i<=j$ ifUPLO $=\mathrm{L}, \quad \mathrm{BP}\left(i+(j 1)^{*}\left(2{ }^{\star} \mathrm{n}-\mathrm{j}\right) / 2\right)=$ B $(i, 7)$ for $j=i<=n$.

On exit, the triangular factor $U$ or $L$ from the Cholesky factorization $\mathrm{B}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ orB $=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$, in the sam e storage form atas $B$.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE = A 'or I'.

VU (input)
See the description of V L .
II (input)
IfRA N G E= 'I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{Z}<=\mathbb{U}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{H}=1$
and $\mathbb{U}=0$ if $\mathrm{N}=0$. N otreferenced ifRANGE $=$ A 'or V'.

IU (input)
See the description of II .

ABSTOL (input)
The absolute emortolerance forthe eigenvalues.
A $n$ approxim ate eigenvalue is accepted as converged w hen it is determ ined to lie in an interval [a,b]
of w idth less than orequal to
$A B S T O L+E P S * \max (|a|, \mid)$,
w here EPS is them achine precision. IfABSTOL is less than or equal to zero, then EPS* $\mid /$ w illbe used in its place, w here $|T|$ is the 1 -norm of the tridiagonal $m$ atrix obtained by reducing $A$ to tridiagonalform.

E igenvalues w illbe com puted m ostaccurately when A BSTOL is set to tw ige the underflow threshold 2*SLAM CH (S ), notzero. If this routine retums w ith $\mathbb{N} \mathrm{FO}>0$, indicating that.som e eigenvectors did not converge, try setting ABSTOL to $2 *$ SLAM CH (S ).

M (output)
The totalnum ber ofeigenvalues found. $0<=\mathrm{M}<=$ N . IfRANGE=A', M=N, and ifRANGE= 'I'M = $\mathbb{U}-\mathbb{L}+1$ 。

W (output)
Realarray, dim ension $(\mathbb{N})$ On norm al exit, the first $M$ elem ents contain the selected eigenvahues in ascending order.

Z (output)
Realanay, dim ension ( $\mathrm{LD} \mathrm{Z}, \mathrm{m} \operatorname{ax}(1, \mathrm{M})$ ) If $\mathrm{OBZ}=$ $N^{\prime}$, then $Z$ is notreferenced. If $O B Z=V$ ', then if $\mathbb{N F O}=0$, the firstM colum ns of $Z$ contain the orthonorm al eigenvectors of the $m$ atrix $A$ corresponding to the selected eigenvalues, with the i-th colum $n$ of $Z$ holding the eigenvector associated w ith W (i). The eigenvectors are norm alized as follows: if ITYPE $=1$ or $2, Z * * T * B * Z=I$; if ITYPE $=3, Z * * T * \operatorname{inv}(B) * Z=I$ 。

If an eigenvector fails to converge, then that colum $n$ of $Z$ contains the latestapproxim ation to the eigenvector, and the index of the eigenvector
is retumed in $\mathbb{F A} \mathbb{I}$. N ote: the userm ustensure that at leastm ax $(1, M)$ ) 0 lum ns are supplied in the array $Z$; ifRANGE = V', the exactvalue ofM is not know $n$ in advance and an upper bound $m$ ust be used.

LD $Z$ (input)
The leading dim ension of the array $\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD Z $>=\mathrm{max}(1, \mathrm{~N})$.

W ORK (w orkspace)
Realarray, dim ension ( $8 \star \mathrm{~N}$ )

IW ORK (w orkspace)
$\mathbb{N}$ TEGER anay, dim ension (5*N )
FAII (output)
$\mathbb{I N}$ TEGER array, dim ension $\mathbb{N}$ ) If $\operatorname{JO}$ B Z = $V$ ', then
if $\mathbb{N} F O=0$, the firstM elem ents of $\mathbb{F} A \mathbb{I}$ are
zero. If $\mathbb{N F O}>0$, then $\mathbb{F A} \mathbb{I}$ contains the indices of the eigenvectors that failed to con-
verge. If $J O B Z=N$ ', then $\mathbb{F A} \mathbb{I}$ is not referenced.
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0$ : if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvahue
> 0: SPPTRF orSSPEVX retumed an errorcode:
$<=\mathrm{N}:$ if $\mathbb{N} F O=\mathrm{i}, \mathrm{SSPEVX}$ failed to converge; i eigenvectors failed to converge. Their indices are stored in array $\mathbb{F} A \mathbb{I} .>N$ : if $\mathbb{N} F O=\mathrm{N}+$ $i$, for $1<=i<=N$, then the leading $m$ inor of orderiofB is notpositive definite. The factorization of B could not be com pleted and no eigenvalues or eigenvectors w ere com puted.

## FURTHER DETAILS

B ased on contributions by
M ark Fahey, Departm entofM athem atics, U niv. of K entucky, U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sspm $v$-perform them atrix-vector operation $y:=$ alpha* $A * x$

+ beta* $y$


## SYNOPSIS

```
SUBROUT\mathbb{NE SSPMV (UPLO,N,ALPHA,A,X,NNCX,BETA,Y,INCY)}
CHARACTER * 1 UPLO
INTEGERN,INCX,\mathbb{NCY}
REAL ALPHA,BETA
REAL A (*),X (*),Y (*)
SU BROUT\mathbb{NE SSPM V_64 (UPLO,N,A LPHA,A,X, INCX,BETA,Y, INCY)}
CHARACTER * 1 UPLO
INTEGER*8N, \mathbb{NCX,INCY}
REAL ALPHA,BETA
REAL A (*),X (*),Y (*)
```


## F95 INTERFACE

```
SU BROUTINE SPMV (UPLO,N,ALPHA,A,X, [NCX],BETA,Y,[INCY])
CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y\)
REAL ::ALPHA,BETA
REAL,D \(\mathbb{M}\) ENSION (:) ::A, X,Y
SU BROUTINE SPMV_64 (UPLO, N,ALPHA, A, X, [ \(\mathbb{N C X}], B E T A, Y,[\mathbb{N} C Y])\)
CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R(8):: N, \mathbb{N} C X, \mathbb{N} C Y\)
REAL ::ALPHA,BETA
```

REAL,D $\mathbb{M}$ ENSION (:) ::A $, \mathrm{X}, \mathrm{Y}$

## C INTERFACE

\#include <sunperfh>
void sspm v (charuplo, intn, floatalpha, float *a, float * $x$, int incx, floatbeta, float *y, intincy);
void sspm v_64 (charuplo, long n, float alpha, float *a, float *x, long incx, floatbeta, float *y, long incy);

## PURPOSE

sspm v perform s the $m$ atrix-vector operation $y:=$ alpha*A *x + beta* $y$, w here alpha and beta are scalars, $x$ and $y$ are $n$ ele$m$ entvectors and $A$ is an $n$ by $n$ sym $m$ etric $m$ atrix, supplied in packed form.

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the m atrix A is supplied in the packed aray A as follow s:
$\mathrm{UPLO}=\mathrm{U}$ 'or U ' The uppertriangularpartof $A$ is supplied in A.
$\mathrm{UPLO}=\mathrm{L}$ 'or $\mathrm{I}^{\prime}$ ' The low er triangularpart of A is supplied in A.

U nchanged on exit.

N (input)
O n entry, $N$ specifies the order of the $m$ atrix A. $\mathrm{N}>=0$. U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
$((n *(n+1)) / 2)$. Before entry w ith UPLO = $U$ ' or $G$ ', the aray A m ustcontain the upper triangularpart of the sym $m$ etric $m$ atrix packed sequentially, colum $n$ by colum $n$, so thatA (1)
containsa(1,1), A (2) and A (3) contain a( 1,2 ) and a( 2,2 ) respectively, and so on. Before entry w ith UPLO = L'or 1', the array A $m$ ustcontain the low er triangularpartof the sym $m$ etric $m$ atrix packed sequentially, column by colum $n$, so thatA (1) contains a ( 1,1 ), A (2) and A (3) contain a $(2,1)$ and $a(3,1)$ respectively, and so on. U nchanged on exit.

X (input)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C X))$. Before entry, the increm ented array $X$ must contain the $n$ elem ent vectorx. U nchanged on exit.
$\mathbb{N} C X$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents ofX. $\mathbb{N} C X<>0$. U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then $Y$ need notbe seton input. U nchanged on exit.

Y (input/output)
$(1+(n-1) \star a b s(\mathbb{N} C Y))$. Before entry, the increm ented array $Y$ m ust contain the $n$ elem ent vectory. On exit, $Y$ is overw rilten by the updated vectory.
$\mathbb{N} C Y$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $Y . \mathbb{N} C Y$ <> 0 . U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sspr-perform the symmetric rank 1 operation A := alpha*x*x'+A

## SYNOPSIS

```
SUBROUT\mathbb{NE SSPR (UPLO,N,ALPHA,X,INCX,A)}
CHARACTER * 1 UPLO
\mathbb{NTEGER N, INCX}
REAL ALPHA
REALX (*),A (*)
SU BROUTINE SSPR_64(UPLO,N,ALPHA,X,\mathbb{NCX ,A)}
CHARACTER * 1 UPLO
INTEGER*8 N, INCX
REALALPHA
REALX (*),A (*)
F95 INTERFACE
```



```
CHARACTER (LEN=1)::UPLO
\mathbb{NTEGER ::N,\mathbb{NCX}}\mathbf{N}=
REAL ::ALPHA
REAL,DIM ENSION (:) ::X,A
```



```
CHARACTER (LEN=1)::UPLO
INTEGER (8)::N,\mathbb{NCX}
REAL ::ALPHA
```

REAL,D $\mathbb{I}$ ENSION (:) :: X,A

## C INTERFACE

\#include <sunperfh>
void sspr(charuplo, intn, floatalpha, float *x, intincx, float*a);
void sspr_64 (charuplo, long n, floatalpha, float *x, long incx, float*a);

## PURPOSE

ssprperform sthe sym $m$ etric rank 1 operation $A:=a \prod p h a * x^{*} x^{\prime}$ $+A$, where alpha is a realscalar, $x$ is an $n$ elem ent vector and $A$ is an $n$ by $n$ sym $m$ etric $m$ atrix, supplied in packed form.

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the $m$ atrix $A$ is supplied in the packed array A as follow s:
$\mathrm{UPLO}=\mathrm{U}$ 'or $\mathrm{L}^{\prime}$ ' The uppertriangularpartofA is supplied in A.

UPLO = L'or I' The low ertriangularpartof A is supplied in A.

U nchanged on exit.

N (input)
On entry, N specifies the order of the m atrix A . $\mathrm{N}>=0$. U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
$(1+(n-1) * a b s(\mathbb{N} C X))$. Before entry, the increm ented array $X$ m ust contain the $n$ elem ent vectorx. U nchanged on exit.

On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents ofX. $\mathbb{N} C X<>0$. U nchanged on exit.

A (input/output)
$\left(\left(n^{*}(n+1)\right) / 2\right)$. Before entry $w$ ith UPLO $=$ U ' or G ', the anay A m ustcontain the upper triangularpartof the symm etric $m$ atrix packed sequentially, column by colum n, so thatA (1) containsa(1,1), A (2) and A (3) contain a( 1,2 ) and a (2,2) respectively, and so on. On exit, the array A is overw rilten by the upper triangular part of the updated $m$ atrix. Before entry with UPLO = L'or I', the array A m ust contain the low er triangularpart of the sym $m$ etric $m$ atrix packed sequentially, colum $n$ by colum $n$, so that A ( 1 ) contains a ( 1,1 ), A (2) and A (3) contain $a(2,1)$ and $a(3,1)$ respectively, and so on. On exit, the array A is overw ritten by the low er triangular part of the updated $m$ atrix.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sspr2-perform the symmetric rank 2 operation A := alpha*x*y'+ alpha*y*x'+ A

## SYNOPSIS

```
SUBROUT\mathbb{NE SSPR2 (UPLO,N,ALPHA,X, NNCX,Y, INCY,AP)}
CHARACTER * 1 UPLO
INTEGERN,INCX,\mathbb{NCY}
REAL ALPHA
REAL X (*),Y (*),AP (*)
SU BROUT\mathbb{NE SSPR2_64(UPLO,N,ALPHA,X, INCX,Y, INCY,AP)}
CHARACTER * 1 UPLO
INTEGER*8N, INCX,INCY
REAL A LPHA
REAL X (*),Y (*),AP (*)
```


## F95 INTERFACE

```
SU BROUTINE SPR2 (UPLO, \(\mathbb{N}], A L P H A, X,[\mathbb{N} C X], Y,[\mathbb{N} C Y], A P)\)
CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y\)
REAL ::ALPHA
REAL,D \(\mathbb{I}\) ENSION (:) ::X,Y,AP
SUBROUTINE SPR2_64 (UPLO, \(\mathbb{N}\) ],ALPHA, \(X,[\mathbb{N C X}], Y,[\mathbb{N C Y}], A P)\)
CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R(8):: N, \mathbb{I N C X}, \mathbb{N} C Y\)
REAL ::ALPHA
```

REAL,D $\mathbb{I}$ ENSION (:) :: X,Y,AP

## C INTERFACE

\#include <sunperfh>
void sspr2 (char uple, intn, float alpha, float *x, int incx, float *y, intincy, float *ap);
void sspr2_64 (charuplo, long n, floatalpha, float *x, long incx, float *y, long incy, float *ap);

## PURPOSE

sspr2 performs the symmetric rank 2 operation $A:=$ alpha* $x^{*} y^{\prime}+$ alpha* $y^{\star} x^{\prime}+A$, w here alpha is a scalar, $x$ and $y$ are $n$ elem ent vectors and $A$ is an $n$ by $n$ sym $m$ etric $m$ atrix, supplied in packed form .

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the m atrix $A$ is supplied in the packed anay A as follow s:
$\mathrm{UPLO}=\mathrm{U}$ 'or G ' The uppertriangularpartof $A$ is supplied in A P .

UPLO = L 'or I' The low ertriangularpart of A is supplied in A P .

U nchanged on exit.

N (input)
O n entry, $N$ specifies the order of the $m$ atrix A. $\mathrm{N}>=0$. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
Realaray, dim ension $(1+(n-1) \star \operatorname{abs}(\mathbb{N} C X))$ B efore entry, the increm ented array $X$ m ust contain the $n$ elem entvector $x$. U nchanged on exit.

On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of $. \mathbb{I N} C X<>0$. U nchanged on exit.

Y (input)
Realarray, dim ension ( $1+(n-1) * a b s(\mathbb{N} C Y)$ )
Before entry, the increm ented array $Y$ m ust contain the $n$ elem entvectory. U nchanged on exit.
$\mathbb{N C Y}$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $Y . \mathbb{N} C Y$ <> 0 . Unchanged on exit.
AP (input/output)
Realarray, dim ension (( $n *(n+1)) / 2$ ) Before
entry with UPLO = U 'or L ', the aray AP must contain the upper triangular part of the sym $m$ etric $m$ atrix packed sequentially, colum $n$ by colum n, so that A P (1) contains a ( 1,1 ), AP (2) and AP (3 ) contain $a(1,2)$ and $a(2,2)$ respectively, and so on. On exit, the array A is overw rilten by the upper triangularpart of the updated $m$ atrix. Before entry w ith UPLO = L 'or I', the amay AP $m$ ust contain the low er triangularpart of the sym $m$ etric $m$ atrix packed sequentially, column by colum $n$, so thatAP (1) contains a ( 1,1 ), AP (2 ) and AP (3) contain $a(2,1)$ and $a(3,1)$ respectively, and so on. On exit, the array AP is overw ritten by the low er triangular part of the updated $m$ atrix.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssprfs -im prove the com puted solution to a system of linear equations $w$ hen the coefficientm atrix is sym $m$ etric indefinite and packed, and provides errorbounds and backw ard error estim ates for the solution

## SYNOPSIS

```
SUBROUT\mathbb{NE SSPRFS (UPLO,N,NRHS,AP,AF,\mathbb{PIVOT,B,LDB,X,LDX,FERR,}}\mathbf{N},\textrm{N},\textrm{L}
    BERR,W ORK,W ORK2,INFO)
CHARACTER * 1 UPLO
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*),W ORK2(*)}
REALAP (*),AF (*), B (LDB ,*), X (LDX ,*), FERR (*), BERR (*),
W ORK (*)
SU BROUTINE SSPRFS_64 (UPLO,N,NRHS,AP,AF,\mathbb{PIVOT,B,LDB,X,LDX,}
    FERR,BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 UPLO
INTEGER*8N,NRHS,LDB,LDX,\mathbb{NFO}
INTEGER*8 P\mathbb{IVOT (*),W ORK2 (*)}
REALAP (*),AF (*), B (LDB,*), X (LDX ,*), FERR (*), BERR (*),
W ORK (*)
```


## F95 INTERFACE

SU BROUTINE SPRFS (UPLO,N, $\mathbb{N} R H S], A P, A F, \mathbb{P} \mathbb{I} O T, B,[L D B], X,[L D X]$, FERR, BERR, [W ORK], [W ORK2], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1)::UPLO
$\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}, \mathrm{W} O R K 2$

SUBROUTINE SPRFS_64 (UPLO, N, $\mathbb{N} R H S], A P, A F, \mathbb{P} \mathbb{I} V O T, B,[L D B], X,[L D X]$, FERR, BERR, [W ORK], [W ORK 2], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) :: UPLO
$\mathbb{N}$ TEGER (8) :: N , NRHS,LDB,LDX, $\mathbb{N}$ FO
$\mathbb{I N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T$, W ORK 2
REAL, D $\mathbb{M} E N S I O N(:):: A P, A F, F E R R, B E R R, W$ ORK
REAL,D $\mathbb{M}$ ENSION (: : : : : B , X

## C INTERFACE

\#include < sunperfh>
void ssprfs (charuplo, intn, int nrhs, float *ap, float *af, int *ipivot, float *b, int ldb, float *x, int ldx, float *fers, float *berr, int *info);
void ssprfs_64 (charuplo, long n, long nihs, float *ap, float *af, long *ipivot, float *b, long ldb, float *x, long ldx, float *ferr, float *berr, float *w ork, long *iw ork, long *info);

## PURPOSE

ssprfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is sym $m$ etric indefinite and packed, and provides errorbounds and backw ard error estim ates for the solution.

## ARGUMENTS

UPLO (input)
= U ': U ppertriangle ofA is stored;
= LL': Low ertriangle ofA is stored.

N (input) The order of the $m$ atrix $\mathrm{A} . \mathrm{N}>=0$.
NRHS (input)
The num ber of righthand sides, i.e., the num ber
of collm ns of the m atrioes B and X. NRHS $>=0$.
A (input) Realarray, dim ension $\mathbb{N}^{*}(\mathbb{N}+1) / 2$ ) The upper or low er triangle of the sym $m$ etric $m$ atrix $A$, packed colum nw ise in a linearamay. The jth colum $n$ of A is stored in the amay A as follow S : if UPLO $=$ U',AP ( $i+(j-1) * j 2)=A(i, j)$ for $1<=i<=j$ if
$\mathrm{UPLO}=\mathrm{L}^{\prime}, \mathrm{AP}(i+(j-1) \star(2 \star \mathrm{n}-\mathrm{j} / 2)=\mathrm{A}(i, j)$ for $j<=\mathrm{i}<=\mathrm{n}$ 。

AF (input)
Realanray, dim ension $\left.\mathbb{N}^{*}(\mathbb{N}+1) / 2\right)$ The factored form of the m atrix A. AF contains the block diagonalm atrix $D$ and the $m$ ultipliers used to obtain the factor $U$ or $L$ from the factorization $A=$ $\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$ orA $=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ as com puted by SSPTRF , stored as a packed triangularm atrix.

IPIVOT (input)
Integer array, dim ension $(\mathbb{N})$ D etails of the interchanges and the block structure of D as determ ined by SSPTRF .

B (input) Realamay, dim ension (LD B NRHS) The right hand side m atrix B .

LD B (input)
The leading dim ension of the array B . LD B $>=$ $\max (1, N)$.

X (input/output)
Realarray, dimension (LDX,NRHS) On entry, the solution matrix X , as com puted by SSPTRS. On exit, the im proved solution $m$ atrix $X$.

## LD X (input)

The leading dim ension of the array X. LD X >= $\max (1, N)$.

## FERR (output)

Realaray, dim ension $\mathbb{N}$ RH S) The estim ated forw ard error bound foreach solution vectorX ( $\mathcal{1}$ ) (the $j$ th colum $n$ of the solution $m$ atrix $X$ ). IfX TRUE is the true solution corresponding to $X(7), \operatorname{FERR}(7)$ is an estim ated upperbound for the $m$ agnitude of the largest elem entin (X ( 7 ) - X TRUE) divided by them agnitude of the largestelem ent in $X$ ( $\mathcal{I}$ ). The estim ate is as reliable as the estim ate forRCOND, and is alm ostalw ays a slight overestim ate of the true enror.

BERR (output)
Realarray, dimension $\mathbb{N} R H S)$ The com ponentw ise relative backw ard emor ofeach solution vector X (i) (ie., the sm allestrelative change in any elem ent of A orB thatm akes X (i) an exactsolution).

W ORK (w orkspace)
Realanray, dim ension ( $3 * N$ )

W ORK 2 (w orkspace)
Integer amay, dim ension $(\mathbb{N})$
INFO (output) Integer
= 0: successfulexit
$<0:$ if $\mathbb{N ~ F O ~}=-$ i, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

sspsv - com pute the solution to a real system of linear equations $A * X=B$,

## SYNOPSIS



```
CHARACTER * 1 UPLO
\mathbb{NTEGER N,NRHS,LDB,INFO}
INTEGER \mathbb{PIVOT (*)}
REAL AP (*),B (LDB,*)
SUBROUTINE SSPSV_64(UPLO,N,NRHS,AP,\mathbb{P}\mathbb{IVOT,B,LDB,\mathbb{NFO)}}\mathbf{(N,N}
CHARACTER * 1 UPLO
INTEGER*8N,NRHS,LDB,INFO
INTEGER*8 \mathbb{PIVOT (*)}
REAL AP (*),B (LDB,*)
```


## F95 INTERFACE

```
SU BROUTINE SPSV (UPLO,N, NRHS],AP, \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}],[\mathbb{N} F O])\)
CHARACTER (LEN=1)::UPLO
\(\mathbb{N}\) TEGER :: N, NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V} O T\)
REAL,D IM ENSION (:) ::AP
REAL,D IM ENSION (:,:) ::B
SU BROUT \(\left.\mathbb{N} E S P S V \_64(U P L O, N, N R H S], A P, \mathbb{P} \mathbb{I V} O T, B,[L D B],[\mathbb{N} F O]\right)\)
CHARACTER (LEN=1) ::UPLO
```

$\mathbb{N}$ TEGER (8) :: N , NRHS,LD B, $\mathbb{N} F O$
$\mathbb{I N}$ TEGER (8), D $\mathbb{M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT}$
REAL,D $\mathbb{I}$ ENSION (:) ::AP
REAL,D IM ENSION (:,:) ::B

## C INTERFACE

\#include <sunperfh>
void sspsv (charuplo, int n, int nrhs, float *ap, int *ipivot, float *b, int ldb, int *info);
void sspsv_64 (char uple, long n, long nrhs, float *ap, long *ịivot, float *b, long ldb, long *info);

## PURPOSE

sspsv com putes the solution to a realsystem of linear equations
$A$ * $X=B$, where $A$ is an $N$ boy $N$ sym m etric $m$ atrix stored in packed form at and X and B are N -by -N RH S m atrices.

The diagonalpivoting $m$ ethod is used to factorA as
$A=U * D * U * * T$, if $U P L O=U$ ', or
$A=L * D * L * * T$, if $U P L O=L '$,
where U (orL) is a productofperm utation and unit upper (low er) triangularm atrioes, $D$ is sym m etric and block diagonalw th 1-by-1 and 2 -by-2 diagonal blocks. The factored form ofA is then used to solve the system of equations A * $X=B$.

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': U pper triangle of A is stored;
= L': Low er triangle ofA is stored.

N (input) The num ber of linearequations, ie., the order of them atrix $A . N>=0$.

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the m atrix B. NRHS $>=0$.

A P (input/output)
Realamay, dim ension $(\mathbb{N} *(\mathbb{N}+1) / 2$ ) On entry, the upper or low er triangle of the sym $m$ etric $m$ atrix $A$, packed colum nw ise in a linear anray. The jth
column of A is stored in the array AP as follow s : ifUPLO $=U$ ',AP $(i+(j-1) * j 2)=A(i, j)$ for $1<=\mathrm{i}<=\dot{j}$ if UPLO = L',AP (i+ (j-1)* (2n-j)/2)= A $(i, j)$ for $j=i<=n$. See below for further details.

On exit, theblock diagonalm atrix $D$ and the $m u l$ tipliers used to obtain the factor $U$ orl from the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$ orA $=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ as com puted by SSPTRF, stored as a packed triangular $m$ atrix in the sam e storage form atasA.

## IPIVOT (output)

Integer anray, dim ension $\mathbb{N}$ ) D etails of the interchanges and the block structure of $D$, as determ ined by SSPTRF. If IPIV OT $(k)>0$, then row $s$ and colum nsk and $\mathbb{P} \mathbb{I} O T(k)$ were interchanged, and $D(k, k)$ is a 1-by-1 diagonalblock. If $U P L O=U^{\prime}$ and $\mathbb{P} \mathbb{V} O T(k)=\mathbb{P} \mathbb{I V} O T(k-1)<0$, then row sand colum nsk-1 and $-\mathbb{P} \mathbb{I V O T}(k)$ were interchanged and D ( $k-1$ k, $k-1$ k) is a 2 -by-2 diagonalblock. If $\mathrm{UPLO}=\mathrm{L}$ 'and $\mathbb{P} \mathbb{I V O T}(k)=\mathbb{P} \mathbb{V} O \mathrm{~T}(k+1)<0$, then row s and colum ns $k+1$ and $-\mathbb{P}$ IV OT ( $k$ ) were interchanged and $D(k: k+1, k: k+1)$ is a $2-b y-2$ diagonal block.

B (input/output)
Realarray, dim ension (LDB , NRHS) On entry, the N -by-NRHS right hand side matrix B. On exit, if $\mathbb{N} F O=0$, the $N$-by-N RH S solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the array $\mathrm{B} . \mathrm{LD} \mathrm{B}>=$ $\max (1, N)$.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{I N}$ FO = -i, the i-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N F O}=\mathrm{i}, \mathrm{D}(\mathrm{i}, \mathrm{i})$ is exactly zero. The factorization has been com pleted, but the block diagonalm atrix $D$ is exactly singular, so the solution could notbe com puted.

## FURTHER DETAILS

The packed storage schem e is illustrated by the follow ing exam ple when $N=4$, UPLO = U':

Tw o-dim ensional storage of the sym m etric m atrix A:
a11 a12 a13 a14

```
a22 a23 a24
    a33 a34 (aij= aji)
        a44
```

Packed storage of the upper triangle ofA :

$$
A P=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]
$$

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

```
sspsvx - use the diagonal pivoting factorization A =
U *D *U **T or A = L *D *L**T to com pute the solution to a real
system of linearequations A * X = B,where A is an N by N
symm etric m
by-N RH S m atrices
```


## SYNOPSIS

```
SUBROUT\mathbb{NE SSPSVX (FACT,UPLO,N,NRHS,AP,AF, IPIVOT,B,LDB,X,LDX,}
    RCOND,FERR,BERR,W ORK,W ORK2,INFO)
CHARACTER * 1 FACT,UPLO
\mathbb{NTEGERN,NRHS,LDB,LDX,}\mathbb{N}FO
INTEGER \mathbb{PIVOT (*),W ORK2(*)}
REAL RCOND
REALAP (*),AF (*), B (LDB,*), X (LDX ,*), FERR (*), BERR (*),
W ORK (*)
```



```
    LDX,RCOND,FERR,BERR,WORK,W ORK 2, INFO)
CHARACTER * 1FACT,UPLO
\mathbb{NTEGER*8N,NRHS,LDB,LDX,}\mathbb{N}FO
INTEGER*8 P\mathbb{IVOT (*),W ORK2 (*)}
REAL RCOND
REALAP (*),AF (*), B (LDB,*), X (LDX ,*), FERR (*), BERR (*),
W ORK (*)
```


## F95 INTERFACE

SUBROUTINE SPSVX (FACT, UPLO,N, $\mathbb{N R H S}], A P, A F, \mathbb{P} \mathbb{I V O T}, \mathrm{~B},[\operatorname{LDB}], \mathrm{X}$, [LDX],RCOND,FERR,BERR, [W ORK], [W ORK2], [ $\mathbb{N F O}])$

CHARACTER (LEN=1) ::FACT, UPLO
$\mathbb{N}$ TEGER :: N,NRHS,LDB,LDX, $\mathbb{N} F O$
$\mathbb{N}$ TEGER,D $\mathbb{I M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}, \mathrm{W} O R K 2$
REAL ::RCOND
REAL, D $\mathbb{M}$ ENSION (:) ::AP, AF,FERR,BERR,WORK
REAL,D $\mathbb{M}$ ENSION (: $:$ : : : $\mathrm{B}, \mathrm{X}$

SU BROUTINE SPSVX_64 (FACT, UPLO,N, $\mathbb{N} R H S], A P, A F, \mathbb{P} \mathbb{I} O T, B,[L D B], X$, [LDX ],RCOND ,FERR,BERR, [WORK], [WORK2], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) ::FACT,UPLO
$\mathbb{N}$ TEGER (8) :: N , NRHS,LDB,LDX, $\mathbb{N}$ FO
$\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T, W$ ORK 2
REAL ::RCOND
REAL, D $\mathbb{M}$ ENSION (:) ::AP, AF,FERR,BERR,W ORK
REAL,D $\mathbb{M}$ ENSION (: : : : : B , X

## C INTERFACE

\#include <sunperfh>
void sspsvx (char fact, charuplo, int n, int nrhs, float
*ap, float *af, int*ipivot, float*b, int ldb, float*x, int ldx, float *rcond, float *ferr, float *berr, int*info);
void sspsvx_64 (char fact, char uplo, long n, long nrhs, float *ap, float *af, long *íivot, float *b, long ldb, float *x, long ldx, float *rcond, float * ferr, float *berr, long *info);

## PURPOSE

SSPSVX uses the diagonalpivoting factorization $A=U * D * U * * T$ or $A=L * D * L * * T$ to com pute the solution to a realsystem of linear equations $A * X=B$, where $A$ is an $N$ by $-N$ symm etric $m$ atrix stored in packed form atand $X$ and $B$ are $N$ boy-NRHS $m$ atrices.

Errorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:

1. IfFACT $=N$ ', the diagonalpivoting $m$ ethod is used to factorA as
$A=U * D * U * * T$, if $U P L O=U$ ', or
$A=L * D * L * * T$, if $U P L O=L^{\prime}$ ',
where $U$ (orL) is a productofperm utation and unitupper (low er)
triangularm atrioes and $D$ is sym $m$ etric and block diagonal w ith

1-by-1 and 2 -by-2 diagonalblocks.
2. If som eD $(i, i)=0$, so thatD is exactly singular, then the routine
retums w ith $\mathbb{N}$ FO $=$ i. $O$ therw ise, the factored form of $A$ is used
to estim ate the condition num ber of the $m$ atrix $A$. If the reciprocal of the condition num ber is less than $m$ achine precision,
$\mathbb{N} F O=N+1$ is retumed as a waming, but the routine stillgoes on
to solve for $X$ and com pute error bounds as described below.
3. The system ofequations is solved for $X$ using the factored form of A.
4. Iterative refinem ent is applied to im prove the com puted solution
$m$ atrix and calculate error bounds and backw ard error estim ates
for it.

## ARGUMENTS

FACT (input)
Specifies whether ornot the factored form of A has been supplied on entry. = F ': On entry, A F and $\mathbb{P} \mathbb{I V O T}$ contain the factored form of A. AP, AF and $\mathbb{P} \mathbb{I} O T \mathrm{w}$ ill not be modified. $=\mathrm{N}$ : The $m$ atrix A w illlbe copied to A F and factored.

UPLO (input)
$=\mathrm{U}:$ : U pper triangle ofA is stored;
= L': Low ertriangle ofA is stored.
N (input) The num ber of linear equations, ie., the order of them atrix $A . N>=0$.

NRHS (input)
The num ber of righthand sides, ie., the num ber
of collum ns of the $m$ atrices $B$ and $X$. NRHS $>=0$.

AP (input)
Realarray, dimension $\mathbb{N}^{*}(\mathbb{N}+1) / 2$ ) The upper or low er triangle of the sym $m$ etric $m$ atrix $A$, packed
colum nw ise in a linearamay. The jth colum n of A is stored in the amray AP as follow s: if UPLO = U',AP $(i+(j-1) * j 2)=A(i, j)$ for $1<=i<=j$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{AP}(i+(j-1) \star(2 \star \mathrm{n}-\mathrm{J}) / 2)=\mathrm{A}(i, 7)$ for $j=i<=n$. See below for furtherdetails.

AF (input/output)
Realarray, dim ension $(\mathbb{N} *(N+1) / 2)$ If $F A C T=F^{\prime}$, then $A F$ is an inputargum ent and on entry contains the block diagonalm atrix D and the multipliers used to obtain the factor $U$ orL from the factorization $A=U * D * U * * T$ or $A=L * D{ }^{*} L * * T$ as com puted by SSPTRF, stored as a packed triangularm atrix in the sam e storage form at as A.

IfFACT $=\mathrm{N}$ ', then AF is an output argum ent and on exit contains the block diagonalm atrix D and the multipliers used to obtain the factorU or L from the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$ or $\mathrm{A}=$ $\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ as com puted by SSPTRF, stored as a packed triangularm atrix in the sam e storage form at as A.

## IPIVOT (inputoroutput)

Integer aray, dim ension $\mathbb{N}$ ) IfFACT=F', then
$\mathbb{P} \mathbb{I} O T$ is an inputargum entand on entry contains details of the interchanges and the block structure ofD, as determ ined by SSPTRF. If $\mathbb{P} \mathbb{I V} O T(k)$ $>0$, then row sand colum nsk and $\mathbb{P} \mathbb{I} O T(k)$ were interchanged and $\mathrm{D}(\mathrm{k}, \mathrm{k})$ is a 1 -by -1 diagonal block. If $\mathrm{PLO}=\mathrm{U}$ 'and $\mathbb{P} \mathbb{I V O T}(\mathrm{k})=\mathbb{P} \mathbb{I V O T}(\mathrm{k}-1)$ $<0$, then row s and colum ns k-1 and - $\mathbb{P} \mathbb{I V}$ O T (k) w ere interchanged and $D(k-1 k, k-1 k)$ is a 2 -by-2 diagonal block. If UPLO = L ' and $\mathbb{P} \mathbb{I V O T}(\mathrm{k})=$ $\mathbb{P I V O T}(k+1)<0$, then row $s$ and colum ns $k+1$ and $-\mathbb{P}$ IV O T (k) w ere interchanged and $D(k \cdot k+1, k \cdot k+1)$ is a 2-by-2 diagonalblock.

IfFACT = $N$ ', then $\mathbb{P I V O T}$ is an output argum ent and on exit contains details of the interchanges and the block structure of D, as determ ined by SSPTRF.
$B$ (input) Realarray, dimension (LDB,NRHS) The $N$ boy-NRHS righthand side $m$ atrix $B$.

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, \mathbb{N})$.

X (output)

Realarray, dim ension (LDX,NRHS) If $\mathbb{N} F O=0$ or $\mathbb{N} F O=N+1$, the $N$-by-N RH S solution $m$ atrix $X$.

LD X (input)
The leading dim ension of the array X . LD X >= $\max (1, N)$.

RCOND (output)
The estim ate of the reciprocalcondition num ber of the $m$ atrix $A$. IfRCOND is less than the $m$ achine precision (in particular, if RCOND $=0$ ), the $m$ atrix is singular to working precision. This condition is indicated by a retum code of $\mathbb{N}$ FO > 0.

FERR (output)
Realarray, dim ension $\mathbb{N} R H S$ ) The estim ated forw ard error bound foreach solution vectorX ( $\mathcal{j}$ ) the $j$
th colum $n$ of the solution $m$ atrix $X$ ). IfX TRUE is the true solution comesponding to $X(\mathcal{j}), \operatorname{FERR}(\mathcal{)}$ is an estim ated upperbound for the $m$ agnitude of the largest elem ent in ( $X(\mathcal{j})$-X TRUE) divided by the $m$ agnitude of the largestelem ent in $X(\mathcal{j})$. The estim ate is as reliable as the estim ate forRCOND, and is alm ostalw ays a slightoverestim ate of the true error.

## BERR (output)

Realarray, dim ension $\mathbb{N} R H S$ ) The com ponentw ise relative backw ard error ofeach solution vector
$\mathrm{X}(\mathcal{)})$ (i.e., the sm allestrelative change in any elem ent of A orB thatm akes X ( $\mathcal{j}$ ) an exact solution).

W ORK (w orkspace)
Realamay, dim ension ( $3{ }^{*} \mathrm{~N}$ )
W ORK 2 (w orkspace)
Integer array, dim ension (N)
$\mathbb{I N F O}$ (output)
$=0$ : successfulexit
$<0$ : if $\mathbb{I N F O}=-$ i, the $i$-th argum enthad an illegalvalue
>0: if $\mathbb{N F O}=i$, and $i$ is
$<=\mathrm{N}: \mathrm{D}(i, i)$ is exactly zero. The factorization has been completed but the factorD is exactly singular, so the solution and error bounds could not be com puted. RCOND $=0$ is retumed. $=\mathrm{N}+1: \mathrm{D}$ is nonsingular, butRCOND is less than machine
precision, $m$ eaning that the $m$ atrix is singular to w orking precision. N evertheless, the solution and error bounds are com puted because there are a num ber of situations w here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

## FURTHER DETAILS

The packed storage schem e is illustrated by the follow ing exam ple when $N=4$, UPLO $=U$ ':
Tw o-dim ensional storage of the sym $m$ etric $m$ atrix A :

```
al1 a12 al3 a14
    a22 a23 a24
        a33 a34 (aij= aj̈)
            a44
```

Packed storage of the upper triangle ofA :
$A P=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]$

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

ssptrd - reduce a realsym m etric m atrix A stored in packed form to sym $m$ etric tridiagonal form $T$ by an orthogonalsim ilarity transform ation

## SYNOPSIS

```
SUBROUT\mathbb{NE SSPTRD(UPLO,N,AP,D,E,TAU,INFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGERN, INFO}
REALAP (*),D (*),E (*),TAU (*)
SUBROUT\mathbb{NE SSPTRD_64(UPLO,N,AP,D ,E,TAU , INFO)}
CHARACTER * 1 UPLO
INTEGER*8N,\mathbb{NFO}
REALAP (*),D (*),E (*),TAU (*)
F95 INTERFACE
    SUBROUT\mathbb{NE SPTRD (UPLO,N,AP,D,E,TAU, [NFO])}
    CHARACTER (LEN=1) ::UPLO
    INTEGER::N,\mathbb{NFO}
    REAL,D IM ENSION (:) ::AP,D,E,TAU
```



```
    CHARACTER (LEN=1) ::UPLO
    \mathbb{NTEGER (8)::N,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
    REAL,D IM ENSION (:) ::AP,D,E,TAU
```


## C INTERFACE

\#include <sunperfh>
void ssptrd (charuplo, intn, float *ap, float *d, float *e,
float *tau, int *info);
void ssptrd_64 (charuple, long n, float *ap, float *d, float
*e, float *tau, long *info);

## PURPOSE

ssptrd reduces a realsym m etric matrix A stored in packed form to symm etric tridiagonal form T by an orthogonalsim ilarity transform ation: $\mathrm{Q} * * \mathrm{~T} * \mathrm{~A} * \mathrm{Q}=\mathrm{T}$.

## ARGUMENTS

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low er triangle of A is stored.

N (input) The order of the matrix A. $\mathrm{N}>=0$.

AP (input)
Realarray, dim ension $(\mathbb{N} *(N+1) / 2$ ) On entry, the upper or low er triangle of the sym $m$ etric $m$ atrix $A$, packed colum nw ise in a linear array. The $j$ th colum $n$ ofA is stored in the array AP as follow s : ifUPLO $=U{ }^{\prime}, \mathrm{AP}(i+(j-1) * j 2)=A(i, j)$ for $1<=i<=j$ ifUPLO $=\mathrm{L}$ ', AP $\left(i+(j-1)^{*}\left(2{ }^{\star} \mathrm{n}-j\right) / 2\right)=$ A $(i, j)$ for $j=i<=n$. On exit, ifUPLO $=U '$, the diagonal and first superdiagonal of $A$ are overw ritten by the conesponding elem ents of the tridiagonal $m$ atrix $T$, and the elem ents above the first superdiagonal, w th the array TA U, represent the orthogonalm atrix $Q$ as a product ofelem entary reflectors; if UPLO $=\mathrm{L}$ ', the diagonal and first subdiagonal of A are over-w ritten by the corresponding elem ents of the tridiagonal m atrix T , and the elem ents below the firstsubdiagonal, w ith the array TAU, represent the orthogonal $m$ atrix $Q$ as a productofelem entary reflectors. See FurtherD etails.

D (output)
Realarray, dim ension $(\mathbb{N})$ The diagonalelem ents of the tridiagonalm atrix $T: D(i)=A(i, i)$.

E (output)
Realaray, dim ension $(\mathbb{N}-1)$ The off-diagonal ele$m$ ents of the tridiagonal $m$ atrix $T: E(i)=$ $A(i, i+1)$ if $U P L O=U^{\prime}, E(i)=A(i+1, i)$ if $U P L O=$ L'.

TAU (output)
Realaray, dim ension $\mathbb{N}-1$ ) The scalar factors of the elem entary reflectors (see FurtherD etails).
$\mathbb{I N} F O$ (output)
= 0 : successfiulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the i-th argum enthad an illegalvalue

## FURTHER DETAILS

If $U$ PLO $=U$ ', the $m$ atrix $Q$ is represented as a product of elem entary reflectors

$$
Q=H(n-1) \ldots H(2) H(1) .
$$

Each $H$ (i) has the form

$$
H(i)=I-\tan * v^{*} v^{\prime}
$$

where tau is a real scalar, and $v$ is a realvectorw ith $v(i+1 \mathrm{~m})=0$ and $v(i)=1$; $v(1: i-1)$ is stored on exitin AP, overw riting A ( $1:-1, i+1$ ), and tau is stored in TA U (i).

If $\mathrm{ULO}=\mathrm{L}$ ', the $m$ atrix $Q$ is represented as a product of elem entary reflectors

$$
Q=H(1) H(2) \ldots H(n-1) .
$$

Each $H$ (i) has the form

$$
\mathrm{H}(\mathrm{i})=\mathrm{I}-\tan * \mathrm{v}^{*} \mathrm{v}^{\prime}
$$

where tau is a real scalar, and $v$ is a realvectorw ith $v(1: i)=0$ and $v(i+1)=1$; $v(i+2 n)$ is stored on exitin AP, overw riting A (i+2 $2 n, i$ ), and tau is stored in TAU (i).

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

ssptrf-com pute the factorization of a real sym $m$ etric
$m$ atrix A stored in packed form at using the B unch-K aufm an diagonalpivoting $m$ ethod

## SYNOPSIS

SUBROUTINE SSPTRF (UPLO,N,AP, $\mathbb{P} \mathbb{I V O T}, \mathbb{N} F O)$
CHARACTER * 1 UPLO
$\mathbb{N}$ TEGER $N, \mathbb{N} F O$
$\mathbb{I N} T E G E R \mathbb{P} \mathbb{I V} O T\left({ }^{( }\right)$
REALAP (*)
SU BROUTINE SSPTRF_64 (UPLO,N,AP, $\mathbb{P} \mathbb{V} O T, \mathbb{N} F O$ )
CHARACTER * 1 UPLO
$\mathbb{N}$ TEGER*8 $\mathrm{N}, \mathbb{I N F O}$
$\mathbb{N}$ TEGER*8 $\mathbb{P} \mathbb{I V O T}$ ( ${ }^{*}$ )
REALAP (*)

## F95 INTERFACE

SU BROUTINE SPTRF (UPLO,N,AP, $\mathbb{P} \mathbb{I V O T},[\mathbb{N} F O]$ )
CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER ::N, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V} O T$
REAL,D $\mathbb{M}$ ENSION (:) ::AP
SUBROUTINE SPTRF_64 (UPLO,N,AP, $\mathbb{P} \mathbb{I} \operatorname{OT},[\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::UPLO

```
\(\mathbb{N}\) TEGER ( 8 ) :: N, \(\mathbb{N} F O\)
```

$\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V O T}$
REAL,D $\mathbb{M}$ ENSION (:) ::AP

## C INTERFACE

\#include <sunperfh>
void ssptrf(charuplo, intn, float *ap, int *ipijot, int *info);
void ssptrf_64 (charuplo, long n, float *ap, long *ipivot, long *info);

## PURPOSE

ssptrf com putes the factorization of a realsym $m$ etric $m$ atrix A stored in packed form at using the Bunch $-K$ aufm an diagonal pivoting $m$ ethod:

$$
\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T} \text { or } \mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}
$$

where $U$ (orL) is a product of perm utation and unit upper (low er) triangular $m$ atrioes, and $D$ is sym $m$ etric and block diagonalw ith 1 -by-1 and 2 -by- 2 diagonalblocks.

## ARGUMENTS

```
UPLO (input)
```

= U ': U pper triangle ofA is stored;
= L': Low er triangle ofA is stored.
$N$ (input) The order of the matrix $A . N>=0$.

AP (input/output)
Realarray, dim ension $(\mathbb{N} *(\mathbb{N}+1) / 2$ ) On entry, the upper or low er triangle of the sym $m$ etric $m$ atrix $A$, packed colum nw ise in a linear array. The $j$ th colum $n$ ofA is stored in the array AP as follow s : ifUPLO $=U{ }^{\prime}, \mathrm{AP}(i+(j-1) \star j 2)=A(i, j)$ for $1<=\mathrm{i}<=\dot{j}$ if UPLO $=\mathrm{L}, \mathrm{A} A P(i+(j-1) *(2 n-j) / 2)=$ A ( $i, 7$ ) for $\dot{j}=\dot{j}=n$.

On exit, the block diagonalm atrix $D$ and the $m u l$ tipliers used to obtain the factorU orL, stored as a packed triangularm atrix overw riting A (see below for further details).

## IPIVOT (output)

Integer amay, dim ension $\mathbb{N}$ ) D etails of the interchanges and the block structure of D. If $\mathbb{P} \mathbb{I V O T}(\mathrm{k})>0$, then rows and colum ns k and IP IV O T $(k)$ w ere interchanged and $D(k, k)$ is a 1 -by-1 diagonalblock. IfUPLO $=\mathrm{U}^{\prime}$ and $\mathbb{P} \mathbb{I V O T}(k)=$ $\mathbb{P} \mathbb{I V O T}(k-1)<0$, then row s and colum nsk-1 and - $\mathbb{P}$ IV O T $(k)$ w ere interchanged and $D(k-1 * k, k-1 k)$ is a 2 -by-2 diagonal block. If UPLO = L'and $\mathbb{P} \mathbb{I V} \circ T(k)=\mathbb{P} \mathbb{I} \circ T(k+1)<0$, then row s and colum $n s$ $\mathrm{k}+1$ and $-\mathbb{P} \mathbb{I V O T}(\mathrm{k})$ were interchanged and D $(k: k+1, k: k+1)$ is a 2 -by -2 diagonalblock.

## $\mathbb{N} F O$ (output)

= 0: successfiulexit
$<0:$ if $\mathbb{N F O}=-i$, the $i$-th argum ent had an illegal value
$>0:$ if $\mathbb{N F O}=i, D(i, i)$ is exactly zero. The factorization has been com pleted, but the block diagonalm atrix D is exactly singular, and division by zero w illoccur if it is used to solve a system of equations.

## FURTHER DETAILS

5-96 - B ased on m odifications by J. Lew is, B oeing C om puter Services

C om pany

If $\mathrm{U} P \mathrm{LO}=\mathrm{U}$ ', then $A=U * D * U$ ', where
$U=P(n) \star U(n)^{\star} \ldots{ }^{*} P(k) U(k)^{\star} \ldots$,
i.e., $U$ is a productof term $s P(k) * U(k)$, where $k$ decreases from $n$ to 1 in steps of 1 or 2 , and $D$ is a block diagonal $m$ atrix $w$ ith 1 -by-1 and 2 -by-2 diagonalblocks $D(k) . P(k)$ is a perm utation $m$ atrix as defined by $\mathbb{P} \mathbb{I V} O T(k)$, and $U(k)$ is a unituppertriangularm atrix, such that if the diagonal
block $D(k)$ is of orders $(s=1$ or 2 ), then

```
    ( I v 0 ) k-s
U (k)=(0 I 0 ) s
    ( 0 0 I ) n-k
        k-s s n-k
```

If $s=1, D(k)$ overw rites $A(k, k)$, and $v$ overw rites $A(1, k-$ $1, k)$. If $s=2$, the uppertriangle of $(k)$ overw rites $A(k-$ $1, k-1), A(k-1, k)$, and $A(k, k)$, and $v$ overw rites $A(1 k-2, k-$ 1 k).

```
IfU PLO = L', then A = L *D *L',w here
    L = P (l)*L (l)* ... *P (k)*L (k)* ...,
```

i.e., $L$ is a productofterm $S P(k) * L(k)$, where $k$ increases from 1 to $n$ in steps of 1 or2, and $D$ is ablock diagonal $m$ atrix $w$ th 1 -by -1 and 2 -by- 2 diagonalblocks $D(k) . P(k)$ is a perm utation $m$ atrix as defined by $\mathbb{P} \mathbb{I V O T}(k)$, and $L(k)$ is a unit low ertriangularm atrix, such that if the diagonal block $D(k)$ is of orders ( $s=1$ or 2 ), then

```
    ( I 0 0 ) k-1
    L (k)=( 0 I 0 ) s
    ( 0 v I ) n-k-s+1
        k-1 s n-k-s+1
```

If $s=1, D(k)$ overw rites $A(k, k)$, and $v$ overw rites A $(k+1 n, k)$. If $s=2$, the low er triangle ofD ( $k$ ) overw rites A $(k, k), A(k+1, k)$, and $A(k+1, k+1)$, and $v$ overw rites $A(k+2 n, k k+1)$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssptri-com pute the inverse of a realsym $m$ etric indefinite $m$ atrix $A$ in packed storage using the factorization $A=$ $\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$ orA $=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ com puted by SSPTRF

## SYNOPSIS



```
CHARACTER * 1 UPLO
INTEGERN,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}
REALAP(*),W ORK (*)
SU BROUT\mathbb{NE SSPTRI_64(UPLO,N,AP,\mathbb{PIVOT,W ORK,INFO)}}\mathbf{(})=
CHARACTER * 1 UPLO
INTEGER*8 N,\mathbb{NFO}
\mathbb{NTEGER*8 P\mathbb{IVOT (*)}}\mathbf{(*)}
REALAP (*),W ORK (*)
F95 INTERFACE
```



```
    CHARACTER (LEN=1) ::UPLO
    \mathbb{NTEGER ::N,\mathbb{NFO}}0=0
    \mathbb{NTEGER,D IM ENSION (:) :: \mathbb{PIVOT}}\mathbf{T}\mathrm{ (:)}
    REAL,D IM ENSION (:) ::AP,W ORK
```



```
    CHARACTER (LEN=1)::UPLO
    \mathbb{NTEGER (8)::N,\mathbb{NFO}}\mathbf{~}\mathrm{ ( }
```

$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT}$
REAL,D $\mathbb{M} E N S I O N(:):: A P, W O R K$

## C INTERFACE

\#include <sunperfh>
void ssptri(charuple, intn, float *a, int *ipivot, int *info);
void ssptri 64 (charuplo, long n, float *a, long *ipivot, long *info);

## PURPOSE

ssptricom putes the inverse of a real sym m etric indefinite $m$ atrix $A$ in packed storage using the factorization $A=$ $U * D * U * * T$ orA $=L * D * L * * T$ com puted by SSPTRF .

## ARGUMENTS

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
$=\mathrm{U}$ :: Upper triangular, form is $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$;
$=L^{\prime}$ : Low er triangular, form is $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

A P (input/output)
Realarray, dim ension $(\mathbb{N} *(N+1) / 2)$ On entry, the block diagonal $m$ atrix D and the m ultipliers used to obtain the factorU orL as com puted by SSPTRF, stored as a packed triangularm atrix.

On exit, if $\mathbb{N F O}=0$, the (sym metric) inverse of the originalm atrix, stored as a packed triangular $m$ atrix. The $j$ th colum $n$ of inv ( $A$ ) is stored in the array AP as follow s: ifUPLO = U',AP (i+ (j 1) $\left.{ }^{\prime} \mathcal{j} 2\right)=\operatorname{inv}(A)(i, j)$ for $1<=i<=j ;$ ifUPLO $=\mathbb{L}^{\prime}$ ', $A P\left(i+(j-1)^{\star}(2 n-j / 2)=\operatorname{inv}(A)(i, j)\right.$ for $j=i<=n$.

IPIVOT (input)
Integer array, dim ension $\mathbb{N}$ ) D etails of the interchanges and the block structure ofD as determ ined by SSPTRF.

W ORK (w orkspace)

Realaray, dim ension $(\mathbb{N})$
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N} F O=i, D(i, i)=0$; the $m$ atrix is singular and its inverse could notbe com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

SSPTRS -solve a system of linearequations $A * X=B$ with a real symm etric $m$ atrix $A$ stored in packed form atusing the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ com puted by SSPTRF

## SYNOPSIS


CHARACTER * 1 UPLO
$\mathbb{N}$ TEGER N,NRHS,LDB, $\mathbb{N}$ FO
$\mathbb{N} T E G E R \mathbb{P} \mathbb{V} O T\left({ }^{( }\right)$
REALAP (*), B (LDB,*)
SU BROUTINE SSPTRS_64 (UPLO,N,NRHS,AP, $\mathbb{P} \mathbb{I V O T}, \mathrm{B}, \mathrm{LD} B, \mathbb{N} F O$ )
CHARACTER * 1 UPLO
$\mathbb{N}$ TEGER*8N,NRHS,LDB, $\mathbb{N} F O$
$\mathbb{N}$ TEGER * $8 \mathbb{P} \mathbb{I V O T}$ ( ${ }^{*}$ )
REALAP (*), B (LDB,*)

## F95 INTERFACE

SU BROUTINE SPTRS (UPLO,N, NRHS],AP, $\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\operatorname{LDB}],[\mathbb{N F O}])$
CHARACTER (LEN=1)::UPLO
$\mathbb{N}$ TEGER ::N,NRHS,LDB, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}$
REAL,D $\mathbb{I}$ ENSION (:) ::AP
REAL,D IM ENSION (: : : : : B
SU BROUTINE SPTRS_64 (UPLO ,N, NRHS],AP, $\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}],[\mathbb{N} F O])$

CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER (8) :: N,NRHS,LDB, $\mathbb{N} F O$
$\mathbb{N}$ TEGER (8),D $\mathbb{I M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V}$ OT
REAL,D $\mathbb{M}$ ENSION (:) ::AP
REAL,D $\mathbb{M}$ ENSION (: : : : : : B

## C INTERFACE

\#include <sunperfh>
void ssptres (char uplo, int n, int nrhs, float *ap, int *ipivot, float *b, int ladb, int*info);
void ssptrs_64 (charuplo, long n, long nrhs, float *ap, long *ịíivot, float *b, long ldb, long *info );

## PURPOSE

ssptrs solves a system of linear equations A *X = B w ith a real sym $m$ etric $m$ atrix $A$ stored in packed form atusing the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$ or $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ com puted by SSPTRF.

## ARGUMENTS

## UPLO (input)

Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
$=\mathrm{U}$ : : U ppertriangular, form is $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$;
$=\mathrm{L}$ ': Low ertriangular, form is $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$.

N (input) The order of the matrix A. $\mathrm{N}>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS >=0.

AP (input)
Realarray, dim ension $\mathbb{N} * \mathbb{N}+1) / 2$ ) The block diagonal $m$ atrix $D$ and the $m$ ultipliers used to obtain the factorU orL as com puted by SSPTRF, stored as a packed triangularm atrix.

PIVOT (input)
Integer array, dim ension $\mathbb{N}$ ) D etails of the interchanges and the block structure ofD as determ ined by SSPTRF.

B (input/output)
Realarray, dim ension (LDB,NRHS) On entry, the right hand sidem atrix $B$. On exit, the solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the aray B. LD B >= $\max (1, N)$.
$\mathbb{I N F O}$ (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N} F O=-$ i, the $i$-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sstebz -com pute the eigenvalues of a sym m etric tridiagonal $m$ atrix $T$

## SYNOPSIS

```
SUBROUT\mathbb{NE SSTEBZ RANGE,ORDER,N,VL,VU,IL,\mathbb{U,ABSTOL,D,E,M,}}\mathbf{N},\textrm{M},\textrm{L}
```



```
CHARACTER * 1 RANGE,ORDER
INTEGERN,\mathbb{L},\mathbb{U},M,NSPL\mathbb{T},\mathbb{N}FO
\mathbb{NTEGER IBLOCK (*),ISPLIT (*), IN ORK (*)}
REALVL,VU,ABSTOL
REALD (*),E (*),W (*),W ORK (*)
SUBROUT\mathbb{NE SSTEBZ_64 RANGE,ORDER,N,VL,VU,\mathbb{I},\mathbb{U},ABSTOL,D,E,}
    M,NSPLIT,W,\mathbb{BLOCK, ISPLIT,WORK,\mathbb{N ORK, INFO)}}\mathbf{N}=()
```

CHARACTER * 1 RANGE, ORDER
$\mathbb{N} T E G E R * 8 N, \mathbb{I}, \mathbb{U}, M, N S P L \mathbb{T}, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{B L O C K}(*)$, $\operatorname{ISPLIT}(*)$, $\mathbb{I N}$ ORK (*)
REALVL,VU,ABSTOL
REALD (*), E (*), W (*), W ORK (*)

## F95 INTERFACE

SUBROUTINE STEBZ RANGE,ORDER,N,VL,VU, II, IU,ABSTOL,D,E,M, NSPLIT,W, $\mathbb{B} L O C K, I S P L I T,[W O R K],[\mathbb{W} O R K],[\mathbb{N F O}])$

CHARACTER (LEN=1) ::RANGE,ORDER
$\mathbb{N} T E G E R:: N, \mathbb{L}, \mathbb{U}, \mathrm{M}, \mathrm{N}$ SPLIT, $\mathbb{N} F \mathrm{O}$
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: $\mathbb{B L O C K}, \operatorname{ISPLIT}, \mathbb{I N}$ ORK
REAL ::VL,VU,ABSTOL
REAL,D $\mathbb{M}$ ENSION (:) ::D, $\mathrm{E}, \mathrm{W}, \mathrm{W} O R K$

SU BROUT $\mathbb{N} E$ STEBZ_64 (RANGE, ORDER, N, VL, VU, $\mathbb{I}, ~ \mathbb{U}, ~ A B S T O L, D, E, M$, N SPLIT, W, $\mathbb{B} L O C K, I S P L \mathbb{T},[W O R K],[\mathbb{W} O R K],[\mathbb{N} F O])$

CHARACTER (LEN=1) ::RANGE,ORDER
$\mathbb{N}$ TEGER (8) :: $N, \mathbb{I}, \mathbb{U}, \mathrm{M}, \mathrm{N}$ SPL $\mathbb{I}, \mathbb{I N F O}$
$\mathbb{N} \operatorname{TEGER}$ (8), D $\mathbb{M} \operatorname{ENSION}$ (:) :: $\mathbb{B L O C K}, \operatorname{ISPL} \mathbb{I}, \mathbb{I N} O R K$
REAL ::VL,VU,ABSTOL
REAL,D $\mathbb{I M} E N S I O N(:):$, $, \mathrm{E}, \mathrm{W}, \mathrm{W}$ ORK

## C INTERFACE

\#include <sunperfh>
void sstebz (char range, char order, intn, float vl, float
vu, int il, intiu, floatabstol, float *d, float
*e, int*m, int*nsplit, float *w, int *iblock, int *isplit, int*info);
void sstebz_64 (char range, char order, long n, float vl, float vu, long il, long iu, floatabstol, float *d, float*e, long *m, long *nsplit, float *w, long *iblock, long *isplit, long *info);

## PURPOSE

sstebz com putes the eigenvalues of a sym m etric tridiagonal $m$ atrix T. The userm ay ask for alleigenvalues, alleigenvalues in the half-open interval $(\mathbb{L}, \mathrm{VU}]$, or the $\mathbb{I}$-th through $\mathbb{I U}$-th eigenvalues.

T o avoid overflow, the m atrix m ustbe scaled so that its largestelem ent is no greater than overflow ** (1/2) * underflow ** (1/4) in absolute value, and forgreatest accuracy, it should not.be m uch sm aller than that.

See W . K ahan "A ccurate E igenvalues of a Sym m etric TridiagonalM atrix", ReportC S41, C om puterScience D ept., Stanford U niversity, July 21, 1966.

## ARGUMENTS

RANGE (input)
= A : ("A ll") alleigenvalues w illbe found.
= V ': ("V alue") alleigenvalues in the half-open interval $\mathrm{VL}, \mathrm{V}$ U ] w illbe found. = ' I ': ("Index") the $\mathbb{I}$-th through $\mathbb{I U}$-th eigenvalues (of the entire m atrix) w illbe found.

ORDER (input)
= B ': ("By B lock") the eigenvalues will be grouped by split-off block (see $\mathbb{B L O C K}$, ISPLIT) and ordered from sm allest to largest w ithin the block. = E ': ("Entire m atrix") the eigenvalues for the entire m atrix w illbe ordered from sm allest to largest.

N (input) The order of the tridiagonalm atrix $\mathrm{T} . \mathrm{N}>=0$.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvahues. Eigenvalues less than or equal to $V L$, or greater than VU, will not be retumed. VL < VU. N ot referenced ifRANGE=A'or 'I'.

VU (input)
See the description of L .

IL (input)
If RA N GE= ' I ', the indices (in ascending order) of the smallest and largest eigenvahues to be retumed. $1<=\mathbb{L}<=\mathbb{U}<=N$, if $N>0$; $\mathbb{H}=1$ and $\mathbb{U}=0$ if $N=0$. $N$ otreferenced ifRANGE $=$ A'or V'.

IU (input)
See the description of $\Pi$.

ABSTOL (input)
The absolute tolerance for the eigenvalues. A n eigenvalue (or cluster) is considered to be located if ithas been determ ined to lie in an intervalw hose w idth is A BSTOL or less. IfA BSTOL is less than orequal to zero, then $U L P *|\mid w i l l$ be used, w here $I \mid m$ eans the 1 -norm of $T$.

E igenvalues w illbe com puted m ost accurately w hen ABSTOL is set to tw ige the underflow threshold 2*SLAM CH (S ), notzero.

D (input) The $n$ diagonalelem ents of the tridiagonal $m$ atrix T.

E (input) The $(n-1)$ off-diagonalelem ents of the tridiagonal m atrix T.

M (output)

The actual num berofeigenvahues found. $0<=\mathrm{M}$ <= N. (Se also the description of $\mathbb{N} F O=2,3$.)

NSPLIT (output)
The num ber of diagonalblocks in the m atrix T. 1
<= NSPLIT <= N .

W (output)
On exit, the firstM elem ents of $W$ will contain the eigenvalues. (SSTEBZ may use the rem aining N M elem ents as w orkspace.)

BBLOCK (output)
A teach row /collm n jw here $E()$ is zero or sm all, the m atrix T is considered to split into ablock diagonalm atrix. On exit, if $\mathbb{N F O}=0, \mathbb{B L O C K}$ (i) specifies to which block (from 1 to the num ber of blocks) the eigenvalue W (i) belongs. (SSTEBZ may use the rem aining N M elem ents as w orkspace.)

ISPLIT (output)
The splitting points, atw hich $T$ breaks up into subm atrices. The first subm atrix consists of row s/columns 1 to ISPLIT (1), the second of row s/colum ns ISPL IT (1)+1 through ISPLIT (2), etc., and the NSPLIT-th consists of row s/colum ns ISPLIT $(\mathbb{N}$ SPLIT-1) +1 through ISPLIT $(\mathbb{N} S P L I T)=N$. (Only the firstN SPLIT elem entsw ill actually be used, but since the user cannot know a prioriw hat value NSPLIT w ill have, N w ordsm ust be reserved for ISPLIT.)

W ORK (w orkspace)
dim ension ( $4 * \mathrm{~N}$ )
IN ORK (w orkspace)
dim ension $(3 * N)$
$\mathbb{N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{N}$ FO $=-i$, the $i$-th argum ent had an illegalvalue
> 0: som e orall of the eigenvalues failed to converge or
w ere not com puted:
$=1$ or 3 : B isection failed to converge for som e eigenvalues; these eigenvalues are flagged by a negative block num ber. The effect is that the eigenvalues $m$ ay notbe as accurate as the absolute and relative tolerances. This is generally caused
by unexpectedly inaccurate arithm etic. $=2$ or 3 :
RANGE=I'only:N otallof the eigenvalues $\mathbb{I}: \mathbb{U}$ w ere found.
Effect: $M<\mathbb{U}+1-\mathbb{I}$
C ause: non-m onotonic arithm etic, causing the Sturm sequence to be non-m onotonic. Cure: recalculate, using $R$ A N G E= A ', and pick
outeigenvalues $\mathbb{I L}: \mathbb{Z U} .=4: \quad$ RANGE= I', and the
G ershgorin interval initially used w as too sm all.
N o eigenvahues w ere com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

sstedc - com pute alleigenvalues and, optionally, eigenvectors of a sym $m$ etric tridiagonalm atrix using the divide and conquerm ethod

## SYNOPSIS

```
SUBROUT\mathbb{NE SSTEDC (COMPZ,N,D,E,Z,LD Z,W ORK,LW ORK,IN ORK,L\mathbb{N ORK,}}\mathbf{N},\textrm{L}
    INFO)
CHARACTER * 1 COMPZ
\mathbb{NTEGERN,LD Z,LW ORK,LIN ORK,INFO}
INTEGER IV ORK (*)
REALD (*),E (*),Z (LD Z,*),W ORK (*)
SU BROUT\mathbb{NE SSTED C_64 (COM PZ,N,D,E,Z,LD Z,W ORK,LW ORK,IN ORK,}
    LIN ORK,\mathbb{NFO)}
CHARACTER * 1 COMPZ
\mathbb{NTEGER*8N,LD Z,LW ORK,LIN ORK,INFO}
INTEGER*8 IN ORK (*)
REALD (*),E (*),Z (LD Z,*),W ORK (*)
```


## F95 INTERFACE

```
SU BROUT \(\mathbb{N} E \operatorname{STEDC} \mathbb{C O M} P Z, N, D, E, Z,[L D Z],[W O R K],[L W\) ORK ], [IW ORK ], \(\left[\begin{array}{ll}\mathbb{I} & \mathrm{ORK}],[\mathbb{N F O}])\end{array}\right.\)
CHARACTER (LEN=1) ::COM PZ
\(\mathbb{N}\) TEGER ::N,LDZ,LW ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL,D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK
REAL,D \(\mathbb{I}\) ENSION (:,:) :: Z
```

SU BROU TINE STEDC_64 (COM PZ,N,D,E,Z, [LD Z], [W ORK ], [LW ORK], [IW ORK ], $\left[\begin{array}{l}\mathbb{I} \\ \text { ORK ], }[\mathbb{N} F O])\end{array}\right.$

CHARACTER (LEN=1) ::COM PZ
$\mathbb{N}$ TEGER (8) ::N,LD Z,LW ORK, LIN ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{I N}$ ORK
REAL,D IM ENSION (:) ::D , E,W ORK
REAL,D IM ENSION (:,:) ::Z

## C INTERFACE

\#include < sunperfh>
void sstedc (char com pz, intn, float *d, float *e, float *z, int ldz, int*info);
void sstedc_64 (charcom pz, long n, float *d, float*e, float *z, long ldz, long *info);

## PURPOSE

sstedc com putes alleigenvalues and, optionally, eigenvectors of a sym $m$ etric tridiagonalm atrix using the divide and conquerm ethod. The eigenvectors of a full or band real sym metric $m$ atrix can also be found ifSSY TRD orSSPTRD or SSBTRD hasbeen used to reduce this $m$ atrix to tridiagonal form.

This code $m$ akes very $m$ ild assum ptions about floating point arithm etic. It w illw ork on $m$ achines $w$ ith a guard digit in add/subtract, or on those binary machines w thout guard digits which subtract like the C ray X $-\mathrm{M} P$, C ray $Y$ M P , C ray C-90, or C ray-2. Itcould conœívably fail on hexadecim al or decin al machines w thout guard digits, butw e know of none. Se SLAED 3 for details.

## ARGUMENTS

COMPZ (input)
$=\mathrm{N}^{\prime}:$ C om pute eigenvalues only .
= I': C om pute eigenvectors of tridiagonalm atrix also.
$=\mathrm{V}$ : C om pute eigenvectors of original dense symmetric $m$ atrix also. On entry, $Z$ contains the orthogonalm atrix used to reduce the original $m$ atrix to tridiagonal form .

N (input) The dim ension of the sym $m$ etric tridiagonalm atrix.
$\mathrm{N}>=0$.

D (input/output)
On entry, the diagonalelem ents of the tridiagonal m atrix. On exit, if $\mathbb{N} F O=0$, the eigenvalues in ascending order.

E (input/output)
O n entry, the subdiagonalelem ents of the tridiagonalm atrix. On exit, E hasbeen destroyed.

Z (input) On entry, if COMPZ = V', then Z contains the orthogonal m atrix used in the reduction to tridiagonal form. On exit, if $\mathbb{I N F O}=0$, then if $C O M P Z$ $=\mathrm{V}$ ', Z contains the orthonorm aleigenvectors of the original sym $m$ etric $m$ atrix, and if $C O M P Z=I$ ', $Z$ contains the orthonorm al eigenvectors of the sym $m$ etric tridiagonalm atrix. If $\mathrm{COMPZ}=\mathrm{N}$ ', then Z is not referenced.

LD Z (input)
The leading din ension of the array Z . LD $\mathrm{Z}>=1$.
If eigenvectors are desired, then LD Z $>=\mathrm{max}(1, \mathrm{~N})$.
W ORK (w orkspace)
dim ension (LW ORK)Onexit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim allW ORK.

LW ORK (input)
The dim ension of the aray $W$ ORK. IfCOMPZ $=N^{\prime}$ orN <= 1 then LW ORK m ustbe at least1. If COM PZ $=\mathrm{V}$ 'and $\mathrm{N}>1$ then LW ORK m ustbe at least ( $1+$ $3 * \mathrm{~N}+2{ }^{*} \mathrm{~N} * \lg \mathrm{~N}+3 * \mathrm{~N} * * 2$ ), where $\lg (\mathrm{N})=\mathrm{sm}$ allest integerk such that $2 * * k>=\mathrm{N}$. If $\mathrm{COM} \operatorname{PZ}=$ 'I' and N > 1 then LW ORK m ustbe at least ( $1+$ $4 * \mathrm{~N}+\mathrm{N} * * 2$ ).

IfLW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IN ORK (w orkspace/output)
On exit, if $\mathbb{N F} F=0, \mathbb{I N}$ ORK (1) retums the optim al LIN ORK.

LIV ORK (input)
The dim ension of the array $\mathbb{I N} O R K$. IfCOMPZ $=N^{\prime}$ or $N$ <= 1 then LIW ORK m ustbe at least 1. If

COMPZ $=V$ 'and $N>1$ then $L \mathbb{I V}$ ORK m ustbe at least

then LIN ORK must.be at least ( $3+5{ }^{*} \mathrm{~N}$ ).

If LIV ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim al size of
the $\mathbb{I V}$ ORK array, retums this value as the first entry of the $\mathbb{I V}$ ORK amay, and no errorm essage related to LIN ORK is issued by XERBLA.
$\mathbb{I N} F O$ (output)
= 0: successfulexit.
$<0$ : if $\mathbb{N}$ FO $=-$ i, the $i$-th argum ent had an illegalvalue.
>0: The algonithm failed to com pute an eigenvalue while w orking on the subm atrix lying in row s and colum $n s \mathbb{N} F O / \mathbb{N}+1$ ) through $m$ od ( $\mathbb{N} F O, N+1$ ).

## FURTHER DETAILS

B ased on contributions by
JeffR utter, C om puter Science D ívision, U niversity of C alifomia
at B erkeley, U SA
M odified by Francoise T isseur, U niversity of Tennessee.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

sstegr- (a) C om pute $T$-sigm a_i= L_iD_i L_i^T, such that L_iD_iL_i^T is a relatively robustrepresentation

## SYNOPSIS



```
    Z,LDZ,ISUPPZ,W ORK,LW ORK,IN ORK,LIW ORK,\mathbb{NFO)}
```

```
CHARACTER * 1 JOBZ,RANGE
\mathbb{N}TEGERN,\mathbb{L},\mathbb{U},M,LDZ,LW ORK,L\mathbb{N ORK,\mathbb{NFO}}\mathbf{M}\mathrm{ , L}
INTEGER ISUPPZ (*), IN ORK (*)
REAL VL,VU,ABSTOL
REALD (*),E (*),W (*),Z (LD Z ,*),W ORK (*)
SU BROUTINE SSTEGR_64 (JOBZ,RANGE,N,D,E,VL,VU,\mathbb{I},\mathbb{U},ABSTOL,M,
    W,Z,LDZ,ISUPPZ,W ORK,LW ORK,IN ORK,LIN ORK,INFO)
```

CHARACTER * 1 JOBZ,RANGE
$\mathbb{N}$ TEGER*8N, $\mathbb{I}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W$ ORK,LIN ORK, $\mathbb{N} F \mathrm{~F}$
$\mathbb{N} T E G E R * 8 \operatorname{ISUPPZ}$ (*), $\mathbb{I N}$ ORK (*)
REALVL,VU,ABSTOL
REALD (*), E (*), W (*), Z (LD Z ,*), W ORK (*)

## F95 INTERFACE

SU BROUTINE STEGR (JOBZ,RANGE, $\mathbb{N}], D, E, V L, V U, \mathbb{I}, \mathbb{U}, A B S T O L, M$, W , Z, [LD Z], ISUPPZ, [W ORK ], [LW ORK ], [IN ORK ], [LINORK], [ $\mathbb{N} F O])$

CHARACTER (LEN=1): : JOBZ,RANGE
$\mathbb{N} T E G E R:: N, \mathbb{L}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W$ ORK,LIN ORK, $\mathbb{N} F \mathrm{O}$
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: ISUPPZ, $\mathbb{I N}$ ORK
REAL ::VL,VU,ABSTOL

REAL,D $\mathbb{M}$ ENSION (:) ::D , E, W ,W ORK
REAL,D IM ENSION (:,:) ::Z

SUBROUTINE STEGR_64 (OBZ,RANGE, $\mathbb{N}], D, E, V L, V U, \mathbb{L}, \mathbb{I}, A B S T O L$, $M, W, Z,[L D Z], I S U P P Z,[W O R K],[L W O R K],[\mathbb{W} O R K],[L \mathbb{N} O R K],[\mathbb{N} F O])$

CHARACTER (LEN=1) :: JOBZ,RANGE
$\mathbb{N}$ TEGER (8) :: $N, \mathbb{I}, \mathbb{U}, \mathrm{M}, \mathrm{LD} \mathrm{Z}, \mathrm{LW}$ ORK,LIWORK, $\mathbb{N} F O$
$\mathbb{N}$ TEGER (8), D $\mathbb{I M} E N S I O N$ (:) :: ISUPPZ , $\mathbb{I N}$ ORK
REAL ::VL,VU,ABSTOL
REAL,D $\mathbb{I M} E N S I O N(:):: D, E, W, W O R K$
REAL,D IM ENSION (:,:) ::Z

## C INTERFACE

\#include <sunperfh>
void sstegr(char jobz, char range, intn, float *d, float
*e, float vl, float vu, int il, intiu, float abstol, int *m, float *w , float * $z$, int ldz, int *isuppz, int *info);
void sstegr_64 (char j̀jbz, char range, long n, float *d, float *e, float vl, floatvu, long il, long iu, floatabstol, long *m, float*w, float *z, long ldz, long *isuppz, long *info);

## PURPOSE

sstegrb) C om pute the eigenvahues, lam bda_j of L_i D_i L_i^T to high
relative accuracy by the dqds algorithm,
(c) If there is a cluster of close eigenvalues, "choose"
sigm a_i
close to the cluster, and go to step (a),
(d) G iven the approxim ate eigenvalue lam bda_jofL_i D _i L_i^T, com pute the corresponding eigenvectorby form ing a rank-revealing tw isted factorization. The desired accuracy of the output can be specified by the inputparam eterABSTOL.

Form ore details, see "A new O ( $n^{\wedge} 2$ ) algorithm for the sym $m$ etric tridiagonal eigenvahue/eigenvector problem ", by Inder进D hillon, C om puterScience D ivision TechnicalR epont N o. U CB C SD -97-971, U C B erkeley, M ay 1997.

N ote 1 : Cumently SSTEGR is only setup to find ALL the $n$ eigenvalues and eigenvectors of $T$ in $O\left(n^{\wedge} 2\right)$ tim e N ote 2 : Cumently the routine SSTE $\mathbb{N}$ is called when an appropriate sigm a_i cannot be chosen in step (c) above.

SSTE $\mathbb{I N}$ invokes m odified Gram -Schm idt when eigenvalues are close.
N ote 3 :SSTEGR w orks only on $m$ achines which follow ieec-754 floating-point standard in their handling of infinities and NaN s. N orm alexecution of SSTEGR $m$ ay create $N a N s$ and infinities and hence $m$ ay abort due to a floating point exception in environm ents w hich do not conform to the ieee standard.

## ARGUMENTS

JOBZ (input)
= N ': C om pute eigenvalues only;
$=\mathrm{V}$ : C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvalues w illibe found.
= V : alleigenvalues in the half-open interval
( L L, VU ] will be found. = I': the $\mathbb{I}$-th through
IU -th eigenvaluesw illube found.

N (input) The order of the m atrix. $\mathrm{N}>=0$.
D (input/output)
O n entry, the n diagonalelem ents of the tridiagonalm atrix T.On exit, D is overw ritten.

E (input/output)
O $n$ entry, the ( $n-1$ ) subdiagonal elem ents of the tridiagonal m atrix T in elem ents 1 to $\mathrm{N}-1$ of E ; $\mathrm{E}(\mathbb{N})$ need notbe set. On exit, E is overw rilten.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRA N GE = A' 'or I'.

VU (input)
Se the description of V L .

II (input)
IfRA N G E= I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{H}<=\mathbb{U}<=N$, ifn $>0$; $\mathbb{H}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N ot referenced ifRANGE= A 'or V'.

IU (input)
See the description of II.

ABSTOL (input)
The absolute error tolerance for the eigenvalues/eigenvectors. $\mathbb{F} J \mathrm{OBZ}=\mathrm{V}$ ', the eigenvalues and eigenvectors outputhave residual norm s bounded by ABSTOL, and the dotproducts betw een different eigenvectors are bounded by ABSTOL. If ABSTOL is less than N *EPS*|T $\mid$, then N *EPS*|T|w illbe used in its place, where EPS is the $m$ achine precision and $F$ is the 1 -norm of the tridiagonalm atrix. The eigenvalues are com puted to an accuracy ofEPS* 1 |imespective of A BSTOL . If high relative accuracy is im portant, setA B STO L to DLAM CH (Safem inim um '). See Barlow and Dem mel "C om puting A ccurate Eigensystem s of Scaled D iagonally D om inantM atrioes", LA PA CK W orking N ote \#7 for a discussion of $w$ hich $m$ atrioes define their eigenvalues to high relative accuracy .

M (output)
The total num berofeigenvalues found. $0<=\mathrm{M}$ <= N . IfRANGE $=A \prime, \mathrm{M}=\mathrm{N}$, and ifRANGE $=\mathrm{I}^{\prime}, \mathrm{M}=$ $\mathbb{I U}-\mathbb{H}+1$.

W (output)
The firstM elem ents contain the selected eigenvalues in ascending order.
$Z$ (input) If $J O B Z=V^{\prime}$, then if $\mathbb{N F O}=0$, the first $M$ colum ns of $Z$ contain the orthonorm aleigenvectors of them atrix T comesponding to the selected eigenvalues, $w$ ith the i-th colum $n$ of $Z$ holding the eigenvector associated w ith W (i). If $\mathrm{JO} \mathrm{BZ}=\mathrm{N}$ ', then $Z$ is not referenced. N ote: the userm ust ensure that at leastm ax ( $1, M$ ) colum ns are supplied in the array $Z$; ifRANGE = V', the exact value of M is not know n in advance and an upperbound m ust be used.

LD $Z$ (input)
The leading dim ension of the array $Z . L D Z>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD Z >= $\mathrm{max}(1, N)$.

ISU PPZ (output)
The support of the eigenvectors in $Z$, i.e., the indices indicating the nonzero elem ents in $Z$. The i-th eigenvector is nonzero only in elem ents ISU PPZ (2*i-1 ) through ISU PPZ (2*i).

W ORK (w orkspace)

On exit, if $\mathbb{N F O}=0, W$ ORK ( 1 ) retums the optim al (and minim al) LW ORK .

## LW ORK (input)

The dimension of the aray W ORK. LW ORK >= $\max (1,18 * N)$
IfLW O RK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK aray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IW ORK (w orkspace/output)
On exit, if $\mathbb{N} F O=0, \mathbb{I W}$ ORK (1) retums the optim al LIN ORK.

LIV ORK (input)
The dim ension of the aray $\mathbb{I W}$ ORK. L $\mathbb{I W}$ ORK >= $\max (1,10 \star N)$

IfLIW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the $\mathbb{I W}$ ORK array, retums this value as the first entry of the $\mathbb{I W}$ ORK array, and no errorm essage related to $L \mathbb{I W}$ ORK is issued by X ERBLA.

INFO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum ent had an illegalvalue
$>0:$ if $\mathbb{N F O}=1$, intemalemor in SLARRE, if $\mathbb{N} F O=2$, intemalemor in SLARRV.

## FURTHER DETAILS

B ased on contributions by
Inder屰D hillon, $\mathbb{B M}$ A $\operatorname{lm}$ aden, U SA
O sniM arques, LBNL N ER SC , U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sstein -com pute the eigenvectors of a realsym $m$ etric tridiagonal $m$ atrix $T$ comesponding to specified eigenvalues, using inverse iteration

## SYNOPSIS

```
SUBROUT\mathbb{NE SSTE\mathbb{N N,D,E,M,W, BLOCK,ISPLIT,Z,LD Z,W ORK,IN ORK,}}\mathbf{N},\textrm{N},\textrm{N}
    \mathbb{FA}|,\mathbb{NNOO}
\mathbb{N TEGER N,M,LD Z, IN FO}
```



```
REALD (*),E (*),W (*),Z (LD Z ,*),W ORK (*)
```



```
    IN ORK,\mathbb{FA}\mathbb{I},\mathbb{N}FO)
\mathbb{NTEGER*8 N,M,LD Z,INFO}
\mathbb{NTEGER*8 \mathbb{BLOCK}(*), ISPLIT (*), IN ORK (*),\mathbb{FA I[ (*)}}\mathbf{(})
REALD (*),E (*),W (*),Z (LD Z,*),W ORK (*)
```


## F95 INTERFACE

SU BROUT $\mathbb{N E} \operatorname{STE} \mathbb{N} \mathbb{N}, D, E, M, W, \mathbb{B L O C K}, \mathbb{I S P L I T}, \mathrm{Z},[\operatorname{LD} Z],[W$ ORK ], [ $\mathbb{I V}$ ORK ], $\mathbb{F A} \mathbb{I},[\mathbb{N} F O])$
$\mathbb{N} T E G E R:: N, M, L D Z, \mathbb{N F O}$
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: $\mathbb{B L O C K}, \operatorname{ISPLIT}, \mathbb{I}$ ORK, $\mathbb{F} A \mathbb{I}$
REAL,D $\mathbb{M}$ ENSION (:) ::D ,E,W ,W ORK
REAL,D $\mathbb{M}$ ENSION (:,:)::Z

SU BROUTINE STE $\mathbb{N} \_64 \mathbb{N}, D, E, M, W, \mathbb{B L O C K}, \operatorname{ISPL} \mathbb{I}, \mathrm{Z},[\operatorname{LD} Z],[\mathbb{W}$ ORK ], [ $\mathbb{I N}$ ORK], $\mathbb{F A} \mathbb{I},[\mathbb{N} F O]$ )
$\mathbb{N}$ TEGER (8) :: $\mathrm{N}, \mathrm{M}, \mathrm{LD} \mathrm{Z}, \mathbb{N}$ FO
$\mathbb{N}$ TEGER (8),D $\mathbb{I M}$ ENSION (:) :: $\mathbb{B L O C K}, \operatorname{ISPL} \mathbb{I}, ~ \mathbb{I V}$ ORK, $\mathbb{F} A \mathbb{I}$ REAL,D $\mathbb{M}$ ENSION (:) ::D , E,W ,W ORK
REAL,D $\mathbb{M}$ ENSION (:,:)::Z

## C INTERFACE

\#include <sunperfh>
void sstein (intn, float *d, float *e, intm, float *w, int *iblock, int *isplit, float *z, int ldz, int *ifail, int *info);
void sstein_ 64 (long n, float *d, float *e, long m,float *w , long *iblock, long *isplit, float*z, long ldz, long *ifail, long *info);

## PURPOSE

sstein com putes the eigenvectors of a realsym $m$ etric tridiagonal $m$ atrix $T$ comesponding to specified eigenvalues, using inverse iteration.

The m axim um num ber of terations allow ed foreach eigenvector is specified by an intemal param eterM A X ITS (currently set to 5).

## ARGUMENTS

N (input) The order of the $m$ atrix. $\mathrm{N}>=0$.

D (input) The n diagonalelem ents of the tridiagonal $m$ atrix
T.

E (input) The ( $\mathrm{n}-1$ ) subdiagonalelem ents of the tridiagonal $m$ atrix $T$, in elem ents 1 to $N-1 . E(\mathbb{N})$ need notbe set.

M (input) The num ber of eigenvectors to be found. $0<=\mathrm{M}<=$ N .

W (input) The firstM elem ents ofW contain the eigenvalues for which eigenvectors are to be com puted. The eigenvalues should be grouped by split-off block and ordered from smallest to largestw ithin the block. (The output array $W$ from SSTEBZ w th ORDER = B'is expected here.)

BLOCK (input)
The subm atrix indiges associated with the corresponding eigenvalues in $W$; $\mathbb{B L O C K}(i)=1$ if eigenvalue $W$ (i) belongs to the first subm atrix from the top, $=2$ ifW (i) belongs to the second subm atrix, etc. (The output array $\mathbb{B L O C K}$ from SSTEBZ is expected here.)

## ISPLIT (input)

The splitting points, atw hich $T$ breaks up into subm atrices. The first subm atrix consists of row s/columns 1 to ISPLIT ( 1 ), the second of row s/Colum ns ISPLIT ( 1 )+1 through ISPLIT (2), etc. (The outputaray ISPLIT from SSTEBZ is expected here.)
Z (output)
The com puted eigenvectors. The eigenvector associated w ith the eigenvalue $W$ (i) is stored in the $i-t h$ colum $n$ of $Z$. A ny vectorw hich fails to converge is set to its current iterate afterM AX ITS iterations.

LD $Z$ (input)
The leading dim ension of the aray $Z$. LD $Z \quad>=$ $\max (1, N)$.

W ORK (w orkspace)
dim ension ( $5 * \mathrm{~N}$ )
IV ORK (w orkspace)
dim ension (N)
FFA II (output)
On norm alexit, allelem ents of $\mathbb{F} A \mathbb{I}$ are zero. If one orm ore eigenvectors fail to converge after
M AXITS iterations, then their indices are stored in array $\mathbb{F A} \mathbb{I}$.
$\mathbb{I N F O}$ (output)
$=0$ : successfulexit.
$<0:$ if $\mathbb{N F O}=-$ i, the $i$-th argum enthad an illegalvalue
$>0$ : if $\mathbb{N} F O=i$, then $i$ eigenvectors failed to converge in M AX ITS iterations. Their indiges are stored in array $\mathbb{F}$ A II.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssteqr-com pute alleigenvalues and, optionally, eigenvectors of a sym $m$ etric tridiagonalm atrix using the im plicit $Q L$ orQ R m ethod

## SYNOPSIS

```
SUBROUT\mathbb{NE SSTEQR (COM PZ,N,D,E,Z,LDZ,W ORK,INFO)}
CHARACTER * 1 COMPZ
\mathbb{NTEGER N,LD Z, INFO}
REALD (*),E (*),Z (LD Z ,*),W ORK (*)
SUBROUT\mathbb{NE SSTEQR_64(COMPZ,N,D,E,Z,LD Z,W ORK,INFO )}
CHARACTER * 1 COMPZ
\mathbb{NTEGER*8N,LD Z,INFO}
REALD (*),E (*),Z (LD Z,*),W ORK (*)
F95 INTERFACE
    SU BROUT\mathbb{NE STEQR COMPZ,N,D,E,Z, [LD Z], [W ORK ], [NFO])}
    CHARACTER (LEN=1) ::COM PZ
    INTEGER ::N,LD Z, INFO
    REAL,DIM ENSION (:) ::D ,E,W ORK
    REAL,DIM ENSION (:,:)::Z
```

    SU BROUTINE STEQR_64 (COM PZ,N,D ,E,Z, [LD Z ], [W ORK ], [NFO ])
    CHARACTER (LEN=1) ::COMPZ
    \(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} \mathrm{Z}, \mathbb{I N F O}\)
    REAL,D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK
    REAL,D \(\mathbb{M}\) ENSION (:,:)::Z
    
## C INTERFACE

\#include <sunperfh>
void ssteqr(char com pz, intn, float *d, float *e, float *z, int $1 d z$, int *info);
void ssteqr_64 (char com pz, long n, float *d, float*e, float
*z, long ldz, long *info);

## PURPOSE

ssteqram putes alleigenvalues and, optionally, eigenvectors of a sym $m$ etric tridiagonalm atrix using the in pliciti Q L orQ R m ethod. The eigenvectors of a fullorband sym m etric $m$ atrix can also be found ifSSY TRD orSSPTRD orSSBTRD has been used to reduce thism atrix to tridiagonal form .

## ARGUMENTS

COMPZ (input)
$=\mathrm{N}$ ': C om pute eigenvalues only .
$=\mathrm{V}$ : Com pute eigenvalues and eigenvectors of the original sym $m$ etric $m$ atrix. On entry, $Z \mathrm{~m}$ ust contain the orthogonalm atrix used to reduce the originalm atrix to tridiagonal form . = I': C om pute eigenvalues and eigenvectors of the tridiagonal m atrix. Z is initialized to the identity $m$ atrix.

N (input) The order of the m atrix. $\mathrm{N}>=0$.
D (input/output)
On entry, the diagonalelem ents of the tridiagonal m atrix. On exit, if $\mathbb{N} F O=0$, the eigenvalues in ascending order.

E (input/output)
O $n$ entry, the ( $n-1$ ) subdiagonal elem ents of the tridiagonal matrix. On exit, E hasbeen destroyed.

Z (input) On entry, if $\mathrm{COMPZ}=\mathrm{V}$ ', then Z contains the orthogonal m atrix used in the reduction to tridiagonal form. On exit, if $\mathbb{N F O}=0$, then if COM PZ $=\mathrm{V}^{\prime}, \mathrm{Z}$ contains the orthonorm aleigenvectors of the original sym $m$ etric $m$ atrix, and ifCOMPZ = 'I',
$Z$ contains the orthonorm al eigenvectors of the sym $m$ etric tridiagonalm atrix. If COM PZ $=N^{\prime}$ ', then $Z$ is not referenced.

LD $Z$ (input)
The leading dim ension of the array Z . LD $\mathrm{Z}>=1$, and if eigenvectors are desired, then LD Z >= $\max (1, N)$.

W ORK (w orkspace)
dim ension (max ( $1,2 \star \mathrm{~N}-2$ )) If $\mathrm{COMPZ}=\mathrm{N}$ ', then W ORK is not referenced.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue
>0: the algorithm has failed to find all the eigenvalues in a total of $30 *$ N terations; if $\mathbb{N}$ FO
$=i$, then ielem ents of $E$ have not converged to zero; on exit, $D$ and $E$ contain the elem ents of a
sym $m$ etric tridiagonalm atrix which is orthogonally
sim ilar to the originalm atrix.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssterf-com pute alleigenvalues of a sym $m$ etric tridiagonal $m$ atrix using the Pal-w alkerK ahan variantof the $Q L$ or $Q R$ algorithm

## SYNOPSIS

```
SU BROUT\mathbb{NE SSTERF N,D,E, NNFO)}
```

$\mathbb{N}$ TEGER $N, \mathbb{N} F O$
REALD (*), E (*)
SUBROUTINE SSTERF_64 $\mathbb{N}, D, E, \mathbb{N} F O$ )
$\mathbb{N}$ TEGER*8 N, $\mathbb{N}$ FO
REALD ( ${ }^{*}$ ), E (*)

F95 INTERFACE
SU BROUT $\mathbb{N} E \operatorname{STERF}(\mathbb{N}], D, E,[\mathbb{N} F O])$
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
REAL,D IM ENSION (:) ::D,E

SU BROUTINE STERF_64 (N ],D ,E, [ $\mathbb{N} F O$ ])
$\mathbb{N} T E G E R(8):: N, \mathbb{N} F O$
REAL,D $\mathbb{I M}$ ENSION (:) ::D ,E

## C INTERFACE

\#include <sunperfh>
void ssterf(intn, float*d, float*e, int*info);

## PURPOSE

ssterf com putes alleigenvalues of a sym m etric tridiagonal $m$ atrix using the Pal-W alkerK ahan variantof the $Q \mathrm{~L}$ orQR algorithm .

## ARGUMENTS

N (input) The order of the m atrix. $\mathrm{N}>=0$.
D (input/output)
O n entry, the n diagonalelem ents of the tridiagonalm atrix. On exit, if $\mathbb{N} F O=0$, the eigenvalues in ascending order.

E (input/output)
O $n$ entry, the ( $n-1$ ) subdiagonal elem ents of the tridiagonal matrix. On exit, E hasbeen destroyed.
$\mathbb{N} F O$ (output)
= 0: successfulexit
< 0: if $\mathbb{N}$ FO = -i, the i-th argum enthad an illegalvalue
>0: the algorithm failed to find all of the eigenvalues in a total of $30 *$ N iterations; if $\mathbb{N}$ FO
$=i$, then ielem ents of $E$ have not converged to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sstev -com pute alleigenvalues and, optionally, eigenvectors of a realsym m etric tridiagonalm atrix A

## SYNOPSIS

```
SUBROUT\mathbb{NE SSTEV (OOBZ,N,D IA G,OFFD,Z,LD Z,W ORK, NNFO)}
CHARACTER * 1 JOBZ
```



```
REALDIAG (*),OFFD (*),Z (LD Z,*),W ORK (*)
SU BROUT\mathbb{NE SSTEV_64(JOBZ,N,D IAG,OFFD,Z,LD Z,W ORK, INFO )}
CHARACTER * 1 JOBZ
INTEGER*8N,LDZ,\mathbb{NFO}
REALDIAG (*),OFFD (*),Z (LD Z ,*),W ORK (*)
```


## F95 INTERFACE


CHARACTER (LEN=1)::JOBZ
$\mathbb{N} T E G E R:: N, L D Z, \mathbb{N} F O$
REAL,D $\mathbb{I M}$ ENSION (:) ::D $\mathbb{A} G, O F F D, W$ ORK
REAL,D $\mathbb{M}$ ENSION (:,:) ::Z
SUBROUTINE STEV_64 (OOBZ,N,D IAG,OFFD, Z, [LD Z ], $\mathbb{W}$ ORK ], [ $\mathbb{N} F O]$ )
CHARACTER (LEN=1)::JOBZ
$\mathbb{N}$ TEGER (8) :: $\mathrm{N}, \mathrm{LD} \mathrm{Z}, \mathbb{I N F O}$
REAL,D $\mathbb{I M}$ ENSION (:) ::D IA G,OFFD ,W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) ::Z

## C INTERFACE

\#include <sunperfh>
void sstev (char jobz, intn, float *diag, float *offd, float

* $z$, int $l d z$, int *info);
void sstev_64 (char j̀jbz, long n, float*diag, float *offfd, float * z, long ldz, long *info);


## PURPOSE

sstev com putes alleigenvahues and, optionally, eigenvectors of a realsym m etric tridiagonalm atrix A.

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}^{\prime}:$ C om pute eigenvalues only;
$=\mathrm{V}$ ': C om pute eigenvalues and eigenvectors.

N (input) The order of them atrix. $\mathrm{N}>=0$.

D IA G (input/output)
O n entry, the $n$ diagonalelem ents of the tridiago-
nal matrix A. On exit, if $\mathbb{N F O}=0$, the eigenvalues in ascending order.

OFFD (input/output)
O $n$ entry, the $(n-1)$ subdiagonal elem ents of the tridiagonal $m$ atrix $A$, stored in elem ents 1 to $N-1$ of OFFD ; OFFD $\mathbb{N}$ ) need notbe set, but is used by the routine. On exit, the contents of OFFD are destroyed.
$Z$ (input) $\operatorname{If} \mathrm{OBZ}=\mathrm{V}^{\prime}$, then if $\mathbb{N} F O=0, Z$ contains the orthonorm aleigenvectors of the m atrix $A$, w ith the i-th colum $n$ of $Z$ holding the eigenvector associated w ith D IA G (i). If $\mathrm{OOBZ}=\mathrm{N}$ ', then Z is not referenced.

LD Z (input)
The leading $d$ im ension of the array Z. LD Z $>=1$, and if $\mathrm{OBZ}=\mathrm{V}^{\prime}$, LD Z $>=\mathrm{m}$ ax $(1, N)$.

W ORK (w orkspace)
If $\mathrm{JOBZ}=\mathrm{N}^{\prime}, \mathrm{W} O R K$ is notreferenced.
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N F O}=i$, the algorithm failed to converge; i off-diagonal elem ents of OFFD did not converge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sstevd -com pute alleigenvalues and, optionally, eigenvectors of a real sym $m$ etric tridiagonalm atrix

## SYNOPSIS

```
SUBROUT\mathbb{NE SSTEVD (OBBZ,N,D,E,Z,LD Z,W ORK,LW ORK,IN ORK,L\mathbb{N ORK,}}\mathbf{N},\textrm{L}
    INFO)
CHARACTER * 1 JOBZ
\mathbb{NTEGER N,LD Z,LW ORK,LIN ORK,INFO}
INTEGER IN ORK (*)
REALD (*),E (*),Z (LD Z ,*),W ORK (*)
SU BROUTINE SSTEVD_64(JOBZ,N,D,E,Z,LD Z,W ORK,LW ORK,IN ORK,
    LIN ORK,INFO)
```

CHARACTER * 1 JOBZ
$\mathbb{N}$ TEGER*8N,LD Z,LW ORK,LIW ORK, $\mathbb{N} F$ O
$\mathbb{N} T E G E R * 8 \mathbb{I N}$ ORK ( ${ }^{*}$ )
REALD (*), E (*), Z (LD Z, $\left.{ }^{\star}\right), \mathrm{W} O R K(*)$

## F95 INTERFACE

SU BROUTINE STEVD (JOBZ,N,D,E,Z, [LD Z], [W ORK ], [LW ORK ], [IW ORK ], $\left[\begin{array}{l}\mathbb{N} \\ \text { ORK ], }[\mathbb{N} F O])\end{array}\right.$

CHARACTER (LEN=1)::JOBZ
$\mathbb{N} T E G E R:: N, L D Z, L W$ ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{I N}$ ORK
REAL,D IM ENSION (:) ::D , E,W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) :: Z
SU BROUTINE STEVD_64 (JOBZ,N,D,E,Z, [LD Z], [W ORK ], [LW ORK], [IW ORK ],
$[\mathrm{L} \mathbb{N} O R K],[\mathbb{N} F O])$

CHARACTER (LEN=1) :: OBZ
$\mathbb{N}$ TEGER (8) :: $N$, LD Z, LW ORK, LIW ORK, $\mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N} O R K$
REAL,D $\mathbb{M}$ ENSION (:) ::D , E, W ORK
REAL,D $\mathbb{I}$ ENSION (:,:) ::Z

## C INTERFACE

\#include <sunperfh>
void sstevd (char jobz, intn, float *d, float *e, float *z, int ldz, int*info);
void sstevd_64 (char jobz, long n, float *d, float *e, float
*z, long ldz, long *info);

## PURPOSE

sstevd com putes alleigenvalues and, optionally, eigenvectors of a realsym $m$ etric tridiagonalm atrix. If eigenvectors are desired, it uses a divide and conquer algo rithm .

The divide and conqueralgorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on $m$ achines w ith a guard digit in add/subtract, or on those binary $m$ achines $w$ thout guard digits w hich subtract like the $C$ ray $X-M P, C$ ray $Y \neq M P, C$ ray $C-90$, orC ray-2. It could conceivably fail on hexadecim al or decim al $m$ achines $w$ ithout guard digits, butw e know of none.

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}$ ': C om pute eigenvalues only;
$=\mathrm{V}$ ': C om pute eigenvalues and eigenvectors.

N (input) The order of the m atrix. $\mathrm{N}>=0$.
D (input/output)
O n entry, the $n$ diagonalelem ents of the tridiagonal m atrix A . On exit, if $\mathbb{N F} \mathrm{FO}=0$, the eigenvalues in ascending order.

E (input/output)
O $n$ entry, the ( $n-1$ ) subdiagonal elem ents of the tridiagonal m atrix A, stored in elem ents 1 to $N-1$ of $E$; $E \mathbb{N}$ ) need notbe set, but is used by the
routine. On exit, the contents ofe are destroyed.
$Z$ (input) If $J O B Z=V '$, then if $\mathbb{N} F O=0, Z$ contains the orthonorm aleigenvectors of the $m$ atrix $A, w$ th the $i$-th colum $n$ of $Z$ holding the eigenvector associated w th D (i). If JOBZ $=N^{\prime}$ ', then $Z$ is not referenced.

LD $Z$ (input)
The leading din ension of the amay Z . LD $\mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD Z >= $\max (1, N)$.
W ORK (w orkspace)
dim ension ( $\mathbb{L} W$ ORK)On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim alLW O RK .

LW ORK (input)
The dim ension of the array $W$ ORK. If $J O B Z=N^{\prime}$ or $\mathrm{N}<=1$ then LW ORK m ustbe at least1. If JO B Z $=\mathrm{V}$ 'and $\mathrm{N}>1$ then LW ORK m ust.be at least ( $1+$ $4 * N+N * * 2)$.

If LW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of theW ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IN ORK (w orkspace/output)
On exit, if $\mathbb{N} F O=0, \mathbb{I V} O R K(1)$ retums the optim al LIN ORK.

LIV ORK (input)
The dim ension of the array $\mathbb{I N} O R K$. If $J O B Z=N^{\prime}$ orN <= 1 then LIW ORK m ustbe at least1. If JO B Z $=V$ 'and $N>1$ then $L \mathbb{I}$ ORK must be at least $3+5 * N$.

If LIV ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the IV ORK array, retums this value as the first entry of the $\mathbb{I N}$ ORK amray, and no errorm essage related to $L \mathbb{I N} O R K$ is issued by XERBLA.
$\mathbb{N}$ FO (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N F O}=-i$, the i-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N F O}=\mathrm{i}$, the algorithm failed to con-
verge; i off-diagonalelem ents ofe did not con-
verge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

sstevr - com pute selected eigenvalues and, optionally,
eigenvectors of a realsym $m$ etric tridiagonalm atrix $T$

## SYNOPSIS

```
SUBROUT\mathbb{NE SSTEVR(JOBZ,RANGE,N,D,E,VL,VU, U, IU,ABSTOL,M,W,}
```



```
CHARACTER * 1 JOBZ,RANGE
\mathbb{N}TEGERN,\mathbb{L},\mathbb{U},M,LDZ,LWORK,L\mathbb{N ORK,\mathbb{NFO}}\mathbf{M}\mathrm{ , L}
INTEGER ISUPPZ (*), IN ORK (*)
REAL VL,VU,ABSTOL
REALD (*),E (*),W (*),Z (LD Z ,*),W ORK (*)
SUBROUTINE SSTEVR_64(JOBZ,RANGE,N,D,E,VL,VU,IL,IU,ABSTOL,M,
    W,Z,LDZ,ISUPPZ,W ORK,LW ORK,IN ORK,LIN ORK,INFO)
```

CHARACTER * 1 JOBZ,RANGE
$\mathbb{N}$ TEGER*8N, $\mathbb{I}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W$ ORK,LIN ORK, $\mathbb{N} F \mathrm{~F}$
$\mathbb{N} T E G E R * 8 \operatorname{ISUPPZ}$ (*), $\mathbb{I N}$ ORK (*)
REALVL,VU,ABSTOL
REALD (*), E (*), W (*), Z (LD Z ,*), W ORK (*)

## F95 INTERFACE

SU BROUTINE STEVR (JOBZ,RANGE, $\mathbb{N}], D, E, V L, V U, \mathbb{I}, \mathbb{U}, A B S T O L, M$, W , Z, [LD Z], ISUPPZ, [W ORK ], [LW ORK ], [IN ORK ], [LINORK], [ $\mathbb{N} F O])$

CHARACTER (LEN=1): : JOBZ,RANGE
$\mathbb{N} T E G E R:: N, \mathbb{I}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W$ ORK,LIN ORK, $\mathbb{N} F \mathrm{O}$
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: ISUPPZ, $\mathbb{I N}$ ORK
REAL ::VL,VU,ABSTOL

REAL,D $\mathbb{M}$ ENSION (:) ::D , E, W ,W ORK
REAL,D IM ENSION (:,:) ::Z

SU BROUTINE STEVR_64 (OBZ,RANGE, $\mathbb{N}], D, E, V L, V U, \mathbb{L}, \mathbb{U}, A B S T O L$, $M, W, Z,[L D Z], I S U P P Z,[W O R K],[L W O R K],[\mathbb{W} O R K],[L \mathbb{N} O R K],[\mathbb{N} F O])$

CHARACTER (LEN=1) :: JOBZ,RANGE
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{\Pi}, \mathbb{Z}, \mathrm{M}, \mathrm{LD} \mathrm{Z}, \mathrm{LW}$ ORK, LIW ORK, $\mathbb{N} F O$
$\mathbb{N}$ TEGER (8), D $\mathbb{I M} E N S I O N$ (:) :: ISUPPZ , $\mathbb{I N}$ ORK
REAL ::VL,VU,ABSTOL
REAL,D $\mathbb{M} E N S I O N(:):$, $, \mathrm{E}, \mathrm{W}, \mathrm{W} O R K$
REAL,D $\mathbb{M}$ ENSION (:,:) :: Z

## C INTERFACE

\#include <sunperfh>
void sstevr(char jobz, char range, intn, float *d, float
*e, float vl, float vu, int il, intiu, float abstol, int *m, float * w , float * z , int ld z , int *isuppz, int *info);
void sstevr_64 (char j̀jbz, char range, long n, float *d, float *e, float vl, floatvu, long il, long iu, floatabstol, long *m, float*w, float *z, long ldz, long *isuppz, long *info);

## PURPOSE

sstevr com putes selected eigenvahues and, optionally, eigenvectors of a realsym m etric tridiagonalm atrix T. Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

W heneverpossible, SSTEVR calls SSTEGR to com pute the eigenspectrum using Relatively Robust Representations. SSTEGR computes eigenvalues by the dqds algorithm, while orthogonaleigenvectors are com puted from various "good" L D $\mathrm{L}^{\wedge} \mathrm{T}$ representations (also known as Relatively Robust Representations).G ram -Schm idtorthogonalization is avoided as far as possible. M ore specifically, the various steps of the algorithm are as follow s.For the $i$-th unreduced block of $T$,
(a) C om pute $\mathrm{T}-$ sigm a_i= L_iD_iL_i^T, such that L_i D_iL_i^T
is a relatively robust representation,
(b) C om pute the eigenvalues, lam bda_j, of L_i D _i L_i^T to high
relative accuracy by the dqds algorithm,
(c) If there is a cluster of close eigenvahues, "choose"
sigm a_i
close to the cluster, and go to step (a),
(d) G iven the approxim ate eigenvalue lam bda_jofL_i D_i L_i^T,
com pute the comesponding eigenvectorby form ing a rank-revealing tw isted factorization. The desired accuracy of the output can be specified by the inputparam eterABSTOL.

Form ore details, see "A new $\mathrm{O}\left(\mathrm{n}^{\wedge} 2\right)$ algorithm for the sym m etric tridiagonal eigenvalue/eigenvector problem ", by Inder \#̈tD hillon, C om puter Science D ìvision TechnicalReport No.UCB/C SD -97-971, UC Berkeley, M ay 1997.

N ote 1 :SSTEVR calls SSTEGR when the full spectrum is requested on $m$ achines $w$ hich conform to the iee-754 floating pointstandard. SSTEVR callsSSTEBZ and SSTE IN on non-ieee m achines and when partial spectrum requests are $m$ ade.

N orm alexecution of SSTEGR m ay create NaNs and infinities and hence $m$ ay abort due to a floating point exception in environm ents which do nothandle N aN s and infinities in the ieee standard defaultm anner.

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}$ ': C om pute eigenvalues only;
= V : C om pute eigenvalues and eigenvectors.
RANGE (input)
= A : alleigenvalues $w$ illbe found.
= V : alleigenvahues in the half-open interval
( $\mathrm{L}, \mathrm{V} \mathrm{U}]$ will be found. = 'I': the $\mathbb{I}$-th through
$\mathbb{I U}$-th eigenvaluesw illlbe found.

N (input) The order of the m atrix. $\mathrm{N}>=0$.

D (input/output)
O n entry, the n diagonalelem ents of the tridiagonal $m$ atrix A. On exit, D m ay be multiplied by a constant factor chosen to avoid over/underflow in com puting the eigenvalues.

E (input/output)
O $n$ entry, the ( $n-1$ ) subdiagonal elem ents of the tridiagonal m atrix A in elem ents 1 to $\mathrm{N}-1$ ofe;

E(N) need notbe set. On exit, E may be multiplied by a constant factor chosen to avoid over/underflow in com puting the eigenvahues.

## VL (input)

IfRANGE=V ', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE=A'or I'.

VU (input)
See the description of V L .

II (input)
IfRA NGE= 'I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{Z}<=\mathbb{U}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{H}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N ot referenced ifRANGE $=$ A'or V'.
$\mathbb{U}$ (input)
See the description of II.

ABSTOL (input)
The absolute error tolerance for the eigenvalues.
A $n$ approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b] of w idth less than orequal to

ABSTOL + EPS * $\max (|k|, \mid)$ ),
$w$ here EPS is the m achine precision. IfA BSTOL is less than or equalto zero, then EPS* $\mid$ | w illbe used in its place, where $\mid T$ is the 1 -norm of the tridiagonal $m$ atrix obtained by reducing A to tridiagonal form .

See "C om puting Sm allSingularV ahues of B idiagonal M atrices w ith G uaranteed H igh Relative A ccuracy," by Dem m eland K ahan, LA PA CK W orking N ote \#3.

If high relative accuracy is im portant, setA BSTO L to SLAM CH (Safe minim um '). D oing so will guarantee thateigenvalues are com puted to high relative accuracy when possible in future releases. The current code does not make any guarantees about high relative accuracy, but future releases w ill. See J. B arlow and J. D em m el, "C om puting A ccurate E igensystem s of Scaled D iagonally D om inantM atrioes", LA PA CK W orking N ote \#7, for a discussion of $w$ hich $m$ atrices define their
eigenvalues to high relative accuracy .

M (output)
The total num berofeigenvalues found. $0<=\mathrm{M}$ <=
N . IfRANGE $=A^{\prime}, \mathrm{M}=\mathrm{N}$, and ifRANGE $=\mathrm{I}^{\prime}, \mathrm{M}=$ $\mathbb{U}-\mathbb{U}+1$.

W (output)
The firstM elem ents contain the selected eigenvalues in ascending order.
$Z$ (input) If $\mathcal{J O B Z}=\mathrm{V}^{\prime}$, then if $\mathbb{N F O}=0$, the first $M$ colum ns of $Z$ contain the orthonorm aleigenvectors of them atrix A comesponding to the selected eigenvalues, w ith the i -th colum n of $Z$ holding the eigenvector associated $w$ ith $W$ (i). $N$ ote: the user $m$ ust ensure that at leastm ax ( $1, \mathrm{M}$ ) colum ns are supplied in the array $Z$; ifRANGE = $V$ ', the exact value of $M$ is not know $n$ in advance and an upper bound $m$ ustbe used.

LD Z (input)
The leading dim ension of the array $\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}$ ', LD $Z>=\mathrm{max}(1, \mathrm{~N})$.

## ISU PPZ (output)

The support of the eigenvectors in $Z$, i.e., the indices indicating the nonzero elem ents in $Z$. The $i$-th eigenvector is nonzero only in elem ents ISU PPZ (2*i-1 ) through ISU PPZ (2*i).

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al (andminim al) LW ORK .

LW ORK (input)
The dim ension of the array $W$ ORK. LW ORK $>=20 * N$.
If LW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IV ORK (w orkspace/output)
On exit, if $\mathbb{N} F O=0, \mathbb{I N}$ ORK (1) retums the optim al (andminim al) LINORK.

LIN ORK (input)
The dim ension of the anay $\mathbb{I N}$ ORK. LIN ORK $>=10 * N$.

If LIW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the $\mathbb{I V}$ ORK anray, retums this value as the first entry of the $\mathbb{I W}$ ORK array, and no errorm essage related to $L \mathbb{I W}$ ORK is issued by XERBLA.
$\mathbb{I N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the $i$ th argum enthad an illegalvalue
> 0: Intemalemor

## FURTHER DETAILS

B ased on contributions by
Inder]̈̈t D hillon, $\mathbb{B M}$ A $\operatorname{lm}$ aden, U SA
O sniM arques, LBNL N ER SC , U SA
K en Stanley, C om puterScience D ivision, U niversity of C alifomia at B erkeley, U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sstevx - com pute selected eigenvalues and, optionally, eigenvectors of a realsym $m$ etric tridiagonalm atrix A

## SYNOPSIS

```
SUBROUT\mathbb{NE SSTEVX(OBZ,RANGE,N,DIAG,OFFD,VL,VU, IL,IU,ABTOL,}
    NFOUND,W,Z,LDZ,WORK,INORK2,\mathbb{FA}\mathbb{L},\mathbb{N}FO)
CHARACTER * 1 JOBZ,RANGE
\mathbb{NTEGERN,\mathbb{L,}\mathbb{U},NFOUND,LD Z,}\mathbb{N}FO
\mathbb{NTEGER IN ORK2(*),\mathbb{FA L (*)}}\mathbf{(*)}
REALVL,VU,ABTOL
REALDIAG (*),OFFD (*),W (*),Z (LD Z,*),W ORK (*)
```




```
CHARACTER * 1 JOBZ,RANGE
\mathbb{NTEGER*8N,\mathbb{N},\mathbb{U},NFOUND,LD Z,INFO}
\mathbb{NTEGER*8 IN ORK2 (*), \mathbb{FA LH (*)}}\mathbf{(*)}
REALVL,VU,ABTOL
REALDIAG (*),OFFD (*),W (*),Z (LD Z ,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE STEVX (OBZ,RANGE,N,D $\mathbb{A} G, O F F D, V L, V U, \mathbb{I}, \mathbb{U}, A B T O L$, NFOUND, W, Z, [LD Z], [W ORK], [IW ORK 2], $\mathbb{F A} \mathbb{I},[\mathbb{N} F O])$

CHARACTER (LEN=1)::JOBZ,RANGE
$\mathbb{N} T E G E R:: N, \mathbb{L}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: $\mathbb{I W}$ ORK2, $\mathbb{F A} \mathbb{I}$
REAL ::VL,VU,ABTOL
REAL,D $\mathbb{I M}$ ENSION (:) ::D $\mathbb{A} G, O F F D, W, W$ ORK

SUBROUTINE STEVX_64 (DOBZ,RANGE,N,DIAG,OFFD,VL,VU, $\mathbb{I}, \mathbb{I U}$, ABTOL,NFOUND,W,Z,[LDZ], [WORK],[IWORK2], $\mathbb{F} A \mathbb{I},[\mathbb{N} F O$ ])

CHARACTER (LEN=1): : OBZ,RANGE
$\mathbb{N}$ TEGER (8) :: N , $\mathbb{I}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N F O}$
$\mathbb{N}$ TEGER (8), D $\mathbb{I M}$ ENSION (:) :: $\mathbb{I N} O R K 2, \mathbb{F} A \mathbb{L}$
REAL ::VL,VU,ABTOL
REAL,DIMENSION (:) ::DIAG,OFFD,W ,W ORK
REAL,D $\mathbb{I M}$ ENSION (:,:) ::Z

## C INTERFACE

\#include <sunperfh>
void sstevx (char jobz, char range, intn, float *diag, float
*offd, float vl, floatvu, int il, intiu, float
abtol, int *nfound, float *W, float * $z$, int ldz, int*ifail, int*info);
void sstevx_64 (char jobz, char range, long n, float *diag, float *offd, floatvl, float vu, long il, long iu, float abtol, long *nfound, float *w, float *z, long ldz, long *ifail, long *info);

## PURPOSE

sstevx com putes selected eigenvahues and, optionally , eigenvectors of a realsym m etric tridiagonalm atrix A. Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

## ARGUMENTS

JO BZ (input)
$=\mathrm{N}^{\prime}:$ C om pute eigenvalues only;
$=\mathrm{V}$ ': C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvalues w illbe found.
$=\mathrm{V}$ ': alleigenvalues in the half-open interval
(VL, VU] will be found. = I': the IL th through IU th eigenvalues w illbe found.

N (input) The order of the m atrix. $\mathrm{N}>=0$.

D IA G (input/output)

O n entry, the n diagonalelem ents of the tridiagonal $m$ atrix A. On exit, D IA G m ay be multiplied by a constant factor chosen to avoid over/undenflow in com puting the eigenvalues.

OFFD (input/output)
O $n$ entry, the ( $n-1$ ) subdiagonal elem ents of the tridiagonalm atrix $A$ in elem ents 1 to $N-1$ of FFD ; OFFD $\mathbb{N}$ ) need notbe set. On exit, OFFD may be m ultiplied by a constant factor chosen to avoid over/underflow in com puting the eigenvalues.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE = A 'or I'.

VU (input)
See the description of V L .
II (input)
IfRA N G E= 'I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{U}<=\mathbb{U}<=N$, if $N>0$; $\mathbb{H}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N ot referenced ifRANGE $=$ A 'or V'.

IU (input)
See the description of II .
ABTOL (input)
The absolute error tolerance for the eigenvalues. A $n$ approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b] of w idth less than orequal to

ABTOL + EPS * $\max (\nmid,||$,$) ,$
where EPS is them achine precision. If ABTOL is less than or equal to zero, then EPS* $\mid$ | w illbe used in its place, where $F \mid$ is the 1 -norm of the tridiagonalm atrix.

E igenvalues w illbe com puted m ostaccurately when ABTOL is set to tw ice the underflow threshold 2*SLAM CH (S ), notzero. If this routine retums w ith $\mathbb{N}$ FO $>0$, indicating that som e eigenvectors did not converge, try setting ABTOL to $2 *$ SLAM CH ( S ).

See "C om puting Sm allSingularV ahes ofB idiagonal
$M$ atrices w th G uaranteed H igh Relative A ccuracy," by Dem m eland $K$ ahan, LA PA CK W orking $N$ ote \#3.

NFOUND (output)
The total num ber of eigenvalues found. $0<=$ NFOUND <= N. IfRANGE = A',NFOUND = N, and if RANGE $=I ', N F O U N D=\mathbb{U}-\mathbb{H}+1$.

W (output)
The firstNFOUND elem ents contain the selected eigenvalues in ascending order.

Z (input) If $\mathrm{JOBZ}=\mathrm{V}^{\prime}$, then if $\mathbb{N} F O=0$, the first NFOUND colum ns of $Z$ contain the orthonorm aleigenvectors of them atrix A comesponding to the selected eigenvalues, $w$ ith the i-th colum $n$ of $Z$ holding the eigenvector associated w ith $W$ (i). If an eigenvector fails to converge ( $\mathbb{N}$ FO $>0$ ), then that colum $n$ of $Z$ contains the latest approxim ation to the eigenvector, and the index of the eigenvector is retumed in $\mathbb{F A} I I$. If $\mathrm{JOBZ}=\mathrm{N}$ ', then Z is not referenced. N ote: the userm ustensure that at leastm ax (1,NFOUND) colum ns are supplied in the array $Z$; ifRANGE = $V$ ', the exactvalue ofNFOUND is not know $n$ in advance and an upperbound $m$ ust.be used.

LD Z (input)
The leading dim ension of the array $\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1$, and if $\mathrm{JOBZ}=V^{\prime}$, LD $Z>=\max (1, N)$.

W ORK (w orkspace)
dim ension ( $5 * \mathrm{~N}$ )

IW ORK 2 (w orkspace)
FFII (output)
If $\mathrm{JOBZ}=\mathrm{V}$ ', then if $\mathbb{N F O}=0$, the first NFOUND
elem ents of $\mathbb{F A} I I$ are zero. If $\mathbb{N F O}>0$, then
FAII contains the indices of the eigenvectors
that failed to converge. If $\mathrm{JOB}=\mathrm{N}^{\prime}$, then
IFA II is not referenced.
$\mathbb{N} F O$ (output)
= 0 : successfinlexit
<0: if $\mathbb{N N}$ FO $=-$ i, the i-th argum enthad an illegalvalue
> 0 : if $\mathbb{N F O}=$ i, then ieigenvectors failed to converge. Their indioes are stored in array
ㅍFAI.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

sstsv - com pute the solution to a system of linear equations
$A * X=B$ where $A$ is a sym m etric tridiagonalm atrix

## SYNOPSIS

```
SU BROUT\mathbb{NE SSTSV N,NRHS,L,D,SUBL,B,LDB,IPIV,\mathbb{NFO)}}\mathbf{N},\textrm{N},\textrm{S}
\mathbb{NTEGERN,NRHS,LDB,INFO}
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(*)}
REALL (*),D (*),SUBL (*),B (LD B ,*)
```



```
\mathbb{NTEGER*8N,NRHS,LDB,INFO}
\mathbb{NTEGER*8 \mathbb{P IV (*)}}\mp@subsup{}{(}{*}
REALL (*),D (*),SUBL (*),B (LDB,*)
```


## F95 INTERFACE

SU BROUTINE STSV $\mathbb{N}, N R H S, L, D, S U B L, B,[L D B], \mathbb{P} \mathbb{I},[\mathbb{N} F O])$
$\mathbb{N}$ TEGER ::N,NRHS,LDB, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}$
REAL,D IM ENSION (:) ::L,D ,SUBL
REAL,D $\mathbb{I M}$ ENSION (:,:) ::B
SUBROUTINESTSV_64 $\mathbb{N}, N R H S, L, D, S U B L, B,[L D B], \mathbb{P} \mathbb{I},[\mathbb{N} F O])$
$\mathbb{N}$ TEGER (8) ::N,NRHS,LD B, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I M}$ ENSION(:) :: $\mathbb{P} \mathbb{V}$
REAL,D IM ENSION (:) ::L,D ,SUBL
REAL,D $\mathbb{M}$ ENSION (:,:) ::B

## C INTERFACE

\#include <sunperfh>
void sstsv (intn, intnihs, float *l, float *d, float *subl, float *b, int ldb, int *ípiv, int *info);
void sstsv_64 (long n, long nins, float *l, float *d, float *subl, float *b, long ldb, long *ịiv, long
*info);

## PURPOSE

sstsv com putes the solution to a system of linear equations $A * X=B$ where $A$ is a symm etric tridiagonalm atrix.

## ARGUMENTS

N (input) $\mathbb{N} T E G E R$
The order of them atrix $A . N>=0$.

NRHS (input)
The num ber of righthand sides in B.

L (input/output)
REAL array, dim ension $\mathbb{N}$ )
O n entry, the n-1 subdiagonalelem ents of the tridiagonal m atrix A. On exit, part of the factorization ofA .

D (input/output)
REA L aray, dim ension $\mathbb{N}$ )
O n entry, the $n$ diagonalelem ents of the tridiagonalm atrix A. On exit, the $n$ diagonalelem ents of the diagonalm atrix D from the factorization of $A$.

SUBL (output)
REA L array, dim ension $(\mathbb{N})$
On exit, part of the factorization of A.

B (input/output)
The colum ns ofB contain the righthand sides.

LD B (input)
The leading dim ension ofB as specified in a type orD $\mathbb{I M}$ ENSION statem ent.

IP IV (output)
$\mathbb{N}$ TEGER array, dim ension $\mathbb{N}$ )
On exit, the pivot indices of the factorization.
$\mathbb{I N F O}$ (output)
$\mathbb{N}$ TEGER
= 0: successfulexit
$<0:$ if $\mathbb{N N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue
$>0$ : if $\mathbb{N} F O=i, D(k, k)$ is exactly zero. The factorization has been com pleted, but the block diagonalm atrix $D$ is exactly singular and division by zero w illoccur if it is used to solve a system of equations.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssturf-com pute the factorization of a sym $m$ etric tridiagonalm atrix A

## SYNOPSIS

SU BROUTINE SSTTRF $\mathbb{N}, L, D, S U B L, \mathbb{P} \mathbb{I}, \mathbb{N} F O$ )
$\mathbb{N}$ TEGER $N, \mathbb{N} F O$
$\left.\mathbb{N} T E G E R \mathbb{P} \mathbb{I}{ }^{( }{ }^{*}\right)$
REALL ( ${ }^{*}$ ) , D ( $\left.{ }^{*}\right)$, SUBL (*)
SU BROUTINE SSTTRF_64 $\mathbb{N}, L, D, S U B L, \mathbb{P} \mathbb{I}, \mathbb{N} F O)$
$\mathbb{N}$ TEGER*8 $\mathrm{N}, \mathbb{I N F O}$
$\mathbb{N T E G E R *} \mathbb{P}^{\mathbb{P}} \mathbb{I}$ ( $\left.^{\star}\right)$
REALL (*) , D (*) , SUBL (*)

## F95 INTERFACE

SU BROUTINE STTRF ( $\mathbb{N}], L, D, S U B L, \mathbb{P} \mathbb{I}, ~[\mathbb{N} F O])$
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}$
REAL,D $\mathbb{I}$ ENSION (:) ::L,D ,SUBL
SU BROUTINE STTRF_64 ( $\mathbb{N}$ ],L,D ,SUBL, $\mathbb{P} \mathbb{I V},[\mathbb{N} F O]$ )
$\mathbb{N}$ TEGER (8) :: $\mathrm{N}, \mathbb{N}$ FO
$\mathbb{N} T E G E R(8), D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}$
REAL,D IM ENSION (:) :: L, D ,SUBL

## C INTERFACE

\#include <sunperfh>
void ssturf(intn, float*l, float *d, float *subl, int *ípív, int *info);
void ssttrf_64 (long n, float *l, float *d, float *subl, long *ipiv, long *info);

## PURPOSE

ssturf com putes the factorization of a com plex H erm itian tridiagonalm atrix A .

## ARGUMENTS

N (input) $\mathbb{N} T E G E R$
The order of them atrix $A . N>=0$.

L (input/output)
REAL aray, dim ension $\mathbb{N}$ )
O n entry, the $\mathrm{n}-1$ subdiagonalelem ents of the tridiagonal m atrix A. On exit, part of the factorization of A .

D (input/output)
REAL array, dim ension $\mathbb{N}$ )
O n entry, the $n$ diagonalelem ents of the tridiagonalm atrix A. On exit, the $n$ diagonalelem ents of the diagonalm atrix $D$ from the $L * D * L * * H$ factorization ofA.

SU BL (output)
REA L array, dim ension $\mathbb{N}$ )
O n exit, part of the factorization ofA.

IP IV (output)
$\mathbb{N}$ TEGER array, dim ension $(\mathbb{N})$
On exit, the pivot indices of the factorization.
$\mathbb{N}$ FO (output)
IN TEGER
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the $i-t h$ argum enthad an ille-
galvalue
$>0:$ if $\mathbb{N F O}=\mathrm{i}, \mathrm{D}(\mathrm{k}, \mathrm{k})$ is exactly zero. The
factorization has been com pleted, but the block
diagonalm atrix $D$ is exactly singular and division
by zero w illoccur if it is used to solve a system
of equations.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssturs - com putes the solution to a real system of linear equations $A * X=B$

## SYNOPSIS



```
\mathbb{NTEGERN,NRHS,LDB,INFO}
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(*)}
REALL (*),D (*),SUBL (*),B (LD B ,*)
```



```
\mathbb{NTEGER*8N,NRHS,LDB,INFO}
\mathbb{NTEGER*8 \mathbb{P IV (*)}}\mp@subsup{}{(}{*}
REALL (*),D (*),SUBL (*),B (LDB,*)
```


## F95 INTERFACE

SUBROUTINE STTRS $\mathbb{N}, N R H S, L, D, S U B L, B,[L D B], \mathbb{P} \mathbb{I},[\mathbb{N} F O])$
$\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}$
REAL,D IM ENSION (:) :: L, D ,SUBL
REAL,D $\mathbb{I M}$ ENSION (:,:) ::B
SU BROUTINE STTRS_64 $\mathbb{N}, N R H S, L, D, S U B L, B,[L D B], \mathbb{P} \mathbb{V},[\mathbb{N} F O])$
$\mathbb{N}$ TEGER (8) ::N,NRHS,LD B, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I M}$ ENSION(:) :: $\mathbb{P} \mathbb{V}$
REAL,D IM ENSION (:) ::L,D ,SUBL
REAL,D $\mathbb{M}$ ENSION (:,:) ::B

## C INTERFACE

\#include <sunperfh>
void ssttrs (intn, int nihs, float *l, float *d, float
*subl, float *b, int ldb, int *ípiv, int *info);
void ssttrs_64 (long n, long nrhs, float *l, float *d, float
*subl, float *b, long ldb, long *ịiv, long
*info);

## PURPOSE

ssttrs com putes the solution to a real system of linear equations $A * X=B$, where $A$ is an $N$ boy $-N$ symm etric tridiagonalm atrix and X and B are N boy-N R H S m atrices.

## ARGUMENTS

N (input) $\mathbb{N} T E G E R$
The order of them atrix $A . N>=0$.

NRHS (input)
$\mathbb{N}$ TEGER
The num ber of right hand sides, i.e., the num ber of colum ns of the m atrix B.NRH S >=0.
$L$ (input) REAL array, dim ension $(\mathbb{N}-1)$
O n entry, the subdiagonalelem ents ofLL and D D.

D (input) REAL array, dim ension $\mathbb{N}$ )
O n entry, the diagonalelem ents ofD D .

SUBL (input)
REA L anray, dim ension $(\mathbb{N}-2)$
O n entry, the second subdiagonalelem ents of LL .

B (input/output)
REA L array, dim ension
(LDB, NRHS) On entry, the N boy NRHS right hand side $m$ atrix $B$. On exit, if $\mathbb{N F O}=0$, the N -byN RH S solution m atrix X .

LD B (input)
IN TEGER
The leading dim ension of the aray B. LD B >= $\max (1, N)$

IPIV (output)
$\mathbb{N}$ TEGER array, dim ension $\mathbb{N}$ )
D etails of the interchanges and block pivot. If $\mathbb{P} \mathbb{V}(\mathbb{K})>0,1$ by 1 pìvot, and if $\mathbb{P} \mathbb{V}(\mathbb{K})=K+1$ an interchange done; If $\mathbb{P} \mathbb{I V}(\mathbb{K})<0,2$ by 2
pivot, no interchange required.
$\mathbb{I N F O}$ (output)
$\mathbb{N}$ TEGER
= 0: successfulexit
$<0$ : if $\mathbb{N} F O=-k$, the $k$-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssw ap -Exchange vectors $x$ and $y$.

## SYNOPSIS



```
INTEGER N,\mathbb{NCX,INCY}
REALX (*),Y (*)
```



```
INTEGER*8N,\mathbb{NCX,INCY}
REALX (*),Y (*)
F95 INTERFACE
```



```
    \mathbb{NTEGER ::N,\mathbb{NCX,}\mathbb{NCY}}\mathbf{N}={
    REAL,D IM ENSION (:) ::X,Y
```



```
    \mathbb{NTEGER (8)::N,\mathbb{NCX,}\mathbb{NCY}}\mathbf{N}={
    REAL,D IM ENSION (:) ::X,Y
C INTERFACE
    #include <sunperfh>
    void ssw ap (intn, float *x, intincx, float *y, int incy);
    void ssw ap_64 (long n, float *x, long incx, float *y, long
        incy);
```


## PURPOSE

ssw ap Exchange $x$ and $y$ where $x$ and $y$ are $n$-vectors.

## ARGUMENTS

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.
X (input/output)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C X))$. On entry, the increm ented array $X$ m ust contain the vector $x$. On exit, the $y$ vector.
$\mathbb{N C X}$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents of $\mathrm{X} . \mathbb{N} C X$ m ustnotbe zero. U nchanged on exit.

Y (input/output)
(1+(n-1)*abs( $\mathbb{N} C Y)$ ). On entry, the increm ented array $Y$ m ust contain the vectory. On exit, the x vector.
$\mathbb{N} C Y$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of Y. $\mathbb{N} C Y$ m ustnotbe zero. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssycon -estim ate the reciprocalof the condition num ber (in the 1 -norm ) of a realsym $m$ etric $m$ atrix A using the factorization $A=U * D * U * * T$ orA $=L * D * L * * T$ com puted by SSY TRF

## SYNOPSIS

```
SUBROUT\mathbb{NE SSYCON (UPLO,N,A,LDA, \mathbb{PIVOT,ANORM,RCOND,W ORK,}}\mathbf{N},\textrm{A},\textrm{A}
    IN ORK2,INFO)
CHARACTER * 1 UPLO
NNTEGERN,LDA,}\mathbb{N}F
\mathbb{NTEGER \mathbb{PIVOT (*), IN ORK2 (*)}}\mathbf{(*)}
REAL ANORM,RCOND
REAL A (LDA,*),W ORK (*)
SU BROUT\mathbb{NE SSYCON_64(UPLO,N,A,LDA,\mathbb{PIVOT,ANORM,RCOND,W ORK,}}\mathbf{N},
    IN ORK2, INFO)
CHARACTER * 1 UPLO
\mathbb{N TEGER*8N,LDA,}\mathbb{N}FO
\mathbb{NTEGER*8 \mathbb{PIVOT (*), IN ORK2 (*)}}\mathbf{(*)}
REALANORM,RCOND
REAL A (LDA,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE SYCON (UPLO,N,A, [LDA], $\mathbb{P} \mathbb{I V O T}, A N O R M, R C O N D,[W O R K]$, [ $\mathbb{I W}$ ORK2], [ $\mathbb{N F O}$ ])

CHARACTER (LEN=1)::UPLO
$\mathbb{N}$ TEGER ::N,LDA, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V O T}, \mathbb{I}$ ORK2
REAL ::ANORM,RCOND

SUBROUTINE SYCON_64 (UPLO,N,A, [LDA], $\mathbb{P} \mathbb{I} O$ OT,ANORM,RCOND, $\mathbb{W}$ ORK $]$, [IW ORK2], [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER (8) :: N , LDA, $\mathbb{N}$ FO
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T, \mathbb{I W}$ ORK 2
REAL ::ANORM,RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::W ORK
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A

## C INTERFACE

\#include < sunperfh>
void ssycon (char uple, int n, float *a, int lda, int *ipijot, float anorm, float *roond, int *info);
void ssycon_64 (char uplo, long n, float*a, long lda, long *ipivot, floatanorm, float *rcond, long *info);

## PURPOSE

ssycon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a realsym $m$ etric $m$ atrix A using the factorization $A=U * D * U * * T$ orA $=L * D * L * * T$ com puted by SSY TRF.

A $n$ estim ate is obtained fornorm (inv (A ) ), and the reciprocal of the condition num ber is com puted as RCOND $=1 /(A N O R M *$ norm (inv (A))).

## ARGUMENTS

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
$=\mathrm{U}$ : : Upper triangular, form is $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$;
$=L^{\prime}:$ Low er triangular, form is $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input) The block diagonalm atrix $D$ and the m ultipliers
used to obtain the factorU orL as com puted by SSY TRF.

LD A (input)
The leading dim ension of the array A. LD A >=
$\max (1, \mathbb{N})$.
PIVOT (input)
D etails of the interchanges and the block structure ofD as determ ined by SSY TRF.

ANORM (input)
The 1-norm of the originalm atrix A.
RCOND (output)
The reciprocal of the condition number of the
$m$ atrix $A$, com puted as RCOND $=1 /(A N O R M * A \mathbb{N} V N M)$, where $A \mathbb{N} V N M$ is an estim ate of the 1 -nom of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension $(2 * N)$
IN ORK 2 (w orkspace)
$\mathbb{N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{I N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssyev - com pute alleigenvalues and, optionally, eigenvectors of a realsym m etric m atrix A

## SYNOPSIS

```
SUBROUT\mathbb{NE SSYEV (JOBZ,UPLO,N,A,LDA,W ,W ORK,LDW ORK,INFO)}
CHARACTER * 1 JOBZ,UPLO
INTEGERN,LDA,LDWORK,\mathbb{NFO}
REALA (LDA,*),W (*),WORK (*)
SU BROUT\mathbb{NE SSYEV_64(JOBZ,UPLO,N,A,LDA,W ,W ORK,LDW ORK, INFO)}
CHARACTER * 1 JOBZ,UPLO
INTEGER*8N,LDA,LDW ORK, INFO
REALA (LDA,*),W (*),WORK (*)
```


## F95 INTERFACE

SUBROUTINE SYEV (JOBZ, UPLO,N,A, [LDA ],W, [W ORK], [LDW ORK], [NFO ])
CHARACTER (LEN=1): : JOBZ, UPLO
$\mathbb{N} T E G E R:: N, L D A, L D W$ ORK, $\mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A
SU BROUTINE SYEV_64 (OOBZ, UPLO,N,A, [LDA ],W, [W ORK], [LDW ORK ], [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1): : JOBZ, UPLO
$\mathbb{N}$ TEGER (8) :: N,LDA,LDW ORK, $\mathbb{N} F O$
REAL,D $\mathbb{I M}$ ENSION (:) ::W ,W ORK
REAL,D $\mathbb{M}$ ENSION (: : : : : A

## C INTERFACE

\#include <sunperfh>
void ssyev (char jobz, charuple, intn, float *a, int lda, float * ${ }_{\mathrm{w}}$, int *info);
void ssyev_64 (char jobz, char uplo, long n, float *a, long lda, float *w , long *info);

## PURPOSE

ssyev com putes alleigenvalues and, optionally, eigenvectors of a realsym m etric m atrix A.

## ARGUMENTS

JO BZ (input)
$=\mathrm{N}^{\prime}$ : C om pute eigenvalues only;
$=\mathrm{V}$ : C om pute eigenvalues and eigenvectors.

UPLO (input)
$=\mathrm{U}:$ : U pper triangle of A is stored;
= L': Low er triangle ofA is stored.
N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.
A (input/output)
On entry, the symm etric $m$ atrix $A$. If UPLO $=U$ ', the leading $\mathrm{N}-$ by -N upper triangularpartofA contains the upper triangular part of the $m$ atrix $A$. If UPLO = L', the leading N -by -N low er triangular partofA contains the low er triangular part of the $m$ atrix $A$. On exit, if $J O B Z=V$ ', then if $\mathbb{N} F O=0, A$ contains the orthonorm al eigenvectors of them atrix $A$. If $J O B Z=N$ ', then on exit the low er triangle (if $\mathrm{PLO}=\mathrm{L}$ ) or the upper triangle (if $\mathrm{UPLO}=\mathrm{U}$ ) of A , including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, \mathbb{N})$.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length of the amray W ORK. LDW ORK >= $\mathrm{max}(1,3 \star \mathrm{~N}-1)$. For optim alefficiency, LDW ORK $>=$ $(\mathbb{N B}+2)^{\star} \mathrm{N}$, where NB is the blocksize for SSY TRD retumed by ILAENV.

IfLD W ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
$\mathbb{I N F O}$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i-$ th argum enthad an illegalvalue
$>0:$ if $\mathbb{N F O}=$ i, the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

ssyevd - com pute alleigenvalues and, optionally, eigenvectors of a realsym m etricm atrix A

## SYNOPSIS

```
SUBROUT\mathbb{NE SSYEVD (JOBZ,UPLO,N,A,LDA,W ,W ORK,LW ORK,IN ORK,}
    LIN ORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGERN,LDA,LW ORK,LIN ORK,INFO}
INTEGER IN ORK (*)
REALA (LDA,*),W (*),W ORK (*)
SUBROUT\mathbb{NE SSYEVD_64(JOBZ,UPLO,N,A,LDA,W,W ORK,LW ORK, IN ORK,}
    LIN ORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGER*8N,LDA,LW ORK,LIN ORK,INFO}
INTEGER*8 IN ORK (*)
REALA (LDA,*),W (*),WORK (*)
```


## F95 INTERFACE

SU BROUTINE SYEVD (JOBZ,UPLO ,N,A, [LDA],W , $\mathbb{W}$ ORK ], [LW ORK ], [ $\mathbb{W}$ ORK ], $\left[\begin{array}{ll}\mathbb{N} & O R K],[\mathbb{N} F O])\end{array}\right.$

CHARACTER (LEN=1) :: JOBZ,UPLO
$\mathbb{N} T E G E R:: N, L D A, L W$ ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{I N}$ ORK
REAL,D IM ENSION (:) ::W ,W ORK
REAL,D $\mathbb{M}$ ENSION (: : : : : A

SU BROUTINE SYEVD_64 (OBZ,UPLO,N,A, [LDA],W, [W ORK], [LW ORK], [IW ORK ], [LIV ORK ], [ $\mathbb{N F O}])$

CHARACTER (LEN=1)::JOBZ,UPLO
$\mathbb{N}$ TEGER ( 8 ) :: N, LDA, LW ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{M}$ ENSION (:) :: $\mathbb{I N}$ ORK
REAL,D $\mathbb{M}$ ENSION (:) ::W,W ORK
REAL,D IM ENSION (: : : : : A

## C INTERFACE

\#include <sunperfh>
void ssyevd (char jobz, charuplo, intn, float *a, int lda, float * ${ }_{W}$, int *info);
void ssyevd_64 (char jobz, charuplo, long n, float *a, long lda, float *w , long *info);

## PURPOSE

ssyevd com putes alleigenvalues and, optionally, eigenvectors of a real symm etric matrix A. Ifeigenvectors are desired, it uses a divide and conquer algorithm .

The divide and conquer algorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on $m$ achines w ith a guard digit in add/subtract, or on those binary $m$ achines $w$ thout guard digits w hich subtract like the $C$ ray X M P , C ray Y M P , C ray C-90, orC ray-2. Itcould conceivably fail on hexadecim al or decim al $m$ achines $w$ ithout guard digits, butw e know of none.

Because of large use ofB LA S of level3, SSY EV D needs N**2 m ore w orkspace than SSY EVX .

## ARGUMENTS

JOBZ (input)
= N ': C om pute eigenvalues only;
$=\mathrm{V}$ : C om pute eigenvalues and eigenvectors.

UPLO (input)
= U : : U pper triangle ofA is stored;
= L': Low er triangle ofA is stored.
N (input) The order of the matrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)

On entry, the symm etric $m$ atrix $A$. If $\mathrm{PLO}=\mathrm{U}$ ', the leading N -by -N uppertriangularpart of A contains the upper triangular part of the $m$ atrix $A$. If U PLO = L', the leading $N$ by -N low er triangular partofA contains the low er triangular part of the $m$ atrix $A$. On exit, if $J O B Z=V$ ', then if $\mathbb{N} F O=0, A$ contains the orthonorm al eigenvectors of them atrix $A$. If $J O B Z=N$ ', then on exit the low er triangle (if $\mathrm{PLLO}=\mathrm{L}$ ) or the upper triangle (if $\mathrm{UPLO}=\mathrm{U}$ ) of A , including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, N)$.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.

## W ORK (w orkspace)

dim ension (LW ORK)On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim alLW ORK.

## LW ORK (input)

The dim ension of the array W ORK. If $\mathrm{N}<=1$, LW ORK must be at least1. If $\mathrm{JOBZ}=\mathrm{N}$ 'and $\mathrm{N}>$ 1, LW ORK mustbe at least $2{ }^{*} \mathrm{~N}+1$. If $\mathrm{JOBZ}=\mathrm{V}^{\prime}$ and $\mathrm{N}>1$, LW ORK must be atleast $1+6 \star \mathrm{~N}+$ $2 \star \mathrm{~N} * * 2$.

If LW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

IV ORK (w orkspace/output)
On exit, if $\mathbb{N} F O=0, \mathbb{I N}$ O RK (1) retums the optim al
LIV ORK.
LIV ORK (input)
The dim ension of the array $\mathbb{I N}$ ORK. If $\mathrm{N}<=1$, LIN ORK mustbe at least1. If JOBZ $=\mathrm{N}$ 'and $\mathrm{N}>$ $1, L \mathbb{I}$ ORK m ustbe at least 1. If $\mathrm{JOBZ}=\mathrm{V}^{\prime}$ and $\mathrm{N}>1, \mathrm{LIV}$ ORK must.be at least3 $+5 * \mathrm{~N}$.

If LIV ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the IW ORK array, retums this value as the first entry of the $\mathbb{I W}$ ORK array, and no errorm essage
related to LIN ORK is issued by XERB LA.
$\mathbb{N} F O$ (output)
$=0$ : successfulexit
< 0: if $\mathbb{N}$ FO = -i, the i-th argum enthad an illegalvalue
$>0$ : if $\mathbb{N} F O=$ i, the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

## FURTHER DETAILS

B ased on contributions by
JeffR utter, C om puter Science D ìvision, U niversity of C alifomia
atB erkeley, U SA
M odified by Francoise $T$ isseur, U niversity of Tennessee.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

ssyevr - com pute selected eigenvalues and, optionally, eigenvectors of a realsym $m$ etric tridiagonalm atrix $T$

## SYNOPSIS

$$
\begin{aligned}
& \text { SU BROUTINE SSYEVR (JOBZ,RANGE,UPLO,N,A,LDA,VL,VU, II, IU, } \\
& \text { ABSTOL,M,W,Z,LD Z, ISUPPZ,W ORK,LW ORK, } \mathbb{W} O R K, L \mathbb{I W} O R K, \mathbb{N} F O \text { ) }
\end{aligned}
$$

CHARACTER * 1 JOBZ,RANGE, UPLO
$\mathbb{N} T E G E R N, L D A, \mathbb{I}, \mathbb{U}, M, L D Z, L W O R K, L \mathbb{I N} O R K, \mathbb{N} F O$
$\mathbb{N}$ TEGER $\operatorname{ISUPPZ}$ ( $\left.^{*}\right)$, $\mathbb{I W}$ ORK (*)
REALVL,VU,ABSTOL
REALA (LDA, $), \mathrm{W}$ (*), Z (LD Z, *), W ORK (*)
SU BROUTINE SSYEVR_64 (JOBZ,RANGE,UPLO,N,A,LDA,VL,VU, II, $\mathbb{I}$, ABSTOL,M,W,Z,LDZ,ISUPPZ,WORK,LWORK, IN ORK,LIW ORK, $\mathbb{N} F O$ )

CHARACTER * 1 JOBZ,RANGE,UPLO
$\mathbb{N} T E G E R * 8 N, L D A, \mathbb{L}, \mathbb{U}, M, L D Z, L W O R K, L \mathbb{I N} O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \operatorname{ISUPPZ}(*), \mathbb{I N}$ ORK ( ${ }^{*}$ )
REALVL,VU,ABSTOL


## F95 INTERFACE

SUBROUTINE SYEVR (JOBZ,RANGE,UPLO, $\mathbb{N}], A,[L D A], V L, V U, \mathbb{I}, \mathbb{I}$, ABSTOL,M,W,Z,[LDZ], ISUPPZ, [W ORK], [LW ORK], [IN ORK ], [LINORK], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1):: JOBZ,RANGE,UPLO
$\mathbb{N}$ TEGER :: $N, L D A, \mathbb{L}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W$ ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N}$ TEGER,D $\mathbb{M}$ ENSION (:) :: ISU PPZ, $\mathbb{I N}$ ORK

REAL ::VL,VU,ABSTOL
REAL,D IM ENSION (:) ::W ,W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) ::A, Z
SU BROUTINE SYEVR_64 (JOBZ,RANGE, UPLO, $\mathbb{N}], A,[L D A], V L, V U, \mathbb{I}, \mathbb{U}$, ABSTOL,M,W,Z,[LD Z], ISUPPZ, [W ORK ], [LW ORK], [IW ORK], [LIW ORK], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
$\mathbb{N} T E G E R(8):: N, L D A, \mathbb{L}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W$ ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{M}$ ENSION (:) :: $\mathbb{I S U P P Z , I \mathbb { I }}$ ORK
REAL ::VL,VU,ABSTOL
REAL,D IM ENSION (:) ::W ,W ORK
REAL,D IM ENSION (:,:)::A, Z

## C INTERFACE

\#include < sunperfh>
void ssyevr(char jobz, char range, charuplo, int $n$, float
*a, int lda, float vl, float vu, int il, int in, float abstol, int *m, float * ${ }_{\mathrm{w}}$, float * z , int ldz, int*isuppz, int*info);
void ssyevr_64 (char jobz, char range, char uplo, long n, float *a, long lda, float vl, float vu, long il, long iu, float abstol, long ${ }^{m}$, float ${ }^{\text {w }}$, float *z, long ldz, long *isuppz, long *info);

## PURPOSE

ssyevr com putes selected eigenvalues and, optionally, eigenvectors of a realsym $m$ etric tridiagonalm atrix $T$. Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

W heneverpossible, SSY EVR calls SSTEGR to com pute the eigenspectrum using Relatively Robust Representations. SSTEGR com putes eigenvalues by the dqds algorithm, while orthogonaleigenvectors are com puted from various "good" L D $L^{\wedge} \mathrm{T}$ representations (also known as Relatively Robust Representations). G ram -Schm idtorthogonalization is avoided as far as possible. M ore specifically, the various steps of the algorithm are as follow s. For the $i$-th unreduced block of T ,
(a) Com pute $T$-sigm a_i= L_iD_iL_i^T, such that L_i D_iL_i^T
is a relatively robustrepresentation,
(b) C om pute the eigenvalues, lam bda_j, of L_i D_i L_i^T to high
relative accuracy by the dqds algorithm,
(c) If there is a cluster of close eigenvalues, "choose"
sigm a_i
close to the cluster, and go to step (a),
(d) G iven the approxim ate eigenvalue lam bda_jofL_i D_i L_i^T, com pute the corresponding eigenvectorby form ing a rank-revealing tw isted factorization.
The desired accuracy of the output can be specified by the inputparam eterABSTOL.

Form ore details, see "A new O ( $n^{\wedge} 2$ ) algorithm for the sym $m$ etric tridiagonal eigenvahue/eigenvector problem ", by Inder屰D hillon, $C$ om puter Science D ivision TechnicalR eport N o.U CB /C SD -97-971, U C Berkeley, M ay 1997.
N ote 1 :SSYEVR calls SSTEGR when the full spectum is requested on $m$ achines $w$ hich conform to the ieee-754 floating pointstandard. SSYEVR calls SSTEBZ and SSTE $\mathbb{N}$ on non-ieee $m$ achines and
when partialspectrum requests are m ade.

N orm alexecution of SSTEGR m ay create NaNs and infinities and hence $m$ ay abort due to a floating pointexception in environm ents which do nothandle N aN s and infinities in the ieee standard defaultm anner.

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}$ : C om pute eigenvalues only;
$=\mathrm{V}$ ': C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvalues w illbe found.
$=\mathrm{V}$ ::alleigenvalues in the half-open interval
( $\mathrm{L}, \mathrm{V} \mathrm{U}] \mathrm{w}$ ill be found. = I ': the II th through
IU th eigenvahes w illbe found.

UPLO (input)
$=\mathrm{U}$ ': Upper triangle ofA is stored;
$=\mathbb{L}$ ': Low er triangle of A is stored.

N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
O n entry, the symm etric $m$ atrix A. If UPLO $=U '$, the leading N -by N uppertriangularpartof u contains the upper triangular part of the $m$ atrix $A$.

If UPLO $=\mathrm{L}$ ', the leading N -by N low er triangular partofA contains the low er triangular part of the matrix A. On exit, the low ertriangle (if $\mathrm{UPLO}=\mathrm{L}$ ) or the uppertriangle (if $\mathrm{UPLO}=\mathrm{U}$ ) of A , including the diagonal, is destroyed.

LDA (input)
The leading dim ension of the amay A. LDA >= $\max (1, \mathbb{N})$.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE=A'or I'.

VU (input)
See the description of V L .

II (input)
If RA NGE= I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{L}<=\mathbb{U}<=N$, if $N>0$; $\mathbb{L}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N ot referenced ifRANGE= A 'or V'.

IU (input)
See the description of II.

ABSTOL (input)
The absolute error tolerance for the eigenvalues. A $n$ approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b] of w idth less than orequal to

ABSTOL + EPS * max ( $\mid$ |, $\mid$ |),
where EPS is them achine precision. IfABSTOL is less than or equalto zero, then EPS* $\mid$ |w illbe used in its place, where $T$ is the 1 -nom of the tridiagonal $m$ atrix obtained by reducing $A$ to tridiagonal form .

See "C om puting Sm allSingularV alues of $B$ idiagonal M atrices with G uaranteed H igh Relative A ccuracy," by D em meland K ahan, LA PA CK W orking N ote \#3.

If high relative accuracy is im portant, setA B STO L to SLAM CH (Safe minim um' ). D oing so will guarantee thateigenvalues are com puted to high relative accuracy when possible in future
releases. The cument code does not $m$ ake any guarantees abouthigh relative accuracy, but funutre releasesw ill. See J.B arlow and J. Demmel, "C om puting A ccurate E igensystem s of Scaled D iagonally D om inantM atrices", LA PA CK W orking N ote \#7, for a discussion of which $m$ atrices define their eigenvalues to high relative accuracy .

M (output)
The total num berofeigenvalues found. $0<=\mathrm{M}$ <= N . IfRANGE $=A \prime, \mathrm{M}=\mathrm{N}$, and ifRANGE $=\mathrm{I}^{\prime}, \mathrm{M}=$ $\mathbb{U}-\mathbb{L}+1$.

W (output)
The firstM elem ents contain the selected eigenvalues in ascending order.
$Z$ (input) If $\mathcal{O B Z}=\mathrm{V}$ ', then if $\mathbb{N F O}=0$, the first M colum ns of $Z$ contain the orthonorm aleigenvectors of them atrix A comesponding to the selected eigenvalues, $w$ ith the $i$-th colum $n$ of $Z$ holding the eigenvector associated w ith W (i). If $\mathrm{JOBZ}=\mathrm{N}$ ', then $Z$ is not referenced. N ote: the userm ust ensure that at leastm ax ( $1, \mathrm{M}$ ) colum ns are supplied in the amay $Z$; ifRANGE = V', the exact value of M is not know n in advance and an upperbound m ust be used.

LD Z (input)
The leading din ension of the array $\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1$, and if $\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{max}(1, N)$.

ISU PPZ (output)
The support of the eigenvectors in $Z$, ie., the indices indicating the nonzero elem ents in Z . The i-th eigenvector is nonzero only in elem ents $\operatorname{ISUPPZ}(2 \star i-1)$ through $\operatorname{ISU} \operatorname{PPZ}(2 * i)$.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array W ORK. LW ORK >= $\max (1,26 * N)$. For optim al efficiency, LW ORK >= (NB+6)*N, where NB is the m ax of the blocksize for SSY TRD and SORM TR retumed by $\mathbb{L} A E N V$.

If LW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim al size of
the W ORK amray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA .
IV ORK (w orkspace/output)
On exit, if $\mathbb{N} F O=0, \mathbb{I N}$ O RK (1) retums the optim al LW ORK.

LIV ORK (input)
The dim ension of the array $\mathbb{I N}$ ORK. LIV ORK >= $\max (1,10 \star N)$.

If LIV ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the $\mathbb{I W}$ ORK array, retums this value as the first entry of the IV ORK array, and no errorm essage related to LIN ORK is issued by XERBLA.
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N N}$ FO $=-$ i, the $i$-th argum enthad an illegalvalue
> 0: Intemalerror

## FURTHER DETAILS

B ased on contributions by
Inder\#̈̈D hillon, $\mathbb{B M}$ A $\operatorname{lm}$ aden, U SA
O sniM arques, LBNL LNERSC ,U SA
K en Stanley, C om puterScience D ivision, U niversity of C alifomia atB erkeley, U SA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssyevx - com pute selected eigenvalues and, optionally, eigenvectors of a real sym $m$ etric $m$ atrix $A$

## SYNOPSIS

```
SUBROUT\mathbb{NE SSYEVX (ODBZ,RANGE,UPLO,N,A,LDA,VL,VU,IL,\mathbb{U,}}\mathbf{N},
    ABTOL,NFOUND,W ,Z,LDZ,W ORK,LDW ORK,IW ORK2,FFA IL, \mathbb{NFO)}
CHARACTER * 1 JOBZ,RANGE,UPLO
\mathbb{N}TEGERN,LDA,\mathbb{I},\mathbb{U},NFOUND,LDZ,LDW ORK,\mathbb{NFO}
\mathbb{NTEGER IN ORK2(*),\mathbb{FA [H (*)}}\mathbf{(})
REALVL,VU,ABTOL
REALA (LDA,*),W (*),Z (LD Z,*),W ORK (*)
```



CHARACTER * 1 JOBZ,RANGE,UPLO
$\mathbb{N} T E G E R * 8 N, L D A, \mathbb{L}, \mathbb{U}, N F O U N D, L D Z, L D W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I N}$ ORK 2 (*), $\mathbb{F A} \mathbb{L}\left({ }^{*}\right)$
REALVL,VU,ABTOL
REALA (LDA, *), W (*), Z (LD Z,*), W ORK (*)

## F95 INTERFACE

SU BROUTINE SYEVX (JOBZ,RANGE,UPLO,N,A, [LDA],VL,VU, $\mathbb{I}, \mathbb{U}$, ABTOL,NFOUND,W,Z, [LDZ], [W ORK], [LDW ORK], [W ORK2], $\mathbb{F} A \mathbb{I}$, [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
$\mathbb{N} T E G E R:: N, L D A, \mathbb{L}, \mathbb{U}, N F O U N D, L D Z, L D W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{I N} O R K 2, \mathbb{F A} \mathbb{I}$
REAL ::VL,VU,ABTOL

REAL,D $\mathbb{I M} E N S I O N(:):: W, W O R K$
REAL,D IM ENSION (:,:) ::A , Z

SU BROUTINE SYEVX_64 (OBB , RANGE, UPLO, N, A, [LDA ], VL, VU, $\mathbb{I}, \mathbb{I}$,
 [ $\mathbb{N} \mathrm{FO}]$ )

CHARACTER (LEN=1) :: JOBZ,RANGE, UPLO
$\mathbb{N}$ TEGER (8) :: N , LDA $, \mathbb{I}, \mathbb{Z}, N F O U N D, L D Z, L D W O R K, \mathbb{N F O}$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} E N S I O N(:):: \mathbb{I N} O R K 2, \mathbb{F} A \mathbb{I}$
REAL ::VL,VU,ABTOL
REAL,D $\mathbb{I M}$ ENSION (:) ::W ,W ORK
REAL, D $\mathbb{M}$ ENSION (:,:) ::A , Z

## C INTERFACE

\#include <sunperfh>
void ssyevx (char jंbz, char range, charuplo, int n, float
*a, int lla, floatvl, floatvu, intil, intin,
floatabtol, int *nfound, float *W, float* $z$, int ldz, int *ifail, int *info);
void ssyevx_64 (char j̀jbz, char range, char uplo, long n, float *a, long lda, floatvl, floatvu, long il, long in, float abtol, long *nfound, float *W , float *z, long ldz, long *ifail, long *info);

## PURPOSE

ssyevx com putes selected eigenvalues and, optionally, eigenvectors of a real symm etric $m$ atrix A. Eigenvalues and eigenvectors can be selected by specifying either a range of values or range of indices for the desired eigenvalues.

## ARGUMENTS

JOBZ (input)
$=\mathrm{N}$ : C om pute eigenvalues only;
$=\mathrm{V}$ ': C om pute eigenvalues and eigenvectors.

RANGE (input)
= 'A ': alleigenvalues w illbe found.
$=\mathrm{V}$ : alleigenvalues in the half-open interval
(VL, VU] w ill be found. = I': the II th through
$\mathbb{I U}$-th eigenvaluesw illbe found.

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
On entry, the symm etric $m$ atrix $A$. If $\operatorname{PLO}=U '$, the leading N -by N uppertriangularpartof A contains the upper triangular part of the $m$ atrix $A$. If UPLO = L', the leading N -by N low er triangular partofA contains the low er triangular part of the $m$ atrix $A$. On exit, the low ertriangle (if $\mathrm{UPLO}=\mathrm{L}$ ) or the uppertriangle (if $\mathrm{UPLO}=\mathrm{U}$ ) of A , including the diagonal, is destroyed.
LDA (input)
The leading dim ension of the array A. LD A >= $\max (1, \mathbb{N})$.

VL (input)
IfRANGE=V', the low er and upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE = A 'or I'.

VU (input)
See the description of V L .

II (input)
IfRA NGE= I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{L}=1$ and $\mathbb{U}=0$ if $\mathrm{N}=0$. N ot referenced ifRANGE $=$ A 'or V'.
$\mathbb{I U}$ (input)
See the description of II.

ABTOL (input)
The absolute error tolerance for the eigenvalues. A $n$ approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b]
of w idth less than orequal to
ABTOL + EPS * max ( $\mathfrak{k}|, \mathrm{p}|)$,
where EPS is the m achine precision. If ABTOL is less than orequal to zero, then EPS* $\mid$ | w illbe used in its place, where $T$ | is the 1 -norm of the tridiagonal $m$ atrix obtained by reducing $A$ to tridiagonal form .

E igenvalues w illbe com putedm ostaccurately when

ABTOL is set to tw ice the underflow threshold $2 *$ SLAM CH (S ), notzero. If this routine retums w ith $\mathbb{N}$ FO $>0$, indicating that som e eigenvectors did not converge, try setting ABTOL to $2 *$ SLAM CH (S ).

See "C om puting Sm allSingularV ahues of B idiagonal M atrices w ith G uaranteed H igh Relative A ccuracy," by D em m eland K ahan, LA PA CK W orking N ote \#3. NFOUND (output)

The total num ber of eigenvalues found. $0<=$ NFOUND <= N. IfRANGE = A',NFOUND = N, and if RANGE $=I^{\prime}$, NFOUND $=\mathbb{U}-\mathbb{L}+1$.

W (output)
On norm alexit, the firstN FOUND elem ents contain the selected eigenvalues in ascending order.
$Z$ (input) If $J O B Z=V^{\prime}$, then if $\mathbb{N F O}=0$, the first $N F O U N D$ colum ns of $Z$ contain the orthonorm aleigenvectors of them atrix A comesponding to the selected eigenvalues, $w$ ith the i-th colum $n$ of $Z$ holding the eigenvector associated with $W$ (i). If an eigenvector fails to converge, then that colum n of $Z$ contains the latest approxim ation to the eigenvector, and the index of the eigenvector is retumed in FAII. If $J 0 B Z=N$ ', then $Z$ is not referenced. $N$ ote: the user must ensure that at least $m$ ax ( 1, NFO UND ) colum ns are supplied in the array $Z$; if RANGE = V', the exactvalue ofNFOUND is not know $n$ in advance and an upperbound $m$ ust.be used.

LD Z (input)
The leading dim ension of the array $Z . L D Z>=1$, and if $\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{max}(1, N)$.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length of the array $W$ ORK. LDW ORK >= $\max (1,8 * N)$. For optim al efficiency, LDW ORK >= ( $\mathrm{N} B+3)^{*} \mathrm{~N}$, where $N B$ is the $m$ ax of the blocksize for SSY TRD and SORM TR retumed by IUAENV.

If LD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

IV ORK 2 (w orkspace)
ㅍFAII (output)
If $\mathrm{OBZ}=\mathrm{V}^{\prime}$, then if $\mathbb{N F O}=0$, the first NFOUND
elem ents of $\mathbb{F A} I I$ are zero. If $\mathbb{N} F O>0$, then
IFA II contains the indices of the eigenvectors
that failed to converge. If $\mathrm{OBZ}=\mathrm{N}$ ', then
$\mathbb{F} A \mathbb{I}$ is notreferenced.

INFO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$ th argum enthad an illegalvalue
$>0:$ if $\mathbb{N} F O=$ i, then ieigenvectors failed to converge. Their indices are stored in anay 팦.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssygs2 -reduce a realsym m etric-definite generalized eigenproblem to standard form

## SYNOPSIS

```
SU BROUT\mathbb{NE SSYGS2(TTYPE,UPLO,N,A ,LDA ,B,LDB, NNFO )}
```

CHARACTER * 1 UPLO
$\mathbb{N} T E G E R \mathbb{T T Y P E , N , L D A , L D B , ~} \mathbb{N} F O$
REALA (LDA,*), B (LDB,*)
SU BROUTINE SSYGS2_64 (TTYPE, UPLO ,N,A,LDA,B,LDB, INFO)
CHARACTER * 1 UPLO
$\mathbb{N} T E G E R * 8 \mathbb{T} Y P E, N, L D A, L D B, \mathbb{N} F O$
REALA (LDA,*), B (LDB,*)

## F95 INTERFACE

SU BROUTINE SYGS2 (TTYPE, UPLO ,N,A, [LDA],B, [LDB], [NFO])
CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D A, L D B, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:,:): ::A,B
SU BROUTINE SYGS2_64 (TTYPE, UPLO ,N,A, [LDA ], B , [LD B], [NFO ])
CHARACTER (LEN=1)::UPLO
$\mathbb{N}$ TEGER (8) :: $\mathbb{T} Y \mathrm{PE}, \mathrm{N}, \mathrm{LDA}, L D B, \mathbb{N} F O$
REAL,D IM ENSION (: :: : ::A, B

## C INTERFACE

\#include < sunperfh>
void ssygs2 (int itype, char uple, intn, float *a, int lda, float*b, int ldb, int *info);
void ssygs2_64 (long itype, charuplo, long n, float *a, long lda, float *b, long ldb, long *info);

## PURPOSE

ssygs2 reduces a realsym m etric-definite generalized eigenproblem to standard form .

If ITY PE $=1$, the problem is $A * x=$ lam bda*B ${ }^{*} X_{\text {, }}$ and $A$ is overw rilten by inv (U)*A *inv (U) orinv ( (L) *A *inv (L) If ITYPE $=2$ or 3 , the problem is $\mathrm{A} * \mathrm{~B} * \mathrm{x}=$ lam bda* x or
 B m usthave been previously factorized as $U$ " $U$ or $L^{*}$ L' by SPOTRF.

## ARGUMENTS

ITYPE (input)
$=1:$ com pute inv (U)*A *inv (U) orinv (L)*A *inv (L);
$=2$ or 3 : com pute $U$ *A *U 'orL *A *L .

UPLO (input)
Specifies w hether the upper or low er triangular
part of the sym $m$ etric $m$ atrix $A$ is stored, and how
B has been factorized. = U ': Upper triangular
= LL : Low er triangular

N (input) The order of the m atriges A and $\mathrm{B} . \mathrm{N}>=0$.

A (input/output)
O n entry, the sym m etric m atrix A. If UPLO $=U '$, the leading $n$ by $n$ upper triangularpart of A contains the upper triangularpart of the $m$ atrix $A$, and the strictly low er triangularpart of $A$ is not referenced. If $\mathrm{UPLO}=\mathrm{L}$ ', the leading n by n low er triangularpart of A contains the low er triangularpart of the m atrix A, and the strictly upper triangularpart of $A$ is notreferenced.

On exit, if $\mathbb{N F O}=0$, the transform ed $m$ atrix, stored in the sam e form atas A.

## LD A (input)

The leading dim ension of the array A. LDA >= $\max (1, N)$.
$B$ (input) The triangular factor from the C holesky factorization ofB, as retumed by SPO TRF.

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.
$\mathbb{I N} F O$ (output)
= 0: successfulexit.
< 0: if $\mathbb{N}$ FO = -i, the i-th argum ent had an illegalvahue.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssygst-reduce a realsym m etric-definite generalized eigenproblem to standard form

## SYNOPSIS

SU BROUTINE SSYGST (TTYPE, UPLO,N,A,LDA,B,LDB, $\mathbb{N} F O$ )
CHARACTER * 1 UPLO
$\mathbb{N} T E G E R \mathbb{I T Y P E , N , L D A , L D B , ~} \mathbb{N} F O$
REALA (LDA, $), B(L D B, *)$
SU BROUTINE SSYGST_64 (TTYPE,UPLO, N,A,LDA,B,LDB, $\mathbb{N} F O$ )

CHARACTER * 1 UPLO
$\mathbb{N} T E G E R * 8 \mathbb{T} Y P E, N, L D A, L D B, \mathbb{N} F O$
REALA (LDA, $), B(L D B, *)$

## F95 INTERFACE

SU BROUTINE SYGST (TTYPE, UPLO, N, A , [LDA], B, [LDB], [NFO])
CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D A, L D B, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:,:): ::A,B
SUBROUTINESYGST_64 (TTYPE, UPLO ,N,A, [LDA],B, [LDB], [ $\mathbb{N F F O}]$ )

CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER (8) :: $\mathbb{T} Y \mathrm{PE}, \mathrm{N}, \mathrm{LDA}, L D B, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (: : : : :: A, B

## C INTERFACE

\#include <sunperfh>
void ssygst(int itype, charuplo, intn, float *a, int lda, float *b, int ldb, int *info);
void ssygst 64 (long itype, charuplo, long n, float *a, long lda, float *b, long ldb, long *info);

## PURPOSE

ssygst reduces a realsym m etric-definite generalized eigenproblem to standard form .

If ITY PE $=1$, the problem is $A * x=l a m$ bda*B ${ }^{*} x_{\text {, }}$ and $A$ is overw ritten by inv $(U * * T) * A * \operatorname{inv}(U)$ or inv ( $(\mathrm{A})$ *A *inv ( $(\mathrm{L} * * \mathrm{~T})$
If ITYPE $=2$ or 3 , the problem is $A * B * x=$ lam bda* x or $B * A * x=\operatorname{lam}$ bda* $x$, and $A$ is overw rilten by $U * A * U * * T$ or L**T*A*L。

B m usthave been previously factorized as U ** T * U or $\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$ by SPO TRF .

## ARGUMENTS

ITYPE (input)
$=1:$ compute $\quad \operatorname{inv}(\mathrm{U} * * T) \star A * \operatorname{inv}(\mathrm{U})$ or
$\operatorname{inv}(\amalg) \star A$ *inv $(\amalg * * T)$;
$=2$ or 3 : com pute $\mathrm{U} * \mathrm{~A} * \mathrm{U} * * \mathrm{~T}$ orL ${ }^{* *} \mathrm{~T} * \mathrm{~A} * \mathrm{~L}$.

UPLO (input)
$=\mathrm{U}^{\prime}$ : Uppertriangle of $A$ is stored and $B$ is factored as $\mathrm{U} * * \mathrm{~T} * \mathrm{U} ;=\mathrm{L}:$ : Low er triangle of A stored and B is factored as $\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$.

N (input) The order of the $m$ atriges $A$ and $B . N>=0$.

A (input/output)
O n entry, the sym m etric m atrix A. If $\mathrm{ULO}=\mathrm{U}$ ', the leading N -by N uppertriangularpart of $A$ contains the upper triangularpart of the $m$ atrix $A$, and the strictly low er triangularpart of A is not referenced. If $\mathrm{UPLO}=\mathrm{L}$ ', the leading N boy- N low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of $A$ is notreferenced.

On exit, if $\mathbb{N F O}=0$, the transform ed matrix,
stored in the sam e form at as A .

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, N)$.

B (input) The triangular factor from the Cholesky factorization ofB , as retumed by SPO TRF .

LD B (input)
The leading dim ension of the aray $\mathrm{B} . \operatorname{LDB}>=$ $\max (1, N)$.
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the $i-$ th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
ssygv - com pute all the eigenvalues, and optionally, the
eigenvectors of a realgeneralized sym \(m\) etric-definite eigen-
problem, of the form \(A * x=(\operatorname{lam} . b d a) * B * x, A * B x=(\operatorname{lam} . b d a) * x\), or
B *A *x= (lam bda) *x
```


## SYNOPSIS

```
SU BROUT\mathbb{NE SSYGV (TTYPE,JOBZ,UPLO,N,A,LDA,B,LDB,W ,W ORK,}
    LDW ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
\mathbb{NTEGER ITYPE,N,LDA,LDB,LDW ORK, INFO}
REALA (LDA,*),B (LDB ,*),W (*),W ORK (*)
SU BROUT\mathbb{NE SSYGV_64 (TTYPE,JOBZ,UPLO,N,A ,LDA,B,LDB,W ,W ORK,}
    LDWORK,\mathbb{NFO)}
```

CHARACTER * 1 JOBZ, UPLO
$\mathbb{N} T E G E R * 8 \mathbb{I T Y P E}, N, L D A, L D B, L D W O R K, \mathbb{N} F O$
REALA (LDA,*), B (LDB,*),W (*),WORK (*)

## F95 INTERFACE

 [LDW ORK ], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1): : JOBZ, UPLO
$\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D A, L D B, L D W O R K, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::W ,W ORK
REAL,D $\mathbb{M}$ ENSION (: : : : :: A, B
SU BROUTINE SYGV_64 (TTYPE, JOBZ, UPLO ,N,A, [LDA ], B, [LDB],W, [W ORK], [LDW ORK], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1):: DBB Z UPLO
$\mathbb{N}$ TEGER (8) :: ITYPE,N,LDA,LDB,LDWORK, $\mathbb{N} F O$
REAL,D $\mathbb{I M} E N S I O N(:):: W$,W ORK
REAL,D $\mathbb{I M}$ ENSION (: : : : : A , B

## C INTERFACE

\#include <sunperfh>
void ssygv (int itype, char jंbz, charuplo, intn, float *a, int lda, float *b, int ldb, float *W , int *info);
void ssygv_64 (long itype, char jobz, char uplo, long n, float *a, long lda, float *b, long ldb, float *w , long *info);

## PURPOSE

ssygv com putes all the eigenvalues, and optionally, the eigenvectors of a realgeneralized sym m etric-definite eigenproblem, of the form $A{ }^{*} x=(\operatorname{lam} . b d a){ }^{*} B{ }^{*} x, A * B x=(\operatorname{lam} . b d a){ }^{*} x$, or $B *_{A} *_{x}=(\operatorname{lam} . b d a){ }^{*} x$. H ere $A$ and $B$ are assum ed to be sym $m$ etric and $B$ is also positive definite.

## ARGUMENTS

## ITYPE (input)

Specifies the problem type to be solved:
$=1: A *_{x}=(\operatorname{lam} . b d a){ }^{*} B{ }^{*}{ }_{x}$
$=2: A * B{ }^{*} \mathrm{x}=\left(\operatorname{lam}\right.$ bda) ${ }^{\star} \mathrm{x}$
$=3: B * A{ }^{*} \mathrm{X}=(\operatorname{lam} \mathrm{bda}){ }^{\star} \mathrm{x}$

JO B Z (input)
$=\mathrm{N}$ ': C om pute eigenvalues only;
$=V^{\prime}:$ C om pute eigenvalues and eigenvectors.

UPLO (input)
$=U$ ': U ppertriangles of A and B are stored;
$=\mathbb{L}$ ': Low ertriangles of $A$ and $B$ are stored.

N (input) The order of the $m$ atriges $A$ and $B . N>=0$.

A (input/output)
O n entry, the sym m etric m atrix A. If UPLO = U', the leading N boy N uppertriangularpartofA contains the upper triangularpart of the $m$ atrix $A$.

If UPLO $=\mathrm{L}$ ', the leading N -by N low er triangular part ofA contains the low er triangular part of them atrix A.

On exit, if $\mathrm{JOBZ}=\mathrm{V}$ ', then if $\mathbb{N F O}=0$, A contains the $m$ atrix $Z$ of eigenvectors. The eigenvectors are norm alized as follow s: if ITYPE =1 or 2, $Z * * T * B * Z=I$; if $T T Y P E=3, Z * * T * i n v(B) * Z=I$. If $\mathrm{JOBZ}=\mathrm{N}$ ', then on exit the uppertriangle (if $\mathrm{UPLO}=\mathrm{U}$ ) or the low er triangle (if $\mathrm{UPLO}=\mathrm{L}$ ) of A , including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the array A. LDA >= max ( $1, \mathbb{N}$ ).
B (input/output)
O $n$ entry, the sym $m$ etric positive definite $m$ atrix $B$. If $\mathrm{UPLO}=\mathrm{U}$ ', the leading N -by N uppertriangularpartofB contains the upper triangular part of the $m$ atrix $B$. If $\mathrm{PLO}=\mathrm{L}$ ', the leading N by-N lowertriangularpart of B contains the low er triangularpart of the m atrix B .

On exit, if $\mathbb{N F O}<=N$, the part of $B$ containing the $m$ atrix is overw rilten by the triangular factor U orL from the Cholesky factorization $\mathrm{B}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $\mathrm{B}=\mathrm{L} * \mathrm{~L} * * \mathrm{~T}$.

LD B (input)
The leading dim ension of the amay $B$. LD B >= $\max (1, N)$.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length of the array $W$ ORK. LDW ORK >= m ax $(1,3 * \mathrm{~N}-1)$. For optim alefficiency, LD W ORK >= $(N B+2)^{\star} N$, where $N B$ is the blocksize for SSYTRD retumed by ILAENV .

If LD W ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
$\mathbb{I N} F O$ (output)
= 0: successfiulexit
$<0:$ if $\mathbb{N F O}=-$ i, the i-th argum enthad an illegalvalue
> 0: SPO TRF orSSYEV retumed an errorcode: $<=\mathrm{N}:$ if $\mathbb{N} F \mathrm{O}=\mathrm{i}$, SSYEV failed to converge; i off-diagonalelem ents of an interm ediate tridiagonalform did notconverge to zero; > N : if $\mathbb{N F O}$ $=N+i$, for $1<=i<=N$, then the leading $m$ inor of orderiofB is not positive definite. The factorization of B could notbe com pleted and no eigenvalues or eigenvectors w ere com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

```
ssygvd -com pute all the eigenvalues, and optionally, the
eigenvectors of a realgeneralized sym m etric-definite eigen-
problem, of the form A *x= (lam bda)*B *x, A *B x= (lam bda)*x, or
B *A *X= (lam boda)* }\mp@subsup{\textrm{X}}{}{\prime
```


## SYNOPSIS

```
SU BROUT\mathbb{NE SSYGVD (TTYPE,NOBZ,UPLO,N,A,LDA,B,LDB,W,W ORK,}
    LW ORK,IN ORK,LIN ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
```



```
INTEGER 㱐 ORK (*)
REALA (LDA,*),B (LDB,*),W (*),W ORK (*)
SU BROUT\mathbb{NE SSYGVD_64 (TTYPE,JOBZ,UPLO,N,A LD A , B ,LDB,W ,W ORK,}
```


CHARACTER * 1 JOBZ, UPLO
$\mathbb{N} T E G E R * 8 \mathbb{I T} Y P E, N, L D A, L D B, L W O R K, L \mathbb{I W} O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I N}$ ORK ( $\left.{ }^{( }\right)$
REALA (LDA,*), B (LDB,*),W (*),WORK (*)

## F95 INTERFACE

SU BROUTINE SYGVD (TTYPE, JOBZ,UPLO, $\mathbb{N}], A,[L D A], B,[L D B], W,[W O R K]$, [LW ORK], [ $\mathbb{W}$ ORK], [LIN ORK ], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1):: JOBZ,UPLO
$\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D A, L D B, L W$ ORK,LIN ORK, $\mathbb{N} F O$
$\mathbb{N}$ TEGER,D $\mathbb{I M}$ ENSION (:) :: $\mathbb{I N}$ ORK
REAL,D IM ENSION (:) ::W ,W ORK

REAL,D $\mathbb{M}$ ENSION (: : : : : A , B

SU BROUTINE SYGVD_64 (TTYPE, JOBZ, UPLO, $\mathbb{N}]$, A, [LDA ], B , [LD B ], W , $[\mathbb{W} O R K],[L W O R K],[\mathbb{I}$ ORK $],[L \mathbb{I N} O R K],[\mathbb{N} F O])$

CHARACTER (LEN=1) :: JOBZ, UPLO
$\mathbb{N} T E G E R(8):: \operatorname{ITY} P E, N, L D A, L D B, L W O R K, L \mathbb{N} O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{I W} O R K$
REAL,D $\mathbb{I M} E N S I O N(:):: W$,W ORK
REAL,D $\mathbb{M}$ ENSION (: : : : : A , B

## C INTERFACE

\#include <sunperfh>
void ssygvd (int itype, char jobz, charuplo, int n, float
*a, int lda, float *b, int ldlo, float *w, int
*info);
void ssygvd_64 (long itype, char j̀jbz, char uple, long n, float *a, long lda, float*b, long ldb, float *w , long *info);

## PURPOSE

ssygvd com putes all the eigenvahues, and optionally, the eigenvectors of a realgeneralized sym m etric-definite eigenproblem, of the form $A * x=(\operatorname{lam} . b d a) * B{ }^{*} x, A * B x=(\operatorname{lam} b d a){ }^{*} x$, or $B * A * x=(\operatorname{lam} . b d a){ }^{\star} x$. H ere $A$ and $B$ are assum ed to be sym $m$ etric and $B$ is also positive definite. If eigenvectors are desired, ituses a divide and conqueralgorithm .

The divide and conquer algorithm $m$ akes very $m$ ild assum ptions about floating point arithm etic. Itw illw ork on $m$ achines w ith a guard digit in add/subtract, or on those binary $m$ achines $w$ thout guard digits $w$ hich subtract like the $C$ ray X M P , C ray Y M P , C ray C-90, orC ray-2. It could conceivably fail on hexadecim al or decim al $m$ achines $w$ ithout guard digits, butw e know of none.

## ARGUMENTS

ITYPE (input)
Specifies the problem type to be solved:
$=1: A *_{\mathrm{x}}=(\operatorname{lam} \mathrm{bda}){ }^{*} \mathrm{~B}{ }^{*} \mathrm{x}$
$=2: A * B{ }^{*} \mathrm{x}=(\operatorname{lam} \mathrm{bda}){ }^{\star} \mathrm{x}$
$=3: B *^{*} *_{X}=\left(\operatorname{lam}\right.$ bda) ${ }^{*} \mathrm{X}$

JOBZ (input)
$=N^{\prime}:$ C om pute eigenvalues only;
$=\mathrm{V}$ : C om pute eigenvalues and eigenvectors.

UPLO (input)
$=\mathrm{U}$ ': U ppertriangles of $A$ and $B$ are stored;
$=\mathrm{L}$ ': Low ertriangles of $A$ and $B$ are stored.
N (input) The order of the m atrices A and $\mathrm{B} . \mathrm{N}>=0$.
A (input/output)
On entry, the sym m etric $m$ atrix $A$. If $U P L O=U '$, the leading $\mathrm{N}-$ by -N uppertriangularpartofA contains the upper triangular part of the $m$ atrix $A$. If UPLO = L', the leading N by -N low er triangular partofA contains the low er triangular part of the $m$ atrix $A$.

On exit, if $\mathrm{JOBZ}=\mathrm{V}$ ', then if $\mathbb{N F O}=0$, A contains the $m$ atrix $Z$ ofeigenvectors. The eigenvectors are norm alized as follow s: if ITYPE = 1 or 2, $\mathrm{Z} * * \mathrm{~T} * \mathrm{~B} * \mathrm{Z}=\mathrm{I}$; if $\mathrm{IT} Y \mathrm{PE}=3, \mathrm{Z} * * \mathrm{~T} * \operatorname{inv}(\mathrm{~B}) * \mathrm{Z}=\mathrm{I}$. If $\mathrm{JOBZ}=\mathrm{N}$ ', then on exit the upper triangle (if $\mathrm{UPLO}=\mathrm{U}$ ) or the low er triangle (if $\mathrm{UPLO}=\mathrm{L}$ ) of A, including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, \mathbb{N})$.

B (input/output)
On entry, the symm etric $m$ atrix $B$. If $\operatorname{PLO}=U$ ', the leading N -by N uppertriangularpartofB contains the upper triangular part of the $m$ atrix $B$. If U PLO = L', the leading N by -N low er triangular partofB contains the low er triangular part of the $m$ atrix $B$.

On exit, if $\mathbb{N} F O<=N$, the part of $B$ containing the $m$ atrix is overw rilten by the triangular factor $U$ orL from the C holesky factorization $B=U * * T * U$ orB $=\mathrm{L}$ * $\mathrm{L} * * \mathrm{~T}$.

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, N)$.

W (output)
If $\mathbb{N} F O=0$, the eigenvalues in ascending order.

W ORK (w orkspace)
On exit, if $\mathbb{N F O}=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array $W$ ORK. If $N<=1$, LW ORK $>=1$. If $\mathrm{OBZ}=\mathrm{N}^{\prime}$ and $\mathrm{N}>1$, LW ORK $>=$ $2 \star \mathrm{~N}+1$. If $\mathrm{OBZ}=\mathrm{V}^{\prime}$ and $\mathrm{N}>1$,LW ORK $>=1+6 * \mathrm{~N}$ $+2 \star \mathrm{~N} * * 2$ 。

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IW ORK (w orkspace/output)
On exit, if $\mathbb{N} F O=0, \mathbb{I N}$ ORK (1) retums the optim al LIW ORK.

LIV ORK (input)
The dim ension of the amay $\mathbb{I W} O R K$. If $\mathrm{N}<=1$, LIW ORK >=1. If $O B Z=N$ 'andN $>1$,LIW ORK $>=$ 1. If $\mathrm{OBZ}=\mathrm{V}$ 'and $\mathrm{N}>1$, LIN $\mathrm{ORK}>=3+5 \star \mathrm{~N}$.

IfLIW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the $\mathbb{I W}$ ORK array, retums this value as the first entry of the $\mathbb{I W}$ ORK aray, and no errorm essage related to $L \mathbb{I} W$ ORK is issued by XERBLA.
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0$ : if $\mathbb{N}$ FO $=-i$, the $i$ th argum ent had an illegalvalue
> 0: SPO TRF orSSY EVD retumed an errorcode:
$<=\mathrm{N}:$ if $\mathbb{N F O}=\mathrm{i}, \mathrm{SSYEVD}$ failed to converge; i offf-diagonalelem ents of an interm ediate tridiagonal form did notconverge to zero; > N : if $\mathbb{N N F O}$ $=N+i$, for $1<=i<=N$, then the leading $m$ inor oforderiofB is not positive definite. The factorization of $B$ could notbe com pleted and no eigenvalues or eigenvectors w ere com puted.

## FURTHER DETAILS

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv. of K entucky, USA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

```
ssygvx - com pute selected eigenvalues, and optionally,
eigenvectors of a realgeneralized symm etric-definite eigen-
problem, of the form A *x= (lam bda)*B *x, A *B x= (lam bda)*x, or
B *A *X= (lam boda)* }\mp@subsup{\textrm{X}}{}{\prime
```


## SYNOPSIS

```
SU BROUTINE SSYGVX (ITY PE,NOBZ,RANGE,UPLO,N,A,LDA,B,LDB,VL,
    VU,\mathbb{I},\mathbb{U},ABSTOL,M,W,Z,LD Z,W ORK,LW ORK,IN ORK,\mathbb{FA}\mathbb{M},
    \mathbb{NFO)}
CHARACTER * 1 JOBZ,RANGE,UPLO
\mathbb{NTEGER ITYPE,N,LDA,LDB,\mathbb{L},\mathbb{U},M,LD Z,LW ORK,\mathbb{NFO}}\mathbf{M}\mathrm{ , L',}
INTEGER IN ORK (*),\mathbb{FA}|(*)
REALVL,VU,ABSTOL
REALA (LDA,*),B (LDB,*),W (*),Z (LD Z ,*),W ORK (*)
SUBROUTINE SSYGVX_64 (TTYPE,JOBZ,RANGE,UPLO,N,A,LDA,B,LDB,VL,
        VU,\mathbb{I},\mathbb{U},ABSTOL,M,W,Z,LD Z,W ORK,LW ORK,IN ORK,\mathbb{FA}\mathbb{I},
        \mathbb{NFO)}
```

CHARACTER * 1 JOBZ,RANGE, UPLO
$\mathbb{N} T E G E R * 8 \mathbb{I T} Y P E, N, L D A, L D B, \mathbb{I}, \mathbb{U}, M, L D Z, L W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{I N}$ ORK (*), $\mathbb{F A} \mathbb{L}(*)$
REALVL,VU,ABSTOL
REALA (LDA, $\left.{ }^{\star}\right), \mathrm{B}(\mathrm{LDB}, \star), \mathrm{W}(*), \mathrm{Z}(\mathrm{LD} Z, \star), \mathrm{W} O R K(\star)$

## F95 INTERFACE

SU BROUTINE SYGVX (TTYPE, JOBZ,RANGE,UPLO, $\mathbb{N}], A,[L D A], B,[L D B]$,
 $\mathbb{F A} \mathbb{L},[\mathbb{N} F O]$ )

CHARACTER (LEN=1) :: OBZ,RANGE, UPLO
$\mathbb{N} T E G E R:: \mathbb{I T}$ YE, $N, L D A, L D B, \mathbb{L}, \mathbb{U}, M, L D Z, L W O R K, \mathbb{N} F O$
$\mathbb{N}$ TEGER,D $\mathbb{M}$ ENSION (:) :: $\mathbb{I W}$ ORK, $\mathbb{F} A \mathbb{I}$
REAL ::VL,VU,ABSTOL
REAL,D IM ENSION (:) ::W ,W ORK
REAL,D $\mathbb{M}$ ENSION (: : : : : : A, B, Z

SU BROU T INE SY GVX_64 (TTYPE, JOBZ,RANGE,UPLO, $\mathbb{N}], A,[L D A], B,[L D B]$, $\mathrm{VL}, \mathrm{VU}, \mathbb{I}, \mathbb{I}, \mathrm{ABSTOL}, \mathrm{M}, \mathrm{W}, \mathrm{Z},[\mathrm{LD} \mathrm{Z}],[\mathrm{W} O R K],[\mathrm{LW} O R K],[\mathbb{I W} O R K]$, $\mathbb{F A} \mathbb{I},[\mathbb{N} F O])$

CHARACTER (LEN=1) :: OBZ,RANGE,UPLO
$\mathbb{N} T E G E R(8):: \mathbb{I T} Y P E, N, L D A, L D B, \mathbb{I}, \mathbb{U}, M, L D Z, L W O R K$,
$\mathbb{N}$ FO
$\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{I N O R K}$, $\mathbb{F} A \mathbb{I}$
REAL ::VL,VU,ABSTOL
REAL,D $\mathbb{M} E N S I O N(:):: W, W O R K$
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A, B , Z

## C INTERFACE

\#include <sunperfh>
void ssygvx (int itype, char jobz, char range, charuple, int n , float *a, int lda, float *b, int ldo, float vl, floatvu, int il, intiu, float abstol, int *m, float *W, float *z, int ldz, int*ifail, int *info);
void ssygvx_64 (long itype, char jobz, char range, charuplo, long $n$, float *a, long lda, float *b, long ldb, floatvl, float vu, long il, long iu, float abstol, long *m, float *W, float *z, long ldz, long *ifail, long *info);

## PURPOSE

ssygvx com putes selected eigenvalues, and optionally, eigenvectors of a realgeneralized sym $m$ etric-definite eigenproblem, of the form $A * x=(\operatorname{lam} b d a) * B{ }^{*} x, A * B x=(\operatorname{lam} b d a) * x$, or $B * A * x=(\operatorname{lam} . b d a){ }^{\star} x$. H ere $A$ and $B$ are assum ed to be sym $m$ etric and $B$ is also posilive definite. Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvahes.

## ARGUMENTS

## ITYPE (input)

Specifies the problem type to be solved:
$=1: A{ }^{*} \mathrm{X}=(\operatorname{lam} \mathrm{bda}){ }^{\mathrm{B}}{ }^{*} \mathrm{X}$
$=2: \mathrm{A} * \mathrm{~B} * \mathrm{X}=\left(\operatorname{lam}\right.$ bda) ${ }^{*} \mathrm{x}$
$=3: B * A * x=\left(\operatorname{lam}\right.$ bda) ${ }^{*} \mathrm{x}$
JOBZ (input)
$=\mathrm{N}^{\prime}$ : C om pute eigenvalues only;
$=\mathrm{V}:$ : C om pute eigenvalues and eigenvectors.
RANGE (input)
= A ': alleigenvalues w ill.be found.
= V : alleigenvahues in the half-open interval ( $\mathrm{L}, \mathrm{V}, \mathrm{J}]$ will be found. = I ': the I -th through $\mathbb{I U}$-th eigenvaluesw illlbe found.

UPLO (input)
= U ': U ppertriangle of A and B are stored;
= L': Low ertriangle of A and B are stored.
N (input) The order of the $m$ atrix pencil $(A, B) . N>=0$.
A (input/output)
On entry, the symm etric m atrix $A$. If UPLO = U', the leading N -by -N uppertriangularpart of A contains the upper triangular part of the $m$ atrix $A$. If UPLO $=\mathrm{L}$ ', the leading N -by N low er triangular partofA contains the low er triangular part of them atrix A.

On exit, the low er triangle (ifU PLO = L) or the upper triangle (if $\mathrm{PLO}=\mathrm{U}$ ) ofA, including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, N)$.

B (input/output)
On entry, the symm etric $m$ atrix $B$. If $U P L O=U$ ', the leading N -by N uppertriangular partofB contains the upper triangularpart of the $m$ atrix $B$. If UPLO = L', the leading N -by -N low er triangular partofB contains the low er triangular part of the matrix B .

On exit, if $\mathbb{N F O}<=N$, the part of $B$ containing the $m$ atrix is overw rilten by the triangular factor U orL from the Cholesky factorization $\mathrm{B}=\mathrm{U} * * \mathrm{~T} * \mathrm{U}$ or $B=L{ }^{\star} L^{\star *}{ }^{\mathrm{T}}$.

LD B (input)
The leading dim ension of the aray B. LD B >= $\max (1, N)$.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. $\mathrm{VL}<\mathrm{VU}$. N ot referenced ifRANGE=A 'or 'I'.

VU (input)
See the description ofV L .
IU (input)
IfRA NGE= I', the indiges (in ascending order) of the smallest and largest eigenvalues to be retumed. $1<=\mathbb{Z}<=\mathbb{U}<=\mathrm{N}$, if $\mathrm{N}>0$; $\mathbb{U}=1$ and $\mathbb{U}=0$ if $N=0$. $N$ otreferenced ifRANGE $=$ A'or V'.

IU (input)
See the description of II.

ABSTOL (input)
The absolute emortolerance forthe eigenvalues.
A $n$ approxim ate eigenvalue is accepted as converged $w$ hen it is determ ined to lie in an interval [a,b] of w idth less than orequal to
$A B S T O L+E P S * \max (|a|,||$,$) ,$
where EPS is them achine precision. IfA BSTOL is less than or equal to zero, then EPS*|I w illbe used in its place, w here $T$ | is the 1 -norm of the tridiagonal $m$ atrix obtained by reducing $A$ to tridiagonalform.

E igenvahues w ill.be com puted m ostaccurately w hen ABSTOL is set to tw ige the underflow threshold 2*D LAM CH (S ), notzero. Ifthis routine retums w ith $\mathbb{N} \mathrm{FO}>0$, indicating thatsom e eigenvectors did not converge, try setting ABSTOL to $2 *$ SLAM CH (S ).

M (output)
The totalnum ber ofeigenvalues found. $0<=\mathrm{M}<=$ N . IfRANGE=A', M=N, and ifRANGE= 'I'M = $\mathbb{U}-\mathbb{L}+1$ 。

W (output)
O n norm alexit, the firstM elem ents contain the selected eigenvahues in ascending order.
$Z$ (input) If $\mathrm{OBB}=\mathrm{N}^{\prime}$, then Z is notreferenced. If OBZ $=V$ ', then if $\mathbb{N} F O=0$, the firstM colum ns of $Z$ contain the orthonorm aleigenvectors of the $m$ atrix A corresponding to the selected eigenvalues, w ith the i-th colum $n$ of $Z$ holding the eigenvector associated w ith W (i). The eigenvectors are norm alized as follow s: if ITYPE $=1$ or $2, Z * * T * B * Z=I$; if $\operatorname{ITYPE}=3, Z * * T * \operatorname{inv}(B) * Z=I$.
If an eigenvector fails to converge, then that colum $n$ of $Z$ contains the latestapproxim ation to the eigenvector, and the index of the eigenvector is retumed in $\mathbb{F} A \mathbb{I}$. N ote: the userm ustensure that at leastm ax ( $1, M$ ) colum ns are supplied in the amay $Z$; ifRANGE = V', the exactvalue ofM is notknow $n$ in advance and an upper bound m ust be used.

LD Z (input)
The leading dim ension of the array $\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1$, and if $\mathrm{OBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{max}(1, N)$.

W ORK (w orkspace)
On exit, if $\mathbb{N F O}=0, W$ ORK ( 1 ) retums the optim al LW ORK.

## LW ORK (input)

The length of the array $W$ ORK. LW ORK >= $\mathrm{max}(1,8 * \mathrm{~N})$. For optim al efficiency, LW ORK >= $(\mathbb{N B}+3) \star \mathrm{N}$, where NB is the blocksize for SSY TRD retumed by $\amalg \mathrm{LA} E \mathrm{EN}$.

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IW ORK (w orkspace)
dim ension ( $5 \star \mathrm{~N}$ )

FAII (output)
If $O B Z=V^{\prime}$, then if $\mathbb{N F O}=0$, the firstM ele$m$ ents of $\mathbb{F} A \mathbb{I}$ are zero. If $\mathbb{N} F O>0$, then $\mathbb{F} A \mathbb{H}$ contains the indices of the eigenvectors that failed to converge. If $\mathrm{OOBZ}=\mathrm{N}$ ', then $\mathbb{F} A \mathbb{I}$ is notreferenced.
$\mathbb{N}$ FO (output)
= 0: successfulexit
<0: if $\mathbb{N F O}=-$ i, the $i$-th argum enthad an illegalvahue
> 0: SPO TRF orSSY EVX retumed an errorcode:
$<=N:$ if $\mathbb{N} F O=i, S S Y E V X$ failed to converge; i eigenvectors failed to converge. Their indices are stored in array $\mathbb{F A} \mathbb{I} .>N$ : if $\mathbb{N} F O=N+$ $i$, for $1<=i<=N$, then the leading $m$ inor of orderiofB is notpositive definite. The factorization of B could not be com pleted and no eigenvalues or eigenvectors w ere com puted.

## FURTHER DETAILS

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv. of K entucky, USA

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssymm -perform one of the $m$ atrix m atrix operations $C:=$ alpha*A *B + beta*C orC := alpha*B *A + beta*C

## SYNOPSIS

```
SUBROUT\mathbb{NE SSYMM (SDE,UPLO,M,N,ALPHA,A,LDA,B,LDB,BETA,C,}
    LD C )
CHARACTER * 1SDDE,UPLO
INTEGERM,N,LDA,LDB,LDC
REALALPHA,BETA
REALA (LDA **),B (LD B,*),C (LDC,*)
SU BROUT\mathbb{NE SSYMM _64 (S\mathbb{DE,UPLO ,M ,N,ALPHA,A,LDA,B,LDB,BETA,C,}}\mathbf{~},\textrm{L}
    LD C)
CHARACTER * 1 SIDE,UPLO
INTEGER*8M,N,LDA,LDB,LD C
REALALPHA,BETA
REAL A (LDA,*),B (LDB,*),C (LDC ,*)
F95 INTERFACE
    SUBROUT\mathbb{NE SYMM (SDE,UPLO, M ], N ],ALPHA,A,[LDA ],B,[LDB],}
        BETA,C,[LDC])
    CHARACTER (LEN=1)::SDE,UPLO
    INTEGER ::M ,N,LDA,LDB,LDC
    REAL ::ALPHA,BETA
    REAL,D IM ENSION(:,:)::A,B,C
```



```
        BETA,C, [LDC])
```

CHARACTER ( $几 E N=1$ ) : : SDE E , UPLO
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{LD} A, L D B, L D C$
REAL ::ALPHA,BETA
REAL, D $\mathbb{M}$ ENSION (: : : : : A, B, C

## C INTERFACE

\#include <sunperfh>
void ssym m (charside, charuplo, intm , intn, float alpha, float *a, int lda, float *b, int ldb, floatbeta, float* *, int ldc);
void ssym m _64 (charside, charuplo, long m, long n, float alpha, float *a, long lda, float*b, long ldb, floatbeta, float *c, long ldc);

## PURPOSE

ssym $m$ perform sone of the $m$ atrix-m atrix operations $C=$ alpha*A *B + beta* C or $\mathrm{C}:=$ alpha*B *A + beta* C where alpha and beta are scalars, $A$ is a sym $m$ etric $m$ atrix and $B$ and $C$ are $m$ by $n m$ atrices.

## ARGUMENTS

SIDE (input)
O n entry, SIDE specifiesw hether the sym metric $m$ atrix A appears on the leftorright in the operation as follow s:
$S \mathbb{D E}=$ L'or I' $C:=$ a耳pha*A *B + beta* $C$,

SIDE = R 'or 'r' C : alpha*B *A + beta*C ,

U nchanged on exit.

## UPLO (input)

On entry, UPLO specifies whether the upper or lower triangular part of the symm etric $m$ atrix $A$ is to be referenced as follow $s$ :
$\mathrm{UPLO}=\mathrm{U}$ 'or L ' Only the upper triangularpart of the sym $m$ etric $m$ atrix is to be referenced.
$\mathrm{UPLO}=\mathrm{L}$ 'or $\mathrm{I}^{\prime}$ ' O nly the low ertriangularpart of the sym $m$ etric $m$ atrix is to be referenced.

U nchanged on exit.
M (input)
O $n$ entry, M specifies the num ber of row sof the $m$ atrix $C . M$ M $=0$. U nchanged on exit.

N (input)
O n entry, N specifies the num ber of colum ns of the $m$ atrix $C . N>=0$. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.
A (input)
REAL array ofD $\mathbb{M}$ ENSION (LDA, ka), where ka ism when $S \mathbb{D} E=\mathbb{L}$ 'or I ' and is n otherw ise.

Before entry with SDE = L'or I', the m by $m$ part of the array A mustcontain the sym $m$ etric $m$ atrix, such thatw hen UPLO $=U$ 'or $L^{\prime}$ ', the leading $m$ by $m$ uppertriangular part of the array A mustcontain the upper triangular part of the symm etric $m$ atrix and the strictly low er triangularpart of A is not referenced, and $w$ hen UPLO = L' or I', the leading $m$ by $m$ lowertriangularpart of the array A m ust contain the lower triangular part of the sym $m$ etric $m$ atrix and the strictly upper triangular part of A is not referenced.

Before entry with $S \mathbb{D E}=\mathrm{R}$ 'or 'r', the n by $n$ part of the amray A $m$ ustcontain the sym$m$ etric $m$ atrix, such thatw hen UPLO $=U$ 'or L ', the leading $n$ by $n$ upper triangularpart of the array A mustcontain the upper triangular part of the symm etricm atrix and the strictly low er triangularpartof $A$ is not referenced, and when UPLO = L' or I', the leading $n$ by $n$ low er triangularpart of the anray A must contain the lower triangular part of the sym $m$ etric $m$ atrix and the strictly upper triangular part of A is not referenced.

U nchanged on exit.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen $S \mathbb{D} E=\mathbb{L}$ 'or I ' then LD A $>=\mathrm{max}(1, \mathrm{~m})$, other-
w ise LD $\mathrm{A}>=\mathrm{max}(1, \mathrm{n})$. U nchanged on exit.
B (input)
REAL array ofD $\mathbb{M}$ ENSION (LDB, $n$ ). Before entry, the leading $m$ by $n$ partof the array $B$ ust contain the $m$ atrix B. U nchanged on exit.

LD B (input)
O n entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program. LD $B>=m a x(1, m)$. Unchanged on exit.
BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then C need not.be set on input. U nchanged on exit.

C (input/output)
REAL array ofD $\mathbb{I M}$ ENSION (LDC,n). Before entry, the leading $m$ by $n$ partofthe aray $C$ must contain the $m$ atrix $C$, except when beta is zero, in which case $C$ need notbe seton entry. On exit, the array $C$ is overw ritten by the $m$ by n updated m atrix.

LD C (input)
O n entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program.
LD C $>=\max (1, m)$. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
ssym v -perform them atrix-vectoroperation y := alpha*A *x
+ beta*Y
```


## SYNOPSIS

```
SU BROUTINE SSYMV (UPLO,N,ALPHA,A,LDA,X, NNCX,BETA,Y, INCY)
CHARACTER * 1 UPLO
\mathbb{NTEGERN,LDA,INCX,}\mathbb{N}CY
REAL ALPHA,BETA
REAL A (LDA,*),X (*),Y (*)
SUBROUT\mathbb{NE SSYMV_64(UPLO,N,ALPHA,A,LDA,X,INCX,BETA,Y,INCY)}
CHARACTER * 1UPLO
\mathbb{NTEGER*8N,LDA,}\mathbb{N}CX,\mathbb{NCY}
REAL ALPHA,BETA
REALA (LDA,*),X (*),Y (*)
```


## F95 INTERFACE



```
CHARACTER (LEN=1)::UPLO
INTEGER ::N,LDA,}\mathbb{NCX,}\mathbb{N}C
REAL ::ALPHA,BETA
REAL,D IM ENSION (:) ::X,Y
REAL,D IM ENSION (:,:) ::A
```



```
        [\mathbb{NCY ])}
```

$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, L D A, \mathbb{N} C X, \mathbb{N} C Y$
REAL ::ALPHA,BETA
REAL,D $\mathbb{I M}$ ENSION (:) :: X , Y
REAL,D $\mathbb{M}$ ENSION (:,:) ::A

## C INTERFACE

\#include <sunperfh>
void ssym v (charuplo, intn, float alpha, float *a, int lda, float *x, int incx, float beta, float*y, int incy);
void ssym v_64 (charuplo, long n, floatalpha, float *a, long lda, float *x, long incx, floatbeta, float *y, long incy);

## PURPOSE

ssym v perform sthe $m$ atrix-vector operation $y:=a \xi p h a \star A * x+$ beta* $y$, w here alpha and beta are scalars, $x$ and $y$ are $n$ ele$m$ entvectors and $A$ is an $n$ by $n$ sym $m$ etric $m$ atrix .

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the array $A$ is to be referenced as follow s:
$\mathrm{UPLO}=\mathrm{U}$ 'or L ' Only the upper triangularpart of A is to be referenced.
$\mathrm{UPLO}=\mathrm{L}$ 'or l' O nly the low ertriangularpart of A is to be referenced.

U nchanged on exit.

N (input)
O $n$ entry, $N$ specifies the order of the $m$ atrix $A$. $\mathrm{N}>=0$. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha.
U nchanged on exit.

A (input)
Before entry w ith UPLO = U 'or L ', the leading
n by n upper triangularpart of the array A m ust contain the upper triangular part of the sym $m$ etric $m$ atrix and the strictly low ertriangularpartofA is not referenced. Before entry w ith UPLO = L' or '1', the leading $n$ by $n$ low er triangularpart of the array A m ust contain the low er triangular part of the sym $m$ etric $m$ atrix and the strictly uppertriangularpart of $A$ is not referenced. U nchanged on exit.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD A >= $\max (1, n)$. U nchanged on exit.

X (input)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C X))$. Before entry, the increm ented array $X$ must contain the $n$ elem ent vectorx. U nchanged on exit.
$\mathbb{N} C X$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents ofX . $\mathbb{N} C X<>0$. U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then Y need notbe seton input. U nchanged on exit.

Y (input/output)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))$. B efore entry, the increm ented array $Y$ m ust contain the $n$ elem ent vectory. On exit, $Y$ is overw rilten by the updated vectory.
$\mathbb{N C C Y}$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $. \mathbb{N} C Y<>0$. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssyr-perform the symmetric rank 1 operation A:= alpha*x*x'+A

## SYNOPSIS

```
SUBROUT\mathbb{NE SSYR (UPLO,N,ALPHA,X, INCX,A,LDA)}
CHARACTER * 1 UPLO
INTEGERN,\mathbb{NCX,LDA}
REAL ALPHA
REALX (*),A (LDA,*)
SU BROUTINE SSYR_64(UPLO,N,A LPHA,X,NNCX,A,LDA)
CHARACTER * 1 UPLO
NNTEGER*8N,\mathbb{NCX,LDA}
REAL ALPHA
REALX (*),A (LDA,*)
```


## F95 INTERFACE

SU BROUTINE SYR (UPLO, $\mathbb{N}], A L P H A, X,[\mathbb{N} C X], A,[L D A])$
CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R:: N, \mathbb{N C X}, L D A$
REAL ::ALPHA
REAL,D $\mathbb{I M}$ ENSION (:) ::X
REAL,D $\mathbb{M}$ ENSION (:,:) ::A

SU BROUTINE SYR_64 (UPLO, $\mathbb{N}], A L P H A, X,[\mathbb{N C X}], A,[L D A])$
CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R(8):: N, \mathbb{N C X}, L D A$

REAL ::ALPHA
REAL,D IM ENSION (:) ::X
REAL,D IM ENSION (:,:) ::A

## C INTERFACE

\#include <sunperfh>
void ssyr(charuplo, intn, float alpha, float *x, int incx, float *a, int lda);
void ssyr_ 64 (charuplo, long n, float alpha, float *x, long incx, float *a, long lda);

## PURPOSE

ssyrperform $s$ the sym $m$ etric rank 1 operation $A:=a l p h a^{*} X^{*} x^{\prime}$
$+A$, where alpha is a real scalar, x is an n elem entvector and $A$ is an $n$ by $n$ sym $m$ etric $m$ atrix.

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the array $A$ is to be referenced as follow s:

UPLO = U 'or L' Only the upper triangularpart ofA is to be referenced.

UPLO = L 'or I' O nly the low ertriangularpart ofA is to be referenced.

U nchanged on exit.

N (input)
O $n$ entry, $N$ specifies the order of the $m$ atrix $A$. $\mathrm{N}>=0$. U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
$(1+(n-1) * a b s(\mathbb{N} C X))$. Before entry, the increm ented array $X$ must contain the $n$ elem ent vectorx. U nchanged on exit.
$\mathbb{N C X}$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents ofX . $\mathbb{N} C X<>0$. U nchanged on exit.

A (input/output)
Before entry w ith UPLO = U 'or 4 ', the leading $n$ by $n$ uppertriangularpart of the array A m ust contain the uppertriangularpart of the sym $m$ etric $m$ atrix and the strictly low er triangularpart of $A$ is not referenced. O n exit, the upper triangular part of the array A is overw ritten by the upper triangularpart of the updated $m$ atrix. Before entry w ith UPLO = L 'or I', the leading $n$ by $n$ low er triangularpart of the array A m ust contain the low er triangularpart of the sym $m$ etric $m$ atrix and the strictly upper triangularpart of A is not referenced. On exit, the low er triangularpart of the array $A$ is overw rilten by the low er triangular part of the updated $m$ atrix.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD $A>=$ $\max (1, n)$. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssyl2 -perform the symmetric rank 2 operation A := alpha*x*y'+ alpha* ${ }^{*}{ }^{\prime}$ '+ A

## SYNOPSIS

```
SU BROUT\mathbb{NE SSYR2 (UPLO,N,ALPHA,X, INCX,Y, INCY,A,LDA )}
CHARACTER * 1 UPLO
INTEGERN,INCX,\mathbb{NCY,LDA}
REAL ALPHA
REALX (*),Y (*),A (LDA,*)
SU BROUT\mathbb{NE SSYR2_64 (UPLO,N,ALPHA,X, INCX,Y, INCY,A,LDA )}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N, NNCX,INCY,LDA}
REAL ALPHA
REALX (*),Y (*),A (LDA,*)
```


## F95 INTERFACE

```
SU BROUTINE SYR2 (UPLO, \(\mathbb{N}], A L P H A, X,[\mathbb{N C X}], Y,[\mathbb{N C Y}], A,[L D A])\)
CHARACTER (LEN=1)::UPLO
\(\mathbb{N} T E G E R:: N, \mathbb{N C X}, \mathbb{I N C Y , L D A}\)
REAL ::ALPHA
REAL,D IM ENSION (:) :: X,Y
REAL,D IM ENSION (:,:) ::A
SU BROUTINE SYR2_64 (UPLO, \(\mathbb{N}], A L P H A, X,[\mathbb{N C X}], Y,[\mathbb{N C Y}], A,[L D A])\)
CHARACTER (LEN=1) ::UPLO
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{N} C X, \mathbb{N} C Y, L D A\)
```

REAL ::ALPHA
REAL,D $\mathbb{I M}$ ENSION (:) ::X,Y
REAL,D $\mathbb{I}$ ENSION (:,:) ::A

## C INTERFACE

\#include <sunperfh>
void ssyl2 (charuple, intn, float alpha, float *x, int incx, float *y, intincy, float *a, intlda);
void ssyl2_64 (charuplo, long n, floatalpha, float *x, long incx, float *y, long incy, float *a, long lda);

## PURPOSE

ssyl2 performs the symmetric rank 2 operation $A:=$ alpha* $x^{\star} y^{\prime}+$ alpha* $y^{\star} x^{\prime}+A$, w here alpha is a scalar, $x$ and $y$ are $n$ elem ent vectors and $A$ is an $n$ by $n$ sym $m$ etric $m$ atrix.

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the array $A$ is to be referenced as follow s:
$\mathrm{UPLO}=\mathrm{U}$ 'or L ' Only the upper triangularpant of $A$ is to be referenced.
$\mathrm{UPLO}=\mathrm{L}$ 'or l' O nly the low er triangularpart of A is to be referenced.

U nchanged on exit.

N (input)
O n entry, $N$ specifies the order of the $m$ atrix $A$. $N>=0$. U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C X))$. Before entry, the increm ented array $X$ must contain the $n$ elem ent vectorx. U nchanged on exit.
$\mathbb{N C X}$ (input)
On entry, $\mathbb{N} C X$ specifies the increm ent for the elem ents ofX . $\mathbb{N} C X<>0$. U nchanged on exit.

Y (input)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))$. Before entry, the increm ented array $Y$ must contain the $n$ elem ent vectory. U nchanged on exit.
$\mathbb{N C C Y}$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $. \mathbb{N} C Y<>0$. U nchanged on exit.
A (input/output)
Before entry w ith UPLO = U 'or L', the leading $n$ by $n$ upper triangularpart of the array A m ust contain the upper triangular part of the sym $m$ etric $m$ atrix and the strictly low er triangularpart of A is not referenced. On exit, the upper triangular part of the array A is overw rilten by the upper triangularpart of the updated $m$ atrix. Before entry w ith UPLO = L 'or I', the leading $n$ by $n$ low er triangularpart of the array A m ust contain the low er triangularpart of the sym $m$ etric $m$ atrix and the strictly uppertriangularpart of A is not referenced. On exit, the low er triangularpart of the array $A$ is overw rilten by the low er triangular part of the updated $m$ atrix.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD A >= $\max (1, n)$. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssyi2k -perform one of the symm etric rank $2 k$ operations $C$ $:=$ alpha*A *B' + alpha*B*A ' + beta*C orC $:=$ alpha*A *B + alpha*B *A + beta*C

## SYNOPSIS

```
SUBROUT\mathbb{NE SSYR2K (UPLO,TRANSA,N,K,ALPHA,A,LDA,B,LDB,BETA,C,}
    LD C )
CHARACTER * 1 UPLO,TRANSA
\mathbb{NTEGERN,K,LDA,LDB,LDC}
REALALPHA,BETA
REALA (LDA,*),B (LDB,*),C (LDC,*)
SUBROUT\mathbb{NE SSYR2K_64 (UPLO,TRANSA,N,K,ALPHA,A,LDA,B,LDB,BETA,}
        C,LDC)
```

CHARACTER * 1 UPLO, TRANSA
$\mathbb{N} T E G E R * 8 N, K, L D A, L D B, L D C$
REALALPHA,BETA
REALA (LDA , *), B (LDB,*), C (LDC,*)

## F95 INTERFACE

SU BROUTINE SYR2K ©PLO, [TRANSA], $\mathbb{N}],[K], A L P H A, A,[L D A], B,[L D B]$, BETA, C, [LDC])

CHARACTER (LEN=1) ::UPLO,TRANSA
$\mathbb{N} T E G E R:: N, K, L D A, L D B, L D C$
REAL ::ALPHA,BETA
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A,B,C
SUBROUTINE SYR2K_64 (UPLO, [TRANSA], $\mathbb{N}],[K], A L P H A, A,[L D A], B$,
[LD B],BETA, C, [LD C ])

CHARACTER (LEN=1) ::UPLO,TRANSA
$\mathbb{N}$ TEGER (8) ::N, $\mathrm{K}, \mathrm{LDA}, \mathrm{LD} \mathrm{B}, \mathrm{LDC}$
REAL ::ALPHA,BETA
REAL,D $\mathbb{M}$ ENSION (:,:) ::A,B,C

## C INTERFACE

\#include <sunperfh>
void ssyr2k (charuplo, chartransa, int n, int $k$, float alpha, float * a , int lda, float *b, int ldb, float beta, float *c, int ldc);
void ssyi2k_64 (charuplo, chartransa, long n, long $k$, float alpha, float *a, long lda, float*b, long ldb, float.beta, float * c , long ldc);

## PURPOSE

ssy 2 k perform s one of the sym $m$ etric rank 2 k operations $\mathrm{C}:=$ alpha*A*B'+ alpha*B*A'+ beta*C or C := alpha*A *B + alpha*B *A + beta*C w here alpha and beta are scalars, C is an $n$ by $n$ symmetric $m$ atrix and $A$ and $B$ are $n$ by $k$ $m$ atrices in the first case and $k$ by $n m$ atrices in the second case.

## ARGUMENTS

## UPLO (input)

On entry, UPLO specifies whether the upper
or lower triangular part of the array $C$ is
to be referenced as follow s:

UPLO = U'or L' Only the upper triangular partof $C$ is to be referenced.

UPLO = L'or I' Only the lower triangular partof C is to be referenced.

U nchanged on exit.
TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA $=\mathrm{N}^{\prime}$ or $\mathrm{h}^{\prime} \mathrm{C}:=$ alpha*A *B '+ alpha*B*A '

+ beta*C.

TRANSA = T'ort' $\mathrm{C}:=$ alpha*A *B + alpha*B *A + beta*C.

TRANSA = C 'or C' C : alpha*A *B + alpha*B *A + beta*C .

U nchanged on exit.

TRANSA is defaulted to $N$ 'forF $95 \mathbb{I N}$ TERFACE.

N (input)
O n entry, $N$ specifies the order of the $m$ atrix C. N m ust.be at least zero. U nchanged on exit.
$K$ (input)
On entry w th TRANSA = N 'or h', K specifies the num ber of colum ns of the $m$ atrioes $A$ and $B$, and on entry $w$ ith TRANSA $=T$ 'or t'or $C^{\prime}$ or $k^{\prime}, \mathrm{K}$ specifies the num ber of row sof the $m$ atrices $A$ and $B$. $K$ must be at least zero. U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha.
U nchanged on exit.

A (input)
REAL array ofD $\mathbb{M} E N S \mathbb{O N}$ (LDA, ka) , where ka isk when TRANSA = N 'or h', and is n otherw ise. Before entry w ith $\mathrm{TRANSA}=\mathrm{N}^{\prime}$ or
h', the leading $n$ by k partof the amay A
m ustcontain the m atrix $A$, otherw ise the leading k by n partof the array $A$ mustcontain the $m$ atrix A. U nchanged on exit.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen TRANSA $=N$ 'or h'then LDA must be at least $\max (1, n)$, otherw ise LD A m ustbe at least $\max (1, k)$. U nchanged on exit.

B (input)
REAL array ofD $\mathbb{M E N S I O N}$ (LDB, kb ), where kb isk when TRANSA = N 'or h', and is n otherw ise. Before entry w ith TRANSA $=\mathrm{N}^{\prime}$ or $h$ ', the leading $n$ by k part of the array $B$ m ust contain the $m$ atrix $B$, otherw ise the leading $k$ by $n$ partof the aray $B$ mustcontain the $m$ atrix B. U nchanged on exit.

LD B (input)
O n entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program. $W$ hen TRANSA $=N$ 'or $h$ 'then LDB must be at least $\max (1, n)$, otherw ise LD B m ustbe at least $\max (1, k)$. U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. U nchanged on exit.

C (input/output)
REAL amay ofD $\mathbb{M} E N S I O N(L D C, n)$.

Before entry w th UPLO = U'or L', the leading $n$ by $n$ uppertriangularpart of the anray $C$ $m$ ustcontain the upper triangular part of the sym $m$ etric $m$ atrix and the strictly low er triangularpartofC is not referenced. On exit, the upper triangularpart of the array $C$ is overw ritten by the upper triangularpart of the updated $m$ atrix.

Before entry w ith UPLO = L 'or I', the leading $n$ by $n$ low er triangular part of the array $C$ $m$ ustcontain the low er triangular part of the sym $m$ etric $m$ atrix and the strictly uppertriangularpartof C is not referenced. On exit, the low er triangularpart of the array $C$ is overw ritten by the low er triangularpart of the updated $m$ atrix.

LD C (input)
O n entry, LD C specifies the first dim ension of C as declared in the calling (sub) program.
LD C m ust be at leastm ax ( $1, \mathrm{n}$ ). U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssynfs - im prove the com puted solution to a system of linear equations $w$ hen the coefficientm atrix is sym $m$ etric indefinite, and provides errorbounds and backw ard enror estim ates for the solution

## SYNOPSIS

```
SU BROUT\mathbb{NE SSYRFS (UPLO,N,NRHS,A,LDA,AF,LDAF, \mathbb{PIVOT,B,LDB,X,}}\mathbf{N},\textrm{L}
    LDX,FERR,BERR,W ORK,W ORK 2,INFO)
```

CHARACTER * 1 UPLO
$\mathbb{N}$ TEGERN,NRHS,LDA,LDAF,LDB,LDX, $\mathbb{N} F O$
$\mathbb{N} T E G E R \mathbb{P} \mathbb{V} O T\left({ }^{( }\right)$, WORK 2 ( ${ }^{*}$ )

$\operatorname{BERR}$ (*), W ORK (*)
SU BROUTINE SSYRFS_64 (UPLO,N,NRHS,A,LDA,AF,LDAF, $\mathbb{P} \mathbb{I V} O T, B, L D B$,
X,LDX,FERR,BERR,W ORK,W ORK2, $\mathbb{N} F O$ )
CHARACTER * 1 UPLO
$\mathbb{N}$ TEGER*8N,NRHS,LDA,LDAF,LDB,LDX, $\mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{I V O T}(*), \mathrm{W}$ ORK $2(*)$
REAL A (LDA, $)$, AF (LDAF,*), B (LDB , *), X (LDX,*), FERR (*),
BERR (*) , W ORK (*)

## F95 INTERFACE

SU BROUTINE SYRFS (UPLO,N,NRHS,A, [LDA],AF, [LDAF], $\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LD} B]$, X, [LDX],FERR,BERR, [W ORK], [W ORK2], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1)::UPLO
$\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V O T}, \mathrm{W} O R K 2$

SU BROUT $\mathbb{N} E S Y R F S \_64(U P L O, N, N R H S, A,[L D A], A F,[L D A F], \mathbb{P} \mathbb{I} O T, B$, $[$ [LD B ], X, [ LDX ], FERR, BERR, $\mathbb{W} O R K],[W O R K 2],[\mathbb{N F O}])$

CHARACTER (LEN=1) :: UPLO
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} A, L D A F, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T$, W ORK 2
REAL,D $\mathbb{M}$ ENSION (:) ::FERR,BERR,W ORK
REAL,D $\mathbb{I M} E N S I O N(:,:): A, A F, B, X$

## C INTERFACE

\#include <sunperfh>
void ssyrfs (charuplo, intn, intnrhs, float *a, int lda, float *af, int ldaf, int*ipivot, float*b, int ldb, float *x, int ldx, float * ferr, float *berr, int*info);
void ssyrfs_64 (charuplo, long n, long nihs, float*a, long lda, float *af, long ldaf, long *ipivot, float *b, long ldb, float *x, long ldx, float *ferr, float *berr, long *info);

## PURPOSE

ssyufs im proves the com puted solution to a system of linear equations w hen the coefficientm atrix is sym m etric indefinite, and provides errorbounds and backw ard error estim ates for the solution.

## ARGUMENTS

UPLO (input)
= U ': U pper triangle ofA is stored;
= L': Low er triangle ofA is stored.

N (input) The order of the matrix A. $\mathrm{N}>=0$.
NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the $m$ atrioes $B$ and $X$. NRH $S>=0$.

A (input) The sym m etric $m$ atrix $A$. If $\mathrm{PLO}=\mathrm{U}$ ', the leading N -by N uppertriangularpart of A contains the upper triangularpart of the $m$ atrix $A$, and the strictly low ertriangularpartofA is not refer-
enced. If UPLO = L', the leading N -by-N lower triangularpartofA contains the low er triangular part of the $m$ atrix A, and the strictly upper triangularpart of A is not referenced.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, N)$.

AF (input)
The factored form of them atrix A. AF contains the block diagonal matrix D and themultipliers used to obtain the factor $U$ or $L$ from the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$ orA $=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ as com puted by SSY TRF.

LDAF (input)
The leading dim ension of the array AF. LDAF >= $\max (1, N)$.
$\mathbb{P I V O T}$ (input)
D etails of the interchanges and the block structure ofD as determ ined by SSY TRF.
$B$ (input) The righthand side m atrix $B$.
LD B (input)
The leading dim ension of the array $B$. LD B >= $\max (1, N)$.

X (input/output)
On entry, the solution $m$ atrix $X$, as com puted by SSY TRS. On exit, the im proved solution $m$ atrix $X$.

LD X (input)
The leading dim ension of the array X . LD X >= $\max (1, N)$.

FERR (output)
The estim ated forw ard errorbound for each solution vector $X(\mathcal{j})$ the $j$ th column of the solution $m$ atrix X). If XTRUE is the true solution corresponding to $X(\mathcal{H})$, FERR ( $)$ ) is an estim ated upperbound for the $m$ agnitude of the largest ele$m$ ent in ( $X(\mathcal{D})-X$ TRUE) divided by the $m$ agnitude of the largestelem ent in $X(\mathcal{)}$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

BERR (output)

The com ponentw ise relative backw ard error of each solution vector $X(\mathcal{)}$ (ie., the sm allest relative change in any elem entofA orB thatm akes X ( $\mathcal{(}$ ) an exactsolution).

W ORK (w orkspace)
dim ension ( $3 * \mathrm{~N}$ )
W ORK 2 (w orkspace)
dim ension (N)
$\mathbb{N} F O$ (output)
= 0: successfulexit
$<0:$ if $\mathbb{N F O}=-$ i, the $i$-th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssyrk -perform one of the sym m etric rank $k$ operations C $:=$ alpha*A ${ }^{*}$ A ' beta*C orC $:=$ alpha*A *A + beta*C

## SYNOPSIS

```
SUBROUTINE SSYRK(UPLO,TRANSA,N,K,ALPHA,A,LDA,BETA,C,LDC)
CHARACTER * 1 UPLO,TRANSA
INTEGERN,K,LDA,LDC
REAL ALPHA,BETA
REAL A (LDA,*),C (LDC ,*)
SU BROUT\mathbb{NE SSYRK_64 (UPLO,TRANSA,N,K,ALPHA,A,LDA,BETA,C,LDC)}
CHARACTER * 1 UPLO,TRANSA
\mathbb{NTEGER*8N,K,LDA,LDC}
REALALPHA,BETA
REALA (LDA,*),C (LDC **)
```


## F95 INTERFACE

SU BROUTINE SYRK (UPLO, [TRANSA], $\mathbb{N}],[K], A L P H A, A,[L D A], B E T A, C$, [LD C])

CHARACTER (LEN=1) ::UPLO,TRANSA
$\mathbb{N} T E G E R:: N, K, L D A, L D C$
REAL ::ALPHA,BETA
REAL,D $\mathbb{M}$ ENSION (: :: : ::A, C
SU BROUTINE SYRK_64 (UPLO, [TRANSA], $\mathbb{N}],[K], A L P H A, A,[L D A], B E T A$, C, (LD C ])

CHARACTER (LEN=1) ::UPLO,TRANSA
$\mathbb{N}$ TEGER (8) :: N, K, LDA, LDC
REAL ::ALPHA,BETA
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A, C

## C INTERFACE

\#include <sunperfh>
void ssyrk (char uplo, char transa, int $n$, int $k$, float aloha, float *a, intlda, floatbeta, float * C , int ldc);
void ssyrk_64 (charuplo, chartransa, long n, long k, float alpha, float *a, long lda, floatbeta, float * C , long ldc);

## PURPOSE

ssyik perform s one of the sym $m$ etric rank $k$ operations $C:=$ alpha*A *A '+ beta*C orC $:=$ alpha*A *A + beta*C where alpha and beta are scalars, $C$ is an $n$ by $n$ sym $m$ etric $m$ atrix and $A$ is an $n$ by $k m$ atrix in the first case and $a k$ by $n$ $m$ atrix in the second case.

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whether the upper
or lower triangular part of the amay $C$ is
to be referenced as follow s:
$U P L O=U$ 'or $G^{\prime}$ Only the upper triangular part of $C$ is to be referenced.

UPLO = L'or I' Only the lower triangular part of $C$ is to be referenced.

U nchanged on exit.

TRANSA (input)
O n entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA $=N^{\prime}$ or $h^{\prime} \mathrm{C}:=$ alpha*A *A ' + beta* C.

TRANSA $=$ T'ort' $\mathrm{C}:=$ alpha*A *A + beta*C.

TRANSA $=$ C 'or $C^{\prime} C:=$ alpha*A *A + beta*C.

U nchanged on exit.
TRANSA is defaulted to $N$ 'forF $95 \mathbb{I N}$ TERFACE.

N (input)
On entry, $N$ specifies the order of the $m$ atrix $C$. N m ustbe at least zero. U nchanged on exit.

K (input)
On entry w ith TRANSA $=N$ 'or $h$ ', $K$ specifies the number of columns of the matrix $A$, and on entry $w$ ith TRANSA $=$ ' 'or t'or $C^{\prime}$ or $\mathrm{E}^{\prime}$, $K$ specifies the num ber of row sof the m atrix $\mathrm{A} . \mathrm{K}$ m ustibe at least zero. U nchanged on exit.
A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
REAL aray ofD $\mathbb{I M}$ ENSION (LDA, ka ), where ka isk when TRANSA = N'orh', and is
n otherw ise. Before entry w th TRANSA $=\mathrm{N}$ ' or h ', the leading n by k part of the array $A$ $m$ ust contain the $m$ atrix $A$, otherw ise the leading k by n partofthe array A mustcontain the m atrix A. U nchanged on exit.

LDA (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen TRANSA $=N$ 'orh'then LDA must be at least $\max (1, n)$, otherw ise LDA m ustbe at least $\max (1, k)$. U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. U nchanged on exit.

C (input/output)
REAL aray ofD $\mathbb{I}$ ENSION (LDC,n).
Before entry w ith UPLO = U 'or L ', the leading $n$ by $n$ uppertriangularpart of the array $C$ $m$ ustcontain the upper triangular part of the sym $m$ etric $m$ atrix and the strictly low ertriangularpartofC is not referenced. On exit, the upper triangularpart of the array $C$ is overw ritten by the upper triangularpart of the updated
m atrix.

Before entry w ith UPLO = L 'or I', the leading $n$ by $n$ low ertriangular part of the anray $C$ m ust contain the low er triangular part of the sym $m$ etric $m$ atrix and the strictly upper triangularpartof C is not referenced. On exit, the low er triangularpart of the amay $C$ is overw ritten by the low er triangularpart of the updated $m$ atrix.

LD C (input)
O n entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program. LD C must be at leastm ax ( $1, \mathrm{n}$ ). U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssysv -com pute the solution to a real system of linear equations $A * X=B$,

## SYNOPSIS

```
SUBROUTINE SSYSV (UPLO,N,NRHS,A,LDA,\mathbb{PIV,B,LDB,W ORK,LW ORK,}
    \mathbb{NFO)}
CHARACTER * 1 UPLO
INTEGERN,NRHS,LDA,LDB,LW ORK,INFO
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(})
REALA (LDA,*),B (LDB,*),W ORK (*)
SU BROUT\mathbb{NE SSY SV_64 (UPLO ,N,NRHS,A,LDA, \mathbb{P IV ,B,LD B,W ORK,}}\mathbf{~}\mathrm{ , N,}
    LW ORK,\mathbb{NFO)}
CHARACTER * 1 UPLO
```



```
\mathbb{NTEGER*8 \mathbb{P IV (*)}}\mathbf{(})
REALA (LDA,*),B (LDB,*),W ORK (*)
```


## F95 INTERFACE

```
SU BROUTINE SYSV (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I}, B,[L D B],[W\) ORK], [LW ORK], [ \(\mathbb{N F O}\) ])
CHARACTER (LEN=1) ::UPLO
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}\)
REAL,D IM ENSION (:) ::W ORK
REAL,D IM ENSION (:,:) ::A,B
```

SU BROUTINE SYSV_64 (UPLO, $\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I}, B,[L D B],[W$ ORK],

$$
[\mathrm{LW} O R K],[\mathbb{N} F O])
$$

CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D B, L W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V}$
REAL,D $\mathbb{M}$ ENSION (:) ::W ORK
REAL,D $\mathbb{M}$ ENSION (: : : : ::A, B

## C INTERFACE

\#include <sunperfh>
void ssysv (charuplo, intn, intnrhs, float *a, int lda, int *ipivot, float *b, int ldb, int *info);
void ssysv_64 (char uplo, long n, long nrhs, float *a, long lda, long *ipivot, float *b, long ldb, long *info);

## PURPOSE

ssysv com putes the solution to a realsystem of linearequations
$A * X=B$, where $A$ is an $N$-by-N symm etric $m$ atrix and $X$ and $B$ are $N$-by $-N$ RH S $m$ atrices.

The diagonalpivoting $m$ ethod is used to factorA as
$A=U * D * U * *$, if $U P L O=U$ ', or
$A=L * D * L * T$, if UPLO $=L$ ',
where $U$ (orL) is a productof perm utation and unit upper (low er) triangular $m$ atrices, and $D$ is sym $m$ etric and block diagonalw ith 1 -by -1 and 2 -by -2 diagonalblocks. The factored form of $A$ is then used to solve the system of equations $A * X=B$.

## ARGUMENTS

## UPLO (input)

= U : U pper triangle ofA is stored;
= L': Low er triangle ofA is stored.
N (input) The num ber of linear equations, ie., the order of them atrix $A . N>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS $>=0$.

A (input/output)

On entry, the symm etric $m$ atrix $A$. If $\mathrm{PLO}=\mathrm{U}$ ', the leading N -by -N uppertriangularpart of A contains the upper triangular part of the $m$ atrix $A$, and the strictly low er triangularpart of A is not referenced. If $\mathrm{UPLO}=\mathrm{L}$ ', the leading N -by -N low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangular partofA is not referenced.

On exit, if $\mathbb{N} F O=0$, the block diagonalm atrix $D$ and the multipliers used to obtain the factor $U$ or L from the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$ or $\mathrm{A}=$ L *D *L**T as com puted by SSY TRF.
LDA (input)
The leading dim ension of the array A. LDA >= $\max (1, \mathbb{N})$.

IPIV (output)
D etails of the interchanges and the block structure ofD, as determ ined by SSY TRF. If $\mathbb{P} \mathbb{I V}(k)>$ 0 , then row $s$ and colum ns $k$ and $\mathbb{P} \mathbb{I V}(k)$ were interchanged, and $\mathrm{D}(\mathrm{k}, \mathrm{k})$ is a 1-by-1 diagonalblock. If $U P L O=U$ 'and $\mathbb{P} \mathbb{I V}(k)=\mathbb{P} \mathbb{I}(k-1)<0$, then rows and colum ns $k-1$ and $-\mathbb{P} I V(k)$ were interchanged and $D(k-1 *, k-1 * k)$ is a $2-b y-2$ diagonal block. IfU PLO = L'and $\mathbb{P} \mathbb{I V}(k)=\mathbb{P} \mathbb{I V}(k+1)<0$, then row $s$ and colum ns $k+1$ and - $\mathbb{P}$ IV ( $k$ ) w ere interchanged and $D(k: k+1, k \cdot k+1)$ is a $2-b y-2$ diagonal block.

B (input/output)
On entry, the N -by-NRH S righthand side m atrix B . On exit, if $\mathbb{N} F O=0$, the $N$-by $-N$ RH $S$ solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the aray B. LD B $>=$ $\max (1, N)$.

W ORK (w orkspace)
On exit, if $\mathbb{N F}$ F $=0, \mathrm{~W}$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The length ofW ORK. LW ORK >=1, and forbestperform ance LW ORK >=N*NB, where NB is the optim al blocksize forSSY TRF.

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim al size of
the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
$\mathbb{I N F O}$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N} F O=-$ i, the $i$-th argum ent had an illegalvalue
$>0:$ if $\mathbb{N} F O=i, D(i, i)$ is exactly zero. The factorization has been com pleted, but the block diagonalm atrix $D$ is exactly singular, so the solution could notbe com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssysvx -use the diagonal pivoting factorization to com pute the solution to a realsystem of linearequations $A * X=B$,

## SYNOPSIS

```
SUBROUT\mathbb{NE SSY SVX (FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,B,}}\mathbf{N},\mp@code{N},
    LDB,X,LDX,RCOND,FERR,BERR,W ORK,LDW ORK,W ORK 2,INFO)
CHARACTER * 1FACT,UPLO
INTEGERN,NRHS,LDA,LDAF,LDB,LDX,LDW ORK,INFO
INTEGER \mathbb{PIVOT (*),W ORK2(*)}
REALRCOND
REAL A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX ,*), FERR (*),
BERR (*),W ORK (*)
SUBROUT\mathbb{NE SSY SVX_64(FACT,UPLO,N,NRHS,A,LDA,AF,LDAF, IPIVOT,}
    B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,LDW ORK,W ORK 2, INFO)
```

CHARACTER * 1 FACT,UPLO
$\mathbb{N}$ TEGER*8N,NRHS,LDA,LDAF,LDB,LDX,LDW ORK, $\mathbb{N} F O$
$\mathbb{N}$ TEGER * $8 \mathbb{P} \mathbb{I V O T}(*), W$ ORK 2 (*)
REALRCOND
REAL A (LDA $\left.{ }^{\star}\right)$, AF (LDAF,*), B (LDB , $\left.{ }^{\star}\right)$, X (LDX,*), FERR (*),
BERR (*) ,W ORK (*)

## F95 INTERFACE

 B, [LDB],X, [LDX],RCOND,FERR,BERR, [W ORK], [LDW ORK], [W ORK2], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::FACT,UPLO
$\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, L D W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{I V O T}, \mathrm{W} O R K 2$
REAL ::RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::FERR,BERR,W ORK
REAL,D $\mathbb{I}$ ENSION (:,:) ::A, AF,B,X

SU BROUTINE SYSVX_64 (FACT, UPLO, N,NRHS,A, [LDA],AF, [LDAF], $\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}], \mathrm{X},[\mathrm{LDX}], R C O N D, F E R R, B E R R,[W O R K],[L D W O R K]$, $[\mathbb{W} O R K 2],[\mathbb{N F O}])$

CHARACTER (LEN=1) ::FACT, UPLO
$\mathbb{N}$ TEGER (8) :: N , NRHS,LDA,LDAF,LDB,LDX,LDW ORK, $\mathbb{I N F O}$
$\mathbb{N}$ TEGER (8), D $\mathbb{I M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V O T}, W$ ORK 2
REAL ::RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::FERR,BERR,W ORK
REAL, D $\mathbb{M}$ ENSION (: : : : : : A, AF, B, X

## C INTERFACE

\#include <sunperfh>
void ssysvx (char fact, charuplo, intn, intnrhs, float*a, int lda, float*af, int ldaf, int *ipivot, float *b, int ldlb, float *x, int ldx, float *roond, float *ferr, float *berr, int *info);
void ssysvx_64 (char fact, char uplo, long n, long nrhs, float *a, long lda, float*af, long ldaf, long *ipivot, float *b, long ldb, float *x, long ldx, float *rcond, float *ferr, float *berr, long *info);

## PURPOSE

ssysvx uses the diagonalpivoting factorization to com pute the solution to a realsystem of linear equations $A * X=B$, $w$ here $A$ is an $N$ boy $N$ sym $m$ etric $m$ atrix and $X$ and $B$ are $N$ boyN R H S m atrices.

Emorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:

1. IfFACT $=N$ ', the diagonalpivoting $m$ ethod is used to factorA.
The form of the factorization is
$A=U * D * U * * T$, if $U P L O=U \prime$, or
$A=L * D * L * * T$, if $U P L O=L \prime$
where U (orL) is a product ofperm utation and unitupper
(low er)
triangularm atrices, and $D$ is sym $m$ etric and block diago-

## nalw th

1-by-1 and 2-by-2 diagonalblocks.
2. If som eD $(i, i)=0$, so thatD is exactly singular, then the routine
retums w ith $\mathbb{N F O}=$ i. O therw ise, the factored form of A is used
to estim ate the condition num ber of the $m$ atrix $A$. If the reciprocal of the condition num ber is less than $m$ achine precision,
$\mathbb{N}$ FO $=\mathrm{N}+1$ is retumed as a waming, but the routine stillgoes on
to solve for $X$ and com pute error bounds as described below .
3.The system ofequations is solyed for $X$ using the factored form of A.
4. Iterative refinem ent is applied to im prove the com puted solution
$m$ atrix and calculate error bounds and backw ard error estim ates
for it.

## ARGUMENTS

FACT (input)
Specifies w hether ornot the factored form of A has been supplied on entry. = $\mathrm{F}^{\prime}:$ O n entry, $\mathrm{A} F$ and $\mathbb{P I V O T}$ contain the factored form of A. AF and $\mathbb{P} \mathbb{I V}$ O T w illnotbe m odified. $=\mathrm{N}$ ': Them atrix A w illbe copied to A F and factored.

UPLO (input)
= U ': U pper triangle ofA is stored;
$=L^{\prime}$ : Low er triangle of $A$ is stored.
N (input) The num ber of linear equations, i.e., the order of them atrix $A . N>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the m atrices B and X. NRHS >=0.

A (input) The symm etricm atrix $A$. If UPLO $=U$ ', the leading $N$ by -N uppertriangularpartofA contains the upper triangularpart of the $m$ atrix $A$, and the strictly low ertriangularpartofA is not refer-
enced. If UPLO = L', the leading N -by-N lower triangularpartofA contains the low er triangular part of the $m$ atrix A, and the strictly upper triangularpart ofA is not referenced.

LDA (input)
The leading dim ension of the anay A. LDA >= $\max (1, N)$.

AF (input/output)
If FACT = F ', then $A F$ is an inputargum ent and on entry contains the block diagonalm atrix D and the m ultipliers used to obtain the factor $U$ orL from the factorization $A=U * D * U * * T$ orA $=L * D * L * * T$ as com puted by SSY TRF .
If FA C T = N ', then AF is an output argum ent and on exit retums the block diagonalm atrix $D$ and the multipliers used to obtain the factorU or L from the factorization $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$ or $\mathrm{A}=$ $\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$.

LDAF (input)
The leading dim ension of the array AF. LDAF >= $\max (1, N)$.

IPIVOT (inputoroutput)
If FACT = $\mathrm{F}^{\prime}$, then $\mathbb{P} \mathbb{I} O T$ is an input argum ent and on entry contains details of the interchanges and the block structure of D, as determ ined by SSY TRF. If $\mathbb{P} \operatorname{IV} O T(k)>0$, then row $s$ and colum ns $k$ and $\mathbb{P} \mathbb{I V O T}(k)$ were interchanged and $D(k, k)$ is a 1 -by-1 diagonal block. If UPLO $=U '$ and $\mathbb{P} \mathbb{I} O T(k)=\mathbb{P} \mathbb{I V} O T(k-1)<0$, then row sand colum ns $\mathrm{k}-1$ and $-\mathbb{P} \mathbb{I V O T}(\mathrm{k})$ were interchanged and $\mathrm{D}(\mathrm{k}-$ $1 \mathrm{k}, \mathrm{k}-1 \mathrm{k}$ ) is a 2 -by-2 diagonalblock. IfUPLO = L 'and $\mathbb{P} \mathbb{I V} O T(k)=\mathbb{P} \mathbb{I V} O T(k+1)<0$, then row $s$ and colum nsk+1 and $-\mathbb{P} \operatorname{IV} O T(k)$ were interchanged and D $k: k+1, k: k+1)$ is a 2 -by-2 diagonalblock.

If $\mathrm{FACT}=\mathrm{N}$ ', then $\mathbb{P} \mathbb{I V O T}$ is an output argum ent and on exitcontains details of the interchanges and the block structure of D, as determ ined by SSY TRF .

B (input) The N by -N R H S righthand side m atrix B .

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, \mathbb{N})$.

X (output)
If $\mathbb{N} F O=0$ or $\mathbb{N} F O=\mathrm{N}+1$, the N -by-NRH S solution
$m$ atrix $X$.

LD X (input)
The leading dim ension of the anay X . LD X >= $\max (1, N)$.

RCOND (output)
The estim ate of the reciprocal condition num ber of the matrix A. IfRCOND is less than them achine precision (in particular, if RCOND $=0$ ), the $m$ atrix is singular to working precision. This condition is indicated by a retum code of $\mathbb{N}$ FO > 0.

## FERR (output)

The estim ated forw ard emrorbound for each solution vector $X()$ ) the $j$ th colum $n$ of the solution $m$ atrix X). If XTRUE is the true solution corresponding to $X(\mathcal{H}), \operatorname{FERR}(\mathcal{)})$ is an estim ated upperbound for the $m$ agnitude of the largest ele$m$ ent in ( $X(\mathcal{D})-X$ TRU $E$ ) divided by the $m$ agnitude of the largestelem ent in $X(\mathcal{J})$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

## BERR (output)

The com ponentw ise relative backw ard error of each solution vector X (i) (i.e., the sm allest relative change in any elem entof $A$ or $B$ thatm akes $X(\mathcal{J})$ an exactsolution).

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length ofW ORK. LDW ORK >= 3*N , and for best perform ance LDW ORK $>=N * N B$, where $N B$ is the optim alblocksize forSSY TRF.

If LDW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 2 (w orkspace)
dim ension $(\mathbb{N})$
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N F O}=-$ i, the $i$-th argum enthad an illegalvalue
$>0$ : if $\mathbb{N F O}=i$, and $i$ is
$<=\mathrm{N}: \mathrm{D}(i, i)$ is exactly zero. The factorization
has been completed but the factorD is exactly singular, so the solution and error bounds could not be com puted. RCOND $=0$ is retumed. $=\mathrm{N}+1$ : D is nonsingular, butRCOND is less than $m$ achine precision, $m$ eaning that the $m$ atrix is singular to w orking precision. N evertheless, the solution and error bounds are com puted because there are a num ber of situationsw here the com puted solution can bem ore accurate than the value ofRC O N D w ould suggest.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

ssytd2 - reduce a realsym m etric m atrix A to sym m etric tridiagonal form T by an orthogonal sim ilarity transform ation

## SYNOPSIS

```
SU BROUT\mathbb{NE SSYTD 2 (UPLO,N,A,LDA,D ,E,TAU , INFO )}
CHARACTER * 1 UPLO
\mathbb{NTEGERN,LDA,}\mathbb{N}FO
REALA (LDA,*),D (*),E (*),TAU (*)
SU BROUT\mathbb{NE SSY TD 2_64 (UPLO,N,A,LDA,D ,E,TAU,INFO)}
CHARACTER * 1 UPLO
INTEGER*8N,LDA,INFO
REALA (LDA,*),D (*),E (*),TAU (*)
F95 INTERFACE
    SUBROUT\mathbb{NE SYTD 2 (UPLO,N,A, [LDA ],D ,E,TAU, [NNFO])}
    CHARACTER (LEN=1) ::UPLO
    \mathbb{NTEGER ::N,LDA,}\mathbb{NFO}
    REAL,DIM ENSION (:) ::D,E,TAU
    REAL,D IM ENSION (:,:) ::A
    SU BROUT\mathbb{NE SYTD 2_64 (UPLO,N,A, [LDA ],D ,E,TAU, [NFO ])}
    CHARACTER (LEN=1) ::UPLO
    \mathbb{NTEGER (8) ::N,LDA,}\mathbb{NFO}
    REAL,D IM ENSION (:) ::D,E,TAU
    REAL,DIM ENSION (:,:) ::A
```


## C INTERFACE

\#include <sunperfh>
void ssytd2 (charuplo, intn, float *a, intlda, float *d, float *e, float *tau, int *info);
void ssytd2_64 (charuplo, long n, float *a, long lda, float *d, float *e, float *tau, long *info);

## PURPOSE

ssytd2 reduces a real sym $m$ etric $m$ atrix A to sym $m$ etric tridiagonal form $T$ by an orthogonal sim ilarity transform ation: Q ' * $A * Q=T$.

## ARGUMENTS

UPLO (input)
Specifies w hether the upper or low er triangular part of the sym $m$ etricm atrix A is stored:
= U ': U ppertriangular
= L': Low ertriangular

N (input) The orderof the matrix A. $\mathrm{N}>=0$.
A (input) $O n$ entry, the sym $m$ etric $m$ atrix $A$. If $U P L O=U '$, the leading $n-b y-n$ upper triangularpart of $A$ contains the uppertriangular part of the $m$ atrix $A$, and the strictly low ertriangularpart of A is not referenced. If UPLO = ' L ', the leading $\mathrm{n}-\mathrm{by}-\mathrm{n}$ low er triangularpart of $A$ contains the low ertriangularpart of the m atrix A, and the strictly upper triangular part of A is not referenced. On exit, if UPLO = U ', the diagonal and first superdiagonalofA are overw rilten by the corresponding elem ents of the tridiagonalm atrix $T$, and the ele$m$ ents above the first superdiagonal, $w$ ith the array TAU, represent the orthogonalm atrix $Q$ as a product of elem entary reflectors; if $\mathrm{IPLO}=\mathrm{L}$ ', the diagonaland firstsubdiagonalofA are overw ritten by the comesponding elem ents of the tridiagonalm atrix $T$, and the elem ents below the first subdiagonal, $w$ ith the amay TA $U$, represent the orthogonalm atrix $Q$ as a product of elem entary reflectors. See FurtherD etails.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, \mathbb{N})$.

D (output)
The diagonalelem ents of the tridiagonalm atrix T :
D $(i)=A(i, i)$.

E (output)
The off-diagonal elem ents of the tridiagonal $m$ atrix $T: E(i)=A(i, i+1)$ if $U P L O=U ', E(i)=$ A $(\mathbf{i}+1, i)$ ifUPLO $=L$.

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).
$\mathbb{N} F O$ (output)
= 0 : successfiulexit
<0: if $\mathbb{N N F O}=-i$, the $i$-th argum ent had an illegalvalue.

## FURTHER DETAILS

If $U P L O=U$ ', the $m$ atrix $Q$ is represented as a product of elem entary reflectors

$$
Q=H(n-1) \ldots H(2) H(1) .
$$

Each H (i) has the form

$$
H(i)=I-\tan * v^{*} v^{\prime}
$$

where tau is a real scalar, and $v$ is a realvectorw ith $v(i+1 m)=0$ and $v(i)=1 ; v(1: i-1)$ is stored on exitin A ( $1:-1, i+1)$, and tau in TAU (i).

If $U P L O=L$ ', the $m$ atrix $Q$ is represented as a product of elem entary reflectors

$$
Q=H(1) H(2) \ldots H(n-1) .
$$

Each H (i) has the form

$$
H(i)=I-\tan * v^{*} v^{\prime}
$$

$w$ here tau is a real scalar, and $v$ is a realvectorw ith $\mathrm{v}(1: i)=0$ and $\mathrm{v}(\mathrm{i}+1)=1 ; \mathrm{v}(\mathrm{i}+2 \mathrm{n})$ is stored on exit in A (i+2n,i), and tau in TAU (i).

```
exam ples w ith n = 5:
```

```
ifUPLO = U ': ifUPLO = L':
    ( d e v2 v3 v4 ) ( d
)
    ( d e v3 v4 ) ( e d
)
    ( d e v4 ) ( v1 e d
)
    ( d e ) ( v1 v2 e d
)
    ( d ) (v1 v2 v3 e d
    )
```

where d and e denote diagonaland off-diagonal elem ents of $T$, and videnotes an elem ent of the vectordefining $H$ (i).

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

ssytf2 - com pute the factorization of a real sym m etric $m$ atrix A using the Bunch $-K$ aufn an diagonalpivoting $m$ ethod

## SYNOPSIS

```
SU BROUT\mathbb{NE SSYTF2(UPLO,N,A,LDA, \mathbb{PIV , INFO)}}\mathbf{N}\mathrm{ (N)}
    CHARACTER * 1 UPLO
    \mathbb{NTEGERN,LDA,}\mathbb{N}FO
    \mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(*)}
    REALA (LDA,*)
```



```
    CHARACTER * 1 UPLO
    INTEGER*8N,LDA,INFO
    \mathbb{NTEGER*8 \mathbb{PIV (*)}}\mathbf{(*)}
    REALA (LDA,*)
F95 INTERFACE
```



```
    CHARACTER (LEN=1)::UPLO
    \mathbb{NTEGER ::N,LDA,}\mathbb{N}FO
    INTEGER,D IM ENSION (:) :: \mathbb{PIV}
    REAL,DIMENSION (:,:)::A
    SUBROUTINE SYTF2_64 (UPLO, NN ],A, [LDA ], \mathbb{PIV ,[\mathbb{NFO ])}}\mathbf{(})
    CHARACTER (LEN=1) ::UPLO
    \mathbb{NTEGER (8) ::N,LDA,}\mathbb{NFO}
```

$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I}$
REAL,D IM ENSION (:,:) ::A

## C INTERFACE

\#include <sunperfh>
void ssytf2 (charuplo, intn, float *a, int lda, int *ipiv, int*info);
void ssytf2_64 (charuplo, long n, float*a, long lda, long *ịiv, long *info);

## PURPOSE

ssytf2 com putes the factorization of a realsym $m$ etric $m$ atrix
A using the Bunch $K$ aufm an diagonalpivoting $m$ ethod:

$$
A=U * D * U^{\prime} \text { or } A=L * D * L^{\prime}
$$

where U (orL) is a productof perm utation and unit upper (low er) triangularm atrioes, U 'is the transpose of U , and D is sym $m$ etric and block diagonalw ith 1 -by -1 and 2 -by -2 diagonalblocks.

This is the unblocked version of the algorithm , calling Level2 BLAS .

## ARGUMENTS

UPLO (input)
Specifies w hether the upper or low er triangular part of the sym $m$ etric $m$ atrix $A$ is stored:
$=\mathrm{U}$ ': U pper triangular
= L': Low ertriangular

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
O $n$ entry, the sym m etric $m$ atrix A. If $\mathrm{U} P \mathrm{O}=\mathrm{U}$ ', the leading $n-b y-n$ upper triangularpart ofA contains the upper triangularpart of the $m$ atrix $A$, and the strictly low er triangularpart of A is not referenced. If $U P L O=L '$, the leading $n-b y-n$ low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of $A$ is notreferenced.

O n exit, the block diagonalm atrix D and the m ul
tipliers used to obtain the factorU orl (see below for further details).

LDA (input)
The leading dim ension of the anray A. LD A >= $\max (1, \mathbb{N})$.

IPIV (output)
D etails of the interchanges and the block structure ofD. If $\mathbb{P} \mathbb{I V}(k)>0$, then row $s$ and colum ns $k$ and $\mathbb{P} \mathbb{I V}(k)$ were interchanged and $D(k, k)$ is a 1 -by-1 diagonalblock. IfUPLO $=U$ 'and $\mathbb{P} \mathbb{I V}(k)$ $=\mathbb{P} \mathbb{V}(k-1)<0$, then row $s$ and colmm ns $k-1$ and $-\mathbb{P} \mathbb{V}(k)$ w ere interchanged and $D(k-1 * k, k-1 *)$ is a 2 -by-2 diagonalblock. IfUPLO $=\mathbb{L}$ 'and $\mathbb{P} \mathbb{I V}(k)$
$=\mathbb{P} \mathbb{I}(k+1)<0$, then row sand colum ns $k+1$ and $-\mathbb{P} \mathbb{V}(k)$ w ere interchanged and $D(k, k+1, k \mathrm{k}+1)$ is a 2-by-2 diagonalblock.
$\mathbb{N F O}$ (output)
$=0$ : successfulexit
$<0:$ if $\mathbb{N} F O=-\mathrm{k}$, the k -th argum enthad an illegalvalue $>0:$ if $\mathbb{N} F O=k, D(k, k)$ is exactly zero. The factorization has been com pleted, but the block diagonalm atrix $D$ is exactly singular, and division by zero w illoccur if it is used to solve a system of equations.

## FURTHER DETAILS

## 1-96 -B ased on m odifications by J.Lew is, Boeing Com puter

## Services

Com pany
If U PLO $=\mathrm{U}$ ', then $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}$ ', where
$U=P(n) \star U(n) * \ldots * P(k) U(k) * \ldots$,
i.e., $U$ is a productof term $S P(k) * U(k)$, where $k$ decreases from $n$ to 1 in steps of 1 or2, and $D$ is ablock diagonal $m$ atrix $w$ ith 1 -by -1 and 2 -by-2 diagonalblocks $D(k) . P(k)$ is a perm utation $m$ atrix as defined by $\mathbb{P} \mathbb{V}(k)$, and $U(k)$ is a unituppertriangularm atrix, such that if the diagonal block $D(k)$ is of orders ( $s=1$ or 2 ), then

$$
\begin{aligned}
& \text { ( I v 0 ) k-s } \\
& \mathrm{U}(\mathrm{k})=(0 \mathrm{I} 0) \mathrm{s} \\
& \text { ( } 0 \text { O I ) } \mathrm{n}-\mathrm{k} \\
& \mathrm{k}-\mathrm{s} \mathrm{~s} \mathrm{n}-\mathrm{k}
\end{aligned}
$$

If $s=1, D(k)$ overw rites $A(k, k)$, and $v$ overw rites $A(1 k-$
$1, k)$. If $s=2$, the upper triangle ofD ( $k$ ) overw rites $A(k-$ $1, k-1)$, A $(k-1, k)$, and $A(k, k)$, and $v$ overw rites A ( 1 k- $2, k-$ $1 \mathrm{k})$.

If $\operatorname{PLO}=\mathrm{L}$ ', then $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L}$ ', where
$\mathrm{L}=\mathrm{P}(1) \star \mathrm{L}(1){ }^{*} \ldots * \mathrm{P}(k) \star \mathrm{L}(k)^{*} \ldots$,
i.e., $L$ is a product of term $S P(k) * L(k)$, where $k$ increases from 1 to $n$ in steps of 1 or 2, and $D$ is a block diagonal $m$ atrix $w$ ith 1 -by -1 and 2 -by-2 diagonalblocks $D(k) . P(k)$ is a perm utation $m$ atrix as defined by $\mathbb{P} \mathbb{V}(k)$, and $L(k)$ is a unit low ertriangularm atrix, such that if the diagonal
block D (k) is of orders ( $s=1$ or2), then

$$
\begin{aligned}
& \text { ( } \left.\begin{array}{llll}
\mathrm{I} & 0 & 0
\end{array}\right) \mathrm{k}-1 \\
& L(k)=\left(\begin{array}{lll}
0 & I & 0
\end{array}\right) s \\
& \text { ( } 0 \text { v I ) } n-k-s+1 \\
& \text { k-1 s n-k-s+1 }
\end{aligned}
$$

If $s=1, D(k)$ overw rites $A(k, k)$, and $v$ overw rites
$A(k+1 n, k)$. If $s=2$, the low er triangle ofD ( $k$ ) overw rites A $(k, k), A(k+1, k)$, and $A(k+1, k+1)$, and $v$ overw rites A $(k+2 m, k k+1)$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

ssytrd - reduce a realsym $m$ etric $m$ atrix A to real sym $m$ etric tridiagonal form T by an orthogonal sim ilarity transform ation

## SYNOPSIS

```
SUBROUT\mathbb{NE SSYTRD(UPLO,N,A,LDA,D,E,TAU,W ORK,LW ORK, INFO)}
CHARACTER * 1 UPLO
INTEGERN,LDA,LW ORK,INFO
REALA (LDA,*),D (*),E (*),TAU (*),WORK (*)
SU BROUT\mathbb{NE SSYTRD_64(UPLO,N,A,LDA,D,E,TAU,W ORK,LW ORK, NNFO)}
CHARACTER * 1 UPLO
\mathbb{NTEGER*8N,LDA,LW ORK, INFO}
REAL A (LDA,*),D (*),E (*),TAU (*),W ORK (*)
```

F95 INTERFACE
SU BROUTINE SYTRD (UPLO, N, A, [LDA],D,E,TAU, [W ORK ], [LW ORK ], [ $\mathbb{N F O}]$ )
CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R:: N, L D A, L W O R K, \mathbb{N} F O$
REAL,D $\mathbb{M}$ ENSION (:) ::D, E,TAU,W ORK
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A
SU BROUTINE SY TRD_64 (UPLO ,N , A, [LDA ],D ,E,TAU, [W ORK ], [LW ORK ],
[ $\mathbb{N} F O$ ])
CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER (8) :: N,LDA,LW ORK, $\mathbb{N}$ FO

REAL,D $\mathbb{I}$ ENSION (:) :: D, E,TAU,WORK
REAL,D IM ENSION (:,:) ::A

## C INTERFACE

\#include <sunperfh>
void ssytrd (charuplo, intn, float *a, intlda, float *d, float*e, float *tau, int *info);
void ssytrd_64 (charuplo, long n, float*a, long lda, float *d, float *e, float *tau, long *info);

## PURPOSE

ssytrd reduces a realsym $m$ etric $m$ atrix A to real sym $m$ etric tridiagonal form T by an orthogonal sim ilarity transform ation: $Q * * T * A * Q=T$.

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ ': U pper triangle ofA is stored;
$=\mathbb{L}$ ': Low ertriangle of A is stored.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input) On entry, the symm etric matrix A. IfUPLO = $\mathrm{U}^{\prime}$, the leading N boy N uppertriangularpartofA contains the uppertriangularpart of the $m$ atrix $A$, and the strictly low er triangularpart of $A$ is not referenced. If $\mathrm{UPLO}=\mathrm{L}$ ', the leading N -by -N low er triangularpant of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangular partofA is notreferenced. On exit, if UPLO = U ', the diagonal and firstsuperdiagonal of A are overw ritten by the comesponding elem ents of the tridiagonalm atrix $T$, and the ele$m$ ents above the first superdiagonal, $w$ ith the anray TAU, represent the orthogonalm atrix $Q$ as a product of elem entary reflectors; if U PLO $=\mathrm{L}$ ', the diagonal and firstsubdiagonal of A are overw ritten by the corresponding elem ents of the tridiagonalm atrix $T$, and the elem ents below the first subdiagonal, w ith the amay TA U, represent the orthogonalm atrix $Q$ as a productofelem entary reflectors. See FurtherD etails.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, N)$.

D (output)
The diagonalelem ents of the tridiagonalm atrix T :
D $(i)=A(i, i)$.

E (output)
The off-diagonal elem ents of the tridiagonal $m$ atrix $T: E(i)=A(i, i+1)$ if $U P L O=U^{\prime}, E(i)=$ A $(\mathbf{i}+1, i)$ if $\mathrm{UPLO}=\mathrm{L}$.

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array $W$ ORK. LW ORK >=1. For optim um perform ance $L W$ ORK $>=N * N B$, where $N B$ is the optim alblocksize.

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
$\mathbb{I N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{N}$ FO $=-$ i, the i-th argum ent had an illegalvalue

## FURTHER DETAILS

If U PLO $=\mathrm{U}$ ', the m atrix $Q$ is represented as a product of elem entary reflectors

$$
Q=H(n-1) \ldots H(2) H(1) .
$$

Each $H$ (i) has the form

$$
H(i)=I-\tan * V^{*} v^{\prime}
$$

where tau is a real scalar, and $v$ is a realvectorw ith $v(i+1 m)=0$ and $v(i)=1 ; v(1: i-1)$ is stored on exitin

A ( $1: i-1, i+1)$, and tau in TAU (i).
IfU PLO $=\mathrm{L}$ ', the m atrix $Q$ is represented as a product of elem entary reflectors

$$
Q=H(1) H(2) \ldots H(n-1) .
$$

Each $H$ (i) has the form
H (i) $=I-\tan { }^{*} V^{*} V^{\prime}$
where tau is a real scalar, and $v$ is a realvectorw ith $\mathrm{v}(1: i)=0$ and $\mathrm{v}(i+1)=1 ; \mathrm{v}(i+2 \mathrm{n})$ is stored on exit in A (i+2m,i), and tau in TAU (i).
The contents of A on exitare illustrated by the follow ing exam plesw ith $\mathrm{n}=5$ :

```
ifUPLO = U ': ifUPLO = L :
```

    ( d e v2 v3 v4 ) ( d
    )
( d e v3 v4 ) ( e d
)
( d e v4 ) (v1 e d
)
( d e ) ( v1 v2 e d
)
( d ) (v1 v2 v3 e d
)
where $d$ and e denote diagonal and off-diagonal elem ents of $T$, and videnotes an elem entof the vectordefining $H$ (i).

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

ssytrf-com pute the factorization of a real symmetric
$m$ atrix A using the Bunch $-K$ aufn an diagonalpivoting $m$ ethod

## SYNOPSIS


CHARACTER * 1 UPLO
$\mathbb{N}$ TEGER $N, L D A, L D W$ ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R \mathbb{P} \mathbb{I} O T(*)$
REALA (LDA, $\left.{ }^{*}\right), \mathrm{W} O R K(\star)$
SU BROUTINE SSYTRF_64 (UPLO,N,A,LDA, $\mathbb{P} \mathbb{I} O T, W$ ORK,LDW ORK, $\mathbb{N} F O$ )
CHARACTER * 1 UPLO
$\mathbb{N} T E G E R * 8 N, L D A, L D W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{I V} O T\left({ }^{*}\right)$
REALA (LDA, $\left.{ }^{\star}\right), \mathrm{W} O R K(\star)$

## F95 INTERFACE

SU BROUTINE SY TRF (UPLO ,N,A, [LDA], $\mathbb{P} \mathbb{I V O T}, \mathbb{W}$ ORK ], [LDW ORK ], [ $\mathbb{N F O}])$
CHARACTER (LEN=1)::UPLO
$\mathbb{N} T E G E R:: N, L D A, L D W O R K, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V O T}$
REAL,D IM ENSION (:) ::W ORK
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A
SU BROUTINE SYTRF_64 (UPLO,N,A, [LDA], $\mathbb{P} \mathbb{V} O T,\left[\begin{array}{l}\text { W ORK ], [LDW ORK ], }\end{array}\right.$ [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER (8) ::N,LDA,LDW ORK, $\mathbb{N}$ FO
$\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T$
REAL,D IM ENSION (:) ::W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) ::A

## C INTERFACE

\#include <sunperfh>
void ssytrf(char uplo, int $n$, float *a, int lda, int *ịívot, int*info);
void ssytrf_64 (charuplo, long n, float *a, long lda, long
*ịívot, long *info);

## PURPOSE

ssytrf com putes the factorization of a realsym $m$ etric $m$ atrix A using the Bunch $-K$ aufm an diagonalpivoting $m$ ethod. The form of the factorization is

$$
\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T} \text { or } \mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}
$$

where $U$ (orL) is a productof perm utation and unit upper (low er) triangular $m$ atrices, and $D$ is sym $m$ etric and block diagonalw ith 1 -by-1 and 2 -by-2 diagonalblocks.

This is the blocked version of the algorithm, calling Level 3 BLAS.

## ARGUMENTS

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.
A (input/output)
O n entry, the sym m etric m atrix A. If UPLO = U', the leading N -by -N uppertriangularpartofA contains the upper triangular part of the $m$ atrix $A$, and the strictly low er triangularpartofA is not referenced. If $\mathrm{UPLO}=\mathrm{L}$ ', the leading N -by -N low er triangularpart of $A$ contains the low ertriangularpartof the m atrix A, and the strictly upper triangularpartofA is not referenced.

On exit, the block diagonalm atrix D and the multipliers used to obtain the factorU orL (see below for further details).

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, N)$.

## IPIVOT (output)

D etails of the interchanges and the block structure of D. If $\mathbb{P I V O T}(k)>0$, then row sand columnsk and $\mathbb{P I V O T}(k)$ were interchanged and $D(k, k)$ is a $1-b y-1$ diagonalblock. If $U P L O=U^{\prime}$ and $\mathbb{P} \mathbb{V} O T(k)=\mathbb{P} \mathbb{V} O T(k-1)<0$, then row $s$ and colum ns $k-1$ and - $\mathbb{P I V O T}(k)$ were interchanged and D ( $k-1 * k, k-1 k)$ is a $2-b y-2$ diagonal block. If UPLO $=\mathrm{L}$ 'and $\mathbb{P I V O T}(k)=\mathbb{P} \mathbb{I V O T}(k+1)<0$, then row sand colum ns $k+1$ and $-\mathbb{P} \mathbb{V} O T(k)$ were interchanged and $D(k k+1, k k+1)$ is a $2-b y-2$ diagonal block.

W ORK (w orkspace)
On exit, if $\mathbb{N F F O}=0, \mathrm{~W}$ ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length ofW ORK. LDW ORK >=1. Forbestperfor$m$ ance LDW ORK >=N *NB, where NB is the block size retumed by $\amalg A E N V$.

IfLDW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
$\mathbb{N} F O$ (output)
= 0 : successfinlexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the $i$-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N F O}=\mathrm{i}, \mathrm{D}(\mathrm{i}, \mathrm{i})$ is exactly zero. The factorization has been com pleted, but the block diagonalm atrix $D$ is exactly singular, and division by zero w illoccur if it is used to solve a system of equations.

## FURTHER DETAILS

If $\mathrm{ULO}=\mathrm{U}$ ', then $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}$ ', where
$U=P(n) \star U(n)^{\star} \ldots{ }^{\star} P(k) U(k)^{\star} \ldots$,
ie., $U$ is a product ofterm $\operatorname{sP}(k) * U(k)$, where $k$ decreases from $n$ to 1 in steps of 1 or 2 , and $D$ is a block diagonal $m$ atrix w ith 1 -by-1 and 2 -by-2 diagonalblocks $D(k) . P(k)$ is a perm utation $m$ atrix as defined by $\mathbb{P} \mathbb{I V O T}(\mathrm{k})$, and $\mathrm{U}(\mathrm{k})$ is a unituppertriangularm atrix, such that if the diagonal block D (k) is of orders ( $s=1$ or 2 ), then

$$
\begin{aligned}
& \text { ( I v 0 ) k-s } \\
& U(k)=\left(\begin{array}{lll}
0 & I
\end{array}\right) s \\
& \text { ( } 000 \text { I ) n-k } \\
& \mathrm{k}-\mathrm{s} \text { s n-k }
\end{aligned}
$$

If $s=1, D(k)$ overw rites $A(k, k)$, and $v$ overw rites $A(1 k-$ $1, k$ ). If $s=2$, the upper triangle ofD $(k)$ overw rites $A(k-$ $1, k-1), A(k-1, k)$, and $A(k, k)$, and $V$ overw rites $A(1 k-2, k-$ $1 \mathrm{k})$.

If $\mathrm{UPLO}=\mathrm{L}$ ', then $A=\mathrm{L} * \mathrm{D} * \mathrm{~L}$ ', where
$L=P(1) \star L(1)^{\star} \ldots * P(k) \star L(k) * \ldots$
ie., $L$ is a productofterm $s P(k) * L(k)$, where $k$ increases
from 1 to n in steps of 1 or 2 , and D is a block diagonal $m$ atrix $w$ th 1 -by -1 and 2 -by-2 diagonalblocks $D(k) . P(k)$ is a perm utation $m$ atrix as defined by $\mathbb{P} \mathbb{I V O T}(k)$, and $L(k)$ is a unitlow ertriangularm atrix, such that if the diagonal
block $D(k)$ is of orders ( $s=1$ or 2 ), then

$$
\begin{aligned}
& \text { ( } \left.\begin{array}{llll}
I & 0 & 0
\end{array}\right) k-1 \\
& L(k)=\left(\begin{array}{lll}
0 & I & 0
\end{array}\right) s \\
& \text { ( } 0 \mathrm{~V} \text { I ) } \mathrm{n}-\mathrm{k}-\mathrm{s}+1 \\
& \mathrm{k}-1 \text { s } \mathrm{n}-\mathrm{k}-\mathrm{s}+1
\end{aligned}
$$

If $s=1, D(k)$ overw rites $A(k, k)$, and $v$ overw rites $A(k+1 m, k)$. If $s=2$, the low ertriangle ofD $(k)$ overw rites $A(k, k), A(k+1, k)$, and $A(k+1, k+1)$, and $v$ overw rites A $(k+2 m, k: k+1)$.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssytri-com pute the inverse of a realsym $m$ etric indefinite $m$ atrix $A$ using the factorization $A=U * D * U * *$ or $A=$ L*D *L**T com puted by SSY TRF

## SYNOPSIS



```
CHARACTER * 1 UPLO
NNTEGER N,LDA, INFO
INTEGER \mathbb{PIVOT (*)}
REALA (LDA,*),W ORK (*)
```



```
CHARACTER * 1 UPLO
\mathbb{N TEGER*8 N,LDA,}\mathbb{N FO}
\mathbb{NTEGER*8 P\mathbb{IVOT (*)}}\mathbf{(})
REALA (LDA,*),WORK(*)
```

F95 INTERFACE
SU BROUTINE SYTRI(UPLO,N,A, [LDA ], $\mathbb{P} \mathbb{I} O T,[\mathbb{O} O R K],[\mathbb{N F O}])$
CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER :: $\mathrm{N}, \mathrm{LDA}, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T$
REAL,D IM ENSION (:) ::W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) ::A
SU BROUTINE SY TRI_64 (UPLO,N,A, [LDA ], $\mathbb{P} \mathbb{I V O T},[\mathbb{W}$ ORK ], [ $\mathbb{N F O}$ ])
CHARACTER (LEN=1)::UPLO
$\mathbb{N}$ TEGER (8) :: $\mathrm{N}, \mathrm{LD} A, \mathbb{N}$ FO
$\mathbb{N} \operatorname{TEGER}$ (8), D $\mathbb{M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT}$
REAL,D $\mathbb{I}$ ENSION (:) ::W ORK
REAL,D $\mathbb{I}$ ENSION (:,:) ::A

## C INTERFACE

\#include <sunperfh>
void ssytri(char uplo, int $n$, float *a, int lda, int *ịivot, int *info);
void ssytri_ 64 (char uplo, long n, float*a, long lda, long *ipivot, long *info);

## PURPOSE

ssytricom putes the inverse of a real sym m etric indefinite $m$ atrix $A$ using the factorization $A=U * D * U * * T$ orA $=$ L *D *L **T com puted by SSY TRF.

## ARGUMENTS

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
$=\mathrm{U}$ ': Uppertriangular, form is $\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}$;
$=\mathrm{L}^{\prime}$ : Low ertriangular, form is $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input/output)
O n entry, the block diagonalm atrix D and the m ultipliers used to obtain the factorU orL as com puted by SSY TRF.

On exit, if $\mathbb{N F O}=0$, the (sym m etric) inverse of the original m atrix. If $\mathrm{UPLO}=\mathrm{U}$ ', the upper triangularpart of the inverse is form ed and the part ofA below the diagonal is notreferenced; if $\mathrm{UPLO}=$ 'L' the low er triangular part of the inverse is formed and the partofA above the diagonal is not referenced.

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, N)$.
$\mathbb{P I V O T}$ (input)
D etails of the interchanges and the block structure ofD as determ ined by SSY TRF.

W ORK (w orkspace)
dim ension (N)
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N} F O=-$ i, the $i$-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N} F O=i, D(i, i)=0$; the $m$ atrix is singular and its inverse could notbe com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

ssytrs-solve a system of linearequations $A * X=B$ with a
real sym $m$ etric $m$ atrix $A$ using the factorization $A=U * D * U * * T$
orA $=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ com puted by SSY TRF

## SYNOPSIS

```
SUBROUT\mathbb{NE SSY TRS (UPLO,N,NRHS,A,LDA, PIVOT,B,LDB,INFO)}
CHARACTER * 1 UPLO
INTEGERN,NRHS,LDA,LDB,INFO
INTEGER \mathbb{PIVOT (*)}
REALA (LDA,*),B (LDB,*)
```



```
CHARACTER * 1 UPLO
INTEGER*8N,NRHS,LDA,LDB,INFO
\mathbb{NTEGER*8 P\mathbb{IVOT (*)}}\mathbf{(})
REALA (LDA,*),B (LDB,*)
```

F95 INTERFACE
SU BROUTINE SYTRS (UPLO,N,NRHS,A, [LDA], $\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LD} B],[\mathbb{N} F O])$
CHARACTER (LEN=1) ::UPLO
$\mathbb{N}$ TEGER ::N,NRHS,LDA,LDB, $\mathbb{N} F O$
$\mathbb{N}$ TEGER,D $\mathbb{I M}$ ENSION (:) :: $\mathbb{P} \mathbb{V} O T$
REAL,D $\mathbb{M}$ ENSION (:,:) ::A, B
SU BROUTINE SYTRS_64 (UPLO,N,NRHS,A, [LDA ], $\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}],[\mathbb{N} F O])$
CHARACTER (LEN=1) ::UPLO
$\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D B, \mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENS} \operatorname{ION}(:):: \mathbb{P} \mathbb{I} O T$
REAL,D $\mathbb{M}$ ENSION (:,:) ::A , B

## C INTERFACE

\#include <sunperfh>
void ssytrs (char uplo, intn, intnrhs, float*a, int lda, int *ipivot, float *b, int ldl , int *info);
void ssytrs_64 (charuplo, long n, long nihs, float *a, long lda, long *ipivot, float *b, long ldb, long *info);

## PURPOSE

ssytrs solves a system of linear equations $A * X=B$ with a realsym $m$ etric $m$ atrix $A$ using the factorization $A=U * D * U * * T$ orA $=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$ com puted by SSY TRF.

## ARGUMENTS

UPLO (input)
Specifies w hether the details of the factorization are stored as an upper or low er triangularm atrix.
$=U$ : U ppertriangular, form is $A=U * D * U * * T$;
$=\mathrm{L}^{\prime}:$ Low ertriangular, form is $\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}$.

N (input) The order of them atrix A. N >=0.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS $>=0$.

A (input) The block diagonalm atrix D and the m ultipliers
used to obtain the factorU orL as com puted by SSY TRF.

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, N)$.

IPIVOT (input)
D etails of the interchanges and the block structure ofD as determ ined by SSY TRF.

B (input/output)
O $n$ entry, the righthand side m atrix B. On exit,
the solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the aray B. LD B $>=$ $\max (1, N)$.

INFO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N} F O=-i$, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

stboon -estim ate the reciprocal of the condition num ber of a triangular band $m$ atrix $A$, in eitherthe 1 -norm orthe infinity-norm

## SYNOPSIS

```
SUBROUT\mathbb{NE STBCON NORM,UPLO,DIAG,N,KD,A,LDA,RCOND,W ORK,}
    W ORK2,\mathbb{NFO)}
CHARACTER * 1 NORM,UPLO,DIAG
INTEGER N,KD,LDA,INFO
\mathbb{NTEGERWORK2(*)}
REALRCOND
REALA (LDA,*),W ORK (*)
SUBROUT\mathbb{NE STBCON_64 NORM,UPLO,DIAG,N,KD,A,LDA,RCOND,W ORK,}
        WORK2, \mathbb{NFO)}
CHARACTER * 1NORM,UPLO,DIAG
\mathbb{NTEGER*8N,KD,LDA,}\mathbb{NFO}
INTEGER*8 W ORK 2 (*)
REALRCOND
REALA (LDA,*),W ORK (*)
```


## F95 INTERFACE

```
SU BROUTINE TBCON \(\mathbb{N} O R M, U P L O, D \mathbb{I} G, N, K D, A,[L D A], R C O N D,[W O R K]\), [W ORK 2], [ \(\mathbb{N} F O\) ])
CHARACTER (LEN=1) ::NORM,UPLO,D IAG
\(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathrm{KD}, L D A, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) ::W ORK2
REAL ::RCOND
```

SUBROUTINE TBCON_64 $\mathbb{N} O R M, U P L O, D I A G, N, K D, A,[L D A], R C O N D$, [W ORK], [WORK2], [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1) ::NORM, UPLO, DIAG
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{KD}, \mathrm{LD} A, \mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M}$ ENSION (:) ::W ORK2
REAL: RCOND
REAL,D $\mathbb{M}$ ENSION (:) ::W ORK
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A

## C INTERFACE

\#include < sunperfh>
void stbocon (charnorm , charuplo, chardiag, intn, int kd, float *a, int lda, float *rcond, int *info);
void stbcon_64 (charnorm , charuplo, chardiag, long n, long kd, float *a, long lda, float *roond, long *info);

## PURPOSE

stocon estim ates the reciprocal of the condition num ber of a triangular band $m$ atrix $A$, in either the 1 -norm orthe infinity-norm.

The norm ofA is com puted and an estim ate is obtained for norm (inv (A )), then the reciprocal of the condition num ber is com puted as
$\operatorname{RCOND}=1 /(\operatorname{norm}(A) * \operatorname{norm}(\operatorname{inv}(A)))$.

## ARGUMENTS

N O RM (input)
Specifies w hether the 1 -norm condition num ber or the infinity-norm condition num ber is required:
= $\mathrm{I}^{\prime}$ or $\mathrm{O}^{\prime}$ : 1-norm;
$=I^{\prime}: \quad$ Infinity-norm .

UPLO (input)
$=\mathrm{U}^{\prime}: \mathrm{A}$ is uppertriangular;
= L ': A is low er triangular.

D IA G (input)
$=\mathrm{N}^{\prime}: \mathrm{A}$ is non-unittriangular;
$=\mathrm{U}: \mathrm{A}$ is unittriangular.

N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.
KD (input)
The num berof superdiagonals or subdiagonals of the triangularband $m$ atrix $A . K D>=0$.

A (input) The upper or low er triangular band matrix A, stored in the firstkd+1 row sof the array. The $j$ th column ofA is stored in the $j$ th column of the array A as follow s: if UPLO = U',A (kd+1+i$j, j)=A(i, j)$ for $\max (1, j k d)<=i<=j$; UPLO $=$ L',A $(1+i-j)=A(i, j)$ for $j=i<=m$ in $(n, j+k d)$. IfD $\mathbb{I A} G=U$ ', the diagonalelem ents of $A$ are not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the anay A. LD A >= K D +1.

RCOND (output)
The reciprocal of the condition number of the $m$ atrix $A$, computed as RCOND $=1 /($ norm (A) * norm (inv (A))).

W ORK (w orkspace)
dim ension $(3 * N)$

W ORK 2 (w orkspace)
dim ension $(\mathbb{N})$
$\mathbb{N}$ FO (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-$ i, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

stom $v$-perform one of the $m$ atrix-vector operations $x:=$ $A * x$, or $x:=A * x$

## SYNOPSIS

```
SUBROUT\mathbb{NE STBMV (UPLO,TRANSA,D IAG,N,K,A,LDA,Y, INCY)}
CHARACTER * 1 UPLO,TRANSA,DIAG
\mathbb{NTEGERN,K,LDA,INCY}
REALA (LDA,*),Y(*)
SUBROUT\mathbb{NE STBM V_64(UPLO,TRANSA,D IAG,N,K,A,LDA,Y, INCY)}
CHARACTER * 1 UPLO,TRANSA,D IAG
\mathbb{NTEGER*8N,K,LDA,INCY}
REALA (LDA,*),Y (*)
F95 INTERFACE
```



```
    CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
    \mathbb{NTEGER ::N,K,LDA,INCY}
    REAL,D IM ENSION (:) ::Y
    REAL,D IM ENSION (:,:) ::A
    SU BROUT\mathbb{NE TBM V_64 (UPLO,[TRANSA ],D IAG, N ],K,A,[LDA ],Y,}
        [\mathbb{NCY])}
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
    \mathbb{NTEGER (8)::N,K,LDA,\mathbb{NCY}}\mathbf{N}=\mp@code{M}
    REAL,D IM ENSION (:) ::Y
    REAL,D IM ENSION (:,:) ::A
```


## C INTERFACE

\#include <sunperfh>
void stom v (charuplo, char transa, chardiag, intn, int $k$, float *a, int lda, float *y, int incy);
void stbm v_64 (charuplo, chartransa, char diag, long n, long $k$, float *a, long lda, float * $y$, long incy);

## PURPOSE

stom $v$ perform s one of them atrix-vector operations $x: A * x$, or $x:=A{ }^{*} x$, where $x$ is an $n$ elem entvectorand $A$ is an $n$ by $n$ unit, or non-unit, upper or low er triangular band $m$ atrix, with ( $k+1$ ) diagonals.

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies w hether the $m$ atrix is an upper or low er triangularm atrix as follow s :
$\mathrm{UPLO}=\mathrm{U}$ 'or G ' $A$ is an upper triangular $m$ atrix.

UPLO = L' or I' A is a lower triangular $m$ atrix.

U nchanged on exit.
TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA $=N^{\prime}$ 'or $h^{\prime} \mathrm{x}:=\mathrm{A} * \mathrm{x}$.

TRANSA $=$ T'ort' $x=A$ * $x$.

TRANSA = C'ort' $\mathrm{x}:=\mathrm{A}$ * x .
U nchanged on exit.

TRANSA is defaulted to N 'forF $95 \mathbb{I N}$ TERFACE.
D IA G (input)
On entry, D IA G specifies w hether or notA is unit
triangular as follow s:

D IA G = U'or 4 ' A is assum ed to be unit triangular.

D $\mathbb{A} G=N$ 'or $h$ ' $A$ is notassum ed to be unit triangular.

U nchanged on exit.
$N$ (input)
On entry, N specifies the order of the m atrix A . $\mathrm{N}>=0$. U nchanged on exit.

K (input)
On entry with UPLO $=$ U 'or L', K specifies the num ber of super-diagonals of them atrix $A$. On entry w ith UPLO = L' or I', K specifies the num ber of sub-diagonals of the $m$ atrix $A . K>=0$. U nchanged on exit.

A (input)
Before entry w th UPLO = U 'or G ', the leading ( $k+1$ ) by $n$ part of the array A $m$ ustcontain the upper triangularband partof the $m$ atrix of coefficients, supplied colum $n$ by colum $n$, w ith the leading diagonal of the $m$ atrix in row ( $k+1$ ) of the array, the firstsuper-diagonal starting at position 2 in row $k$, and so on. The top leftk by $k$ triangle of the array A is not referenced. The follow ing program segm entw ill transfer an upper triangular band $m$ atrix from conventional full $m$ atrix storage to band storage:

$$
\begin{aligned}
& \text { DO } 20, J=1, N \\
& \text { M }=\mathrm{K}+1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{M} A X(1, \mathrm{~J}-\mathrm{K}), \mathrm{J} \\
& \text { A }(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \text { CONTINUE } \\
& 20 \text { CONTINUE }
\end{aligned}
$$

Before entry w ith UPLO = L 'or 1', the leading (
$k+1$ ) by $n$ part of the amay A $m$ ust contain the low er triangularband part of the $m$ atrix of coefficients, supplied colum n by colum n, w th the leading diagonal of the m atrix in row 1 of the array, the firstsub-diagonal starting atposition 1 in row 2 , and so on. The bottom right $k$ by $k$ triangle of the amay A is not referenced. The follow ing program segm entw ill transfer a low er
triangular band $m$ atrix from conventional full $m$ atrix storage to band storage:

$$
\begin{aligned}
& \text { DO } 20, J=1, N \\
& \mathrm{M}=1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{J}, \mathrm{M} \mathbb{N}(\mathrm{~N}, \mathrm{~J}+\mathrm{K}) \\
& \mathrm{A}(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \operatorname{CONTINUE} \\
& 20 \mathrm{CONTINUE}
\end{aligned}
$$

$N$ ote thatw hen D $\mathbb{A} G=U$ 'or L 'the elem ents of the array A comesponding to the diagonalelem ents of the $m$ atrix are not referenced, but are assum ed to be unity. U nchanged on exit.
LDA (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD A $>=$ ( $\mathrm{k}+1$ ). U nchanged on exit.

Y (input/output)
$(1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C Y))$. Before entry, the increm ented array $Y$ must contain the $n$ elem ent vectorx. On exit, $Y$ is overw rilten $w$ th the tranform ed vector $x$.
$\mathbb{N C Y}$ (input)
O n entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of. $\mathbb{N} C Y$ <> 0 . U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

stbrfs - provide emorbounds and backw ard enror estim ates for the solution to a system of linear equations w ith a triangularband coefficientm atrix

## SYNOPSIS

```
SUBROUT\mathbb{NE STBRFS (UPLO,TRANSA,D IAG,N,KD,NRHS,A,LDA,B,LDB,}
    X,LDX,FERR,BERR,W ORK,W ORK2,\mathbb{NFO)}
```

CHARACTER * 1 UPLO, TRANSA, DIAG
$\mathbb{N}$ TEGER N, KD,NRHS,LDA,LDB,LDX, $\mathbb{N} F O$
$\mathbb{I N}$ TEGER W ORK 2 (*)

SU BROUTINE STBRFS_64 (UPLO, TRANSA, D $\mathbb{A} G, N, K D, N R H S, A, L D A, B$,
LD $B, X, L D X, F E R R, B E R R, W$ ORK, W ORK 2, $\mathbb{N} F O$ )
CHARACTER * 1 UPLO, TRANSA, D IA G
$\mathbb{N} T E G E R * 8 N, K D, N R H S, L D A, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R * 8$ W ORK 2 ( ${ }^{*}$ )


## F95 INTERFACE

SU BROUTINE TBRFS (UPLO, [TRANSA],D $\mathbb{I A G}, N, K D, N R H S, A,[L D A], B$, [LDB], X, [LDX ],FERR,BERR, [W ORK], [W ORK 2], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
$\mathbb{N} T E G E R:: N, K D, N R H S, L D A, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) ::W ORK2
REAL,D $\mathbb{I M}$ ENSION (:) ::FERR,BERR,W ORK
REAL,D $\mathbb{I M}$ ENSION (: : : : : A, B, X

SU BROUTINE TBRFS_64 (UPLO, [TRANSA],D $\mathbb{A} G, N, K D, N R H S, A,[L D A]$,


CHARACTER (LEN=1) ::UPLO, TRANSA,D IA G
$\mathbb{N}$ TEGER (8) ::N,KD,NRHS,LDA,LDB,LDX, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I M}$ ENSION (:) ::W ORK2
REAL,D IM ENSION (:) ::FERR,BERR,W ORK
REAL,D $\mathbb{I M}$ ENSION (: : : : : A $\mathrm{A}, \mathrm{B}, \mathrm{X}$

## C INTERFACE

\#include < sunperfh>
void stbrfs (charuplo, chartransa, chardiag, int $n$, int kd, int nihs, float *a, intlda, float*b, int ldb, float * $x$, int ldx , float * ferrs, float *berr, int*info);
void stbrfs_64 (charuplo, chartransa, char diag, long n, long kd, long nihs, float *a, long lda, float *b, long ldo, float *x, long ldx, float *ferr, float *berr, long *info);

## PURPOSE

stbrfs provides emorbounds and backw ard emor estim ates forthe solution to a system of linear equations $w$ th a triangularband coefficientm atrix.

The solution $m$ atrix X m ustbe com puted by STBTRS or some other $m$ eans before entering this routine. STBRFS does not do iterative refinem entbecause doing so cannotim prove the backw ard error.

## ARGUMENTS

## UPLO (input)

$=\mathrm{U}$ ': A is uppertriangular;
= L ': A is low ertriangular.

TRANSA (input)
Specifies the form of the system of equations:
$=\mathrm{N}: \mathrm{A} * \mathrm{X}=\mathrm{B} \quad$ N $\circ$ transpose)
$=T$ ': A ** $T$ * $\mathrm{X}=\mathrm{B}$ ( T ranspose)
$=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad$ (C onjugate transpose $=\mathrm{T}$ ran-
spose)

TRANSA is defaulted to N 'forF95 $\mathbb{I N}$ TERFACE.

D IA G (input)
$=N$ : A is non-unit triangular;
$=U: A$ is unittriangular.

N (input) The order of the m atrix A. $\mathrm{N}>=0$.
KD (input)
The num berof superdiagonals or subdiagonals of the triangularband $m$ atrix $A . K D>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the $m$ atrices $B$ and $X$. NRH $S>=0$.
A (input) The upper or low er triangular band $m$ atrix $A$, stored in the firstkd+1 row sof the amay. The $j$ th column of A is stored in the $j$ th column of the anay A as follow s: if UPLO = U',A (kd+1+ij기) $=A(i, j)$ for $\max (1, j \mathrm{kd})<=i<=j$ if UPLO $=$ L', A $(1+i-j\rangle)=A(i, 7)$ for $j<=i<=m$ in $(n, j+k d)$. IfD $I A G=U$ ', the diagonalelem ents of $A$ are not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the array A. LDA >= K D +1 .
$B$ (input) The righthand side $m$ atrix $B$.
LD B (input)
The leading dim ension of the aray $B$. LD B >= $\max (1, \mathbb{N})$.

X (input) The solution $m$ atrix X .
LD X (input)
The leading dim ension of the array X . LD $\mathrm{X}>=$ $\max (1, \mathbb{N})$.

## FERR (output)

The estim ated forw ard enrorbound for each solution vector $X(\mathcal{)}$ ) the $j$ th colum $n$ of the solution $m$ atrix X). If XTRUE is the true solution comesponding to $X(\mathcal{O}), \operatorname{FERR}(\mathcal{O})$ is an estim ated upperbound for the $m$ agnitude of the largest ele$m$ ent in ( $X(\mathcal{H})$-XTRUE) divided by the $m$ agnitude of the largestelem ent in $X(\mathcal{J})$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

## BERR (output)

The com ponentw ise relative backw ard emor of each
solution vectorX (i) (ie., the sm allest relative
change in any elem entofA orB thatm akes X ( $\mathcal{I}$ ) an exact.solution).

W ORK (w orkspace)
dim ension ( $3 * N$ )
W ORK 2 (w orkspace)
dim ension $\mathbb{N}$ )
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N ~ F O ~}=-i$, the i-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

stbsv - solve one of the system sofequations $A$ * $\mathrm{x}=\mathrm{b}$, or A * $\mathrm{x}=\mathrm{b}$

## SYNOPSIS

```
SUBROUT\mathbb{NE STBSV (UPLO,TRANSA,D IAG,N,K,A,LDA,Y,\mathbb{NCY)}}\mathbf{N}\mathrm{ (T)}
CHARACTER * 1UPLO,TRANSA,DIAG
\mathbb{NTEGERN,K,LDA,INCY}
REAL A (LDA,*),Y (*)
SUBROUT\mathbb{NE STBSV_64(UPLO,TRANSA,D IAG,N,K,A,LDA,Y, INCY)}
CHARACTER * 1UPLO,TRANSA,DIAG
\mathbb{NTEGER*8N,K,LDA,INCY}
REALA (LDA,*),Y (*)
```

F95 INTERFACE
SU BROUTINE TBSV (UPLO, [TRANSA],D $\mathbb{A} G, \mathbb{N}], K, A,[L D A], Y,[\mathbb{N C Y}])$
CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
$\mathbb{N} T E G E R:: N, K, L D A, \mathbb{N C Y}$
REAL,D $\mathbb{M}$ ENSION (:) ::Y
REAL,D IM ENSION (:,:) ::A
SU BROUTINE TBSV_64 (UPLO, [TRANSA ],D $\mathbb{I} G, \mathbb{N}], K, A,[L D A], Y$,
[ $\mathbb{N} C Y$ ])
CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
$\mathbb{N}$ TEGER (8) ::N, K,LDA $\mathbb{I N C Y}$
REAL,D $\mathbb{I M}$ ENSION (:) ::Y
REAL,D IM ENSION (: : : : : A

## C INTERFACE

\#include < sunperfh>
void stbsv (charuplo, char transa, chardiag, intn, int $k$, float *a, int lla, float *y, int incy);
void stbsv_64 (charuplo, chartransa, char diag, long n, long $k$, float *a, long lda, float * $y$, long incy);

## PURPOSE

stbsv solves one of the system sof equations $A * x=b$, or $A{ }^{*} x=b, w$ here $b$ and $x$ are $n$ elem entvectors and $A$ is an $n$ by $n$ unit, or non-unit, upper or low er triangular band $m$ atrix, $w$ ith ( $k+1$ ) diagonals.

N o test forsingularity or near-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies whetherthem atrix is an upper or low er triangularm atrix as follow s:

UPLO = U'or L ' A is an upper triangular $m$ atrix.

UPLO = L' or I' A is a lower triangular $m$ atrix.

U nchanged on exit.
TRANSA (input)
On entry, TRANSA specifies the equations to be solved as follow s:

TRANSA $=N$ 'or $h^{\prime} A * x=b$.
TRANSA $=\mathrm{T}^{\prime}$ or $\mathrm{t}^{\prime} \mathrm{A}{ }^{*} \mathrm{x}=\mathrm{b}$.

TRANSA = C'ort' A*x=b.
U nchanged on exit.

D IA G (input)
O n entry, D IA G specifies w hether ornotA is unit triangular as follow s:

D $\mathbb{A G G}=\mathrm{U}$ 'or $\mathrm{L}^{\prime} \mathrm{A}$ is assum ed to be unit triangular.

D $\mathbb{A G}=N$ 'or $h$ ' $A$ is notassumed to be unit triangular.

U nchanged on exit.

N (input)
O $n$ entry, $N$ specifies the order of the $m$ atrix A. $\mathrm{N}>=0$. U nchanged on exit.

K (input)
On entry with UPLO $=$ U 'or L', $K$ specifies the num ber of super-diagonals of them atrix $A$. On entry w ith UPLO = L' or I', K specifies the num ber of sub-diagonals of the $m$ atrix $A . K>=0$. U nchanged on exit.

A (input)
Before entry w th UPLO $=U$ 'or L ', the leading ( $k+1$ ) by $n$ part of the array A m ust contain the upper triangularband part of the $m$ atrix of coefficients, supplied colum $n$ by colum $n$, w ith the leading diagonal of the $m$ atrix in row ( $k+1$ ) of the aray, the firstsuper-diagonalstarting at position 2 in row $k$, and so on. The top leftk by $k$ triangle of the array A is not referenced. The follow ing program segm entw ill transfer an upper triangular band $m$ atrix from conventional full $m$ atrix storage to band storage:

$$
\begin{aligned}
& \text { DO } 20, J=1, N \\
& \text { M }=\mathrm{K}+1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{MAX}(1, \mathrm{~J}-\mathrm{K}), \mathrm{J} \\
& \mathrm{~A}(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \text { CONTINE } \\
& 20 \text { CONTINUE }
\end{aligned}
$$

Before entry w ith UPLO = L 'or 1', the leading ( $k+1$ ) by $n$ part of the array A must contain the low er triangularband part of the $m$ atrix of coefficients, supplied colum $n$ by colum n, w ith the leading diagonal of the $m$ atrix in row 1 of the
array, the firstsub-diagonal starting atposition 1 in row 2, and so on. The bottom right $k$ by $k$ triangle of the array A is not referenced. The follow ing program segm entw ill transfer a low er triangular band $m$ atrix from conventional full $m$ atrix storage to band storage:

D O 20, J=1, N
M = 1 -J
D O $10, \mathrm{I}=\mathrm{J}, \mathrm{M} \mathbb{N}(\mathrm{N}, \mathrm{J}+\mathrm{K})$
A $(M+I, J)=m \operatorname{atrix}(I, J)$
10 CONTINUE
20 CONTINUE
$N$ ote thatw hen D $\mathbb{A} G=U$ 'or L'the elem ents of the array A comesponding to the diagonalelem ents of the $m$ atrix are not referenced, butare assum ed to be unity. U nchanged on exit.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A $>=$ ( $k+1$ ). U nchanged on exit.

Y (input/output)
$(1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))$. Before entry, the increm ented aray $Y \mathrm{~m}$ ust contain the n elem ent righthand side vectorb. On exit, $Y$ is overw ritten $w$ ith the solution vector $x$.
$\mathbb{N} C Y$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $\mathrm{Y} . \mathbb{N} C Y<>0$. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

stbtrs-solve a triangular system of the form $A * X=B$ orA ${ }^{* *}$ T * $\mathrm{X}=\mathrm{B}$,

## SYNOPSIS

```
SU BROUT\mathbb{NE STBTRS (UPLO,TRANSA,D IAG,N,KD,NRHS,A,LDA,B,LDB,}
    \mathbb{NFO)}
CHARACTER * 1 UPLO,TRANSA,DIAG
\mathbb{NTEGERN,KD,NRHS,LDA,LDB,INFO}
REAL A (LDA,*),B (LDB,*)
SUBROUTINE STBTRS_64 (UPLO,TRANSA,D IA G ,N,KD,NRHS,A LD LA,B,
    LDB,\mathbb{NFO)}
```

CHARACTER * 1 UPLO, TRANSA, D IA G
$\mathbb{N}$ TEGER*8N,KD,NRHS,LDA,LDB, $\mathbb{N} F O$
REALA (LDA,*), B (LDB,*)

## F95 INTERFACE

SU BROUTINE TBTRS (UPLO,TRANSA,D $\mathbb{I A} G, N, K D, N R H S, A,[L D A], B$, [LDB], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G $\mathbb{N} T E G E R:: N, K D, N R H S, L D A, L D B, \mathbb{N F O}$
REAL,D $\mathbb{I M}$ ENSION (: : : : ::A, B
SU BROUTINE TBTRS_64 (UPLO, TRANSA, D $\mathbb{A} G, N, K D, N R H S, A,[L D A], B$, [LDB], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
$\mathbb{N}$ TEGER (8) ::N,KD,NRHS,LDA, LDB, $\mathbb{N} F O$

REAL, D $\mathbb{M}$ ENSION (:,:) ::A,B

## C INTERFACE

\#include <sunperfh>
void stotrs (charuplo, chartransa, chardiag, int n, int kd , int nihs, float *a, intlda, float*b, int ldlo, int*info);
void stbtrs_64 (charuplo, chartransa, char diag, long n, long kd, long nrhs, float *a, long lda, float *b, long ldlo, long *info);

## PURPOSE

stbtes solves a triangular system of the form
$w$ here $A$ is a triangularband $m$ atrix of order $N$, and $B$ is an N boy NRHS m atrix. A check is m ade to verify thatA is nonsingular.

## ARGUMENTS

UPLO (input)
$=\mathrm{U}$ : A is upper triangular;
$=\mathbb{L}$ ': A is low ertriangular.

TRANSA (input)
Specifies the form the system of equations:
$=\mathrm{N}^{\prime}: \mathrm{A} * \mathrm{X}=\mathrm{B} \quad$ (No transpose)
$=T: A * * T X=B \quad$ ( $r$ ranspose)
$=C$ ': $A * * H * X=B \quad$ (C onjugate transpose $=T$ ran -
spose)

D IA G (input)
$=\mathrm{N}$ : A is non-unit triangular;
$=\mathrm{U}: \mathrm{A}$ is unit triangular.

N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.

KD (input)
The num ber of superdiagonals or subdiagonals of the triangularband $m$ atrix A. KD $>=0$.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRH S $>=0$.

A (input) The upper or low er triangular band matrix A, stored in the first kd+1 row sofA. The jth column ofA is stored in the $j$ th column of the aray A as follow s: if UPLO = U',A (kd+1+i-j) = A $(i, j)$ form ax $(1, j k d)<=i<=\dot{j}$ if UPLO $=\mathrm{L}$ ', $A(1+i-j)=A(i, j)$ for $j=i<=m$ in $(n, j+k d)$. If $D \mathbb{A G}=U$ ', the diagonalelem ents of $A$ are not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the array A. LDA >= K D +1.

B (input/output)
On entry, the righthand side m atrix B. On exit, if $\mathbb{N} F O=0$, the solution $m$ atrix $X$.

LD B (input)
The leading dim ension of the anay B . LD B $>=$ $\max (1, N)$.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
< 0: if $\mathbb{N}$ FO = -i, the i-th argum enthad an illegalvalue
$>0:$ if $\mathbb{N} F O=i$, the $i$-th diagonalelem ent of $A$ is zero, indicating that the $m$ atrix is singular and the solutions $X$ have notbeen com puted.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

stgevc - com pute som e orallof the rightand/or left generalized eigenvectors of a pair of real upper triangular $m$ atrices ( $A, B$ )

## SYNOPSIS

```
SUBROUT\mathbb{NE STGEVC (SDE,HOW MNY,SELECT,N,A,LDA,B,LDB,VL,LDVL,}
    VR,LDVR,MM,M,W ORK,INFO)
CHARACTER * 1SIDE,HOW MNY
NNTEGER N,LDA,LDB,LDVL,LDVR,MM,M,\mathbb{NFO}
LOG ICAL SELECT (*)
REALA (LDA,*),B (LDB,*),VL (LDVL,*),VR(LDVR,*),W ORK (*)
SU BROUT\mathbb{NE STGEVC_64 (SDE ,HOW M NY,SELECT,N,A,LDA,B,LDB,VL,}
    LDVL,VR,LDVR,MM,M,W ORK,\mathbb{NFO)}
CHARACTER * 1SDEE,HOW MNY
INTEGER*8N,LDA,LD B,LDVL,LDVR,MM,M,INFO
LOG ICAL*8 SELECT (*)
REALA (LDA,*),B (LDB,*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE TGEVC (STDE,HOW MNY,SELECT,N,A, [LDA],B, [LDB],VL, [LDVL], VR, [LDVR], M M , M, [W ORK], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1) ::SDE,HOW M NY
$\mathbb{N} T E G E R:: N, L D A, L D B, L D V L, L D V R, M M, M, \mathbb{N} F O$
LOG ICAL,D IM ENSION (:) ::SELECT
REAL,D IM ENSION (:) ::W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) ::A,B,VL,VR

SU BROUTINE TGEVC_64 (SDE,HOW M NY,SELECT,N,A, [LDA],B, [LDB],VL, [LDVL], VR, [LDVR], M M , M, [W ORK ], [ $\mathbb{N F O}]$ )

CHARACTER (LEN=1) ::SDE,HOW M NY
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{LD} A, L D B, L D V L, L D V R, M M, M, \mathbb{N} F O$
LOG ICAL (8), D IM ENSION (:) :: SELECT
REAL,D IM ENSION (:) ::W ORK
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A,B,VL,VR

## C INTERFACE

\#include < sunperfh>
void stgevc (char side, char how my, int *select, int $n$, float *a, intlda, float*b, intldb, float *vl, int ldvl, float * $\mathrm{Vr}_{\text {, int }}$ int $\operatorname{ldvr}$, int m , int * m , int *info);
void stgevc_64 (char side, charhow m ny, long *select, long n, float *a, long lda, float *b, long ldb, float *vl, long ldvl, float *vr, long ldvr, long mm, long *m , long *info);

## PURPOSE

stgevc com putes som e orallof the right and/or left generalized eigenvectors of a pair of realupper triangular $m$ atrices $(A, B)$.

The right generalized eigenvectorx and the left generalized eigenvectory of $(A, B)$ corresponding to a generalized eigenvaluew are defined by:

$$
(A-w B) * x=0 \text { and } y^{* * H} *(A-w B)=0
$$

$w$ here $\mathrm{y}^{* *}{ }_{\mathrm{H}}$ denotes the conjugate tranpose of y .
If an eigenvalue w is determ ined by zero diagonal elem ents of both A and B, a unit vector is retumed as the comesponding eigenvector.

If alleigenvectors are requested, the routine $m$ ay either retum the $m$ atrices $X$ and/or $Y$ of rightor lefteigenvectors of $(A, B)$, or the products $Z * X$ and/or $Q * Y$, where $Z$ and $Q$ are input orthogonal $m$ atrices. If ( $A, B$ ) w as obtained from the generalized real-Schur factorization of an original pair of $m$ atrices ( $0, B 0$ ) $=Q * A * Z * * H, Q * B * Z * * H)$, then $Z * X$ and $Q * Y$ are the $m$ atrioes of right or lefteigenvec-

A m ustbe block uppertriangular, w ith 1 -by -1 and $2-b y-2$ diagonal blocks. C orresponding to each $2-b y-2$ diagonal block is a com plex conjugate pair ofeigenvalues and eigenvectors; only one eigenvector of the pair is com puted, nam ely the one corresponding to the eigenvalue $w$ ith posilive im aginary part.

## ARGUMENTS

SIDE (input)
$=R$ : com pute righteigenvectors only;
$=\mathbb{L}$ ': com pute lefteigenvectors only;
= B': com pute both rightand lefteigenvectors.
HOW M NY (input)
= 'A ': com pute all right and/or lefteigenvectors;
$=\mathrm{B}$ ': com pute all rightand/or lefteigenvectors, and backtransform them using the inputm atrices supplied in VR and/orVL; = S ': com pute selected right and/or lefteigenvectors, specified by the logical anay SELECT.

## SELECT (input)

If H OW M NY = $S^{\prime}$, SELECT specifies the eigenvectors
to be com puted. If HOW M NY = A 'or B', SELECT is
notreferenced. To select the real eigenvector
corresponding to the real eigenvalue w ( $)$, SELECT ( 7 ) mustbe setto .TRUE. To select the complex eigenvector corresponding to a complex conjugate pairw ( 1 ) and w ( $j+1$ ), either SELECT ( 7 ) orSELECT (j+1) m ustbe setto .TRUE..

N (input) The order of the m atriges $A$ and $B . N>=0$.

A (input) The upperquasi-triangularm atrix A.

LD A (input)
The leading dim ension of anay A. LD A $>=\max (1$, N) 。

B (input) The uppertriangularm atrix B. IfA has a 2 -by -2 diagonal block, then the corresponding 2 -by-2 block of $B$ m ustbe diagonal $w$ ith positive elem ents.

LD B (input)

The leading dimension of array B. LDB >= $\max (1, \mathbb{N})$.

VL (input/output)
On entry, if $S \mathbb{D} E=L^{\prime}$ 'or $B$ 'and HOW MNY = B', VL must contain an $N$-by N m atrix $Q$ (usually the orthogonalm atrix $Q$ of leftSchurvectors returmed by SHGEQZ). On exit, ifSDE = L'or B',VL contains: if HOW MNY = A', thematrix $Y$ of left eigenvectors of $(A, B)$; if HOW M NY = $B$ ', the matrix $Q * Y$; if HOW M NY $=S^{\prime}$, the left eigenvectors of (A , B ) specified by SELEC T , stored consecutively in the colum nsofVL, in the same order as their eigenvalues. If $S \mathbb{D} E=R \prime, V L$ is not referenced.

A com plex eigenvector comesponding to a com plex eigenvalue is stored in tw o consecutive colum ns, the firstholding the realpart, and the second the im aginary part.

LDVL (input)
The leading dimension of array VL. LDVL >= $\max (1, N)$ if $S \mathbb{D} E=L$ 'or $B$ '; LDVL $>=1$ other w ise.

VR (input/output)
On entry, if $S D E=R$ 'or $B$ 'and $H O W M N Y=B$ ', $V R$ must contain an $N$ by $-N \mathrm{~m}$ atrix $Q$ (usually the orthogonal matrix $Z$ of right Schur vectors retumed by $S H G E Q Z$ ). On exit, if $S \mathbb{D} E=R$ 'or $B^{\prime}, \mathrm{VR}$ contains: if HOW MNY = $A$ ', the $m$ atrix $X$ of right eigenvectors of $(A, B)$; if HOW M NY $=B$ ', them atrix $Z * X$; if H OW M NY $=S^{\prime}$, the right eigenvectors of $(A, B)$ specified by SELECT, stored consecutively in the colum ns of V $R$, in the sam e order as their eigenvalues. If $S \mathbb{D} E=\mathrm{L}, \mathrm{VR}$ is not referenced.

A com plex eigenvector corresponding to a com plex eigenvalue is stored in tw o consecutive colum ns, the firstholding the realpart and the second the im aginary part.

LDVR (input)
The leading dim ension of the array VR. LDVR >= $\max (1, N)$ if $S \mathbb{D} E=R$ 'or $B$ '; LDVR $>=1$ otherw ise.

M M (input) The num ber of colum ns in the arrays $V L$ and/or VR.
$M M>=M$.

M (output)
The num ber of colum ns in the arrays $V \mathrm{~L}$ and/or VR actually used to store the eigenvectors. If HOW M NY = A 'or B', M is set to N. Each selected real eigenvector occupies one colum $n$ and each selected com plex eigenvector occupies tw o colum ns. W ORK (w orkspace)
dim ension $(6 * N)$
$\mathbb{N}$ FO (output)
= 0: successfiulexit.
$<0:$ if $\mathbb{N} F O=-i$, the $i$ th argum enthad an illegalvalue.
 a com plex eigenvalue.

## FURTHER DETAILS

A llocation of w orkspace:

W ORK ( $j$ ) $=1$-norm of $j$ th colum $n$ ofA, above the diagonal

W ORK $(N+j)=1$-norm of $j$ th colum $n$ of $B$, above the diagonal

W ORK $(2 * \mathrm{~N}+1: 3 \star \mathrm{~N})=$ realpartofeigenvector
W ORK $(3 * N+1: 4 * N)=$ im aginary partofeigenvector
W ORK ( $4 * N+1: 5 \star \mathrm{~N})=$ realpartofback-transform ed eigenvector

W ORK $(5 * N+1: 6 * N)=$ im aginary part of back-transform ed eigenvector

R ow w ise vs. colum nw ise solution m ethods:

Finding a generalized eigenvector consists basically of solving the singular triangular system

$$
(A-w B) x=0 \quad \text { (for right) or: }(A-w B)^{\star *} H \quad y=0
$$

(forleft)

C onsider finding the i-th right eigenvector (assume all eigenvalues are real). The equation to be solved is: $0=\operatorname{sum} C(j k) v(k)=\operatorname{sum} C(j k) v(k) \quad$ for $j=i, .$. .1
$k=j \quad k=j$
where $C=(A-w B) \quad$ (The com ponents $v(i+1 n)$ are 0.$)$

The "row w ise" m ethod is:
(1) $\mathrm{v}(\mathrm{i}):=1$
for $j=i-1, \ldots, 1$ :
i
(2) com pute $s=-$ sum $C(j k) \vee(k)$ and $k=j+1$
(3) $\vee(\mathcal{j})=s / C(j)$

Step 2 is som etim es called the "dotproduct" step, since it is an innerproduct.betw een the jth row and the portion of the eigenvector that has been com puted so far.

The "colum nw ise" $m$ ethod consists basically in doing the sum $s$ forall the row $s$ in parallel. A s each $v(j)$ is com puted, the contribution of $v(i)$ tim es the $j$ th colum $n$ of $C$ is added to the partial sum s. Since FO RTRAN amays are stored colum nw ise, this has the advantage that ateach step, the elem ents of $C$ that are accessed are adjacent to one another, w hereas w ith the row w ise m ethod, the elem ents accessed at a step are spaced LD A (and LD B ) w ords apart.

W hen finding lefteigenvectors, the $m$ atrix in question is the transpose of the one in storage, so the row w ise $m$ ethod then actually accesses colum ns of A and B ateach step, and so is the prefered $m$ ethod.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

stgexc - reorder the generalized realSchur decom position of a real $m$ atrix pair ( $A, B$ ) using an orthogonalequivalence transform ation $(A, B)=Q *(A, B) * Z^{\prime}$,

## SYNOPSIS

```
SUBROUTINE STGEXC (N ANTQ,W ANTZ,N,A,LDA,B,LDB,Q,LDQ,Z,LD Z,
    #FST, LLST,W ORK,LW ORK,INFO)
```



```
LOGICALW ANTQ,W ANTZ
REAL A (LDA,*),B (LD B ,*),Q (LD Q ,*), Z (LD Z ,*),W ORK (*)
SUBROUT\mathbb{NE STGEXC_64(N ANTQ,W ANTZ,N,A,LDA,B,LDB,Q,LDQ,Z,LD Z,}
    #FST, [LST,W ORK,LW ORK,INFO)
NNTEGER*8 N,LDA,LD B,LDQ,LD Z,\mathbb{FST},\mathbb{LST,LW ORK, INFO}
LOGICAL*8W ANTQ,W ANTZ
REALA (LDA,*),B (LD B ,*),Q (LD Q ,*), Z (LD Z ,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE TGEXC $\mathbb{N}$ ANTQ,W ANTZ,N,A, [LDA],B, [LDB],Q, [LDQ],Z,

$\mathbb{N}$ TEGER ::N,LDA,LDB,LDQ,LD $\mathrm{L}, \mathbb{F} S T, \mathbb{L} S T, L W$ ORK, $\mathbb{N} F O$
LOG ICAL ::W ANTQ,W ANTZ
REAL,D IM ENSION (:) ::W ORK
REAL,D $\mathbb{M}$ ENSION (:,:) ::A,B,Q,Z
SU BROUTINE TGEXC_64 (NANTQ,WANTZ,N,A,[LDA],B,[LDB],Q,[LDQ],Z, [LD Z], $\mathbb{F S T}, \mathbb{L} S T,\left[\begin{array}{l}\text { W ORK ], [LW ORK ], [ } \mathbb{N F O}])\end{array}\right.$
$\mathbb{N} T E G E R(8):: N, L D A, L D B, L D Q, L D Z, \mathbb{F} S T, \Pi S T, L W O R K, \mathbb{N F O}$
LOGICAL (8) ::W ANTQ,W ANTZ
REAL,D $\mathbb{M}$ ENSION (:) ::W ORK
REAL,D $\mathbb{I}$ ENSION (: : : : : A , B, Q , Z

## C INTERFACE

\#include <sunperfh>
void stgexc (intw antq, intw antz, intn, float*a, int lda, float *b, int ldb, float * q , int ldq, float * z , int ldz, int *ifst, int *ilst, int *info);
void stgexc_64 (long w antr, long w antz, long n, float *a, long lda, float *b, long ldb, float *q, long ldq, float*z, long ldz, long *ifst, long *ilst, long *info);

## PURPOSE

stgexc reorders the generalized realSchurdecom position of a real matrix pair ( $A, B$ ) using an orthogonalequivalence transform ation
so that the diagonalblock of $(A, B)$ w ith row index $\mathbb{F} S T$ is m oved to row $\mathbb{L} S T$.
(A , B ) m ustbe in generalized realSchur canonical form (as retumed by SGGES), ie.A isblock uppertriangularw ith 1-by-1 and 2 -by-2 diagonalblocks.B is uppertriangular.

Optionally, the m atrioes $Q$ and $Z$ ofgeneralized Schur vectors are updated.

Q (in) * A (in) * Z (in) ${ }^{\prime}=\mathrm{Q}$ (out) * A (out) * Z (out)'
Q (in) * B (in) * Z (in) ${ }^{\prime}=\mathrm{Q}$ (out) * B (out) * Z (out) '

## ARGUMENTS

W ANTQ (input)

W ANTZ (input)

N (input) The order of the m atriges A and $\mathrm{B} . \mathrm{N}>=0$.

A (input/output)
O n entry, the m atrix A in generalized real Schur
canonical form. On exit, the updated m atrix A, again in generalized realSchur canonical form .

LD A (input)
The leading dim ension of the aray A. LDA >= $\max (1, N)$.

B (input/output)
O n entry, the m atrix B in generalized real Schur canonical form ( $A, B$ ). On exit, the updated $m$ atrix $B$, again in generalized realSchur canonical form ( $\mathrm{A}, \mathrm{B}$ ).
LD B (input)
The leading dim ension of the array $\mathrm{B} . \mathrm{LD} \mathrm{B}>=$ $\max (1, N)$.

Q (input/output)
On entry, ifW ANTQ = TRUE , the orthogonalm atrix Q . On exit, the updatedmatrix Q . If $\mathrm{W} A N T Q=$ FA LSE., Q is not referenced.

LD Q (input)
The leading dim ension of the array $\mathrm{Q} . \mathrm{LD} \mathrm{Q}>=1$. IfW ANTQ $=$ TRUE., LDQ $>=N$.

Z (input/output)
On entry, ifW ANTZ = TRUE., the orthogonalm atrix $Z$. On exit, the updatedm atrix $Z$. If $\mathrm{ANTZ}=$ FALSE., Z is not referenced.

LD Z (input)
The leading $d i m$ ension of the array $Z$. LD $Z>=1$. IfW ANTZ $=$.TRUE., LD Z $>=N$.

IFST (input/output)
Specify the reordering of the diagonal blocks of ( $A, B$ ). The block $w$ ith row index $\mathbb{F} S T$ ism oved to row ILST, by a sequence ofsw apping betw een adj ${ }^{-}-$ cent.blocks. On exit, if IFST pointed on entry to the second row of a 2 -by- 2 block, it is changed to point to the first row ; ILST alw ays points to the firstrow of the block in its final position (which $m$ ay differ from its inputvalue by +1 or $-1) \cdot 1<=\mathbb{F S T}, \mathbb{L} S T<=\mathrm{N}$.

ㄴST (input/output)
See the description of $\mathbb{F S T}$.
W ORK (w orkspace)
On exit, if $\mathbb{N F F O}=0, \mathrm{~W}$ ORK (1) retums the optim al

LW ORK.

LW ORK (input)
The dim ension of the anay W ORK. LW ORK >= 4*N + 16.

If LW ORK $=-1$, then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
$\mathbb{N} F O$ (output)
=0: successfulexit.
<0: if $\mathbb{N F O}=-$ i, the $i$-th argum ent had an illegalvahue.
=1: The transform ed $m$ atrix pair ( $A, B$ ) w ould be too far from generalized Schur form ; the problem is ill-conditioned. (A,B) may have been partially reordered, and IIST points to the first row of the current position of the block being $m$ oved.

## FURTHER DETAILS

B ased on contributions by
Bo K agstrom and PeterPorom aa, D epartm ent of Com puting Science,
Um ea U niversity, S-901 87 Um ea, Sw eden.
[1] B . K agstrom ; A D irectM ethod forReordering Eigenvahues in the

G eneralized RealSchurForm of a RegularM atrix Pair (A, B), in

M S.M oonen etal (eds), LinearA lgebra for Large Scale and

Real-T in e A pplications, K luw erA cadem ic Publ. 1993, pp 195-218.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

stgsen -reorder the generalized realSchurdecom position of a real $m$ atrix pair ( $A, B$ ) (in term sofan orthonorm al equivalence trans-form ation $Q$ '* $(A, B)$ * $Z$ ), so that a selected cluster of eigenvalues appears in the leading diagonalblocks of the upperquasi-triangularm atrix A and the upper triangularB

## SYNOPSIS

```
SUBROUT\mathbb{NE STGSEN (INOB,W ANTQ,W ANTZ,SELECT,N,A,LDA,B,LDB,}
    ALPHAR,ALPHAI,BETA,Q,LDQ,Z,LD Z,M,PL,PR,D F,W ORK,
```



$\mathbb{N} T E G E R \mathbb{I N}$ ORK (*)
LOG ICALW ANTQ,WANTZ
LO G ICAL SELECT (*)
REAL PL, PR
REAL A (LDA, *), B (LDB,*), ALPHAR (*), ALPHAI(*), BETA (*),
Q (LDQ,$\left.^{\star}\right), \mathrm{Z}(\mathrm{LD} Z, \star), \mathrm{D} \mathbb{F}\left({ }^{\star}\right), \mathrm{W} O R K\left({ }^{\star}\right)$
SU BROUTINE STGSEN_64 (JOB,W ANTQ,W ANTZ,SELECT,N,A,LDA,B,LDB,
A LPHAR,ALPHAI,BETA, $Q, L D Q, Z, L D Z, M, P L, P R, D \mathbb{F}, W$ ORK,
LW ORK, $\mathbb{I W}$ ORK,LIW ORK, $\mathbb{N} F O$ )
$\mathbb{N} T E G E R * 8 \operatorname{LJO}, N, L D A, L D B, L D Q, L D Z, M, L W O R K, L \mathbb{N} O R K$,
$\mathbb{N F O}$
$\mathbb{N}$ TEGER*8 $\mathbb{I N}$ ORK (*)
LOG ICAL*8W ANTQ,WANTZ
LOG ICAL* 8 SELECT (*)
REALPL, PR
 $\mathrm{Q}\left(\mathrm{LD} Q,{ }^{*}\right), \mathrm{Z}(\mathbb{L D} \mathrm{Z}, \star), \mathrm{D} \mathbb{F}(*), \mathrm{W} O R K(*)$

## F95 INTERFACE

SU BROUTINE TG SEN (LOB B,W ANTQ,W ANTZ, SELECT,N,A, [LD A ], B, [LD B ], ALPHAR,ALPHAI,BETA, $Q,[L D Q], Z,[L D Z], M, P L, P R, D \mathbb{F},[W O R K]$, $[[W W O R K],[\mathbb{N} O R K],[L \mathbb{N} O R K],[\mathbb{N} F O])$
$\mathbb{N} T E G E R:: \mathbb{I J O B}, N, L D A, L D B, L D Q, L D Z, M, L W O R K, L \mathbb{I N} O R K$, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N}$ ORK
LOG ICAL ::W ANTQ,W ANTZ
LOG ICAL,D IM ENSION (:) ::SELECT
REAL ::PL,PR
REAL,D $\mathbb{I M} E N S I O N(:):: A L P H A R, A L P H A I, B E T A, D \mathbb{F}, W$ ORK
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A,B,Q,Z
SUBROUTINE TGSEN_64 (LOBB,W ANTQ,W ANTZ,SELECT,N,A, [LDA],B, [LDB], A LPHAR,ALPHAI, BETA, $Q,[L D Q], Z,[L D Z], M, P L, P R, D \mathbb{F},[W O R K]$, [LW ORK], [IN ORK], [LIW ORK], [ $\mathbb{N} F O$ ])
$\mathbb{N} \operatorname{TEGER}(8):: \operatorname{LOB}, N, L D A, L D B, L D Q, L D Z, M, L W O R K, L \mathbb{I N} O R K$, $\mathbb{N}$ FO
$\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N}$ ORK
LOG ICAL (8) ::W ANTQ,W ANTZ
LO G ICAL (8),D IM ENSION (:) ::SELECT
REAL ::PL,PR
REAL,D $\mathbb{M}$ ENSION (:) ::ALPHAR,ALPHAI,BETA,D $\mathbb{F}, \mathrm{W}$ ORK
REAL,D $\mathbb{M}$ ENSION (:,:)::A,B, $\mathrm{Q}, \mathrm{Z}$

## C INTERFACE

\#include < sunperfh>
void stgsen (int ijob, intw antq, intw antz, int *select, int
n , float *a, int lda, float *b, int ldb, float
*alphar, float * alphai, float *beta, float *q, int
ldq, float *z, int ldz, int *m, float *pl, float
*pr, float*dif, int*info);
void stgsen_64 (long ijob, long wantq, long wantz, long
*select, long n , float *a, long lda, float *b, long ldb, float *alphar, float *alphai, float *beta, float *q, long ldq, float*z, long ldz, long *m, float*pl, float *pr, float *dif, long *info);

## PURPOSE

stgsen reorders the generalized realSchurdecom position of a real matrix pair ( $\mathrm{A}, \mathrm{B}$ ) (in term sofan orthonorm al
equivalence trans-form ation $Q^{\prime}$ ( $(A, B)$ * $Z$ ), so that a selected cluster of eigenvalues appears in the leading diagonalblocks of the upperquasi-triangularm atrix $A$ and the upper triangular B. The leading colum ns of $Q$ and $Z$ form orthonorm albases of the corresponding left and righteigenspaces (deflating subspaces). (A,B) m ustbe in generalized realSchurcanonical form (as retumed by SG GES), ie.A is block upper triangular w ith 1 -by-1 and 2 -by-2 diagonal blocks.B is upper triangular.

STG SEN also com putes the generalized eigenvalues

$$
w(\mathcal{j})=\left(\operatorname{ALPHAR}(\mathcal{j})+i^{\star} \operatorname{LPHAI}(\mathcal{j})\right) B E T A(\mathcal{j})
$$

of the reordered $m$ atrix pair ( $A, B$ ).
Optionally, STG SEN computes the estim ates of reciprocal condition num bers foreigenvalues and eigenspaces. These are D ifu [ $A 11, B 11$ ), (A 22, B22)] and D ifll(A 11, B11), (A 22, B22)], i.e. the separation ( $s$ ) betw een the $m$ atrix pairs (A 11, B11) and (A 22,B22) that correspond to the selected cluster and the eigenvalues outside the cluster, resp., and norm sof "pro jections" onto left and right eigenspaces w r.t. the selected cluster in the ( 1,1 )-block.

## ARGUMENTS

```
INO B (input)
    Specifies w hether condition num bers are required
    for the cluster ofeigenvalues (PL and PR) orthe
    deflating subspaces (D ifu and D ifl):
    =0:Only reorderw r.t.SELEC T .N o extras.
    =1:Reciprocal ofnorm s of "pro jections" onto left
    and righteigenspaces w r.t. the selected cluster
    (PL andPR). = 2:U pperbounds on D ifu and D ifl.
    F-norm -based estim ate
    (D \mathbb{F (1:2)).}
    =3:Estim ate ofD ifu and D ifl.1-norm -based esti-
    m}\mathrm{ ate
    (D IF (1:2)). A bout5 tim es as expensive as INO B =
    2. =4: Com pute PL,PR and D FF (ie.0,1 and 2
    above): Econom ic version to get it all. =5: C om -
    putePL,PR and D FF (i.e.0,1 and 3 above)
```

W ANTQ (input)

## SELECT (input)

SELEC T specifies the eigenvalues in the selected cluster. To select a real eigenvalue w ( $\mathcal{j}$ ), SELECT ( $j$ ) must be set to $\mathrm{w}(\mathcal{j})$ and $\mathrm{w}(j+1)$, corresponding to a 2 -by-2 diagonalblock, either SELECT ( $\ddagger$ ) orSELECT ( $j+1$ ) orboth $m$ ust be set to either both included in the cluster or both excluded.

N (input) The order of the m atrioes A and $\mathrm{B} . \mathrm{N}>=0$.
A (input/output)
On entry, the upper quasi-triangular $m$ atrix $A$, w ith (A, B) in generalized realSchur canonical form. On exit, A is overw rilten by the reordered $m$ atrix A.

## LD A (input)

The leading dim ension of the aray A. LD A >= $\max (1, N)$.

B (input/output)
O n entry, the uppertriangularm atrix $B$, $w$ th $A$, B) in generalized realSchurcanonical form. On exit, $B$ is overw ritten by the reordered $m$ atrix $B$.

LD B (input)
The leading dim ension of the array B. LD B >= $\max (1, \mathbb{N})$.

ALPHAR (output)
On exit, (ALPHAR ( ) + ALPHAI ( ) *i) BETA ( ) , $\dot{j} 1, \ldots, N, w i l l$ be the generalized eigenvalues. A LPHAR $(\mathcal{Z})+$ ALPHAI $(\boldsymbol{j} *$ iand BETA $(\mathcal{j}), \dot{于} 1, \ldots, N$ are the diagonals of the com plex Schur form $(S, T)$ that w ould result if the 2 -by-2 diagonalblocks of the real generalized Schur form of ( $A, B$ ) w ere further reduced to triangular form using com plex unitary transform ations. If A LPHA I( $)$ is zero, then the $j$ th eigenvalue is real; ifpositive, then the $j$ th and ( $j+1$ )-steigenvahues are a com plex con $j \mu-$ gate pair, w ith A LPH A I(j+1) negative.

## A LPH A I (output)

See the description of A LPHAR.
BETA (output)
See the description of A LPHAR.
Q (input/output)

On entry, if $W$ ANTQ = TRUE., $Q$ is an $N$-by $N$ $m$ atrix. On exit, $Q$ has been postm ultiplied by the left orthogonal transform ation $m$ atrix which reorder ( $A$, B); The leading $M$ colum ns of f form orthonorm albases for the specified pair of left eigenspaces (deflating subspaces). If $W$ ANTQ $=$ FALSE., Q is notreferenced.

LD Q (input)
The leading dim ension of the array $\mathrm{Q} . \mathrm{LD} Q>=1$; and ifW ANTQ = .TRUE.,LDQ >=N.

Z (input/output)
On entry, if $W$ ANTZ $=$.TRUE., $Z$ is an $N$ by $-N$ $m$ atrix. O n exit, Z has been postm ultiplied by the left orthogonal transform ation $m$ atrix $w$ hich reorder ( $A$, B); The leading $M$ colum ns of $Z$ form orthonorm albases for the specified pair of left eigenspaces (deflating subspaces). If W ANTZ = FALSE., Z is notreferenced.

LD Z (input)
The leading dim ension of the array Z . LD $\mathrm{Z}>=1$; If $W A N T Z=. T R U E, \operatorname{LDZ}>=N$.

M (output)
The dim ension of the specified pair of left and right eigen-spaces (deflating subspaces). $0<=\mathrm{M}$ $<=N$.

PL (output)
If $\mathrm{IJOB}=1,4$ or $5, \mathrm{PL}, \mathrm{PR}$ are low er bounds on the reciprocal of the norm of "projections" onto leftand righteigenspaces w ith respect to the selected cluster. $0<\mathrm{PL}, \mathrm{PR}<=1$. IfM $=0$ orM $=N, P L=P R=1$. IfIOOB $=0,2$ or $3, P L$ and $P R$ are not referenced.

PR (output)
See the description ofPL .

D $\mathbb{F}$ (output)
If IUO B >=2,D $\mathbb{F}(1: 2)$ store the estim ates ofD ifu and $D$ iff.
If $\mathrm{IHO} B=2$ or $4, D \mathbb{F}(1: 2)$ are F -norm Hoased upper bounds on
$D$ ifu and $D$ ifl. If $I N O B=3$ or $5, D \mathbb{F}(1: 2)$ are $1-$
norm -based estim ates ofD ifu and $D$ ifl. $\mathrm{If} M=0$
orN , $D \mathbb{F}(1: 2)=F-n o r m([A, B])$. If $I J O B=0$ or $1, D$ IF is notreferenced.

W ORK (w orkspace)
If $\mathrm{IJOB}=0, \mathrm{~W} O R K$ is not referenced. O therw ise, on exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the amay $W$ ORK. LW ORK $>=4 * N+16$. If $\mathrm{LJO} B=1,2$ or 4, LW ORK $>=\mathrm{MAX}(4 * N+16,2 * \mathrm{M} * \mathbb{N}-$ $\mathrm{M})$ ). If $\mathrm{IJOB}=3$ or 5 , LW ORK $>=\mathrm{MAX}(4 * \mathrm{~N}+16$, $4 * M *(N-M))$.

IfLW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the $W$ ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IV ORK (w orkspace/output)
If $\mathrm{IJO} B=0, \mathbb{I V} O R K$ is not referenced. O therw ise, on exit, if $\mathbb{N} F O=0, \mathbb{I V}$ ORK (1) retums the optim al LIN ORK.

LIV ORK (input)
The dim ension of the array $\mathbb{I N}$ ORK.LIIN ORK >=1. If $\mathrm{IJOB}=1,2$ or $4, \mathrm{~L} \mathbb{I}$ ORK $>=\mathrm{N}+6$. If IJOB $=3$ or $5, L \mathbb{N}$ ORK $>=M A X(2 * M * \mathbb{N}+M), N+6)$.

If LIV ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the $\mathbb{I V}$ ORK array, retums this value as the first entry of the $\mathbb{I N}$ ORK array, and no errorm essage related to $L \mathbb{I N} O R K$ is issued by XERBLA.
$\mathbb{I N} F O$ (output)
=0: Successfulexit.
$<0:$ If $\mathbb{N} F O=-i$, the $i$-th argum enthad an illegal value.
$=1$ : Reordering of ( $A, B$ ) failed because the transform ed $m$ atrix pair ( $A, B$ ) w ould be too far from generalized Schur form ; the problem is very ill-conditioned. (A, B) m ay have been partially reordered. If requested, 0 is retumed in D IF (*), $P L$ and $P R$.

## FURTHER DETAILS

STG SEN firstcollects the selected eigenvalues by com puting orthogonal $U$ and $W$ thatm ove them to the top leftcomerof (A,B). In otherw ords, the selected eigenvalues are the
eigenvalues of (A11, B 11) in:

$$
\begin{gathered}
U *(A, B) * W=(A 11 A 12)(B 11 B 12) n 1 \\
(0 \text { A 22),(0 B22) n2 } \\
n 1 \mathrm{n} 2 \quad \mathrm{n} 1 \mathrm{n} 2
\end{gathered}
$$

w here $\mathrm{N}=\mathrm{n} 1+\mathrm{n} 2$ and U ' m eans the transpose of U . The first n1 colum ns of $U$ and $W$ span the specified pair of left and righteigenspaces (deflating subspaces) of $(A, B)$.

If ( $A, B$ ) has been obtained from the generalized real Schur decom position of a matrix pair ( $C, D$ ) $=Q$ * $(A, B) * Z$ ', then the reordered generalized realSchur form of ( $C, D$ ) is given by

$$
(C, D)=(Q * U)^{\star}(U \star(A, B) * W) *(Z * W)^{\prime},
$$

and the firstn1 colum ns of Q * U and $\mathrm{Z} * \mathrm{~W}$ span the comesponding deflating subspaces of ( $C, D$ ) $Q$ and $Z$ store $Q * U$ and Z *W, resp.).

N ote that if the selected eigenvalue is sufficiently illconditioned, then its value $m$ ay differ significantly from its value before reordering.

The reciprocalcondition num bers of the left and right eigenspaces spanned by the firstn1 colum ns ofU and $W$ (or $\mathrm{Q} * \mathrm{U}$ and $\mathrm{Z} * \mathrm{~W}) \mathrm{m}$ ay be retumed in $\mathrm{D} \mathbb{F}(1: 2)$, corresponding to $D$ ifu and $D$ ifll, resp.

The D ifu and D iflare defined as:
ifu $[$ A 11, B11), (A 22, B22) $]=\operatorname{sigm}$ am in ( Zu )
and
where sigm a-m in ( 2 u ) is the sm allest singular value of the ( $2 *_{n} 1 *{ }_{n} 2$ )-by- $\left(2 *_{n} 1 *_{n} 2\right.$ ) m atrix
$\mathrm{u}=[\mathrm{kron}(\mathrm{In} 2, \mathrm{~A} 11)-\mathrm{kron}(\mathrm{A} 22$ ', In1) ]
[kron(In2,B11) kron (B22', In1)].
H ere, $\operatorname{In} x$ is the identity $m$ atrix of size $n x$ and $A 22$ 'is the transpose of A 22. kron ( $X, Y$ ) is the $K$ roneckerproduct betw een the $m$ atrices $X$ and $Y$.

W hen D IF (2) is sm all, sm all changes in (A, B) can cause large changes in the deflating subspace. A $n$ approxim ate (asym ptotic) bound on them axim um angularemor in the com puted deflating subspaces is PS * norm ( $(A, B)$ )/D $\mathbb{F}(2)$,
where EPS is the $m$ achine precision.

The reciprocal norm of the pro jectors on the left and right eigenspaces associated with (A 11, B 11) m ay be retumed in PL and PR. They are com puted as follow s. First we com pute $L$ and $R$ so that $P$ * $(A, B){ }^{\star} Q$ is block diagonal, w here
$=(I-\amalg) n 1 \quad Q=(I R) n 1$

$$
\begin{array}{lll}
(0 \text { I) n2 } & \text { and } & (0 I) n 2 \\
\text { n1 n2 } & \text { n1 n2 }
\end{array}
$$

and ( $L, R$ ) is the solution to the generalized Sylvester equation $11 * \mathrm{R}-\mathrm{L} *$ A $22=-\mathrm{A} 12$

Then PL $=(F \text {-norm }(L) * * 2+1)^{* *}(-1 / 2)$ and $P R=(F-$ norm $(\mathbb{R}) * * 2+1) * *(-1 / 2)$. A n approxim ate (asym ptotic) bound on the average absolute error of the selected eigenvalues is PS * norm ( $(A, B)) / P L$.

There are also globalemorbounds which valid forperturbations up to a œertain restriction: A low erbound $(x)$ on the sm allest $F$-norm ( $E, F$ ) forw hich an eigenvalue of (A 11, B 11) $m$ ay $m$ ove and coalesce $w$ th an eigenvalue of (A 22,B22) under perturbation $(\mathbb{E})$, (i.e. $(A+E, B+F)$, is
$\mathrm{x}=$


A $n$ approxim ate bound on $x$ can be com puted from $D \mathbb{F}(1: 2)$, PL and PR .

If $y=(F-$ norm $(E, F) / x)<=1$, the angles betw een the per turbed (L', R) and unperturbed ( $L, R$ ) left and right deflating subspaces associated $w$ ith the selected cluster in the $(1,1)$-blocks can be bounded as
max-angle $(L, L)<=\arctan (y * P L /(1-y *(1-P L *$ PL)** (1/2))
$\max -\operatorname{angle}(R, R)<=\arctan (y * P R /(1-y *(1-P R *$
$\operatorname{PR}) * *(1 / 2))$
See LA PA CK U ser's G uide section 4.11 or the follow ing references form ore inform ation.

N ote that if the default $m$ ethod for com puting the Frobenius-norm - based estim ate D $\mathbb{F}$ is not wanted (see SLA TDF), then the param eter ID $\mathbb{F} \cdot \mathcal{B}$ (see below) should be changed from 3 to 4 (routine SLA TDF (LOB $=2 \mathrm{w}$ illbe used)). See STG SY L form ore details.

B ased on contributions by
Bo K agstrom and PeterPorom aa, D epartm ent of Com puting Science,

Um ea U niversity, S-901 87 Um ea, Sw eden.

## References

[1] B .K agstrom ; A D irectM ethod forReordering Eigenvalues in the

G eneralized RealSchurForm of a RegularM atrix Pair (A, B), in

M S.M oonen etal (eds), LinearA lgebra for Large Scale and

Real-T im e A pplications, K luw erA cadem ic Publ. 1993, pp 195-218.
[2] B . K agstrom and P.Porom aa; C om puting Eigenspaces w th Specified E igenvalues of a R egularM atrix Pair ( $A$, $B$ ) and Condition

Estim ation: Theory, A lgorithm s and Softw are, ReportU M IN F - 94.04 , D epartm ent of Computing Science, Um ea

U niversity, S-901 87 Um ea, Sw eden, 1994.A lso as LA PA CK W orking

N ote 87. To appear in N um ericalA lgorithm $\mathrm{s}, 1996$.
[3] B . K agstrom and P .Porom aa, LA PA C K -Style A lgorithm s and Softw are
for Solving the Generalized Sylvester Equation and Estim ating the Separation betw een RegularM atrix Pairs, ReportU M IN F 93 23,

D epartm entofC om puting Science, Um ea U niversity, S-901 87 Um ea,

Sw eden, D ecem ber 1993, Revised A pril 1994, A lso as LAPACK W orking

N ote 75. To appear in A CM Trans. on M ath. Softw are, V ol 22, N ○ 1,
1996.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

stgsja -com pute the generalized singular value decom position (G SV ) of tw o realupper triangular (ortrapezoidal) $m$ atrices $A$ and $B$

## SYNOPSIS

```
SU BROUT\mathbb{NE STGSJA (JOBU,NOBV,NOBQ,M,P,N,K,L,A,LDA,B,LDB,}
    TOLA,TOLB,ALPHA,BETA,U,LDU,V,LDV,Q,LDQ,W ORK,NCYCLE,
    \mathbb{NFO)}
```

CHARACTER * 1 JOBU, JOBV , JOBQ
$\mathbb{N}$ TEGER M, $\mathrm{P}, \mathrm{N}, \mathrm{K}, \mathrm{L}, \mathrm{LD} A, L D B, L D U, L D V, L D Q, N C Y C L E, \mathbb{N} F O$
REALTOLA,TOLB
REAL A (LDA $\left.{ }^{*}\right)$, B (LDB,*), ALPHA (*), BETA (*), U (LDU,*),

SU BROUTINE STGSJA_64 (JOBU, $\mathcal{J O B V}, \mathcal{J O B}, M, P, N, K, L, A, L D A, B, L D B$,
TOLA,TOLB,ALPHA,BETA,U,LDU,V,LDV,Q,LDQ,W ORK,NCYCLE,
$\mathbb{N} F O$ )
CHARACTER * 1 JOBU, JOBV , JOBQ
$\mathbb{N} T E G E R * 8 M, P, N, K, L, L D A, L D B, L D U, L D V, L D Q, N C Y C L E$,
$\mathbb{N}$ FO
REAL TOLA,TOLB
REAL A (LDA $\left.{ }^{\star}\right)$, B (LDB,$\left.\star\right), \operatorname{ALPHA}(\star), \operatorname{BETA}(\star), ~ U(L D U, \star)$,
$\mathrm{V}\left(\mathrm{LDV},{ }^{\star}\right), \mathrm{Q}\left(\mathrm{LD} \mathrm{Q},{ }^{\star}\right), \mathrm{W} O R K(*)$

## F95 INTERFACE

SU BROUTINE TGSJA (JOBU, JOBV, JOBQ, M, $\mathrm{P}, \mathrm{N}, \mathrm{K}, \mathrm{L}, \mathrm{A},[\mathrm{LDA}], \mathrm{B},[\mathrm{LDB}]$, TOLA,TOLB,ALPHA,BETA, U, [LDU],V, [LDV],Q, [LDQ], [W ORK], NCYCLE, [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1) :: JOBU, $0 \mathrm{OBV}, \mathrm{JOBQ}$
$\mathbb{N}$ TEGER ::M, $\mathrm{P}, \mathrm{N}, \mathrm{K}, \mathrm{L}, \mathrm{LDA}, \mathrm{LD} B, L D \mathrm{U}, \mathrm{LDV}, \mathrm{LD} Q, \mathrm{NCYCLE}$,
$\mathbb{N} F O$
REAL ::TOLA,TOLB
REAL,D $\mathbb{M}$ ENSION (:) ::ALPHA,BETA,W ORK
REAL,D IM ENSION (:,:) :: A, B, U, V, Q
SU BROUTINE TGSJA_64 (JOBU, $\mathrm{JOBV}, \mathcal{J O B Q}, \mathrm{M}, \mathrm{P}, \mathrm{N}, \mathrm{K}, \mathrm{L}, \mathrm{A},[\mathrm{LD} A], \mathrm{B}$, [LD B],TOLA,TOLB,ALPHA,BETA, U, [LDU],V, [LDV ], Q, [LDQ], [W ORK ],NCYCLE, [ $\mathbb{N F F O}$ ])

CHARACTER (LEN=1) :: JOBU, NOBV , JOBQ
$\mathbb{N}$ TEGER (8) ::M, $\mathrm{P}, \mathrm{N}, \mathrm{K}, \mathrm{L}, \mathrm{LDA}, \mathrm{LD} \mathrm{B}, \mathrm{LD} \mathrm{U}, \mathrm{LD} \mathrm{V}, \mathrm{LD} Q$, NCY-
CLE, $\mathbb{N} F O$
REAL ::TOLA,TOLB
REAL,D $\mathbb{M}$ ENSION (:) ::ALPHA,BETA,W ORK
REAL,D IM ENSION (:,:) ::A,B,U,V,Q

## C INTERFACE

\#include <sunperfh>
void stgsja (char jobu, char jobv, char jobq, intm, int p, int n , int k , int l, float *a, int lda, float *b, int ldb, floattola, float tolb, float *alpha, float *beta, float *u, int ldu, float *v, int ldv, float *q, int ldq, int *ncycle, int *info);
void stgsj’_64 (char jobu, char jobv, char jobq, long m, long p, long $n$, long $k$, long $l$, float *a, long lda, float*b, long ldb, float tola, float tolb, float *alpha, float *beta, float *u, long ldu, float *v, long ldv, float *q, long ldq, long *ncycle, long *info);

## PURPOSE

stgsja com putes the generalized singularvalue decom position
(G SVD) of two real upper triangular (or trapezoidal) $m$ atrices $A$ and $B$.

On entry, it is assum ed thatm atrices $A$ and $B$ have the follow ing form $s$, which $m$ ay be obtained by the preprocessing subroutine SG G SV P from a generalM by -N m atrix A and P -by N $m$ atrix B :

```
        N-K-L K L
A = K (0 A12 A13) ifM K-工 >= 0;
        L (0 0 A23)
    M K-L(0 0 0 )
```

```
    NK工 K L
A = K (0 A12 A13) ifM K L < 0;
    M K (0 0 A23)
    NK工 K L
B L L (0 0 B13)
    P-L(0 0 0 )
```

w here the K －by K m atrix A 12 and L－by－ m atrix B 13 are non－ singularuppertriangular；A 23 is L－by - uppertriangular if $\mathrm{M}-\mathrm{K}->=0$ ，otherw ise A 23 is $(\mathrm{M}-\mathrm{K})$－by－工 uppertrapezoidal．

On exit，

$$
U{ }^{*} A * Q=D 1 *(0 R), \quad V{ }^{*} B * Q=D 2^{*}(0 R)
$$

where $U, V$ and $Q$ are orthogonal $m$ atrioes，$Z$＇denotes the transpose of $Z, R$ is a nonsingular upper triangularm atrix， and D 1 and D 2 are＇＂diagonal＂m atrices，which are of the follow ing structures：
IfM $K-\longleftarrow>=0$ ，

$$
\begin{aligned}
& \text { K L } \\
& \text { D } 1=K(\mathrm{I} 0) \\
& \text { L ( } 0 \text { C ) } \\
& \text { M K ـ ( } 0 \text { O) }
\end{aligned}
$$

K L
D $2=\mathrm{L} \quad(0 \mathrm{~S})$
$\mathrm{P}-(0 \quad 0)$
$N \mathrm{~K}$ K K
（0R）$=\mathrm{K}$（0 R11 R12）$K$
L（ 0 O R22）L
where

$$
\begin{aligned}
& C=\operatorname{diag}(A L P H A(K+1), \ldots, \text { A LPH A }(\mathbb{K}+L)), \\
& S=\operatorname{diag}(\operatorname{BETA}(\mathbb{K}+1), \ldots, \text { BETA }(K+L)), \\
& C * * 2+S * * 2=I .
\end{aligned}
$$

$R$ is stored in $A(1: K+L, N-K-1+\mathbb{N})$ on exit．

IfM K 工＜0，

K M K K＋L M
$D 1=K\left(\begin{array}{lll}\text { I } 0\end{array}\right)$
$M-K(0 C O)$

K M K K＋L M

```
D2 = MK (0S O )
    K+L-M (0 O I )
    P-(0 0 0 )
```

    N K \(\mathrm{L} \quad \mathrm{K} \quad \mathrm{M}-\mathrm{K} \quad \mathrm{K}+\mathrm{L} \mathrm{M}\)
    \(M-K(0 \quad 0 \quad R 22 R 23)\)
    \(K+L+M(0 \quad 0 \quad 0 \quad R 33)\)
    where
$C=\operatorname{diag}(A L P H A(K+1), \ldots, A L P H A(M))$, $S=\operatorname{diag}(\operatorname{BETA}(K+1), \ldots, \operatorname{BETA}(\mathbb{M}))$, $C * * 2+S * * 2=I$.
$R=(R 11 R 12 R 13)$ is stored in $A(1 \mathbb{M}, N-K-1+1 \mathbb{N})$ and $R 33$ is stored
( 0 R22R23)
in $B(M-K+1: L, N+M-K-1 \pm N)$ on exit.

The com putation of the orthogonaltransform ation $m$ atrices $U$, $V$ or $Q$ is optional. These $m$ atrices $m$ ay eitherbe form ed explicitly, or they $m$ ay be postm ultiplied into input $m$ atrices U 1 , V 1 , or Q 1 .
STG SJA essentially uses a variant of $K$ ogbetliantz algorithm to reduce $m$ in ( $L, M-K$ ) -by -4 triangular (or trapezoidal) $m$ atrix A 23 and L -by -m atrix B 13 to the form :

$$
\mathrm{U} 1{ }^{*} \mathrm{~A} 13^{*} \mathrm{Q} 1=\mathrm{C} 1 * \mathrm{R} 1 ; \mathrm{V} 1 * \mathrm{~B} 13^{*} \mathrm{Q} 1=\mathrm{S} 1 * \mathrm{R} 1
$$ where $\mathrm{U} 1, \mathrm{~V} 1$ and Q 1 are orthogonalm atrix, and $Z^{\prime}$ is the transpose of Z. C1 and S1 are diagonalm atrices satisfying

$$
\mathrm{C} 1 * * 2+\mathrm{S} 1 * * 2=\mathrm{I}
$$

and R1 is an L-by -L nonsingularupper triangularm atrix.

## ARGUMENTS

JOBU (input)
$=\mathrm{U}$ ': U m ustcontain an orthogonalm atrix U 1 on entry, and the productU $1 * \mathrm{U}$ is retumed; = I': U is initialized to the unitm atrix, and the orthogonal matrix U is retumed; = $\mathrm{N}^{\prime}$ : U is notcom puted.

JOBV (input)
$=\mathrm{V}$ ': V m ustcontain an orthogonalm atrix V 1 on entry, and the product V 1*V is retumed; = I ': V is initialized to the unitm atrix, and the orthogonal m atrix V is retumed; $=\mathrm{N}: \mathrm{V}$ is notcom puted.

JOBQ (input)
$=\mathrm{Q}: \mathrm{Q}$ m ustcontain an orthogonalm atrix Q 1 on entry, and the product $\mathrm{Q} \mathrm{I}^{*} \mathrm{Q}$ is retumed; = $\mathrm{I}^{\prime}: \mathrm{Q}$ is initialized to the unitm atrix, and the orthogonal matrix Q is retumed; $=\mathrm{N}$ ': Q is notcom puted.

M (input) The num ber of row s of the $m$ atrix $A . M>=0$.
$P$ (input) The num ber of row sof the $m$ atrix $B . P>=0$.

N (input) The num ber of colum ns of the m atrioes A and B. N $>=0$.
$K$ (input) $K$ and $L$ specify the subblocks in the input $m$ atrices A and B :
$\mathrm{A} 23=\mathrm{A}(\mathrm{K}+1 \mathbb{M} \mathbb{N}(\mathrm{~K}+\mathrm{L}, \mathrm{M}), \mathbb{N}-\mathrm{L}+\mathbb{N})$ and $\mathrm{B} 13=$ $B(1: L, N-\Psi+1 \mathbb{N})$ of $A$ and $B$, whose $G S V D$ is going to be com puted by STG SJA. Se Furtherdetails.
$L$ (input) See the description ofK .

A (input/output)
On entry, the $M-b y-N$ matrix $A$. On exit, $A \mathbb{N}-$ $K+1 \mathbb{N}, \mathbb{M} \mathbb{N}(\mathbb{K}+\mathrm{L}, \mathrm{M})$ ) contains the triangular $m$ atrix $R$ orpartofR. See Purpose for details.

LDA (input)
The leading dim ension of the array A. LD A >= max (1, M).

B (input/output)
On entry, the P -by-N m atrix B. On exit, if necessary, $B(M-K+1: L, N+M-K+1 \mathbb{N}$ ) contains a partofR . See Purpose for details.

LD B (input)
The leading dim ension of the array $\mathrm{B} . \operatorname{LD} \mathrm{B}>=$ $\max (1, \mathrm{P})$.

TO LA (input)
TO LA and TO LB are the convergence criteria for the Jacobi- K ogbetliantz teration procedure. Generally, they are the sam e as used in the preprocessing step, say TOLA $=m a x M, N) *$ norm (A)*M ACHEPS, TOLB $=\max (\mathbb{P}, N) \star$ norm (B)*M ACHEPS.

## TOLB (input)

See the description of TO LA .
ALPHA (output)
On exit, A LPHA and BETA contain the generalized
singular value pairs of $A$ and $B$; ALPHA $(1: K)=1$, BETA $(1: K)=0$, and ifM $K-L>=0$, ALPHA $(K+1 \mathbb{K}+L)$
$=\operatorname{diag}(\mathrm{C})$,
BETA $(K+1 K+L)=\operatorname{diag}(S)$, or if $M-\mathrm{K}<0$,
A LPHA $(\mathbb{K}+1 \mathrm{M})=\mathrm{C}$, A LPHA $\mathrm{M}+1 \mathrm{~K}+\mathrm{L})=0$
BETA $(K+1 M)=S, B E T A(M+1 \mathbb{K}+L)=1$. Furtherm ore,
if $\mathrm{K}+\mathrm{L}<\mathrm{N}, \mathrm{A}$ LPHA $(\mathbb{K}+\mathrm{L}+1 \mathbb{N})=0$ and
BETA $(K+L+1 \mathbb{N})=0$.
BETA (output)
See the description of A LPH A.
U (input) On entry, if $\mathrm{JOBU}=\mathrm{U}$ ', U m ustcontain a matrix U 1 (usually the orthogonal matrix retumed by SGGSVP). On exit, if $\mathrm{JOBU}=\mathrm{I}^{\prime}$, U contains the orthogonalm atrix $U$; if $\operatorname{JOBU}=\mathrm{U}$ ', U contains the productU $1 * \mathrm{U}$. If $\mathrm{JOBU}=\mathrm{N}$ ', U is not referenced.

LD U (input)
The leading dim ension of the aray U. LD U >= $\mathrm{max}(1, \mathrm{M})$ if $\mathrm{OOBU}=\mathrm{U}$ '; LD U >= 1 otherw ise.

V (input) On entry, if $\mathrm{JOBV}=\mathrm{V}^{\prime}, \mathrm{V}$ m ustcontain a matrix
V1 (usually the orthogonal $m$ atrix retumed by SGGSVP). On exit, if $30 \mathrm{BV}=I^{\prime}, \mathrm{V}$ contains the orthogonalm atrix V ; if $\mathrm{JOBV}=\mathrm{V}, \mathrm{V}$ contains the product $\mathrm{V} 1 * \mathrm{~V}$. If $\mathrm{JO} \mathrm{BV}=\mathrm{N}, \mathrm{V}$ is not referenced.

LD V (input)
The leading dim ension of the array V. LDV >= $\mathrm{max}(1, \mathrm{P})$ if $\mathrm{JOBV}=\mathrm{V}$; LDV $>=1$ otherw ise.

Q (input) $O n$ entry, if $J O B Q=Q$ ', Q mustcontain a matrix Q 1 (usually the orthogonal $m$ atrix retumed by SGGSVP). On exit, if $\mathcal{O B Q}=I^{\prime}, Q$ contains the orthogonalm atrix $Q$; if $J B Q=Q$ ', $Q$ contains the product $\mathrm{Q} 1 * \mathrm{Q}$. If $J O B Q=N$ ', $Q$ is not referenced.

LD Q (input)
The leading dim ension of the array $Q . L D Q>=$ $\max (1, N)$ if $J O B Q=Q ; L D Q>=1$ otherw ise.

W ORK (w orkspace)
dim ension ( $2 \star \mathrm{~N}$ )
NCYCLE (output)
The num ber of cycles required for convergence.
$\mathbb{N} F O$ (output)
= 0 : successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an ille-
galvahue.
= 1: the procedure does not converge after M A X IT cycles.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

stgsna -estim ate reciprocal condition num bers for specified eigenvalues and/or eigenvectors of a $m$ atrix pair $(A, B)$ in generalized realSchur canonical form (or of any $m$ atrix pair Q *A * Z ', $\mathrm{Q} * \mathrm{~B} * \mathrm{Z}$ I) w ith orthogonalm atrices Q and Z , where Z ' denotes the transpose of $Z$

## SYNOPSIS

```
SUBROUT\mathbb{NE STGSNA (OOB,HOW MNT,SELECT,N,A,LDA,B,LDB,VL,LDVL,}
    VR,LDVR,S,D\mathbb{F,MM,M,WORK,LWORK,INORK,INFO)}
CHARACTER * 1 OOB,HOW MNT
\mathbb{NTEGER N,LDA,LDB,LDVL,LDVR,MM ,M,LW ORK, INFO}
INTEGER IN ORK (*)
LOG ICAL SELECT (*)
REAL A (LDA,*), B (LDB,*), VL (LDVL,*), VR (LDVR,*), S (*),
D FF (*),W ORK (*)
SU BROUTINE STGSNA_64(JOB,HOW MNT,SELECT,N,A,LDA,B,LDB,VL,
    LDVL,VR,LDVR,S,D\mathbb{F},MM,M,W ORK,LW ORK,\mathbb{N ORK,INFO)}
CHARACTER * 1 JOB,HOW MNT
INTEGER*8N,LDA,LD B,LDVL,LDVR,MM ,M ,LW ORK,\mathbb{NFO}
INTEGER*8 \mathbb{N ORK (*)}
LOG ICAL*8 SELECT (*)
REAL A (LDA,*), B (LDB,*), VL (LDVL,*), VR (LDVR,*), S (*),
D F (*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE TGSNA (JOB,HOW MNT,SELECT, $\mathbb{N}], A,[L D A], B,[L D B], V L$, $[L D V L], V R,[L D V R], S, D \mathbb{F}, M M, M,[W O R K],[L W O R K],[\mathbb{W} O R K]$,

CHARACTER (LEN=1) :: OB, HOW MNT
$\mathbb{N}$ TEGER :: N, LDA, LDB,LDVL,LDVR, M M, M, LW ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{I N} O R K$
LOGICAL, D $\mathbb{I M} E N S I O N(:):$ SELECT
REAL,D $\mathbb{I}$ ENSION (:) :: S, D $\mathbb{E}, W$ ORK
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A,B,VL,VR

SU BROUTINE TG SNA_64 (DBB,HOW M NT,SELECT, $\mathbb{N}], A,[L D A], B,[L D B], V L$, $[[L D V L], V R,[L D V R], S, D \mathbb{F}, M M, M,[W O R K],[L W O R K],[\mathbb{W} O R K]$, [ $\mathbb{N} \mathrm{FO}]$ )

CHARACTER ( $L E N=1$ ) : : JOB, HOW M NT
$\mathbb{N}$ TEGER (8) :: N , LDA ,LDB,LDVL,LDVR,MM,M,LWORK, $\mathbb{N} F O$
$\mathbb{N}$ TEGER (8), D $\mathbb{M}$ ENSION (:) :: $\mathbb{I W}$ ORK
LOG ICAL (8), D $\mathbb{M}$ ENSION (:) :: SELECT
REAL,D $M$ ENSION (:) :: S, D $\mathbb{F}, W$ ORK
REAL,D $\mathbb{I M}$ ENSION (:,:) ::A,B,VL,VR

## C INTERFACE

\#include <sunperfh>
void stgsna (char job, charhow mnt, int *select, intn, float
*a, int lda, float *b, int ldb, float *vl, int
ldvl, float *vr, int ldvr, float*s, float *dif,
intm $m$, int* $m$, int*info);
void stgsna_64 (char job, char how m nt, long *select, long n, float *a, long lda, float *b, long ldb, float *vl, long ldvl, float *vr, long ldvr, float *s, float *dif, long m m , long *m, long *info);

## PURPOSE

stgsna estim ates reciprocal condition num bers for specified eigenvalues and/or eigenvectors of a m atrix pair ( $A, B$ ) in generalized realSchur canonical form (or of any m atrix pair Q *A *Z', Q *B *Z $)$ w ith orthogonalm atrices Q and Z , w here $\mathrm{Z}^{\prime}$ denotes the transpose of $Z$.
( $\mathrm{A}, \mathrm{B}$ ) m ustbe in generalized realSchur form (as retumed by SGGES), i.e.A is block uppertriangularw ith 1 toy -1 and 2-by-2 diagonalblocks.B is upper triangular.

## ARGUMENTS

## JOB (input)

Specifies w hether condition num bers are required
foreigenvalues (S) oreigenvectors (D $\mathbb{F}$ ):
$=\mathrm{E}$ ': foreigenvalues only (S);
$=\mathrm{V}$ ': foreigenvectors only (D $\mathbb{F})$;
= B ': forboth eigenvalues and eigenvectors ( S and D $\mathbb{F}$ ).

HOW MNT (input)
= 'A ': com pute condition num bers for all eigenpairs;
= $S^{\prime}$ : com pute condition num bers for selected eigenpairs specified by the array SELEC T .

## SELECT (input)

If HOW MNT = S', SELECT specifies the eigenpairs for which condition num bers are required. To select condition num bers for the eigenpair comesponding to a realeigenvalue w ( $\mathcal{\nu}$, SELECT ( $)$ m ustibe set to .TRU E ..To selectcondition num bers corresponding to a complex conjugate pair of eigenvaluesw ( 7 ) and w ( $\ddagger+1$ ), either SELECT ( 7 ) or SELECT ( $j+1$ ) or both, mustbe setto .TRUE .. If HOW MNT = A', SELECT is notreferenced.

N (input) The order of the square $m$ atrix pair ( $\mathrm{A}, \mathrm{B}$ ). $\mathrm{N} \quad>=$ 0.

A (input) The upperquasi-triangularm atrix $A$ in the pair $(A, B)$.

LD A (input)
The leading dim ension of the array A. LDA >= $\max (1, N)$.
$B$ (input) The upper triangularm atrix $B$ in the pair $(A, B)$.

LD B (input)
The leading dim ension of the array $B . L D B>=$ $\max (1, N)$.

VL (input)
If $\mathrm{JOB}=\mathrm{E}$ 'or B ', VL mustcontain left eigenvectors of $(A, B)$, comesponding to the eigenpairs specified by H OW M NT and SELEC T. The eigenvectors $m$ ust be stored in consecutive colum ns of $V \mathrm{~L}$, as retumed by STGEVC . If JOB $=V$ ', $V L$ is not referenced.

The leading dim ension of the array VL. LD VL $>=1$. If $\mathrm{JOB}=\mathrm{E}$ 'or $\mathrm{B}^{\prime}$, LDVL $>=\mathrm{N}$.

VR (input)
If $\mathrm{JOB}=\mathrm{E}$ 'or B ', VR m ust contain right eigenvectors of ( $A, B$ ), comesponding to the eigenpairs specified by HOW M NT and SELECT. The eigenvectors m ust be stored in consecutive colum ns ov VR, as retumed by $S T G E V C$. If $J O B=V ', V R$ is not referenced.
LDVR (input)
The leading dim ension of the array VR.LD VR >= 1 . If $\mathrm{JOB}=\mathrm{E}$ 'or B ', LDVR $>=\mathrm{N}$.

S (output)
If $J O B=E$ ' or $B$ ', the reciprocal condition num bers of the selected eigenvalues, stored in consecutive elem ents of the array. For a com plex conjugate pair of eigenvalues tw o consecutive ele$m$ ents of $S$ are set to the sam e value. Thus $S(\mathcal{j})$, D $\mathbb{F}(\mathcal{j})$, and the $j$ th colum ns ofVL and VR all comespond to the sam e eigenpair (butnot in general the jth eigenpair, unless alleigenpairs are selected). If $70 \mathrm{~B}=\mathrm{V}$ ', S is not referenced.

D $\mathbb{F}$ (output)
If $\mathrm{JO} \mathrm{B}=\mathrm{V}$ 'or B ', the estim ated reciprocal condition numbers of the selected eigenvectors, stored in consecutive elem ents of the array. For a complex eigenvector tw o consecutive elem ents of $D \mathbb{F}$ are set to the sam e value. If the eigenvalues cannot be reordered to com pute $D \mathbb{F}(\mathcal{I}), \mathrm{D} \mathbb{F}()$ is set to 0 ; this can only occurw hen the true value would be very sm allanyw ay. If $\mathrm{JO} B=\mathrm{E}, \mathrm{D} \mathbb{F}$ is not referenced.

M M (input)
The num berof elem ents in the arrays $S$ and $D \mathbb{F} . M M$ $>=\mathrm{M}$.

M (output)
The num berof elem ents of the arrays $S$ and $D \mathbb{F}$ used to store the specified condition num bers; for each selected realeigenvalue one elem ent is used, and for each selected com plex conjugate pair of eigenvalues, tw o elem ents are used. If HOW M NT = A', M is set to $N$.

W ORK (w orkspace)
If $\mathrm{JOB}=\mathrm{E}$ ', W ORK is not referenced. O therw ise,
on exit, if $\mathbb{N F} F=0, W$ ORK (1) retums the optim al LW ORK.

## LW ORK (input)

The dim ension of the anay $W$ ORK .LW ORK >= N. If JOB = V 'or B 'LW ORK >= 2*N * $N+2$ ) +16 .
If LW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim al size of theW ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IW ORK (w orkspace)
dim ension $(\mathbb{N}+6)$ If $\mathcal{O B}=\mathrm{E}^{\prime}, \mathbb{I N} \mathrm{ORK}$ is not referenced.
$\mathbb{N} F O$ (output)
=0: Successfulexit
$<0:$ If $\mathbb{N}$ FO $=-$ i, the $i$-th argum enthad an illegal value

## FURTHER DETAILS

The reciprocal of the condition num ber of a generalized eigenvalue $w=(a, b)$ is defined as $(w)=(\mu A v|\star * 2+\mu \mathrm{Bv}| \star * 2)^{* *}(1 / 2) /\left(\right.$ norm $(u){ }^{*}$ norm (v))
$w$ here $u$ and $v$ are the left and righteigenvectors of (A, B) comesponding to $\mathrm{w} ;|z|$ denotes the absolute value of the com plex num ber, and norm (u) denotes the 2 -norm of the vector u. The pair ( $a, b$ ) corresponds to an eigenvalue $w=a \nless \vDash$ $u$ Av/uBv) of the m atrix pair ( $A, B$ ). If both $a$ and $b$ equal zero, then $(A B)$ is singular and $S(I)=-1$ is retumed.

A n approxim ate errorbound on the chordal distance betw een the i-th computed generalized eigenvalue $w$ and the comesponding exacteigenvalue lam bda is hord (w , lam bda) <= EPS * norm (A , B) /S (I)
where EPS is the $m$ achine precision.

The reciprocal of the condition num ber D $\mathbb{F}$ (i) of right eigenvector $u$ and lefteigenvectorv comesponding to the generalized eigenvalue $w$ is defined as follow s:
a) If the $i$-th eigenvalue $w=(a, b)$ is real

Suppose U and V are orthogonal transform ations such that
$U *(A, B) * V=(S, T)=(a *)(b *)$
1

$$
(0 \mathrm{~S} 22),(0 \mathrm{~T} 22)
$$

n-1

$$
1 \mathrm{n}-1 \quad 1 \mathrm{n}-1
$$

Then the reciprocalcondition num berD $\mathbb{F}(i)$ is
D ifl( $(\mathrm{a}, \mathrm{b}),(\mathrm{S} 22, \mathrm{~T} 22))=\operatorname{sigm} \mathrm{a}-\mathrm{m}$ in $(\mathrm{Z} 1)$,
where sigm a-m in (Zl) denotes the sm allest singular value of the
$2(n-1)$-by-2 (n-1) m atrix

```
Zl= [kron (a, In-1) kron (1,S22) ]
            [kron(b, In-1) -kron(1,T22)].
```

H ere $\mathrm{In}-1$ is the identily $m$ atrix of size $\mathrm{n}-1 . \operatorname{kron}(X, Y)$ is the
$K$ roneckerproductbetw een the $m$ atrices $X$ and $Y$.

N ote that if the defaultm ethod for com puting D IF (i) is w anted
(see SLA TDF), then the param eter D FPD RI (see below) should be
changed from 3 to 4 (routine SLATD F (LJO B $=2 \mathrm{w}$ ill be used)).
See STG SY L form ore details.
b) If the $i$-th and (i+1)-th eigenvalues are com plex conjugate pair,

Suppose U and V are orthogonal transform ations such that
$U^{*}(A, B) * V=(S, T)=(S 11 *)(T 11 *$
) 2

$$
(0 \quad S 22),(0
$$

T22) $n-2$

$$
\begin{array}{llll}
2 & n-2 & n-2
\end{array}
$$

and (S11, T11) comesponds to the com plex conjugate eigenvalue
pair ( $w$, con $\dot{g}(w)$ ). There exist unitary $m$ atrices $U 1$ and V1 such
that
U 1 *S11*V1 = (s11 s12 ) and U 1 *T11*V $1=(\mathrm{t} 11 \mathrm{t} 12$
)

$$
\text { ( } 0 \text { s22) ( } 0 \text { t22 }
$$

)
w here the generalized eigenvalues $\mathrm{w}=\mathrm{s} 11$ t11 and con ${ }^{g}(w)=s 22$ t 22 .

Then the reciprocalcondition num berD $\mathbb{F}$ (i) is bounded by

$$
\mathrm{m} \text { in }(\mathrm{d} 1, \mathrm{~m} \text { ax }(1, \text { real(s11)/real(s22)|)*d2 ) }
$$

where, $\mathrm{d} 1=\mathrm{D}$ ifl( $(\mathrm{s} 11, \mathrm{t} 11),(\mathrm{s} 22, \mathrm{t} 22))=$ sigm a-m in $(\mathrm{Z} 1)$, where
Z 1 is the com plex 2 -by -2 m atrix

```
Z1 = [s11 -s22]
    [t11 -t22 ],
```

This is done by com puting (using realarithm etic) the roots of the characteristicalpolynom ialdet(Z1'* Z1 lam bda I),
where Z1'denotes the conjugate transpose of Z1 and $\operatorname{det}(X)$ denotes the determ inantof $X$.
and d2 is an upperbound on D ifl((S11,T11), (S22,T22)), ie.an
upperbound on sigm a-m in ( $Z 2$ ), where $Z 2$ is ( $2 n-2$ )-by-( $2 n-$ 2)

$$
\begin{aligned}
& \mathrm{Z} 2=[\mathrm{kron}(\mathrm{~S} 11 \text { ', In-2) } \mathrm{kron}(\mathrm{I}, \mathrm{~S} 22)] \\
& \text { [kron(T11', In-2) kron (12, T22) ] }
\end{aligned}
$$

$N$ ote that if the default $m$ ethod for com puting $D$ IF is w anted (see
SLA TD F), then the param eterD IFD R I (see below ) should be changed
from 3 to 4 (routine SLA TDF (INO B $=2$ w ill be used)). See STGSYL
form ore details.

For each eigenvaluekector specified by SE LEC T, D IF stores a Frobenius norm -based estim ate ofD ifl.

A $n$ approxim ate errorbound forthe i-th com puted eigenvector VL (i) orVR (i) is given by

> EPS * nom (A, B) /D IF (i).

See ref. [2-3] form ore details and further references.

B ased on contributions by
Bo K agstrom and PeterPorom aa, D epartm ent of C om puting Science,

Um ea U niversity, S-901 87 Um ea, Sw eden.

## References

[1] B .K agstrom ; A D irectM ethod forReordering Eigenvalues in the

G eneralized RealSchurForm of a RegularM atrix Pair (A, B), in

M S.M oonen etal (eds), LinearA lgebra for Large Scale and

Real-T im e A pplications, K luw erA cadem ic Publ. 1993, pp 195-218.
[2] B . K agstrom and P.Porom aa; C om puting Eigenspaces w th Specified E igenvalues of a R egularM atrix Pair ( $A$, $B$ ) and Condition

Estim ation: Theory, A lgorithm s and Softw are, ReportUM IN F-94.04,D epartm ent of Computing Science, Umea

U niversity, S-901 87 Um ea, Sw eden, 1994.A lso as LA PA CK W orking

N ote 87. To appear in N um ericalA lgorithm s, 1996.
[3] B . K agstrom and P . Porom aa, LA PA C K -Style A lgorithm s and Softw are
for Solving the Generalized Sylvester Equation and Estim ating the Separation betw een RegularM atrix Pairs, ReportU M $\mathbb{I N F}$ 93 23,

D epartm entofC om puting Science, Um ea U niversity, S-901 87 Um ea,

Sw eden, D ecem ber 1993, Revised A pril 1994, A lso as LAPACK W orking

N ote 75. To appear in A CM Trans. on M ath. Softw are, V ol 22,

No1, 1996.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

stgsyl-solve the generalized Sylvesterequation

## SYNOPSIS

```
SU BROUT\mathbb{NE STGSYL (TRANS,IJOB,M,N,A,LDA,B,LDB,C,LDC,D,LDD,}
    E,LDE,F,LDF,SCALE,D \mathbb{F,W ORK,LW ORK,IN ORK,INFO)}
CHARACTER * 1 TRANS
INTEGER LIOB,M,N,LDA,LDB,LDC, LDD, LDE, LDF, LW ORK,
NNFO
INTEGER IN ORK (*)
REAL SCALE,D F
REAL A (LDA,*), B (LDB ,*), C (LDC,*), D (LDD ,*), E (LDE,*),
F (LDF,*),WORK (*)
SUBROUTINE STGSYL_64(IRANS,IWOB,M,N,A,LDA,B,LDB,C,LDC,D,
    LDD,E,LDE,F,LDF,SCALE,D \mathbb{F,W ORK,LW ORK,IN ORK,INFO)}
CHARACTER * 1 TRANS
\mathbb{NTEGER*8 IJOB,M,N,LDA,LDB,LDC,LDD,LDE, LDF,LW ORK,}
\mathbb{NFO}
INTEGER*8 \mathbb{IN ORK (*)}
REALSCALE,D F
REAL A (LDA,*), B (LDB,*), C (LDC ,*), D (LDD,*), E (LDE,*),
F (LDF,*),W ORK (*)
```


## F95 INTERFACE

SU BROUTINE TGSYL (TRANS, IOB B, $\mathbb{M}], \mathbb{N}], A,[L D A], B,[L D B], C,[L D C]$, D, [LDD ], E, [LDE],F, [LDF],SCALE,D $\mathbb{F},\left[\begin{array}{l}\text { W ORK ], [LW ORK ], [IW ORK ], }\end{array}\right.$ [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1)::TRANS
$\mathbb{N} T E G E R::$ IJOB, $M, N, L D A, L D B, L D C, L D D, L D E, L D F, L W O R K$,
$\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I}$ ORK
REAL ::SCALE,D $\mathbb{F}$
REAL,D IM ENSION (:) ::W ORK
REAL,D IM ENSION (: : : : : A, B, C, D, E, F

SU BROUTINE TGSY L_64 (TRANS, IJOB, M ], $\mathbb{N}], A,[L D A], B,[L D B], C$, [LDC],D, [LDD ],E, [LDE],F, [LDF],SCALE,D $\mathbb{F},\left[\begin{array}{l}\text { ORK }],[L W ~ O R K], ~\end{array}\right.$ [ $\mathbb{I N}$ ORK], $[\mathbb{N F O}])$

CHARACTER (LEN=1) ::TRANS

LW ORK, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{I N}$ ORK
REAL ::SCALE,D $\mathbb{F}$
REAL,D IM ENSION (:) ::W ORK
REAL,D IM ENSION (:,:) ::A, B, C,D,E,F

## C INTERFACE

\#include <sunperfh>
void stgsyl(chartrans, int ijob, intm, int $n$, float *a, int lda, float *b, int ldb, float * c , int ldc, float*d, int ldd, float*e, int lde, float *f, int ldf, float *scale, float *dif, int*info);
void stgsyl 64 (Char trans, long ijob, long m, long n, float
*a, long lda, float *b, long ldb, float *c, long
ldc, float *d, long ldd, float *e, long lde, float
*f, long ldf, float *scale, float *dif, long
*info);

## PURPOSE

stgsylsolves the generalized Sylvesterequation:
$A * R-L * B=$ scale $* C$
$D * R-L * E=$ scale $* F$
(1)
where $R$ and $L$ are unknow $n m-b y-n m$ atrices, ( $A, D$ ), (B, E) and ( $C, F$ ) are given $m$ atrix pairs of size $m-b y-m, n-b y-n$ and $m$-by- $n$, respectively, $w$ ith realentries. ( $A, D$ ) and ( $B, E$ ) m ust be in generalized (real) Schur canonical form ,ie.A, $B$ are upperquasitriangular and D, E are upper triangular.

The solution $(\mathbb{R}, L)$ overw rites ( $C, F) .0<=$ SCALE $<=1$ is an output scaling factor chosen to avoid overflow .

In $m$ atrix notation (1) is equivalent to solve $\mathrm{Zx}=$ scale b , where $Z$ is defined as

$$
\begin{aligned}
Z= & {\left[\operatorname{kron}(\operatorname{In}, A) \operatorname{kron}\left(B^{\prime}, \operatorname{Im}\right)\right] } \\
& {\left[\operatorname{kron}(\operatorname{In}, D) \operatorname{kron}\left(E^{\prime}, \operatorname{Im}\right)\right] . }
\end{aligned}
$$

H ere $\mathbb{k}$ is the identily $m$ atrix of size $k$ and X 'is the transpose of $X$. kron ( $X, Y$ ) is the $K$ roneckerproduct.betw een the $m$ atrices $X$ and $Y$.

If TRANS = T', STGSY L solves the transposed system Z "y = scale*b, which is equivalent to solve for $R$ and $L$ in

$$
\begin{aligned}
& A^{\prime} * R+D^{\prime *} L=\text { scale * C } \\
& R B^{\prime}+L E^{\prime}=\text { scale * }(F)
\end{aligned}
$$

(3)

This case (TRANS = T) is used to com pute an one-norm -based estim ate of $D$ if $[(A, D),(B, E)]$, the separation betw een the $m$ atrix pairs $(A, D)$ and $(B, E)$, using SLA CON.

If IJO B >= 1, STG SY L com putes a Frobenius norm -based esti$m$ ate ofD if $[(A, D),(B, E)]$. That is, the reciprocal ofa low er
bound on the reciprocal of the sm allest singular value of $Z$.
See [1-2] form ore inform ation.

This is a level3 BLA S algorithm .

## ARGUMENTS

TRANS (input)
= N ', solve the generalized Sylvester equation (1). = T', solve the transposed 'system (3).

IJOB (input)
Specifies whatkind of functionality to be per-
form ed. $=0$ : solve (1) only.
$=1$ : The functionality of 0 and 3 .
$=2$ : The functionality of 0 and 4 .
$=3: 0$ nly an estim ate ofD if $[(A, D),(B, E)]$ is computed. (look ahead strategy IJO B $=1$ is used). $=4: 0$ nly an estim ate ofD if $[(A, D),(B, E)]$ is computed. ( SGECON on sub-system s is used). N ot referenced if TRANS = $T$ '.
$M$ (input) The order of the $m$ atrioes $A$ and $D$, and the row dim ension of the $m$ atrices $C, F, R$ and $L$.
$N$ (input) The order of the $m$ atrices $B$ and $E$, and the $c o l u m n$ dim ension of the $m$ atrioes $C, F, R$ and $L$.

A (input) The upperquasitriangularm atrix A.

LDA (input)
The leading dim ension of the aray A. LDA >= $m a x(1, M)$.
$B$ (input) The upperquasitriangularm atrix $B$.

LD B (input)
The leading dim ension of the aray B. LD B >= $\max (1, N)$.
C (input/output)
On entry, $C$ contains the right-hand-side of the first $m$ atrix equation in (1) or (3). On exit, if IJOB $=0,1$ or $2, C$ has been overw ritten by the solution R. If $\mathrm{IO} B=3$ or 4 and TRANS $=N^{\prime}$, C holds $R$, the solution achieved during the com putation of the $D$ if-estim ate.

LD C (input)
The leading dim ension of the aray C. LD C >= max (1, M).

D (input) The upper triangularm atrix D.

## LD D (input)

The leading dim ension of the aray D. LDD >= $m a x(1, M)$.

E (input) The upper triangularm atrix E .

LDE (input)
The leading dim ension of the aray E. LDE >= $\max (1, N)$.

F (input/output)
On entry, F contains the right-hand-side of the second $m$ atrix equation in (1) or (3). On exit, if IJO B $=0,1$ or $2, F$ has been overw ritten by the solution L. If IOB $=3$ or 4 and TRANS $=N^{\prime}, F$ holds L , the solution achieved during the com putation of the D if-estim ate.

LD F (input)
The leading dim ension of the array F. LDF >= max (1, M).

D $\mathbb{F}$ (output)
On exitSCALE is the reciprocalof a low er bound of the reciprocal of the $D$ if-function, ie. SCA LE is an upper bound of $D$ if $[(A, D),(B, E)]=$ sigm a_m in (Z), where $Z$ as in (2). If $I J O B=0$ or TRANS = T', SCALE is nottouched.
SCALE (output)
On exitSCA LE is the reciprocalofa lower bound of the reciprocal of the $D$ if-function, ie. SCA LE is an upper bound of $D$ if $[(A, D),(B, E)]=$ sigm a_m in (Z), where $Z$ as in (2). If $I J O B=0$ or TRANS = T', SCALE is not touched.

W ORK (w orkspace)
If $\mathrm{IJOB}=0, \mathrm{~W} O R K$ is not referenced. O therw ise, on exit, if $\mathbb{N} F O=0, W$ ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array $W$ ORK. LW ORK $>=1$. If $\mathrm{IJOB}=1$ or 2 and TRANS $=\mathrm{N}^{\prime}$, LW ORK $>=2 * \mathrm{M} * \mathrm{~N}$ 。

If LW ORK $=-1$, then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

IV ORK (w orkspace)
dim ension $M+N+2$ )
$\mathbb{I N F O}$ (output)
$=0$ : successfulexit
$<0:$ If $\mathbb{N}$ FO $=-$ - , the $i$-th argum enthad an illegal
value.
$>0$ : $A, D)$ and $(B, E)$ have com $m$ on orclose eigenvalues.

## FURTHER DETAILS

B ased on contributions by
Bo K agstrom and PeterPorom aa, D epartm ent of Com puting

## Science,

Um ea U niversity, S-901 87 Um ea, Sw eden.
[1] B . K agstrom and P. Porom aa, LA PA C K -Style A lgorithm s and Softw are
for Solving the Generalized Sylvester Equation and Estim ating the

Separation betw een RegularM atrix Pairs, ReportUM $\mathbb{N} F$ -

9323,
D epartm entofC om puting Science, Um ea U niversity, S-901 87 Um ea,

Sw eden, D ecem ber 1993, Revised A pril 1994, A lso as LAPACK W orking

N ote 75. To appear in A CM Trans. on M ath. Softw are, V ol 22,

No1, 1996.
[2] B . K agstrom , A Perturbation A nalysis of the G eneralized Sylvester

Equation (AR -LB,DR-LE ) = (C,F), SIAM J. M atrix A nal.

Appl, 15 (4):1045-1060,1994
[3] B . K agstrom and L. W estin, G eneralized Schur M ethods w th

C ondition Estim ators for Solving the G eneralized Sylvester

Equation, $\mathbb{E E E}$ Transactions on A utom atic C ontrol, V ol. 34,No.7,

July 1989, pp 745-751.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

stpoon - estim ate the reciprocal of the condition num ber of a packed triangular $m$ atrix $A$, in either the 1 -norm or the infinity-norm

## SYNOPSIS

```
SUBROUT\mathbb{NE STPCON NORM,UPLO,DIAG,N,A,RCOND,WORK,W ORK 2,\mathbb{NFO)}}\mathbf{N}\mathrm{ (NOM}
CHARACTER * 1 NORM,UPLO,DIAG
INTEGERN,\mathbb{NFO}
INTEGER W ORK2 (*)
REALRCOND
REALA (*),W ORK (*)
SUBROUT\mathbb{NE STPCON_64 NORM,UPLO,DIAG,N,A,RCOND,W ORK,W ORK2,}
    \mathbb{NFO)}
CHARACTER * 1 NORM,UPLO,DIAG
INTEGER*8 N,\mathbb{NFO}
\mathbb{NTEGER*8 W ORK2 (*)}
REAL RCOND
REALA (*),W ORK (*)
```


## F95 INTERFACE

```
SU BROUTINE TPCON \(\mathbb{N} O R M, U P L O, D \mathbb{I} G, N, A, R C O N D,[\mathbb{W} O R K],[W O R K 2]\), [ \(\mathbb{N} F O\) ])
CHARACTER (LEN=1)::NORM,UPLO,DIAG
\(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
\(\mathbb{I N} T E G E R, D \mathbb{I M}\) ENSION (:) ::W ORK2
REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::A,W ORK
```

SU BROUTINE TPCON_64 $\mathbb{N} O R M, U P L O, D \mathbb{I} G, N, A, R C O N D,[\mathbb{O} O R K],[W O R K 2]$, [ $\mathbb{N}$ FO ])

CHARACTER (LEN=1)::NORM,UPLO,DIAG
$\mathbb{N}$ TEGER (8) :: $\mathrm{N}, \mathbb{I N F O}$
$\mathbb{N} T E G E R(8), D \mathbb{I M}$ ENSION (:) ::W ORK2
REAL ::RCOND
REAL,D $\mathbb{I M}$ ENSION (:) ::A,W ORK

## C INTERFACE

\#include < sunperfh>
void stpcon (charnorm, char uplo, chardiag, int n, float
*a, float *rcond, int *info);
void stpcon_64 (charnorm , char uplo, char diag, long n, float *a, float *roond, long *info);

## PURPOSE

stpcon estim ates the reciprocal of the condition num berof a packed triangular matrix A, in ettherthe 1-norm orthe infinity-norm .

The norm ofA is com puted and an estim ate is obtained for norm (inv (A )), then the reciprocal of the condition num ber is com puted as
$\operatorname{RCOND}=1 /(\operatorname{norm}(A) *$ nom (inv (A))).

## ARGUMENTS

NORM (input)
Specifies w hether the 1-norm condition num ber or the infinity-norm condition num ber is required:
= 1'or $^{0}$ ': 1-nom ;
= I : $\quad$ Infinity-norm .

UPLO (input)
$=\mathrm{U}$ : A is uppertriangular;
$=\mathrm{L}$ ': A is low ertriangular.
D IA G (input)
$=\mathrm{N}$ : A is non-unit triangular;
$=\mathrm{U}$ ': A is unit triangular.
N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.

A (input) The upper or low er triangular m atrix A, packed colum nw ise in a linear array. The jth colum n of A is stored in the aray A as follow s: if UPLO = $U^{\prime}, A(i+(j-1) * j 2)=A(i, j)$ for $1<=i<=j$; if $\mathrm{UPLO}=\mathrm{L}, \mathrm{A}(i+(j-1) *(2 n-j) / 2)=A(i, j)$ for $j=i<=n$. IfD $\mathbb{A} G=U$ ', the diagonalelem ents of $A$ are not referenced and are assum ed to be 1 .

RCOND (output)
The reciprocal of the condition num ber of the $m$ atrix $A$, computed as RCOND $=1 /($ noim (A) * norm (inv (A))).
W ORK (w orkspace)
dim ension $(3 * N)$

W ORK 2 (w orkspace)
dim ension $\mathbb{N}$ )
$\mathbb{N} F O$ (output)
= 0 : successfulexit
<0: if $\mathbb{I N}$ FO $=-$ i, the $i$-th argum ent had an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

stpm $v$-perform one of the $m$ atrix-vector operations $x:=$ $A * x$, or $x:=A * x$

## SYNOPSIS

```
SU BROUTINE STPMV (UPLO,TRANSA,D IA G,N,A,Y, INCY)
CHARACTER * 1 UPLO,TRANSA,DIAG
INTEGERN,\mathbb{NCY}
REALA (*),Y (*)
SU BROUT\mathbb{NE STPM V_64(UPLO,TRANSA,D IAG,N,A ,Y, INCY)}
CHARACTER * 1UPLO,TRANSA,DIAG
INTEGER*8N,\mathbb{NCY}
REALA (*),Y (*)
F95 INTERFACE
    SUBROUT\mathbb{NE TPM V (UPLO, [TRANSA ],D IAG, N ],A,Y,[INCY ])}
    CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
    \mathbb{NTEGER ::N,\mathbb{NCY}}\mathbf{N}=\mp@code{N}
    REAL,D IM ENSION (:) ::A,Y
```



```
    CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
    \mathbb{NTEGER (8)::N,\mathbb{NCY}}\mathbf{}\mathrm{ (})=
    REAL,D IM ENSION (:) ::A,Y
```

C INTERFACE
\#include < sunperfh>
void stpom v (charuplo, char transa, chardiag, int n, float
*a, float *y, int incy);
void stpm v_64 (charuple, chartransa, char diag, long n, float*a, float *y, long incy);

## PURPOSE

stpm $v$ perform s one of them atrix-vector operations $x: A * x$, or $x:=A * x$, where $x$ is an $n$ elem entvectorand $A$ is an $n$ by $n$ unit, ornon-unit, upper or low er triangular $m$ atrix, supplied in packed form .

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies w hether the $m$ atrix is an upper or low er triangularm atrix as follow s:

UPLO = U'or ${ }^{\prime}$ ' A is an upper triangular $m$ atrix.

UPLO = L' or I' A is a lower triangular $m$ atrix.

U nchanged on exit.
TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA $=N^{\prime}$ or $h^{\prime} \mathrm{x}:=\mathrm{A}{ }^{*} \mathrm{x}$.

TRANSA = T'or $t^{\prime} x:=A * x$.

TRANSA $=$ C'or $匕^{\prime} \mathrm{x}:=\mathrm{A} * \mathrm{x}$.
U nchanged on exit.

TRANSA is defaulted to N 'forF $95 \mathbb{I N}$ TERFACE.

D IA G (input)
On entry, D IA G specifies whether ornotA is unit triangular as follow s:

D $\mathbb{I}$ G $=$ U'or $\mathrm{L}^{\prime} A$ is assum ed to be unit tri-
angular.
$D \mathbb{A G}=N$ 'or $h$ ' $A$ is notassum ed to be unit triangular.

U nchanged on exit.

N (input)
O n entry, $N$ specifies the order of the $m$ atrix A. $\mathrm{N}>=0$. U nchanged on exit.

A (input)
( $(n *(n+1)) / 2)$. Before entry with UPLO = $U$ ' or $L$ ', the amay A m ustcontain the upper triangularm atrix packed sequentially, colum $n$ by colum $n$, so thatA (1) contains a (1, 1), A (2) and $A(3)$ contain $a(1,2)$ and $a(2,2)$ respectively, and so on. Before entry w ith UPLO = L' or I', the anray A m ust contain the low er triangular m atrix packed sequentially, colum $n$ by colum $n$, so thatA (1) contains a (1,1),A(2) and $A(3)$ contain $a(2,1)$ and $a(3,1)$ respec tively, and so on. N ote thatw hen D IA G $=U U^{\prime}$ or $G$ ', the diagonal elem ents of A are notreferenced, butare assum ed to be unity. U nchanged on exit.

Y (input/output)
$(1+(n-1) * a b s(\mathbb{N} C Y))$. Before entry, the increm ented array $Y$ must contain the $n$ elem ent vectorx. O n exit, $Y$ is overw ritten $w$ ith the tranform ed vector $x$.
$\mathbb{N C Y}$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of $. \mathbb{N} C Y<>0$. U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

stprfs - provide errorbounds and backw ard error estim ates forthe solution to a system of linearequations $w$ th a triangular packed coefficientm atrix

## SYNOPSIS

```
SUBROUT\mathbb{NE STPRFS (UPLO,TRANSA,D IAG,N,NRHS,A,B,LDB,X,LDX,}
    FERR,BERR,W ORK,W ORK2,INFO)
CHARACTER * 1 UPLO,TRANSA,DIAG
INTEGERN,NRHS,LDB,LDX,INFO
INTEGER W ORK2 (*)
REALA (*),B (LDB,*),X (LDX ,*),FERR (*),BERR (*),W ORK (*)
SUBROUT\mathbb{NE STPRFS_64 (UPLO,TRANSA,D IAG,N,NRHS,A,B,LDB,X,LDX,}
        FERR,BERR,W ORK,W ORK2,NNO)
```

CHARACTER * 1 UPLO, TRANSA, DIAG
$\mathbb{N}$ TEGER*8N,NRHS,LDB,LDX, $\mathbb{N} F O$
$\mathbb{N}$ TEGER *8 W ORK 2 (*)
REALA (*), B (LDB,*), X (LDX,*),FERR (*), BERR (*), W ORK (*)

## F95 INTERFACE

SU BROUTINE TPRFS (UPLO, [TRANSA],D IA G ,N,NRHS,A,B, [LDB],X, [LDX], FERR, BERR, [WORK], [W ORK2], [ $\mathbb{N} F O]$ )

CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
$\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{M}$ ENSION (:) ::W ORK 2
REAL,D IM ENSION (:) ::A,FERR,BERR,W ORK
REAL,D $\mathbb{I}$ ENSION (: : : : : : B , X

SU BROUTINE TPRFS_64 (UPLO, [TRANSA],D $\mathbb{I A G}, N, N R H S, A, B,[L D B], X$, [LD X ],FERR,BERR, [W ORK], [W ORK 2], [ $\mathbb{N} F O$ ])

CHARACTER (LEN=1) ::UPLO,TRANSA,DIAG
$\mathbb{N} T E G E R(8):: N, N R H S, L D B, L D X, \mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I M}$ ENSION (:) ::W ORK2
REAL,D $\mathbb{M}$ ENSION (:) ::A,FERR,BERR,W ORK
REAL,D IM ENSION (:,:) ::B ,X

## C INTERFACE

\#include <sunperfh>
void stprfs (charuplo, chartransa, chardiag, int $n$, int nihs, float *a, float*b, int ldb, float * $x$, int ldx, float * ferr, float *berr, int *info);
void stprfs_64 (charuplo, chartransa, char diag, long n, long nrhs, float *a, float *b, long ldb, float *x, long ldx, float * ferr, float *berr, long *info);

## PURPOSE

stprfs provides errorbounds and backw ard error estim ates for the solution to a system of linear equations w ith a triangular packed coefficientm atrix.

The solution $m$ atrix $X$ m ustbe com puted by STPTRS or some other $m$ eans before entering this routine. STPRFS does not do iterative refinem entbecause doing so cannotim prove the backw ard error.

## ARGUMENTS

UPLO (input)
= U ': A is uppertriangular;
= LL': A is low ertriangular.

TRAN SA (input)
Specifies the form of the system of equations:
$=N$ : A * $\mathrm{X}=\mathrm{B} \quad$ N $\circ$ transpose)
= T': A**T * X = B (T ranspose)
= C': A **H * X = B (C onjugate transpose = Transpose)

TRANSA is defaulted to $N$ 'forF95 $\mathbb{N}$ TERFACE.

D IA G (input)
= N ': A is non-unit triangular;
$=\mathrm{U}: \mathrm{A}$ is unit triangular.
N (input) The order of them atrix $\mathrm{A} . \mathrm{N}>=0$.
NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the m atrices $B$ and X. NRH $S>=0$.

A (input) The upper or low er triangular m atrix A, packed colum nw ise in a linear anray. The jth colum n of A is stored in the array A as follow s: if UPLO = $U^{\prime}, A(i+(j-1) * j 2)=A(i, j)$ for $1<=i<=j$ if $\mathrm{UPLO}=\mathrm{L}, \mathrm{A}(i+(j-1) *(2 \star n-7) / 2)=A(i, 7)$ for $j=i<=n$. IfD $\mathbb{A} G=U '$, the diagonalelem ents of A are not referenced and are assum ed to be 1 .
$B$ (input) The righthand side $m$ atrix $B$.

LD B (input)
The leading dim ension of the aray B. LD B $>=$ $\max (1, N)$.
$X$ (input) The solution $m$ atrix $X$.

LD X (input)
The leading dim ension of the amay X . LD X >= $\max (1, \mathbb{N})$.

FERR (output)
The estim ated forw ard errorbound for each solution vector $X()$ ) the $j$ th colum n of the solution $m$ atrix X). If XTRUE is the true solution comesponding to $X(\mathcal{O})$, FERR ( $)$ is an estim ated upperbound for the $m$ agnitude of the largest ele$m$ ent in ( $X(\mathcal{D})-X$ TRUE) divided by the $m$ agnitude of the largestelem ent in $X(\mathcal{J})$. The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

## BERR (output)

The com ponentw ise relative backw ard error of each
solution vectorX ( $\mathcal{j}$ ) (i.e., the sm allest relative change in any elem entofA orB thatm akes X (i) an exactsolution).

W ORK (w orkspace)
dim ension $\left(3^{*} N\right)$
W ORK 2 (w orkspace)
dim ension ( N )
$\mathbb{N}$ FO (output)
= 0: successfulexit
$<0:$ if $\mathbb{N}$ FO $=-i$, the $i$-th argum enthad an illegalvalue

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

stpsv - solve one of the system sof equations $A$ * $x=b$, or A * $\mathrm{x}=\mathrm{b}$

## SYNOPSIS

```
SUBROUT\mathbb{NE STPSV (UPLO,TRANSA,DIAG,N,A Y, INCY)}
CHARACTER * 1 UPLO,TRANSA,DIAG
INTEGERN,\mathbb{NCY}
REALA (*),Y (*)
SU BROUT\mathbb{NE STPSV_64 (UPLO,TRANSA,D IAG,N,A ,Y, INCY)}
CHARACTER * 1 UPLO,TRANSA,DIAG
INTEGER*8N,\mathbb{NCY}
REALA (*),Y(*)
F95 INTERFACE
```



```
    CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
    \mathbb{NTEGER ::N,\mathbb{NCY}}\mathbf{N}=\mp@code{N}
    REAL,D IM ENSION (:) ::A,Y
```



```
    CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
    \mathbb{NTEGER (8)::N,\mathbb{NCY}}\mathbf{}\mathrm{ (})=
    REAL,D IM ENSION (:) ::A,Y
```

C INTERFACE
\#include < sunperfh>
void stpsv (Charuplo, char transa, chardiag, int n, float
*a, float * $y$, int incy);
void stpsv_64 (charuplo, chartransa, char diag, long n, float*a, float *y, long incy);

## PURPOSE

stpsv solves one of the system sof equations $A *_{x}=b$, or $A$ * $x=b$, where $b$ and $x$ are $n$ elem entvectors and $A$ is an $n$ by $n$ unit, ornon-unit, upper or low er triangular $m$ atrix, supplied in packed form .

N o test forsingularity or near-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.

## ARGUMENTS

UPLO (input)
On entry, UPLO specifies w hether the $m$ atrix is an upper or low er triangularm atrix as follow s:
$\mathrm{UPLO}=\mathrm{U}$ 'or G ' $A$ is an upper triangular $m$ atrix.

UPLO = L' or $\mathrm{I}^{\prime} \mathrm{A}$ is a lower triangular $m$ atrix.

U nchanged on exit.
TRANSA (input)
O n entry, TRANSA specifies the equations to be solved as follow s:

TRANSA $=N^{\prime}$ or $h^{\prime} A * x=b$.
TRANSA $=$ T'ort' $A * x=b$.

TRANSA $=C^{\prime}$ ort' $\mathrm{A}^{*} \mathrm{x}=\mathrm{b}$.
U nchanged on exit.
TRANSA is defaulted to N 'forF $95 \mathbb{I N}$ TERFACE.

D IA G (input)

On entry, D IA G specifies whether ornotA is unit triangular as follow s:

D $\mathbb{A G G}=\mathrm{U}$ 'or $\mathrm{L}^{\prime} \mathrm{A}$ is assum ed to be unit triangular.
$D \mathbb{A} G=N$ 'or $h$ ' $A$ is notassumed to be unit triangular.

U nchanged on exit.

N (input)
O $n$ entry, $N$ specifies the order of the $m$ atrix A. $\mathrm{N}>=0$. U nchanged on exit.
A (input)
$\left(\left(\mathrm{n}^{*}(\mathrm{n}+1)\right) / 2\right)$. Before entry w th $\mathrm{UPLO}=$ $U$ ' or U ', the array A m ustcontain the upper triangularm atrix packed sequentially, colum $n$ by colum n , so that A (1) contains a ( 1,1 ), A (2) and $A(3)$ contain $a(1,2)$ and $a(2,2)$ respectively, and so on. Before entry w ith UPLO = $\mathrm{L}^{\prime}$ or 1 ', the amay A m ust contain the low er triangular $m$ atrix packed sequentially, colum $n$ by colum n, so thatA (1) contains a (1,1), A (2) and $A$ ( 3 ) contain a $(2,1)$ and a $(3,1)$ respectively, and so on. N ote thatw hen D IA G $=$ U' or G ', the diagonal elem ents of A are notreferenced, but are assum ed to be unity. Unchanged on exit.

Y (input/output)
$(1+(n-1) * a b s(\mathbb{N} C Y))$. Before entry, the increm ented array $Y \mathrm{~m}$ ust contain the n elem ent righthand side vectorb. O $n$ exit, $Y$ is overw ritten $w$ ith the solution vector $x$.
$\mathbb{N} C Y$ (input)
On entry, $\mathbb{N} C Y$ specifies the increm ent for the elem ents of. $\mathbb{I N} C Y$ <> 0 . U nchanged on exit.

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS


## NAME

stptri-com pute the inverse of a real upper or low er triangularm atrix A stored in packed form at

## SYNOPSIS

```
SU BROUT\mathbb{NE STPTRI(UPLO,D IAG,N,A, IN FO)}
CHARACTER * 1 UPLO,DIAG
INTEGER N, INFO
REALA (*)
```



```
CHARACTER * 1 UPLO,D IAG
INTEGER*8 N,\mathbb{NFO}
REALA (*)
F95 INTERFACE
SUBROUT\mathbb{NE TPTRI(UPLO,D IAG,N,A,[NNFO])}
CHARACTER (LEN=1) ::UPLO,D IAG
\mathbb{NTEGER ::N,\mathbb{NFO}}\mathbf{N}=
REAL,D IM ENSION (:) ::A
SU BROUT\mathbb{NE TPTRI_64 (UPLO,D IA G ,N,A, [NNFO ])}
CHARACTER (LEN=1)::UPLO,D IAG
\mathbb{NTEGER (8)::N, INFO}
REAL,DIM ENSION (:) ::A
```

void stptri(charuplo, char diag, int n, float *a, int
*info);
void stptri_ 64 (charuplo, chardiag, long n, float *a, long
*info);

## PURPOSE

stptricom putes the inverse of a real upper or low er triangularm atrix A stored in packed form at.

## ARGUMENTS

```
UPLO (input)
    = U ': A is upper triangular;
    = L': A is low ertriangular.
```

D IA G (input)
$=\mathrm{N}: \mathrm{A}$ is non-unit triangular;
$=\mathrm{U}$ ': A is unit triangular.

N (input) The order of the m atrix $\mathrm{A} . \mathrm{N}>=0$.
A (input/output)
O n entry, the upper or low er triangularm atrix A, stored colum nw ise in a linearanay. The jth colum nofA is stored in the array A as follow $s$ : if $U P L O=U ', A(i+(j-1) * j 2)=A(i, 7)$ for $1<=\mathrm{i}<=\dot{j}$ ifUPLO = L', A (i+ $(\mathfrak{j} 1)^{*}((2 * \mathrm{n}-\mathrm{j} / 2)=$ A $(i, 1)$ for $j<=i<=n$. See below for further details. On exit, the (triangular) inverse of the original $m$ atrix, in the sam e packed storage format.
$\mathbb{I N F O}$ (output)
= 0: successfulexit
<0: if $\mathbb{I N}$ FO $=-i$, the $i$-th argum ent had an illegalvalue
$>0$ : if $\mathbb{N F O}=\mathrm{i}, \mathrm{A}(i, i)$ is exactly zero. The triangular $m$ atrix is singular and its inverse can notbe com puted.

## FURTHER DETAILS

A triangularm atrix A can be transferred to packed storage using one of the follow ing program segm ents:
$\mathrm{UPLO}=\mathrm{U} ': \quad \mathrm{UPLO}=\mathrm{L}^{\prime}:$
J $=1$
DO $2 \mathrm{~J}=1$, N
$\pi=1$
DO $2 \mathrm{~J}=1$, N
D○ $1 \mathrm{I}=1$, J
DO $1 \mathrm{I}=\mathrm{J}, \mathrm{N}$
$A(J C+I-1)=A(I, J) \quad A(J C+I-J)=$
A (I, N)
$\begin{array}{ll}1 & \text { CONTINUE } \\ \mathbb{C}=\mathrm{C}+\mathrm{J} & \mathbb{C O N T I N U E} \\ & \mathbb{C}=\mathrm{J}+\mathrm{N}-\mathrm{J}+\end{array}$
1
2 CONTINUE $\quad 2$ CONTINUE

## Contents

- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS


## NAME

```
stptrs - solve a triangularsystem of the form A * X = B
``` orA \({ }^{* *}\) T * \(\mathrm{X}=\mathrm{B}\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE STPTRS (UPLO,TRANSA,D IAG,N,NRHS,A,B,LDB,INFO)}
CHARACTER * 1 UPLO,TRANSA,D IAG
NNTEGERN,NRHS,LDB,NNFO
REALA (*),B (LDB,*)
SUBROUT\mathbb{NE STPTRS_64(UPLO,TRANSA,D IAG,N,NRHS,A,B,LDB,INFO)}
CHARACTER * 1 UPLO,TRANSA,D IAG
INTEGER*8N,NRHS,LDB,INFO
REALA (*),B (LDB,*)

```
F95 INTERFACE
    SU BROUTINE TPTRS (UPLO,TRANSA,D \(\mathbb{A} G, N, N R H S, A, B,[L D B],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
    \(\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O\)
    REAL,D \(\mathbb{I}\) ENSION (:) ::A
    REAL,D \(\mathbb{M}\) ENSION (:,:) ::B
    SU BROUTINE TPTRS_64 (UPLO,TRANSA,D \(\mathbb{I A G}, N, N R H S, A, B,[L D B],[\mathbb{N F O}])\)
    CHARACTER (LEN=1)::UPLO,TRANSA,D IA G
    \(\mathbb{N}\) TEGER (8) :: N,NRHS,LDB, \(\mathbb{N}\) FO
    REAL,D \(\mathbb{I M}\) ENSION (:) ::A
    REAL,D \(\mathbb{M}\) ENSION (:,:) ::B

\section*{C INTERFACE}
\#include <sunperfh>
void stptrs (charuplo, chartransa, chardiag, int \(n\), int nrhs, float *a, float *b, int ldb, int *info);
void stptrs_64 (charuplo, chartransa, char diag, long n, long nihs, float *a, float *b, long ldb, long *info);

\section*{PURPOSE}
stpters solves a triangular system of the form
w here \(A\) is a triangularm atrix of order \(N\) stored in packed form at, and B is an N boy-NRH S m atrix. A check ism ade to verify thatA is nonsingular.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) : A is upper triangular;
\(=\mathbb{L}^{\prime}: A\) is low er triangular.

TRANSA (input)
Specifies the form of the system of equations:
\(=\mathrm{N}^{\prime}: \mathrm{A} * \mathrm{X}=\mathrm{B} \quad\) (Notranspose)
\(=T{ }^{\prime}: A * * T * X=B \quad\) ( ranspose)
\(=C^{\prime}: A * * H * X=B \quad\) (C onjugate transpose \(=T\) ran spose)

D IA G (input)
\(=\mathrm{N}^{\prime}: A\) is non-unittriangular;
\(=\mathrm{U}\) : A is unit triangular.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of right hand sides, i.e., the num ber
of colum ns of the m atrix B. NRHS \(>=0\).

A (input) The upper or low er triangular matrix A, packed colum nw ise in a linear amay. The jth colum n of
A is stored in the aray A as follow s: ifUPLO =
\(U ', A(i+(j-1) \star j 2)=A(i, j)\) for \(1<=i<=j\) if
\(\mathrm{UPLO}=\mathrm{L}, \mathrm{A}\left(i+(j-1)^{\star}\left(2{ }^{\star} \mathrm{n}-\mathrm{j}\right) / 2\right)=A(i, j)\) for
\(j=\mathrm{i}<=\mathrm{n}\) 。

B (input/output)
On entry, the righthand side m atrix B. On exit, if \(\mathbb{N} F O=0\), the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
<0: if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the \(i\)-th diagonalelem ent of \(A\) is zero, indicating that the \(m\) atrix is singular and the solutions \(X\) have notbeen com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}

\section*{strans - transpose and scale source m atrix}

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE STRANS(PLACE,SCALE,SOURCE,M ,N,DEST)}
CHARACTER * 1 PLACE
\mathbb{NTEGERM,N}
REAL SCALE
REAL SOURCE (*),DEST (*)
SU BROUT\mathbb{NE STRANS_64(PLACE,SCALE,SOURCE,M,N,DEST)}
CHARACTER * 1 PLACE
\mathbb{NTEGER*8M,N}
REAL SCALE
REAL SOURCE (*),DEST (*)
F95 INTERFACE
SUBROUTINE TRANS ([PLACE],SCALE,SOURCE,M,N, DEST])
CHARACTER (LEN=1) ::PLACE
\mathbb{NTEGER ::M,N}
REAL ::SCALE
REAL,DIM ENSION (:) ::SOURCE,DEST
SU BROUT\mathbb{NE TRANS_64(PLACE],SCALE,SOURCE,M ,N, DDET ])}
CHARACTER (LEN=1) ::PLACE
\mathbb{NTEGER (8) ::M ,N}
REAL ::SCALE
REAL,DIM ENSION (:) ::SOURCE,DEST

```

\section*{C INTERFACE}
\#include < sunperfh>
void strans(charplace, float scale, float *source, int m, intn, float *dest);
void strans_64 (Charplace, float scale, float *source, long m, long n, float *dest);

\section*{PURPOSE}
strans scales and transposes the source m atrix. The N \(2 \times \mathrm{N} 1\) result is w rilten into SO U RCE when PLACE = I'or l', and DEST when PLACE = 0 'or \(b^{\prime}\) '.
PLACE = 'I'or \({ }^{1}\) ': SOURCE = SCALE * SOURCE'
PLACE = O'orb':DEST = SCALE * SOURCE'

\section*{ARGUMENTS}

PLACE (input)
Type of transpose. 'I'or i'for in-place, \(0^{\prime}\) or \(b\) 'for out-of-place. ' I ' is default.

SCALE (input)
Scale factor on the SO U RCE m atrix.
SOURCE (input/output)
\((M, N)\) on input. A may of \((N, M)\) on output if in-place transpose.
\(M\) (input)
N um ber of row \(s\) in the SO U RCE m atrix on input.
\(N\) (input)
N um berof colum ns in the SOU RCE m atrix on input.
DEST (output)
Scaled and transposed SOURCE matrix if out-ofplace transpose. N ot referenced if in-place transpose.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
stroon -estim ate the reciprocal of the condition num ber of a triangular \(m\) atrix \(A\), in either the 1 -norm or the infinity-norm

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE STRCON NORM ,UPLO,D IAG,N,A,LDA,RCOND,W ORK,W ORK2,}
\mathbb{NFO)}
CHARACTER * 1 NORM,UPLO,DIAG
\mathbb{NTEGER N,LDA,INFO}
\mathbb{NTEGERWORK2(*)}
REALRCOND
REALA (LDA,*),W ORK (*)
SUBROUT\mathbb{NE STRCON_64 NORM,UPLO,DIAG,N,A,LDA,RCOND,WORK,W ORK2,}
\mathbb{NFO )}
CHARACTER * 1 NORM,UPLO,DIAG
\mathbb{NTEGER*8N,LDA,INFO}
INTEGER*8 W ORK 2 (*)
REALRCOND
REALA (LDA,*),W ORK (*)

```

\section*{F95 INTERFACE}

SUBROUTINE TRCON \(\mathbb{N} O R M, U P L O, D \mathbb{I} G, N, A,[L D A], R C O N D,[W\) ORK ], [W ORK 2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::NORM,UPLO,D IAG
\(\mathbb{N}\) TEGER ::N,LDA, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) ::W ORK2
REAL ::RCOND

SU BROUTINE TRCON_64 \(\mathbb{N} O R M, U P L O, D \mathbb{I A G}, N, A,[L D A], R C O N D,[W O R K]\), [WORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::NORM, UPLO, DIAG
\(\mathbb{N} T E G E R(8):: N, L D A, \mathbb{N F O}\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: W\) ORK 2
REAL ::RCOND
REAL,D \(\mathbb{M}\) ENSION (:) ::W ORK
REAL,D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void strcon (char norm , charuplo, chardiag, int n, float *a, int lda, float *rcond, int *info);
void strcon_64 (charnom , char uplo, char diag, long n, float *a, long lda, float *rcond, long *info);

\section*{PURPOSE}
stroon estim ates the reciprocal of the condition num ber of a triangular \(m\) atrix \(A\), in either the 1 -norm orthe infinitynorm .

The norm ofA is com puted and an estim ate is obtained for norm (inv (A )), then the reciprocal of the condition num ber is com puted as
\(\operatorname{RCOND}=1 /(\operatorname{norm}(A) * \operatorname{norm}(\operatorname{inv}(A)))\).

\section*{ARGUMENTS}

N O RM (input)
Specifies w hether the 1 -norm condition num ber or the infinity-norm condition num ber is required:
= 1 'or \(0^{\prime}\) : 1-nom;
= I': Infinity-norm .

UPLO (input)
\(=\mathrm{U}^{\prime}: \mathrm{A}\) is uppertriangular;
\(=\mathbb{L}\) ': A is low er triangular.

D IA G (input)
\(=N^{\prime}: A\) is non-unittriangular;
\(=\mathrm{U}: \mathrm{A}\) is unittriangular.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input) The triangularm atrix \(A\). If \(\mathrm{UPLO}=\mathrm{U}\) ', the leading N -by N upper triangularpart of the array A contains the upper triangular matrix, and the strictly low ertriangularpartofA is not referenced. If UPLO = L', the leading N -by N lower triangular part of the array A contains the low er triangularm atrix, and the strictly uppertriangular part ofA is not referenced. IfD \(\mathbb{I A} G=U\) ', the diagonalelem ents ofA are also not referenced and are assum ed to be 1 .
LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, \mathbb{N})\).

RCOND (output)
The reciprocal of the condition number of the \(m\) atrix \(A\), com puted as RCOND \(=1 /(\) norm (A) * norm (inv (A))).

W ORK (w orkspace)
dim ension ( \(3 *\) N )

W ORK 2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{I N}\) FO = -i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
strevc - com pute som e or allof the right and/or left eigen-
vectors of a real upperquasi-triangularm atrix T

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE STREVC (SDE,HOW MNY,SELECT,N,T,LDT,VL,LDVL,VR,}
LDVR,MM,M,W ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,HOW MNY
INTEGERN,LDT,LDVL,LDVR,MM,M,INFO
LOG ICAL SELECT (*)
REAL T (LDT,*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
SU BROUT\mathbb{NE STREVC_64(SDE,HOW MNY,SELECT,N,T,LDT,VL,LDVL,VR,}
LDVR,MM,M,WORK,\mathbb{NFO)}
CHARACTER * 1SIDE,HOW MNY
INTEGER*8N,LDT,LDVL,LDVR,MM,M,\mathbb{NFO}
LOG ICAL*\& SELECT (*)
REALT (LDT,\star),VL (LDVL,*),VR (LDVR,\star),WORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE TREVC (SDE,HOW MNY,SELECT,N,T, [LDT],VL, [LDVL],VR, \([L D V R], M M, M,[\mathbb{N} O R K],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::SDE,HOW MNY
\(\mathbb{N} T E G E R:: N, L D T, L D V L, L D V R, M M, M, \mathbb{N} F O\)
LOG ICAL,D IM ENSION (:) ::SELECT
REAL,D \(\mathbb{M}\) ENSION (:) ::W ORK
REAL,D IM ENSION (:,:) ::T,VL,VR

SUBROUTINE TREVC_64 (SDE ,HOW M NY, SELECT,N,T, [LDT],VL, [LDVL], VR, [LDVR], M M , M , [WORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::SDE,HOW MNY
\(\mathbb{N}\) TEGER (8) :: N , LD T, LDVL, LDVR, M M , M , \(\mathbb{N}\) FO
LOGICAL (8), D IM ENSION (:) :: SELECT
REAL,D \(\mathbb{I M}\) ENSION (:) ::W ORK
REAL,D \(\mathbb{M}\) ENSION (: : : : : : T, VL, VR

\section*{C INTERFACE}
\#include < sunperfh>
void strevc (charside, char how my, int *select, int n, float *t, int ldt, float *Vl, int ldvl, float *Vr, int ldvr, intm \(m\), int * \(m\), int *info);
void strevc_64 (charside, charhow m ny, long *select, long n, float *t, long ldt, float *vl, long ldvl, float
*Vr, long ldvr, long m m , long *m, long *info);

\section*{PURPOSE}
strevc com putes som e orallof the rightand/or left eigenvectors of a realupperquasi-triangularm atrix \(T\).

The righteigenvectorx and the left eigenvector y of \(T\) corresponding to an eigenvalue w are defined by:
\[
\mathrm{T}^{\star} \mathrm{X}=\mathrm{w}^{\star} \mathrm{x}, \quad \mathrm{y}^{\star} \mathrm{T}=\mathrm{w}^{\star} \mathrm{y}^{\prime}
\]
w here y'denotes the conjugate transpose of the vectory.

If alleigenvectors are requested, the routine \(m\) ay either retum the \(m\) atrioes \(X\) and/or \(Y\) of rightor lefteigenvectors of \(T\), or the products \(Q * X\) and/or \(Q * Y\), where \(Q\) is an input orthogonal
\(m\) atrix. If \(T\) w as obtained from the real-Schur factorization of an originalm atrix \(A=Q * T * Q\) ', then \(Q * X\) and \(Q * Y\) are the \(m\) atrices of right or lefteigenvectors of \(A\).

T m ustbe in Schur canonical form (as retumed by SH SEQR), thatis, block upper triangularw ith 1-boy-1 and \(2-b y-2\) diagonalblocks; each 2-by-2 diagonal block has its diagonal elem ents equal and its off-diagonalelem ents of opposite sign. C orresponding to each 2 -by-2 diagonalblock is a com plex conjugate pair ofeigenvalues and eigenvectors; only one eigenvector of the pair is com puted, nam ely the one corresponding to the eigenvalue \(w\) ith posilive im aginary part.

\section*{ARGUMENTS}

STDE (input)
\(=\mathrm{R}\) : com pute righteigenvectors only;
= L ': com pute lefteigenvectors only;
\(=\mathrm{B}\) ': com pute both right and lefteigenvectors.

HOW MNY (input)
= A ': com pute all right and/or left eigenvec-
tors;
= B ': com pute all right and/or left eigenvec-
tors, and backtransform them using the input m atrices supplied in VR and/orV L; \(=\mathrm{S}\) ': com pute selected right and/or left eigenvectors, specified by the logicalamay SELEC T .

\section*{SELECT (input/output)}

If H OW M NY = S', SELEC T specifies the eigenvectors
to be com puted. IfHOW M NY = A 'or B', SELECT is
not referenced. To select the real eigenvector
corresponding to a realeigenvalue w ( \(\mathcal{j}\), SELECT ( \()\)
m ustibe set to TRUE.. To select the complex eigenvector comesponding to a com plex conjugate
pair w ( \(\mathcal{j}\) ) and w ( \(j+1\) ), either SELECT ( \(\mathcal{j}\) ) or
SELECT (j+1) must be setto TRUE.; then on exit
SELECT ( \(\mathfrak{j}\) ) is TRUE. and SELECT ( \(\mathfrak{j}+1\) ) is FALSE ..

N (input) The order of the m atrix \(\mathrm{T} \cdot \mathrm{N}>=0\).
T (input/output)
The upper quasi-triangular \(m\) atrix \(T\) in Schur canonicalform.

LD T (input)
The leading dim ension of the array T. LD T >= \(\max (1, \mathbb{N})\).

VL (input/output)
On entry, ifSDE = L'or B'and HOW M NY = B',
VL must contain an \(N\)-by N m atrix Q (usually the orthogonalm atrix \(Q\) of Schurvectors retumed by SH SEQR). On ex斗, if \(S \mathbb{D} E=\mathrm{L}\) 'or B',VL contains: if HOW M NY = A', the matrix Y of left eigenvectors of \(T\); VL has the sam equasi-low er triangular form as T'. If T ( \(i, i\) i) is a real eigenvalue, then the \(i\)-th columnVL (i) ofVL is its comesponding eigenvector. If \(T\) ( \(i: i+1, i: i+1\) ) is a 2 -by-2 block whose eigenvalues are complexconjugate eigenvalues of T , then VL (i)+sqıt(
1)*V L (i+1) is the com plex eigenvector comesponding to the eigenvalue \(w\) ith positive realpart. if HOW M NY = B', them atrix Q *Y ; ifHOW M NY = S', the lefteigenvectors of \(T\) specified by SELEC T, stored consecutively in the colum ns ofV \(L\), in the sam \(e\) order as their eigenvalues. A com plex eigenvector comesponding to a com plex eigenvalue is stored in tw o consecutive colum ns, the first holding the real part, and the second the im aginary part. If \(S \mathbb{D E}=\mathrm{R}, \mathrm{VL}\) is not referenced.

LD V L (input)
The leading dim ension of the aray VL. LDVL >= \(\max (1, N)\) if \(S \mathbb{D} E=L\) 'or \(B^{\prime} ;\) LDVL \(>=1\)
otherw ise.

VR (input/output)
On entry, if \(S \mathbb{D} E=R\) 'or \(B\) 'and HOW M NY = B', VR m ust contain an N -by -N m atrix Q (usually the orthogonalm atrix \(Q\) of Schurvectors retumed by SH SEQR). On exit, if \(S \mathbb{D} E=R\) 'or \(B\) ', \(V R\) contains: ifHOW MNY = A', the \(m\) atrix \(X\) of right eigenvectors of \(T\); VR has the sam equasi-upper triangular form as \(T\). If \(T(i, i)\) is a real eigenvalue, then the \(i\)-th colmmnVR (i) ofVR is its comesponding eigenvector. If \(T(i: i+1, i: i+1)\) is a 2 -by-2 block whose eigenvalues are complexconjugate eigenvalues of T, then VR (i)+sqrt(1)*VR(i+1) is the com plex eigenvector corresponding to the eigenvalue w th positive realpart. if HOW M NY = B', them atrix Q *X; if HOW MNY = S', the right eigenvectors of \(T\) specified by SELECT, stored consecutively in the colum ns of \(V R\), in the sam e order as their eigenvalues. A com plex eigenvector corresponding to a com plex eigenvahue is stored in tw o consecutive colum ns, the firstholding the real part and the second the im aginary part. If \(S \mathbb{D} E=\Sigma \prime, V R\) is not referenced.

LDVR (input)
The leading dim ension of the array VR. LDVR >= \(\max (1, N)\) if \(S \mathbb{D} E=R\) 'or B';LDVR >= 1 otherw ise.

M M (input)
The num berof colum ns in the arrays VL and/or VR. M M >=M.

M (output)
The num berof colum ns in the arrays VL and/or VR
actually used to store the eigenvectors. If HOW MNY = A 'or \(B\) ', M is set to N. Each selected real eigenvector occupies one colmm \(n\) and each selected com plex eigenvector occupies tw o colum ns.

W ORK (w orkspace)
dim ension ( \(3{ }^{*}\) N )
\(\mathbb{N}\) FO (output)
= 0 : successfinlexit
< 0 : if \(\mathbb{N F O}=-i\), the \(i\)-th argum ent had an illegal value

\section*{FURTHER DETAILS}

The algorithm used in this program is basically backw ard (forw ard) substitution, \(w\) th scaling to \(m\) ake the the code robustagainst possible overflow .

E ach eigenvector is norm alized so that the elem ent of largest \(m\) agnitude has \(m\) agnitude 1 ; here the \(m\) agnitude of a com plex num ber \((x, y)\) is taken to be \(|x|+|y|\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
strexc - reorder the real Schur factorization of a real
\(m\) atrix \(A=Q * T * Q * * T\), so that the diagonalblock of \(T \mathrm{w}\) ith
row index \(\mathbb{F S T}\) ism oved to row \(\mathbb{L S T}\)

\section*{SYNOPSIS}

```

CHARACTER * 1 COMPQ
\mathbb{NTEGERN,LDT,LDQ,\mathbb{FST,}|ST,INFO}
REALT (LDT,\star),Q (LDQ ,*),W ORK (*)

```

```

    INFO)
    CHARACTER * 1 COMPQ

```

```

REALT (LDT,*),Q (LDQ,*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E \operatorname{TREXC}\left(C O M P Q, N, T,[L D T], Q,[L D Q], \mathbb{F S T}, \mathbb{L} S T,\left[\begin{array}{l}\text { O RK ], }\end{array}\right.\right.\) [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::COMPQ
\(\mathbb{N} T E G E R:: N, L D T, L D Q, \mathbb{F} S T, \mathbb{L} S T, \mathbb{N F O}\)
REAL,D IM ENSION (:) ::W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::T, Q
SU BROUTINE TREXC_64 (COM PQ,N,T, [LDT],Q, [LDQ], \(\mathbb{F S T}, \mathbb{L} S T,\left[\begin{array}{l}\text { O ORK }],\end{array}\right.\) [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1)::COMPQ
\(\mathbb{N} T E G E R(8):: N, L D T, L D Q, \mathbb{F} S T, \Pi S T, \mathbb{N}\) FO
REAL,D \(\mathbb{M}\) ENSION (:) ::W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::T, Q

\section*{C INTERFACE}
\#include <sunperfh>
void strexc (char com pq, int \(n\), float *t, int ldt, float *q, int ldq, int *ifst, int *ilst, int *info);
void strexc_64 (char com pq, long n, float *t, long ldt, float *q, long ldq, long *ifst, long *ilst, long *info);

\section*{PURPOSE}
strexc reorders the real Schur factorization of a real m atrix \(\mathrm{A}=\mathrm{Q} * \mathrm{~T} * \mathrm{Q} * * \mathrm{~T}\), so that the diagonalblock of T w ith row index \(\mathbb{F} S T\) ismoved to row \(\Pi S T\).

The realSchurform \(T\) is reordered by an orthogonalsim ilarity transform ation \(Z * * T * T * Z\), and optionally the \(m\) atrix \(Q\) of Schurvectors is updated by postm ultiplying itw ith Z .

T m ustbe in Schurcanonical form (as retumed by SH SEQR ), that is, block upper triangularw ith 1-by-1 and 2 -by-2 diagonalblocks; each 2-by-2 diagonal block has its diagonal elem ents equal and its off-diagonalelem ents of opposite sign.

\section*{ARGUMENTS}

\section*{COMPQ (input)}
\(=\mathrm{V}\) : update the m atrix Q of Schurvectors;
\(=\mathrm{N}\) : do notupdate Q .

N (input) The order of the m atrix \(\mathrm{T} . \mathrm{N}>=0\).

T (input/output)
O \(n\) entry, the upperquasi-triangularm atrix \(T\), in
Schur Schur canonical form. On exit, the reor-
dered upper quasi-triangular \(m\) atrix, again in
Schur canonical form .

LD T (input)
The leading dim ension of the array \(\mathrm{T} . \mathrm{LD} \mathrm{T}>=\) \(\max (1, N)\).

Q (input) \(O n\) entry, if \(C O M P Q=V\) ', the \(m\) atrix \(Q\) of Schur vectors. On exit, if COMPQ = V', Q has been postm ultiplied by the orthogonal transform ation \(m\) atrix \(Z\) which reorders \(T\). If \(C O M P Q=N\) ', \(Q\) is not referenced.

LD Q (input)
The leading dim ension of the array \(Q . L D Q>=\) \(\max (1, N)\).

FST (input/output)
Specify the reordering of the diagonal blocks of
T. The block w ith row index \(\mathbb{F S T}\) is m oved to row

LST, by a sequence of transpositions betw een adjacent blocks. On exit, if \(\mathbb{F S T}\) pointed on entry to the second row of a 2 -by-2 block, it is changed to point to the firstrow; ILST alw ays points to the first row of the block in its final position (w hich \(m\) ay differ from its input value by +1 or -1 ). \(1<=\mathbb{F S T}<=\mathrm{N} ; 1<=\mathbb{L} S T<=\mathrm{N}\).

LST (input/output)
See the description of IFST .

W ORK (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0: successfulexit
<0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue
= 1: tw o adjacentblocks were too close to sw ap (the problem is very ill-oonditioned); \(\mathrm{T} m\) ay have been partially reordered, and ILST points to the first row of the currentposition of the block being \(m\) oved.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
strm \(m\)-perform one of the \(m\) atrix-m atrix operations \(B:=\) alpha*op (A ) *B , orB \(:=\) alpha*B *op (A )

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE STRMM (S\mathbb{DE,UPLO,TRANSA,D IAG,M,N,ALPHA,A,LDA,B,}}\mathbf{N},\textrm{L}
LD B )
CHARACTER * 1SDE,UPLO,TRANSA,D IAG
INTEGERM,N,LDA,LDB
REAL ALPHA
REALA (LDA,*),B (LDB,*)

```

```

    LD B)
    ```
CHARACTER * 1 SDE , UPLO, TRANSA,D IA G
\(\mathbb{N}\) TEGER*8 M , N,LDA, LD B
REALALPHA
REALA (LDA,*), B (LDB,*)

\section*{F95 INTERFACE}

SU BROUTINE TRMM (SIDE,UPLO, [TRANSA ],D IA G, \(\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A]\), B, [LD B])

CHARACTER (LEN=1)::SDE,UPLO,TRANSA,D IA G
\(\mathbb{N} T E G E R:: M, N, L D A, L D B\)
REAL ::ALPHA
REAL,D IM ENSION (:,:) ::A,B
SUBROUTINE TRMM_64 (SDEE,UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{M}], \mathbb{N}], A L P H A, A\), [LDA], B, [LDB])

CHARACTER ( \(L E N=1\) ) : : SDE E , UPLO, TRAN SA, D IA G
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{LDA}, \operatorname{LDB}\)
REAL ::ALPHA
REAL,D \(\mathbb{M}\) ENSION (: : : : : A , B

\section*{C INTERFACE}
\#include <sunperfh>
void stim m (char side, charuplo, chartransa, chardiag, int
m , int n , floatalpha, float *a, int lda, float
*b, int ldb);
void strm m _64 (charside, charuplo, chartransa, char diag, long m, long n, floatalpha, float*a, long lda, float*b, long lab);

\section*{PURPOSE}
strm \(m\) penform sone of the \(m\) atrix \(m\) atrix operations \(B:=\) alpha*op ( \(A\) ) *B , orB \(:=\) alpha*B*op (A ) where alpha is a scalar, \(B\) is an \(m\) by \(n m\) atrix, \(A\) is a unit, ornon-unit, upper or low er triangularm atrix and op (A ) is one of
\[
o p(A)=A \quad \text { or } \quad o p(A)=A^{\prime}
\]

\section*{ARGUMENTS}

SIDE (input)
On entry, SIDE specifies w hether op (A ) m ultiplies B from the leftor rightas follow s:
\(S \mathbb{D E}=\mathbb{L}\) 'or I' B := alpha*op (A )*B .
\(S \mathbb{D E}=\mathrm{R}^{\prime}\) or \(r^{\prime} \mathrm{B}:=\) alpha*B*op (A) .

U nchanged on exit.

UPLO (input)
O n entry, UPLO specifies w hether the m atrix A is an upper or low er triangularm atrix as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or U ' \(A\) is an upper triangular \(m\) atrix.
\(\mathrm{UPLO}=\mathrm{L}\) ' or \(\mathrm{I}^{\prime}\) A is a lower triangular \(m\) atrix.

U nchanged on exit.
TRANSA (input)
On entry, TRAN SA specifies the form ofop (A ) to be used in the \(m\) atrix \(m\) ultiplication as follow \(s\) :

TRANSA \(=N^{\prime}\) 'or \(h^{\prime}\) op(A) \(=A\).
TRANSA = T'or \(\mathrm{t}^{\prime}\) op(A) \(=\mathrm{A}^{\prime}\).

TRANSA = C'ort' op(A) = A'.
U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF95 \(\mathbb{I N}\) TERFACE.
D IA G (input)
O n entry, D IA G specifies w hether or notA is unit triangular as follow s:

D \(\mathbb{A} G=U\) 'or \(u\) ' \(A\) is assum ed to be unit triangular.
\(D \mathbb{A} G=N\) 'or \(h^{\prime} A\) is notassum ed to be unit triangular.

U nchanged on exit.

M (input)
O \(n\) entry, M specifies the num ber of row s of B. M >= 0 . U nchanged on exit.
\(N\) (input)
O \(n\) entry, \(N\) specifies the num ber of colum ns of \(B\). \(\mathrm{N}>=0\). U nchanged on exit.

ALPHA (input)
On entry, A LPH A specifies the scalar alpha.W hen alpha is zero then A is notreferenced and B need notbe setbefore entry. U nchanged on exit.

A (input)
REAL array ofD \(\mathbb{I}\) ENSION (LDA, \(k\) ), where \(k\) is \(m\) when \(S \mathbb{D} E=\mathbb{L}\) 'or \(\mathbb{I}\) ' and is \(n\) when \(S \mathbb{D} E=\) R 'or 'r'.

Before entry \(w\) ith UPLO = U 'or L ', the leading \(k\) by \(k\) upper triangularpart of the array A \(m\) ustcontain the upper triangularm atrix and the strictly low ertriangularpartofA is not refer-
enced.

Before entry w ith UPLO = L'or I', the leading \(k\) by \(k\) low ertriangularpart of the array \(A\) \(m\) ust contain the low ertriangularm atrix and the strictly uppertriangularpart of \(A\) is not referenced.

N ote thatw hen D IA G = U 'or L', the diagonal elem ents of A are notreferenced either, butare assum ed to be one. U nchanged on exit.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen \(S \mathbb{D E}=' \mathrm{~L}\) 'or \(\mathrm{I}^{\prime}\) then LD \(A>=\mathrm{max}(1, m)\), when \(S \mathbb{D E}=R^{\prime}\) or \(r^{\prime}\) then LDA \(>=\max (1, n)\). U nchanged on exit.

B (input/output)
REAL array ofD \(\mathbb{I M} E N S \mathbb{I O N}\) (LD B , n ). Before entry, the leading \(m\) by \(n\) partof the array \(B m u s t c o n-\) tain the m atrix B, and on exit is overw ritten
by the transform ed \(m\) atrix.

LD B (input)
O n entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program.
LD B \(>=\max (1, m)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
strm \(v\)-perform one of the \(m\) atrix-vector operations \(x:=\) A *x, or \(x:=A * x\)

\section*{SYNOPSIS}
```

SU BROUTINE STRMV (UPLO,TRANSA,D IAG,N,A,LDA,Y,INCY)
CHARACTER * 1 UPLO,TRANSA,DIAG
\mathbb{NTEGERN,LDA,}\mathbb{N}CY
REAL A (LDA,*),Y (*)
SU BROUT\mathbb{NE STRM V_64 (UPLO,TRANSA,D IAG ,N,A ,LDA,Y , INCY)}
CHARACTER * 1 UPLO,TRANSA,DIAG
INTEGER*8N,LDA,}\mathbb{N}C
REAL A (LDA,*),Y (*)

```
F95 INTERFACE
    SUBROUTINE TRMV (UPLO, [TRANSA],D \(\mathbb{I A G}, \mathbb{N}], A,[L D A], Y,[\mathbb{N} C Y])\)
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
    \(\mathbb{N} T E G E R:: N, L D A, \mathbb{N} C Y\)
    REAL,D \(\mathbb{I}\) ENSION (:) ::Y
    REAL,D IM ENSION (:,:) ::A
    SU BROUTINE TRM V_64 (UPLO, [TRANSA ],D \(\mathbb{I A} G, \mathbb{N}], A,[L D A], Y,[\mathbb{N} C Y])\)
    CHARACTER (LEN=1)::UPLO,TRANSA,D IA G
    \(\mathbb{N}\) TEGER (8) ::N,LDA, \(\mathbb{N} C Y\)
    REAL,D \(\mathbb{I M}\) ENSION (:) ::Y
    REAL,D IM ENSION (:,:) ::A

\section*{C INTERFACE}
\#include <sunperfh>
void stim v (char uplo, char transa, chardiag, int n, float
*a, int lda, float *y, int incy);
void strm v_64 (charuplo, chartransa, char diag, long n, float*a, long lda, float *y, long incy);

\section*{PURPOSE}
strm \(v\) perform s one of the \(m\) atrix-vector operations \(x:=A * x\), or \(\mathrm{x}:=A{ }^{*} \mathrm{x}\), where x is an n elem entvectorand \(A\) is an n by \(n\) unit, ornon-unit, upper or low ertriangularm atrix.

\section*{ARGUMENTS}

UPLO (input)
O n entry, UPLO specifies w hether the \(m\) atrix is an upper or low er triangularm atrix as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or \(\mathrm{U}^{\prime} \mathrm{A}\) is an upper triangular \(m\) atrix.
\(\mathrm{UPLO}=\mathrm{L}\) ' or I' A is a lower triangular m atrix.

U nchanged on exit.

TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} x:=A * x\).

TRANSA = T'ort' \(\mathrm{x}:=\mathrm{A}^{*} \mathrm{x}\).

TRANSA = C'orと' \(\mathrm{x}:=A^{*} \mathrm{x}\).

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
O n entry, D IA G specifies w hether ornotA is unit triangular as follow s:

D IA G = U'or L ' \(A\) is assum ed to be unit triangular.
\(D \mathbb{A G}=N\) 'or \(h^{\prime} A\) is notassum ed to be unit triangular.

U nchanged on exit.

N (input)
On entry, N specifies the order of the m atrix A . \(\mathrm{N}>=0\). U nchanged on exit.

A (input)
Before entry w ith UPLO = U 'or L ', the leading \(n\) by \(n\) upper triangular part of the array A m ust contain the upper triangular \(m\) atrix and the strictly low er triangular part of A is not referenced. Before entry with UPLO = L 'or 1', the leading \(n\) by \(n\) low er triangularpart of the aray A m ustcontain the low er triangular \(m\) atrix and the strictly upper triangularpartofA is not referenced. N ote thatw hen \(\mathrm{D} \mathbb{I A G}=\mathrm{U}\) ' or L ', the diagonal elem ents of \(A\) are not referenced either, but are assum ed to be unity. U nchanged on exit.

LDA (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program . LD A \(>=\) \(\max (1, n)\). U nchanged on exit.

Y (input/output)
\((1+(n-1) * a b s(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) must contain the \(n\) elem ent vectorx. On exit, \(Y\) is overw rilten \(w\) ith the tranform ed vectorx.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y . \mathbb{N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
stmfs - provide errorbounds and backw ard error estim ates for the solution to a system of linear equationsw ith a triangular coefficientm atrix

\section*{SYNOPSIS}
```

SUBROUTINE STRRFS (UPLO,TRANSA,D IAG,N,NRHS,A,LDA,B,LDB,X,
LD X,FERR,BERR,W ORK,W ORK 2,INFO)

```
CHARACTER * 1 UPLO, TRANSA, D IAG
\(\mathbb{N}\) TEGER N,NRHS,LDA,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGERWORK 2 (*)
REALA (LDA,\(\left.^{\star}\right), \operatorname{B}\left(\operatorname{LDB},{ }^{\star}\right), \mathrm{X}\left(\mathrm{LD} \mathrm{X},{ }^{\star}\right), \operatorname{FERR}\left({ }^{\star}\right), \operatorname{BERR}\left({ }^{\star}\right), \mathrm{W} O R K\left({ }^{*}\right)\)
SU BROUTINE STRRFS_64 (UPLO, TRANSA, D \(\mathbb{I A G}, N, N R H S, A, L D A, B, L D B, X\),
        LD \(\mathrm{X}, \mathrm{FERR}, \mathrm{BERR}, \mathrm{W}\) ORK, WORK \(2, \mathbb{N} F \mathrm{O}\) )
CHARACTER * 1 UPLO, TRANSA, DIAG
\(\mathbb{N}\) TEGER*8N,NRHS,LDA,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathrm{~W}\) ORK 2 ( \({ }^{*}\) )


\section*{F95 INTERFACE}

SU BROUTINE TRRFS (UPLO, [TRANSA],D IAG,N,NRHS,A, [LDA],B, [LDB],


CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION (:) ::W ORK2
REAL,D IM ENSION (:) ::FERR,BERR,W ORK
REAL,D \(\mathbb{I M}\) ENSION (: : : : : A, B, X

SU BROUTINE TRRFS_64 (UPLO, [TRANSA ], D \(\mathbb{I A G}, N, N R H S, A,[L D A], B,[L D B]\), \(\mathrm{X},[\operatorname{LDX}], F E R R, B E R R,[\mathbb{W}\) ORK \(],[\mathbb{W}\) ORK 2\(],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO, TRANSA,D IAG
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDA,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: W\) ORK2
REAL,D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK
REAL,D IM ENSION (:,:) ::A, B, X

\section*{C INTERFACE}
\#include <sunperfh>
void strnfs (charuplo, chartransa, chardiag, int n, int nihs, float*a, int lda, float*b, int ldb, float
* x , int ldx, float * ferr, float *benr, int *info);
void strnfs_64 (charuplo, chartransa, char diag, long n, long nihs, float *a, long lda, float *b, long ldb, float *x, long ldx, float * ferr, float *berr, long *info);

\section*{PURPOSE}
strmfs provides errorbounds and backw ard enror estim ates for the solution to a system of linear equations \(w\) th a triangular coefficientm atrix.

The solution \(m\) atrix \(X\) m ustbe com puted by STRTRS or some other \(m\) eans before entering this routine. STRRFS does not do iterative refinem entbecause doing so cannotim prove the backw ard error.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) : A is uppertriangular;
\(=\mathrm{L}\) ': A is low er triangular.
TRANSA (input)
Specifies the form of the system of equations:
\(=N\) : A * \(\mathrm{X}=\mathrm{B} \quad \mathrm{N} \circ\) transpose)
\(=T: A * * T\) * \(X=B \quad(T\) ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose \(=\mathrm{T}\) ranspose)

TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.
\(=\mathrm{N}: A\) is non-unit triangular;
\(=\mathrm{U}: \mathrm{A}\) is unit triangular.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber
of colum ns of the m atrioes B and X. NRH S \(>=0\).

A (input) The triangularm atrix A. IfU PLO = U', the lead-
ing N -by -N upper triangularpart of the aray A
contains the upper triangular \(m\) atrix, and the strictly low ertriangularpart of A is notrefer-
enced. IfUPLO = \(\mathbb{L}\) ', the leading N -oy N low er triangular part of the anray A contains the low er triangularm atrix, and the strictly upper triangular part ofA is notreferenced. IfD IA \(G=U^{\prime}\) ', the diagonalelem ents of A are also not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

B (input) The righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).
\(X\) (input) The solution \(m\) atrix \(X\).

\section*{LD X (input)}

The leading dim ension of the aray X. LD X >= \(\max (1, N)\).

\section*{FERR (output)}

The estim ated forw ard emorbound for each solution vectorX ( 1 ) the \(j\) th colum \(n\) of the solution \(m\) atrix \(X\) ). If XTRUE is the true solution corresponding to \(X(\mathcal{J}), \operatorname{FERR}(\mathcal{I})\) is an estim ated upperbound forthem agnitude of the largest ele\(m\) entin \((X(j)-X\) TRUE) divided by the \(m\) agnitude of the largestelem entin X ( 7 ). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slightoverestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vectorX (i) (i.e., the sm allest relative
change in any elem entofA orB thatm akes X ( 7 ) an exactsolution).

W ORK (w orkspace)
dim ension ( \(3 * N\) )

W ORK2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i-\) th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
strsen -reorder the real Schur factorization of a real
\(m\) atrix \(A=Q * T * Q * * T\), so that a selected chuster of eigenvalues appears in the leading diagonalblocks of the upper quasi-triangularm atrix \(T\),

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE STRSEN (JOB,COMPQ,SELECT,N,T,LDT,Q,LDQ,W R,W I,M,}
S,SEP,W ORK,LW ORK,IN ORK,LIN ORK,INFO)
CHARACTER * 1 JOB,COMPQ

```

```

INTEGER IN ORK (*)
LOG ICAL SELECT (*)
REALS,SEP
REALT (LDT,*),Q (LDQ **),W R (*),W I(*),W ORK (*)
SU BROUT\mathbb{NE STRSEN_64(ODB,COMPQ,SELECT,N,T,LDT,Q,LDQ,W R,W I,}
M,S,SEP,W ORK,LW ORK,IN ORK,LIN ORK,\mathbb{NFO)}
CHARACTER * 1 JOB,COMPQ

```

```

INTEGER*8 IN ORK (*)
LO G ICAL*8 SELECT (*)
REALS,SEP
REALT (LDT,*),Q (LDQ ,*),W R (*),W I(*),W ORK (*)

```
F95 INTERFACE
    SU BROUTINE TRSEN (JOB,COMPQ,SELECT,N,T, [LDT],Q, [LDQ],WR,W I,


CHARACTER (LEN=1) :: JOB,COMPQ
\(\mathbb{N}\) TEGER :: N,LDT,LDQ,M,LW ORK,LIV ORK, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
LOG ICAL,D IM ENSION (:) ::SELECT
REAL :: S, SEP
REAL,D \(\mathbb{M}\) ENSION (:) ::WR,W I,W ORK
REAL,D \(\mathbb{I M}\) ENSION (:,:) ::T,Q

SU BROUTINE TRSEN_64 (OB B,COM PQ,SELECT,N,T, [LD T],Q, [LDQ ],W R, W I, M , S, SEP, [W ORK ], [LW ORK ], [WW ORK ], [LIN ORK ], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) :: JOB,COM PQ
\(\mathbb{N}\) TEGER (8) :: N, LD T,LDQ,M,LW ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
LO G ICAL (8), D IM ENSIO N (:) :: SELECT
REAL ::S,SEP
REAL,D \(\mathbb{M}\) ENSION (:) ::WR,W I,W ORK
REAL,D \(\mathbb{M}\) ENSION (:,:) ::T,Q

\section*{C INTERFACE}
\#include <sunperfh>
void strsen (char job, char com pq, int *select, intn, float
*t, int ldt, float *q, intldq, float *w r, float
*W i, int *m, float *s, float *sep, int *info);
void strsen_64 (char j.b. char com pq, long *select, long n, float *t, long ldt, float *q, long ldq, float *w r, float * \({ }_{\mathrm{w}} \mathrm{i}\), long \({ }^{\mathrm{m}} \mathrm{m}\), float *s, float *sep, long *info);

\section*{PURPOSE}
strsen reorders the real Schur factorization of a real \(m\) atrix \(A=Q * T * Q * * T\), so that a selected chuster of eigenvalues appears in the leading diagonalblocks of the upper quasi-triangularm atrix \(T\), and the leading colum ns of form an orthonorm albasis of the corresponding right invariant subspace.

Optionally the routine com putes the reciprocal condition num bers of the cluster of eigenvalues and/or the invariant subspace.

T m ustbe in Schurcanonical form (as retumed by SH SEQR), that is, block uppertriangularw ith 1 -by-1 and 2-by-2 diagonalblocks; each 2-by-2 diagonal block has its diagonal elem nts equal and its off-diagonal elem ents of opposite sign.

\section*{ARGUMENTS}
\(J O B\) (input)
Specifies w hether condition num bers are required
for the cluster ofeigenvalues (S) orthe invari-
antsubspace (SEP):
= N ': none;
= E': foreigenvalues only (S);
\(=\mathrm{V}\) ': for invariant subspace only (SEP);
\(=B\) ': forboth eigenvalues and invariant subspace (S and SEP).

COMPQ (input)
= V ': update the m atrix Q ofSchurvectors;
= N ': do notupdate Q .
SELECT (input)
SELEC T specifies the eigenvalues in the selected cluster. To select a real eigenvalue w ( \(\mathcal{j}\), SELECT ( \(\mathcal{j}\) ) must be set to \(\mathrm{w}(\mathcal{j})\) and \(\mathrm{w}(j+1)\), corresponding to a 2 -by-2 diagonalblock, either SELECT ( \(\ddagger\) ) orSELECT ( \(\ddagger+1\) ) orboth m ust be set to either both included in the cluster or both excluded.

N (input) The order of the m atrix \(\mathrm{T} . \mathrm{N}>=0\).

T (input/output)
On entry, the upperquasi-triangularm atrix T , in Schur canonical form. On exit, T is overw ritten by the reordered \(m\) atrix \(T\), again in Schur canonical form, w ith the selected eigenvalues in the leading diagonalblocks.

LD \(T\) (input)
The leading dim ension of the array \(\mathrm{T} . \mathrm{LD} \mathrm{T}>=\) \(\max (1, \mathbb{N})\).

Q (input) \(O n\) entry, if \(C O M P Q=V\) ', them atrix \(Q\) of Schur vectors. On exit, if \(C O M P Q=V\) ', Q has been postm ultiplied by the orthogonal transform ation \(m\) atrix which reorders \(T\); the leading \(M\) colum ns of \(Q\) form an orthonorm al basis for the specified invariant subspace. If COM PQ \(=N^{\prime}, \mathrm{Q}\) is not referenced.

LD Q (input)
The leading din ension of the array \(\mathrm{Q} . \mathrm{LD} \mathrm{Q}>=1\);
and if \(C O M P Q=V\) ', LD \(Q>=N\).

\section*{W R (output)}

The realand im aginary parts, respectively, of the reordered eigenvalues of \(T\). The eigenvalues are stored in the sam e order as on the diagonal of \(T\), w ith \(\mathrm{W} R(i)=T(i, i)\) and, if \(T(i: i+1, i: i+1)\) is a 2 -by-2 diagonalblock, \(W \mathrm{I}(\) (i) \(>0\) and \(\mathrm{W} \mathrm{I}(\mathrm{i}+1)=\) -W I(i). N ote that if a com plex eigenvalue is sufficiently ill-conditioned, then its value \(m\) ay differ significantly from its value before reordering.

\section*{W I (output)}

See the description of R .
M (output)
The dim ension of the specified invariant subspace. \(0<=M<=N\).

S (output)
If \(J 0 B=E\) 'or \(B ', S\) is a lower bound on the reciprocal condition num ber for the selected clusterofeigenvalues. S cannot underestim ate the true reciprocal condition num berby \(m\) ore than a factorofsqut \(\mathbb{N})\). If \(M=0\) or \(N, S=1\). If \(\operatorname{OBB}=\) N 'or V ', S is not referenced.

SEP (output)
If \(\mathrm{DOB}=\mathrm{V}\) 'or B ', SEP is the estim ated reciprocal condition number of the specified invariant subspace. IfM \(=0\) orN, \(\mathrm{SEP}=\) norm ( T ). If \(\mathrm{JO} B=\) N 'or E ', SEP is not referenced.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array W ORK. If JOB = N ', LW ORK \(>=\max (1, N)\); if \(\operatorname{JOB}=\mathrm{E}^{\prime}, \mathrm{LW}\) ORK \(\left.>=\mathrm{M} * \mathbb{N}-\mathrm{M}\right)\); if \(\mathrm{JOB}=\mathrm{V}\) 'or \(B \prime\) ', LW ORK \(>=2 \star M * \mathbb{N}-M)\).

If LW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

IV ORK (w orkspace/output)
If \(\mathrm{JOB}=\mathrm{N}\) 'or E ', \(\mathbb{I W} O R K\) is not referenced.

LIV ORK (input)
The dim ension of the array \(\mathbb{I N} O R K\). If \(\mathrm{OB}=\mathrm{N}\) 'or E', LIN ORK >=1; if \(\mathrm{O} O \mathrm{~B}=\mathrm{V}\) 'or \(\mathrm{B}^{\prime}\), LIN ORK \(>=\) \(M *(N+M)\).

If \(L \mathbb{I V} O R K=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the \(\mathbb{I N}\) ORK array, retums this value as the first entry of the \(\mathbb{I W}\) ORK arnay, and no errorm essage related to LIV ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an illegalvalue
\(=1\) : reordering of T failed because som e eigenvalues are too close to separate the problem is very ill-conditioned); T m ay have been partially reordered, and W R and W I contain the eigenvalues in the same order as in T; S and SEP (if requested) are set to zero.

\section*{FURTHER DETAILS}

STRSEN first collects the selected eigenvalues by com puting an orthogonal transform ation \(Z\) to \(m\) ove them to the top left comerof T. In otherw ords, the selected eigenvalues are the eigenvalues of T11 in:
\[
\begin{gathered}
\mathrm{Z} * \mathrm{~T} * \mathrm{Z}=(\mathrm{T} 11 \mathrm{~T} 12) \mathrm{n} 1 \\
(0 \mathrm{~T} 22) \mathrm{n} 2 \\
\mathrm{n} 1 \mathrm{n} 2
\end{gathered}
\]
w here \(\mathrm{N}=\mathrm{n} 1+\mathrm{n} 2\) and Z ' m eans the transpose of Z . The first n1 colum ns of \(Z\) span the specified invariant subspace of \(T\).

If \(T\) has been obtained from the realSchur factorization of a matrix \(\mathrm{A}=\mathrm{Q} * \mathrm{~T} * \mathrm{Q}\) ', then the reordered realSchur factorization of \(A\) is given by \(A=(Q * Z) *(Z \text { * } T * Z)^{*}(Q * Z)\) ', and the first n 1 colum ns of \(\mathrm{Q} * \mathrm{Z}\) span the corresponding invariant subspace ofA .

The reciprocalcondition num ber of the average of the eigenvalues of T11 m ay be retumed in S.S lies betw een 0 (very badly conditioned) and 1 (very w ell conditioned). It is com puted as follow s. Firstw e com pute \(R\) so that
\[
\begin{gathered}
P=\left(\begin{array}{l}
\text { I R }) \mathrm{n} 1 \\
(0 \quad 0) n 2 \\
\mathrm{n} 1 \mathrm{n} 2
\end{array}\right.
\end{gathered}
\]
is the projector on the invariant subspace associated w ith T11. R is the solution of the Sylvesterequation:
\[
\mathrm{T} 11 * \mathrm{R}-\mathrm{R} * \mathrm{~T} 22=\mathrm{T} 12 .
\]

LetF-norm (M) denote the Frobenius-norm of \(M\) and 2 -norm \(M\) ) denote the tw o-norm of . Then \(S\) is com puted as the low er bound
\[
(1+F-\operatorname{nom}(R) * * 2)^{\star *}(-1 / 2)
\]
on the reciprocal of 2 -norm ( P ), the true reciprocal condition number. S cannotunderestim ate \(1 / 2\)-norm (P) by m ore than a factorof sqrit \()\).

A n approxim ate errorbound for the com puted average of the eigenvalues of T11 is
```

EPS * norm (T)/S

```
where EPS is the \(m\) achine precision.

The reciprocal condition num ber of the right invariant subspace spanned by the firstn1 colum nsofZ (orofQ *Z) is retumed in SEP. SEP is defined as the separation of T11 and T22:
\[
\operatorname{sep}(T 11, T 22)=\text { sigm } a-m \text { in (C ) }
\]
where sigm a-m in (C) is the sm allest singularvalue of the \(\mathrm{n} 1 \star_{\mathrm{n}} 2-\)-by-n1*n2 m atrix
\[
C=\operatorname{kprod}(I(n 2), T 11)-\operatorname{kprod}(\operatorname{transpose}(I 22), I(n 1))
\]

I( \(m\) ) is an \(m\) by \(m\) identity \(m\) atrix, and kprod denotes the K ronecker product. W e estim ate sigm a-m in (C ) by the reciprocalofan estim ate of the 1 -norm of inverse (C). The true reciprocal 1-norm of inverse ( ) cannot differ from sigm a\(m\) in (C) by \(m\) ore than a factorof sqrt (n1*n2).

W hen SEP is sm all, sm all changes in T can cause large changes in the invariant subspace. A \(n\) approxim ate bound on the maxim um angularerrorin the com puted right invariant subspace is

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
strsm -solve one of the matrix equations op (A )*X = alpha*B , or \(X\) *op ( A ) = alpha*B

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE STRSM(SDE,UPLO,TRANSA,DIAG,M,N,ALPHA,A,LDA,B,}
LD B )
CHARACTER * 1SDE,UPLO,TRANSA,D IAG
INTEGER M ,N,LDA,LDB
REAL ALPHA
REALA (LDA,*),B (LDB,*)
SUBROUT\mathbb{NE STRSM _64(S\mathbb{DE,UPLO,TRANSA,D IAG,M,N,ALPHA,A,LDA,B,}}\mathbf{N},\textrm{N},\textrm{A}
LD B)

```
CHARACTER * 1 SDE , UPLO, TRANSA,D IA G
\(\mathbb{N}\) TEGER*8 M , N,LDA, LD B
REALALPHA
REALA (LDA,*), B (LDB,*)

\section*{F95 INTERFACE}

SU BROUTINE TRSM (SDE, UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{M}], \mathbb{N}], A L P H A, A,[L D A]\), B, [LD B])

CHARACTER (LEN=1)::SDE,UPLO,TRANSA,D IAG
\(\mathbb{N} T E G E R:: M, N, L D A, L D B\)
REAL ::ALPHA
REAL,D \(\mathbb{M}\) ENSION (: : : : :: A, B
SUBROUTINE TRSM_64 (SDE,UPLO, [TRANSA],D \(\mathbb{I} G, \mathbb{M}], \mathbb{N}], A L P H A, A\), [LDA], B, [LDB])

CHARACTER ( \(L E N=1\) ) : : SDE E , UPLO, TRAN SA, D IA G
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{LDA}, \operatorname{LDB}\)
REAL ::ALPHA
REAL,D \(\mathbb{M}\) ENSION (: : : : : A , B

\section*{C INTERFACE}
\#include <sunperfh>
void strsm (char side, charuplo, chartransa, chardiag, int m , int n , float alpha, float*a, int lda, float
*b, int ldb);
void strsm _64 (char side, charuplo, char transa, char diag, long m, long n, floatalpha, float*a, long lda, float*b, long lab);

\section*{PURPOSE}
strsm solves one of the \(m\) atrix equations op (A ) \(A X=\) alpha*B, or \(X\) *op ( \(A\) ) = alpha*B w here alpha is a scalar, \(X\) and \(B\) are \(m\) by \(n m\) atriges, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix and op (A ) is one of
\[
\text { op }(A)=A \text { or } o p(A)=A^{\prime}
\]

Them atrix X is overw rilten on B .

\section*{ARGUMENTS}

SID E (input)
O n entry, SID E specifiesw hetherop (A ) appears on the left or rightofX as follow s:

SIDE = L'or I' op (A ) *X = alpha*B.
\(S \mathbb{D} E=R\) 'or \(r{ }^{\prime} X^{*}\) op (A) \(=\) alpha*B.

U nchanged on exit.

UPLO (input)
O n entry, UPLO specifies w hether the m atrix \(A\) is an upper or low er triangularm atrix as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or \(\mathrm{L}^{\prime}\) ' \(A\) is an upper triangular m atrix .
\(\mathrm{UPLO}=\mathrm{L}\) ' or \(\mathrm{I}^{\prime}\) A is a lower triangular \(m\) atrix.

U nchanged on exit.

TRANSA (input)
On entry, TRAN SA specifies the form of op (A ) to be used in the m atrix \(m\) ultiplication as follow s:

TRANSA \(=N\) 'or \(h^{\prime}\) op (A) \(=A\).

TRANSA = ' 'ort'op (A) =A'.

TRANSA = C'ort' op (A) =A'.

U nchanged on exit.
TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
O n entry, D IA G specifies w hether ornotA is unit triangular as follow s:

D \(\mathbb{A} G=U\) 'or \(U^{\prime} A\) is assum ed to be unit triangular.
\(D \mathbb{A G}=N\) 'or \(h\) ' \(A\) is notassum ed to be unit triangular.

U nchanged on exit.

M (input)
O \(n\) entry, \(M\) specifies the num ber of row s of \(B . M\) \(>=0\). U nchanged on exit.

N (input)
O n entry, \(N\) specifies the num ber of colum ns of \(B\). \(\mathrm{N}>=0\). U nchanged on exit.

ALPHA (input)
On entry, A LPH A specifies the scalar alpha. W hen alpha is zero then \(A\) is notreferenced and \(B\) need notbe setbefore entry. U nchanged on exit.

A (input)
REAL array ofD \(\mathbb{M} E N S I O N(L D A, k)\), where \(k\) is \(m\) when \(S \mathbb{D} E=\mathbb{L}\) 'or \(\mathbb{I}^{\prime}\) and is \(n\) when \(S \mathbb{D} E=R\) 'or \(r^{\prime}\).
Before entry with UPLO = U 'or \(G\) ', the leading \(k\) by \(k\) upper triangularpart of the aray \(A\) \(m\) ustcontain the upper triangularm atrix and the
strictly low ertriangularpartofA is not referenced.
Before entry w ith UPLO = L'or 1', the leading \(k\) by \(k\) low er triangularpart of the array \(A\) \(m\) ustcontain the low er triangularm atrix and the strictly uppertriangularpartofA is not referenced.
\(N\) ote thatw hen D IA G = U 'or U ', the diagonal elem ents of A are not referenced either, butare assum ed to be one. U nchanged on exit.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen
SDDE = L'or I' then LD A \(>=\max (1, m)\), when \(S \mathbb{D} E=R^{\prime}\) or \(\mathrm{r}^{\prime}\) then LDA \(>=\max (1, \mathrm{n})\). U nchanged on exit.

B (input/output)
REAL array ofD \(\mathbb{M}\) ENSION (LDB, n). Before entry, the leading \(m\) by \(n\) partof the array \(B\) must contain the righthand side \(m\) atrix \(B\), and on exit is overw ritten by the solution \(m\) atrix \(X\).

LD B (input)
On entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program. LD B \(>=m a x(1, m)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
strsna -estim ate reciprocal condition num bers for specified
eigenvalues and/or right eigenvectors of a real upper
quasi-triangularm atrix T (orof any m atrix \(\mathrm{Q} * \mathrm{~T} * \mathrm{Q} * * \mathrm{~T}\) w ith Q orthogonal)

\section*{SYNOPSIS}
```

SU BROUTINE STRSNA (JOB,HOW MNY,SELECT,N,T,LDT,VL,LDVL,VR,LDVR,
S,SEP,MM ,M ,W ORK,LDW ORK,W ORK1, NNFO)

```
CHARACTER * 1 Job,HOW MNY
\(\mathbb{N}\) TEGER N,LDT,LDVL,LDVR,MM,M,LDW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R W\) ORK 1 (*)
LOG ICAL SELECT (*)
REAL T (LDT,*), VL (LDVL,*), VR (LDVR,*), S (*), SEP (*),
W ORK (LDW ORK, , )
SU BROUTINE STRSNA_64 (JOB,HOW MNY,SELECT,N,T,LDT,VL,LDVL,VR,
    LDVR,S,SEP,MM,M,WORK,LDW ORK,WORK1, \(\mathbb{N} F O\) )
CHARACTER * 1 JOB,HOW MNY
\(\mathbb{N}\) TEGER*8N,LDT,LDVL,LDVR,MM,M,LDW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathrm{~W}\) ORK 1 ( \({ }^{\star}\) )
LO G ICAL*8 SELECT (*)
REAL T (LDT,*), VL (LDVL,*), VR (LDVR,*), S (*), SEP (*),
W ORK (LDW ORK, , )

F95 INTERFACE
SU BROUTINE TRSNA (JOB,HOW MNY,SELECT,N,T, [LDT],VL, [LDVL],VR, [LDVR],S,SEP,MM,M,[WORK], [LDW ORK], [WORK1], [NFO])

CHARACTER (LEN=1) :: JOB,HOW M NY
\(\mathbb{N}\) TEGER : : N, LD T, LDVL, LDVR, M M , M , LDW ORK , \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: W\) ORK1
LOGICAL, D \(\mathbb{M}\) ENSION (:) ::SELECT
REAL,D \(\mathbb{I}\) ENSION (:) :: S, SEP
REAL,D \(\mathbb{M}\) ENSION (:,:) ::T,VL,VR,W ORK

SU BROUTINE TRSNA_64 (DOB,HOW M NY, SELECT, N, T, [LDT],VL, [LDVL],VR, \([\) [LDVR], S, SEP, M M , M , [W ORK ], [LDW ORK], [W ORK1], [ \(\mathbb{N F O}])\)

CHARACTER ( \(L E N=1\) ) : : JOB , HOW M NY
\(\mathbb{N}\) TEGER (8) :: N , LD T , LDVL, LDVR , M M , M, LDW ORK , \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{I M} \operatorname{ENSION}(:):: W\) ORK1
LOGICAL (8), D \(\mathbb{M}\) ENSION (:) :: SELECT
REAL,D \(\mathbb{I M}\) ENSION (:) :: S, SEP
REAL,D \(\mathbb{M}\) ENSION (:r:) ::T,VL,VR,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void stresna (char job, charhow m ny, int *select, intn, float
*t, int ldt, float*vl, int ldvl, float*vr, int
ldvr, float *s, float *sep, intm m, int *m, int ldw ork, int *info);
void strsna_64 (char j̀.b, charhow my, long *select, long n, float *t, long ldt, float*vl, long ldvl, float
*vr, long ldvr, float*s, float *sep, long mm, long *m, long ldw ork, long *info);

\section*{PURPOSE}
strsna estim ates reciprocal condition num bers for specified eigenvalues and/or right eigenvectors of a real upper quasi-triangularm atrix T (or of any matrix Q * T * \(\mathrm{Q} * * \mathrm{~T}\) w ith Q orthogonal).

T m ustbe in Schurcanonical form (as retumed by SH SEQR), that is, block upper triangularw ith 1-by-1 and 2 -by-2 diagonalblocks; each 2 -by -2 diagonal block has its diagonal elem ents equal and its off-diagonalelem ents of opposite sign.

\section*{ARGUMENTS}

JOB (input)
Specifies w hethercondition num bers are required foreigenvalues (S) oreigenvectors (SEP):
= E ': foreigenvalues only ( S );
= V ': foreigenvectors only (SEP);
= B ': forboth eigenvalues and eigenvectors ( S and SEP).

HOW MNY (input)
= A': com pute condition num bers for all eigenpairs;
\(=S^{\prime}\) : com pute condition num bers for selected eigenpairs specified by the amray SELEC T.

\section*{SELECT (input)}

If H OW M NY = S',SELECT specifies the eigenpairs for which condition numbers are required. To select condition num bers for the eigenpair corresponding to a realeigenvalue w ( \(\mathcal{\nu}\), SELECT ( ) m ustbe set to .TRUE .. To selectcondition num bers conresponding to a complex conjugate pair of eigenvaluesw ( 7 ) and w ( \(j+1\) ), either SELECT ( \(j\) ) or SELECT ( \(j+1\) ) or both, mustbe set to TRUE .. If HOW MNY = A', SELECT is notreferenced.

N (input) The order of the matrix \(\mathrm{T} \cdot \mathrm{N}>=0\).
T (input) The upper quasi-triangular \(m\) atrix \(T\), in Schur canonical form .

LD T (input)
The leading dim ension of the array T. LD T >= \(\max (1, N)\).

VL (input)
If \(\mathrm{JOB}=\mathrm{E}\) 'or B ', VL mustcontain left eigenvectors of \(T\) (orofany \(Q * T{ }^{*} Q * T\) w th \(Q\) orthogonal), comesponding to the eigenpairs specified by HOW M NY and SELECT. The eigenvectorsm ustbe stored in consecutive colum ns of \(V L\), as retumed by SHSEIN orSTREVC. If \(J O B=V\) ', VL is notreferenced.

LDVL (input)
The leading dim ension of the array VL. LD V L >=1; and if \(J 0 B=E\) 'or \(B ', L D V L>=N\).

VR (input)
If \(\mathrm{OB}=\mathrm{E}\) 'or B ', VR m ust contain right eigenvectors of \(T\) (orofany \(Q * T * Q * T\) w ith \(Q\) orthogonal), comesponding to the eigenpairs specified by HOW M NY and SELECT. The eigenvectors m ustbe stored in consecutive columns of \(V R\), as retumed by

SHSEIN orSTREVC. If \(J O B=V\) ', VR is notreferenced.

LDVR (input)
The leading dim ension of the amay VR. LDVR >=1; and if \(\mathrm{OB}=\mathrm{E}\) 'or \(\mathrm{B}^{\prime}, \mathrm{LDVR}>=\mathrm{N}\).

S (output)
If \(\mathrm{OB}=\mathrm{E}^{\prime}\) or \(\mathrm{B}^{\prime}\), the reciprocal condition num bers of the selected eigenvalues, stored in consecutive elem ents of the aray. For a com plex conjugate pair of eigenvalues two consecutive elem ents of \(S\) are set to the same value. Thus \(S(\mathcal{I}, \operatorname{SEP}(\mathcal{I})\), and the \(j\) th colum ns of VL and VR all correspond to the sam e eigenpair but not in general the jth eigenpair, unless alleigenpairs are selected). If \(\mathrm{OB}=V^{\prime}, \mathrm{S}\) is not referenced.

SEP (output)
If \(\mathrm{OB}=\mathrm{V}\) 'or B', the estim ated reciprocal condition numbers of the selected eigenvectors, stored in consecutive elem ents of the array. For a com plex eigenvector two consecutive elem ents of SEP are set to the sam e value. If the eigenvalues cannot be reordered to com pute \(\operatorname{SEP}(7)\), \(\operatorname{SEP}(7)\) is set to 0 ; this can only occurw hen the true value w ould be very sm allanyw ay. If \(\mathrm{JOB}=\mathrm{E}\) ', SEP is notreferenced.

M M (input)
The num berofelem ents in the arrays S (if \(\mathrm{OB}=\)
E' or B') and/orSEP (if \(\mathrm{OB}=\mathrm{V}\) 'or B).M M
\(>=M\).

M (output)
The num ber of elem ents of the arays \(S\) and/or SEP actually used to store the estim ated condition
num bers. If H O W M NY = 'A', M is setto \(N\).

W ORK (w orkspace)
dim ension (LD W ORK, \(N+1\) ) If \(O B=E \prime\), W ORK is not
referenced.

LDW ORK (input)
The leading dim ension of the array W ORK. LDW ORK \(>=1\); and if \(\mathrm{OB}=\mathrm{V}^{\prime}\) or \(\mathrm{B}^{\prime}, \mathrm{LDW}\) ORK \(>=\mathrm{N}\).

W ORK1 (w orkspace)
dim ension \((\mathbb{N})\) If \(O B=E\) ', W ORK 1 is not referenced.
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

The reciprocal of the condition num ber of an eigenvalue lam bda is defined as
\[
S(\operatorname{lam} \text { bda })=N^{*} u \mid /(\text { norm }(u) * \text { norm }(v))
\]
\(w\) here \(u\) and \(v\) are the right and left eigenvectors of \(T\) comesponding to lam bda; v 'denotes the conjugate-transpose of \(v\), and norm (u) denotes the Euclidean norm. These reciprocal condition num bers alw ays lie betw een zero (very badly conditioned) and one (very w ell conditioned). If \(\mathrm{n}=1\), S (lam bda) is defined to be 1.

A \(n\) approxim ate errorbound for a com puted eigenvalue \(W\) (i) is given by
EPS * norm (T) /S (i)
where EPS is the \(m\) achine precision.

The reciprocal of the condition num ber of the right eigenvectoru corresponding to lam bda is defined as follow s. Suppose
\[
\begin{gathered}
\mathrm{T}=(\operatorname{lam} \text { bda } \mathrm{c}) \\
\left(\begin{array}{ll}
\mathrm{O} & \mathrm{~T} 22
\end{array}\right)
\end{gathered}
\]

Then the reciprocalcondtion num ber is
SEP ( lam bda, T22 ) = sigm a-m in ( T22 -lam bda^I )
where sigm a-m in denotes the sm allest singular value. We approxim ate the sm allest singular value by the reciprocal of an estim ate of the one-norm of the inverse of T22 lam bda*I. If \(n=1\), SEP ( 1 ) is defined to be abs ( \((1,1)\) ).

A \(n\) approxim ate errorbound for a com puted right eigenvector VR (i) is given by
```

EPS * norm (T) /SEP (i)

```

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
stresv - solve one of the system sof equations \(A * x=b\), or
A * \(\mathrm{x}=\mathrm{b}\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE STRSV (UPLO,TRANSA,D IAG,N,A,LDA,Y,INCY)}
CHARACTER * 1UPLO,TRANSA,DIAG
\mathbb{NTEGERN,LDA,}\mathbb{N}CY
REALA (LDA,*),Y (*)
SU BROUT\mathbb{NE STRSV_64(UPLO,TRANSA,D IAG,N,A,LDA,Y, INCY)}
CHARACTER * 1 UPLO,TRANSA,DIAG
INTEGER*8N,LDA,}\mathbb{N}C
REAL A (LDA,*),Y (*)

```
F95 INTERFACE
    SU BROUTINE TRSV (UPLO, [TRANSA],D \(\mathbb{I A} G, \mathbb{N}], A,[L D A], Y,[\mathbb{N C Y}])\)
    CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
    \(\mathbb{N} T E G E R:: N, L D A, \mathbb{N} C Y\)
    REAL,D \(\mathbb{I}\) ENSION (:) ::Y
    REAL,D IM ENSION (:,:) ::A
    SU BROUTINE TRSV_64 (UPLO, [TRANSA],D \(\mathbb{I A G}, \mathbb{N}], A,[L D A], Y,[\mathbb{N} C Y])\)
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
    \(\mathbb{N}\) TEGER (8) ::N,LDA, \(\mathbb{N} C Y\)
    REAL,D \(\mathbb{I M}\) ENSION (:) ::Y
    REAL,D \(\mathbb{I M}\) ENSION (:,:) ::A

\section*{C INTERFACE}
\#include < sunperfh>
void strsv (char uplo, char transa, chardiag, int n, float
*a, int lda, float *y, int incy);
void strisv_64 (charuplo, chartransa, char diag, long n, float *a, long lda, float *y, long incy);

\section*{PURPOSE}
strisv solves one of the system sof equations \(A * x=b\), or \(A\) * \(x=b\), where \(b\) and \(x\) are \(n\) elem entvectors and \(A\) is an \(n\) by \(n\) unit, ornon-unit, upper or low er triangularm atrix. N o test forsingularity ornear-singularity is included in this routine. Such testsm ust.be perform ed before calling this routine.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies w hether the \(m\) atrix is an upper or low er triangularm atrix as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or L ' \(A\) is an upper triangular \(m\) atrix.

UPLO = L' or I' A is a lower triangular \(m\) atrix.

U nchanged on exit.
TRANSA (input)
On entry, TRANSA specifies the equations to be solved as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} A * x=b\).
TRANSA \(=\) T'ort' \(A * x=b\).

TRANSA \(=C^{\prime}\) ort' \(\mathrm{A}^{*} \mathrm{x}=\mathrm{b}\).

U nchanged on exit.
TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.

On entry, D IA G specifies whether ornotA is unit triangular as follow s:

D \(\mathbb{A G G}=\mathrm{U}\) 'or \(\mathrm{L}^{\prime} \mathrm{A}\) is assum ed to be unit triangular.
\(D \mathbb{A} G=N\) 'or \(h\) ' \(A\) is notassumed to be unit triangular.

U nchanged on exit.

N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.
A (input)
Before entry w ith UPLO = U 'or L ', the leading n by n upper triangularpart of the array A m ust contain the upper triangular \(m\) atrix and the strictly low ertriangularpartofA is not referenced. Before entry w ith UPLO = L' 'or I', the leading \(n\) by \(n\) low er triangularpart of the array A \(m\) ust contain the low ertriangularm atrix and the strictly uppertriangularpartofA is not referenced. N ote thatw hen D IAG \(=\mathrm{U}\) ' or L ', the diagonal elem ents of \(A\) are not referenced either, but are assum ed to be unity. U nchanged on exit.

LDA (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program . LD A \(>=\) \(m a x(1, n)\). U nchanged on exit.

Y (input/output)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) must contain the \(n\) elem ent right-hand side vectorb. 0 n exit, Y is overw ritten \(w\) th the solution vectorx.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents ofY. \(\mathbb{N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
strsyl-solve the realSylvesterm atrix equation

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE STRSYL (TRANA,TRANB,ISGN,M ,N,A,LDA,B,LDB,C,LDC,}
SCALE,INFO)
CHARACTER * 1 TRANA,TRANB
\mathbb{NTEGER ISGN,M,N,LDA,LDB,LDC,}\mathbb{N}FO
REAL SCALE
REAL A (LDA,*),B (LD B ,*),C (LD C ,*)
SUBROUT\mathbb{NE STRSYL_64(IRANA,TRANB,ISGN,M,N,A,LDA,B,LDB,C,}
LDC,SCALE, INFO)
CHARACTER * 1 TRANA,TRANB
INTEGER*8 ISGN,M,N,LDA,LDB,LDC,INFO
REAL SCALE
REALA(LDA,*),B(LDB,*),C (LDC,*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE TRSYL (IRANA, TRANB, ISGN, M, N, A, [LDA], B, [LDB], C, [LDC],SCALE, [ $\mathbb{N F O}$ ])
CHARACTER (LEN=1) ::TRANA,TRANB
$\mathbb{N} T E G E R:: \mathbb{I S G N}, \mathrm{M}, \mathrm{N}, L D A, L D B, L D C, \mathbb{N} F O$
REAL ::SCALE
REAL,D $\mathbb{M}$ ENSION (: : : : : A, B , C
SU BROUTINE TRSY L_64 (TRANA,TRANB,ISGN,M,N,A, [LDA],B,[LDB],C, (LDC],SCALE, [INFO])

```

CHARACTER (LEN=1) ::TRANA,TRANB
\(\mathbb{N}\) TEGER (8) :: \(\operatorname{ISG} N, M, N, L D A, L D B, L D C, \mathbb{N} F O\)
REAL :: SCALE
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A,B,C

\section*{C INTERFACE}
\#include <sunperfh>
void strsyl(char trana, chartranb, int isgn, intm, int \(n\), float *a, int lda, float *b, intldl, float * C , int ldc, float *scale, int *info);
void strsyl 64 (char trana, chartranb, long isgn, long m, long \(n\), float *a, long lda, float *b, long ldb, float * c, long ldc, float *scale, long *info);

\section*{PURPOSE}
strsylsolves the realSylvesterm atrix equation:
op (A ) \({ }^{*} X+X *\) op \((B)=\) scale \(^{*} C\) or op (A ) *X -X *op (B) = scale*C ,
where op \((A)=A\) orA \(* * T\), and \(A\) and \(B\) are both upper quasitriangular. A is M -by M and B is N -by -N ; the righthand side C and the solution X are M boy N ; and scale is an output scale factor, set<= 1 to avoid overflow in \(X\).

A and B m ustbe in Schur canonical form (as retumed by SH SEQR ), that is, block uppertriangularw th 1 -by-1 and 2-by-2 diagonalblocks; each 2-by-2 diagonal block has its diagonal elem ents equal and its offf-diagonalelem ents of opposite sign.

\section*{ARGUMENTS}

TRANA (input)
Specifies the option op (A):
\(=N^{\prime}: \operatorname{op}(A)=A \quad\) N o transpose)
\(=T\) ':op \((A)=A * * T\) ( ranspose)
= C':op (A) = A **H (C onjugate transpose = Transpose)

TRANB (input)
Specifies the option op (B):
\(=N\) N: op \((B)=B \quad\) N \(\circ\) transpose)
\(=T\) ': op (B) = B**T (Transpose)
\(=C: o p(B)=B * * H \quad\) (Conjugate transpose \(=T\) ran spose)

ISGN (input)
Specifies the sign in the equation:
\(=+1\) : solve op (A )*X \(+X\) *op \((B)=s c a l e * C\)
\(=-1\) : solve op (A)*X \(-X\) *op \((B)=\) scale \({ }^{*} C\)
\(M\) (input) The order of the m atrix \(A\), and the num berof row s in the \(m\) atrioes X and \(\mathrm{C} . \mathrm{M}>=0\).

N (input) The order of the \(m\) atrix \(B\), and the num ber of colum ns in the m atrices X and \(\mathrm{C} . \mathrm{N}>=0\).

A (input) The upper quasi-triangular matrix A, in Schur canonical form.
LD A (input)
The leading dim ension of the aray A. LD A >= max (1, M).
\(B\) (input) The upper quasi-triangular \(m\) atrix \(B\), in Schur canonical form .

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, \mathbb{N})\).

C (input/output)
On entry, the M -by-N righthand side m atrix C. On exit, \(C\) is overw rilten by the solution \(m\) atrix \(X\).

LD C (input)
The leading dim ension of the anay C. LD C >= \(\max (1, M)\)

SCALE (output)
The scale factor, scale, set < = 1 to avoid overflow in \(X\).
\(\mathbb{I N} F \mathrm{O}\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue
\(=1\) : \(A\) and \(B\) have com \(m\) on or very close eigenvalues; perturbed values w ere used to solve the equation (but the \(m\) atrices \(A\) and \(B\) are unchanged).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
strit 2 - com pute the inverse of a real upperor low er triangularm atrix

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE STRTI2 (UPLO,D IAG,N,A,LDA , INFO)}
CHARACTER * 1 UPLO,D IAG
\mathbb{NTEGERN,LDA,INFO}
REALA (LDA,*)
SUBROUTINE STRTI2_64(UPLO,D IAG,N,A,LDA, \mathbb{NFO)}
CHARACTER * 1UPLO,DIAG
INTEGER*8N,LDA, INFO
REALA (LDA,*)
F95 INTERFACE
SUBROUT\mathbb{NE TRTI2 (UPLO,D IAG, N ],A ,[LDA],[NFO])}
CHARACTER (LEN=1) ::UPLO,D IAG
INTEGER ::N,LDA,\mathbb{NFO}
REAL,D IM ENSION (:,:) ::A
SUBROUT\mathbb{NE TRTI2_64(UPLO,DIAG, NN],A,[LDA],[INFO])}
CHARACTER (LEN=1) ::UPLO,D IAG
\mathbb{NTEGER (8)::N,LDA,}\mathbb{NFO}
REAL,DIM ENSION (:,:) ::A

```
C INTERFACE
    \#include <sunperfh>
void strti2 (char uplo, char diag, intn, float *a, int lda, int*info);
void strti2_64 (charuple, chardiag, long n, float*a, long lda, long *info);

\section*{PURPOSE}
strti2 com putes the inverse of a realupper or low er triangularm atrix.

This is the Level2 B LAS version of the algorithm .

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the m atrix A is upper or low er
triangular. = U ': U pper triangular
= L': Low ertriangular

D IA G (input)
Specifies w hether ornot the m atrix A is unittriangular. \(=\mathrm{N}\) ': N on-unittriangular
\(=\mathrm{U}\) ': Unittriangular

N (input) The order of them atrix A. \(\mathrm{N}>=0\).

A (input/output)
O n entry, the triangularm atrix A. If \(\mathrm{U} P \mathrm{O}=\mathrm{U}\) ', the leading \(n\) by \(n\) uppertriangularpart of the array A contains the uppertriangularm atrix, and the strictly low er triangular part of A is not referenced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading n by n low er triangularpart of the array A contains the low ertriangularm atrix, and the strictly upper triangularpart ofA is not referenced. IfD IA G \(=\) U', the diagonal elem ents of A are also not referenced and are assum ed to be 1 .

On exit, the (triangular) inverse of the original \(m\) atrix, in the sam e storage form at.

LD A (input)
The leading dim ension of the aray A. LD A >= \(\max (1, N)\).

IN FO (output)
= 0: successfulexit
< 0: if \(\mathbb{N N}\) FO \(=-\mathrm{k}\), the k -th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
strtri-com pute the inverse of a realupperor low er triangularm atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE STRTRI(UPLO,DIAG,N,A,LDA , NNFO)}
CHARACTER * 1 UPLO,D IAG
INTEGERN,LDA, INFO
REALA (LDA,*)
SU BROUT\mathbb{NE STRTRI_64(UPLO,D IAG,N,A,LDA, INFO)}
CHARACTER * 1 UPLO,D IAG
INTEGER*8N,LDA, INFO
REALA (LDA,*)
F95 INTERFACE
SUBROUT\mathbb{NE TRTRI(UPLO,D IAG,N,A,[LDA ], [NNO ])}
CHARACTER (LEN=1) ::UPLO,D IA G
\mathbb{NTEGER ::N,LDA,}\mathbb{NFO}
REAL,D IM ENSION (:,:) ::A
SUBROUT\mathbb{NE TRTRI_64(UPLO,D IAG,N,A , [LDA],[NFO])}
CHARACTER (LEN=1) ::UPLO,D IAG
\mathbb{NTEGER (8)::N,LDA,}\mathbb{NFO}
REAL,DIM ENSION (:,:) ::A

```
C INTERFACE
    \#include <sunperfh>
void strtri(char uplo, char diag, intn, float *a, int lda, int*info);
void strtri_64 (charuplo, chardiag, long n, float *a, long lda, long *info);

\section*{PURPOSE}
strtricom putes the inverse of a realupper or low er triangularm atrix A.

This is the Level3 B LAS version of the algorithm .

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}:\) A is upper triangular;
\(=\mathbb{L}\) ': A is low ertriangular.

D IA G (input)
\(=\mathrm{N}: A\) is non-unit triangular;
\(=U\) : A is unittriangular.

N (input) The order of them atrix A. \(\mathrm{N}>=0\).

A (input/output)
O n entry, the triangularm atrix A. If UPLO = U', the leading N -by- N uppertriangularpart of the array A contains the upper triangularm atrix, and the strictly low er triangular part of A is not referenced. If UPLO \(=\mathrm{L}\) ', the leading N -by -N low er triangular part of the anray A contains the low er triangularm atrix, and the strictly upper triangularpart ofA is not referenced. IfD IA G = U', the diagonal elem ents of \(A\) are also not referenced and are assum ed to be 1 . O n exit, the (triangular) inverse of the original \(m\) atrix, in the sam e storage form at.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an ille-
galvalue
\(>0:\) if \(\mathbb{N} F O=i, A(i, i)\) is exactly zero. The triangular \(m\) atrix is singular and its inverse can notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

strtes -solve a triangularsystem of the form A * X = B

```
orA \({ }^{* *}\) T \(* \mathrm{X}=\mathrm{B}\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE STRTRS (UPLO,TRANSA,D IAG,N,NRHS,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO,TRANSA,D IAG
INTEGERN,NRHS,LDA,LDB,INFO
REALA (LDA,*),B (LDB,*)
SUBROUT\mathbb{NE STRTRS_64 UPLO,TRANSA,D IAG,N,NRHS,A,LDA,B,LDB,}
\mathbb{NFO)}

```
CHARACTER * 1 UPLO, TRANSA, DIAG
\(\mathbb{N} T E G E R * 8 N, N R H S, L D A, L D B, \mathbb{N F O}\)
REALA (LDA,*),B (LDB,*)

\section*{F95 INTERFACE}

SU BROUTINE TRTRS (UPLO, [TRANSA ], D IA G ,N,NRHS,A, [LDA ], B, [LD B ], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
\(\mathbb{N}\) TEGER ::N,NRHS,LDA,LDB, \(\mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A,B
SU BROUTINE TRTRS_64 (UPLO, [TRANSA ],D \(\mathbb{I A} G, N, N R H S, A,[L D A], B,[L D B]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
\(\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D B, \mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:,:) ::A,B

\section*{C INTERFACE}
\#include < sunperfh>
void strtes (charuplo, chartransa, char diag, int n, int nihs, float *a, int lda, float*b, int ldb, int *info);
void strtes_64 (charuplo, char transa, char diag, long n, long nihs, float *a, long lda, float *b, long ldb, long *info);

\section*{PURPOSE}
strtrs solves a triangular system of the form
where \(A\) is a triangularm atrix of order \(N\), and \(B\) is an \(N\) -by-NRHS matrix. A check ism ade to verify thatA is nonsingular.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}: \mathrm{A}\) is uppertriangular;
= L': A is low ertriangular.
TRANSA (input)
Specifies the form of the system of equations:
\(=N\) : A * X = B N o transpose)
\(=T: A * * T * X=B \quad(T\) ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose \(=\mathrm{T}\) ranspose)

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
\(=\mathrm{N}: \mathrm{A}\) is non-unit triangular;
\(=\mathrm{U}: \mathrm{A}\) is unit triangular.
\(N\) (input) The order of them atrix \(A . N>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of them atrix B. NRHS \(>=0\).

A (input) The triangularm atrix A . If \(\mathrm{PLO}=\mathrm{U}\) ', the leading N -by -N upper triangularpart of the array A contains the upper triangular \(m\) atrix, and the
strictly low ertriangularpantofA is notreferenced. IfUPLO \(=\mathrm{L}\) ', the leading N -by N lower triangular part of the array A contains the low er triangularm atrix, and the strictly upper triangular part of \(A\) is notreferenced. IfD \(\mathbb{A} G=U '\), the diagonal elem ents ofA are also not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).
B (input/output)
On entry, the right hand side \(m\) atrix \(B\). On exit, if \(\mathbb{N} F O=0\), the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray \(\mathrm{B} . \mathrm{LDB}>=\) \(\max (1, N)\).

IN FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N ~ F O ~}=\) i, the \(i\)-th diagonalelem ent of \(A\) is zero, indicating that the \(m\) atrix is singular and the solutions X have notbeen com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
stzrqf-routine is deprecated and has been replaced by routine ST ZR ZF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE STZRQFM,N,A,LDA,TAU,\mathbb{NFO)}}\mathbf{N}\mathrm{ (IN}
INTEGERM,N,LDA,\mathbb{NFO}
REALA (LDA,*),TAU (*)
SUBROUT\mathbb{NESTZRQF_64 M,N,A,LDA,TAU, INFO)}
\mathbb{NTEGER*8M,N,LDA, INFO}
REALA (LDA,*),TAU (*)
F95 INTERFACE
SUBROUT\mathbb{NE TZRQFM,N,A,[LDA ],TAU, [NNO]]}
\mathbb{NTEGER ::M,N,LDA,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REAL,DIM ENSION (:) ::TAU
REAL,D IM ENSION (:,:)::A
SUBROUTINE TZRQF_64M,N,A,[LDA],TAU,[NFO])
\mathbb{NTEGER (8) ::M ,N,LDA,NNFO}
REAL,DIM ENSION (:) ::TAU
REAL,DIM ENSION (:,:) ::A
C INTERFACE
\#include <sunperfh>

```
void stzrqf(intm , intn, float *a, int lda, float *tau, int
*info);
void stzrqf_64 (long m, long n, float *a, long lda, float
*tau, long *info);

\section*{PURPOSE}
stzrqf routine is deprecated and has been replaced by routine STZRZF .

STZRQF reduces the M -by-N ( \(\mathrm{M}<=\mathrm{N}\) ) real upper trapezoidal \(m\) atrix \(A\) to upper triangular form by \(m\) eans of orthogonal transform ations.

The upper trapezoidalm atrix \(A\) is factored as
\(A=\left(\begin{array}{ll}R & 0\end{array}\right) * Z\),
\(w\) here Z is an N boy N orthogonalm atrix and R is an M boy M upper triangularm atrix.

\section*{ARGUMENTS}

M (input) The num ber of row s of the matrix A. M >=0.

N (input) The num ber of colum ns of the \(m\) atrix \(A . N>=M\).

A (input/output)
O n entry, the leading M foy-N upper trapezoidal part of the array A m ustcontain the \(m\) atrix to be factorized. On exit, the leading \(M\) boy -M upper triangularpart of A contains the upper triangular \(m\) atrix \(R\), and elem ents \(M+1\) to \(N\) of the first \(M\) row s of \(A\), \(w\) th the array \(T A U\), represent the orthogonalm atrix \(Z\) as a product of \(M\) elem entary reflectors.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, M)\).

TAU (output)
The scalar factors of the elem entary reflectors.

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an ille-

\section*{FURTHER DETAILS}

The factorization is obtained by H ouseholdersm ethod. The \(k\) th transform ation \(m\) atrix, \(Z(k)\), w hich is used to introduce zeros into the ( \(m-k+1\) )th row ofA, is given in the form
\[
\begin{gathered}
\mathrm{Z}(\mathrm{k})=\left(\begin{array}{ll}
\mathrm{I} & 0
\end{array}\right), \\
\left(\begin{array}{l}
\mathrm{O}(\mathrm{k})
\end{array}\right)
\end{gathered}
\]
where
\[
\begin{gathered}
\left.T(k)=I-\tan { }^{*} u(k) * u(k)\right), \quad u(k)=\left(\begin{array}{ll}
1
\end{array}\right), \\
\left(\begin{array}{l}
0
\end{array}\right) \\
(z(k))
\end{gathered}
\]
tau is a scalar and \(z(k)\) is an ( \(n-m\) ) elem ent vector. tau and \(\mathrm{z}(\mathrm{k})\) are chosen to annihilate the elem ents of the kth row of .

The scalartau is retumed in the \(k\) th elem entofTA \(U\) and the vectoru ( \(k\) ) in the \(k\) th row of \(A\), such that the elem ents of \(z(k)\) are in \(a(k, m+1), \ldots, a(k, n)\). The elem ents ofR are retumed in the upper triangularpartofA.
\(Z\) is given by
\[
Z=Z(1) * Z(2) * \ldots * Z(m) .
\]

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
stzrzf-reduce the \(M-b y-N\) ( \(M<=N\) ) real upper trapezoidal \(m\) atrix \(A\) to upper triangular form by \(m\) eans of orthogonal transform ations

\section*{SYNOPSIS}

SUBROUTINE STZRZFM,N,A,LDA,TAU,WORK,LWORK, \(\mathbb{N} F O\) )
\(\mathbb{N}\) TEGER M, N,LDA,LW ORK, \(\mathbb{N} F O\)
REALA (LDA, *), TAU (*), W ORK (*)
SU BROUTINE STZRZF_64 M,N,A,LDA,TAU,W ORK,LW ORK, \(\mathbb{N} F O\) )
\(\mathbb{N}\) TEGER*8M,N,LDA,LWORK, \(\mathbb{N}\) FO
REALA (LDA,*),TAU (*),W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE TZRZF (M ], \(\mathbb{N}], A,[L D A], T A U,[W O R K],[L W\) ORK ], [ \(\mathbb{N F O}])\)
\(\mathbb{N} T E G E R:: M, N, L D A, L W\) ORK, \(\mathbb{N} F O\)
REAL,D \(\mathbb{I}\) ENSION (:) ::TAU,W ORK
REAL,D IM ENSION (:,:) ::A
SU BROUTINE TZRZF_64 ( \(\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L W\) ORK \(],[\mathbb{N} F O])\)
\(\mathbb{N} T E G E R(8):: M, N, L D A, L W O R K, \mathbb{N} F O\)
REAL,D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
REAL,D \(\mathbb{I}\) ENSION (:,:) ::A
C INTERFACE
\#include < sunperfh>
void stzrzf(intm, intn, float *a, int lda, float *tau, int *info);
void stzrzf_64 (long m, long n, float *a, long lda, float *tau, long *info);

\section*{PURPOSE}
stzrzf reduces the M -by -N ( \(\mathrm{M}<=\mathrm{N}\) ) real upper trapezoidal \(m\) atrix \(A\) to upper triangular form by \(m\) eans of orthogonal transform ations.

The upper trapezoidalm atrix \(A\) is factored as
\[
A=\left(\begin{array}{ll}
R & 0
\end{array}\right)^{*} Z
\]
w here Z is an N -by -N orthogonalm atrix and R is an M boy -M upper triangularm atrix.

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input/output)
On entry, the leading M -by -N upper trapezoidal part of the array A m ustcontain them atrix to be factorized. On exit, the leading M -by -M upper triangularpart ofA contains the upper triangular \(m\) atrix \(R\), and elem ents \(M+1\) to \(N\) of the first \(M\) row s of \(A\), w th the array TAU, represent the orthogonalm atrix Z as a product of M elem entary reflectors.

LD A (input)
The leading dim ension of the array A. LDA >= max (1, M).

TAU (output)
The scalar factors of the elem entary reflectors.
W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

The dimension of the array \(W\) ORK. LW ORK >= \(m\) ax \((1, M)\). Foroptim um perform anœ \(L W O R K>=M * N B\), w here NB is the optim alblocksize.

If LW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{N F O}=-\) i, the \(i\)-th argum ent had an illegal value

\section*{FURTHER DETAILS}

B ased on contributions by
A. Petitet, C om puterScience D ept., U niv . of Tenn., K noxville, U SA

The factorization is obtained by H ouseholdersm ethod. The \(k\) th transform ation \(m\) atrix, \(Z(k)\), which is used to introduce zeros into the ( \(m-k+1\) )th row ofA, is given in the form
\[
\begin{gathered}
\mathrm{Z}(\mathrm{k})=\left(\begin{array}{ll}
\mathrm{I} & 0
\end{array}\right), \\
\binom{\mathrm{O}}{\mathrm{~T}(\mathrm{k})}
\end{gathered}
\]
where
\[
\begin{gathered}
\left.T(k)=I-\tan { }^{*} u(k) * u(k)\right), \quad u(k)=\left(\begin{array}{ll}
1
\end{array}\right), \\
\left(\begin{array}{l}
0
\end{array}\right) \\
(z(k))
\end{gathered}
\]
tau is a scalar and \(z(k)\) is an ( \(n-m\) ) elem ent vector. tau and \(\mathrm{z}(\mathrm{k})\) are chosen to annihilate the elem ents of the kth row of .

The scalartau is retumed in the kth elem entofTAU and the vectoru ( \(k\) ) in the kth row of A, such that the elem ents of \(z(k)\) are in \(a(k, m+1), \ldots, a(k, n)\). The elem ents ofR are retumed in the upper triangularpartofA.
\(Z\) is given by
\[
Z=Z(1) * Z(2) * \ldots * Z(m) .
\]

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
sunpenf_version - gets library inform ation

F95 INTERFACE
C INTERFACE
\#include <sunperfh>
The C version of sunperf_version also retum a pointer to the version string.
char *sunperf_version (int *version, int *patch, int *update);
char *sunperf_version_64 (long *version, long *patch, long
*update);

\section*{ARGUMENTS}

\author{
VERSION (output) \\ \(V\) ersion num ber of library \\ PATCH (output) \\ Patch num ber of library \\ U pdate num ber of library
}

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
subrm m - variable block sparse row form atm atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NESVBRMM(TRANSA,MB,N,KB,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,BINDX,RPNTR,CPNTR,BPNTRB,BPNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,MB,N,KB,DESCRA (5),LDB,LDC,LW ORK}
INTEGER }\mathbb{NDX(*),B\mathbb{NDX (*),RPNTR M B+1),CPNTR(KB+1),}
* BPNTRB M B ),BPNTRE M B)
REAL ALPHA,BETA
REAL VAL (*),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NESVBRMM_64(TRANSA,M B,N,KB,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,BINDX,RPNTR,CPNTR,BPNTRB,BPNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER*8 TRANSA,MB,N,KB,DESCRA (5),LDB,LDC,LW ORK}

```

```

* BPNTRB MB),BPNTREMB)
REAL ALPHA,BETA
REAL VAL (*),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NEVBRMM (TRANSA,MB, N ],KB,ALPHA,DESCRA,}

* VAL, INDX,BINDX,RPNTR,CPNTR,BPNTRB,BPNTRE,
* B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])

```
\(\mathbb{N} T E G E R\) TRANSA,MB,KB
\(\mathbb{N}\) TEGER,D \(\mathbb{I M}\) ENSION (:) :: DESCRA, \(\mathbb{N} D X, B \mathbb{N} D X\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: RPNTR,CPNTR,BPNTRB,BPNTRE
REAL ALPHA,BETA
REAL,D \(\mathbb{I M}\) ENSION (:) ::VAL

REAL,D \(\mathbb{I M}\) ENSION (:, :) :: B , C

SUBROUTINE VBRM M _64 (TRANSA, M B, \(\mathbb{N}], K B, A L P H A, D E S C R A\), * \(\quad V A L, \mathbb{N D X}, B \mathbb{N D X}, R P N T R, C P N T R, B P N T R B, B P N T R E\),
* \(B,[\) LD B \(], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{I N T E G E R * 8}\) TRANSA, M B , KB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: D E S C R A, \mathbb{N} D X, B \mathbb{N} D X\)
\(\mathbb{I N T E G E R * 8 , D \mathbb { M } E N S I O N ( : ) : : R P N T R , C P N T R , B P N T R B , B P N T R E ~}\)
REAL ALPHA,BETA
REAL,D \(\mathbb{I}\) ENSION (:) ::VAL
REAL,D \(\mathbb{I}\) ENSION (:, :) :: B , C

\section*{DESCRIPTION}
C <-alpha op (A ) B + beta C
where A LPHA and BETA are scalar, \(C\) and \(B\) are \(m\) atrioes, \(A\) is a \(m\) atrix represented in variable block sparse row form at and op (A ) is one of
\(o p(A)=A\) or \(o p(A)=A^{\prime}\) or op \((A)=\operatorname{conjg}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRA NSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalent to 1 if the \(m\) atrix is real.

M B \(\quad\) Num ber ofblock row s in m atrix A
\(N \quad N\) um ber of colum \(n s\) in \(m\) atrix \(C\)

K B \(\quad\) Num ber ofblock colum ns in m atrix A

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger aray
D ESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm itian ( \(A=\) CON JG (A ) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(A=-C O N J(A)\) )

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 :non-unit
1 : unit
DESCRA (4) A ray base \(N\) OT \(\mathbb{M}\) PLEM ENTED)
0 :C C++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL ( scalar array of length NN Z consisting of the block entries of A w here each block entry is a dense rectangularm atrix stored colum \(n\) by colum \(n\).
NN Z is the total num berof pointentries in all nonzero block entries of am atrix A.
\(\mathbb{N} D \mathrm{X}\) () integer anray of length BNN Z +1 where BNNZ is the num berof block entries of a m atrix A such that the I-th elem entof \(\mathbb{N}\) D X [] points to the location in VAL of the \((1,1)\) elem ent of the I-th block entry.

B IND X () integer array of length BNN Z consisting of the block colum \(n\) indiges of the block entries of \(A\) where BNNZ is the num berblock entries of a m atrix A.

RPN TR 0) integer amay of length M B+1 such thatRPN TR (I) RPN TR (1) +1 is the row index of the firstpoint row in the \(I\)-th block
row .
RPN TR \(M B+1\) ) is set to \(M+\operatorname{RPN} \operatorname{TR}(1)\) where \(M\) is the num ber of row \(s\) in \(m\) atrix \(A\).
Thus, the num berof point row sin the I-th block row is RPNTR (I+1)RPNTR (I).

CPN TR 0 integer array of length \(K B+1\) such that CPN TR (J)-CPN TR (1)+1 is the colum \(n\) index of the firstpoint colum \(n\) in the \(J\) th block colum n. CPN TR ( \(K B+1\) ) is set to \(K+C P N T R(1)\) where \(K\) is the num ber of \(c o l u m n s\) in \(m\) atrix \(A\). Thus, the num ber of point \(\infty 0\) lum ns in the \(J\) th block colum n is CPNTR ( \(\mathrm{J}+1\) )-CPNTR ( \(J\) ).

BPNTRB () integer array of length \(M B\) such thatBPNTRB (I) BPNTRB (1) +1 points to location in B IND X of the first.block entry of the I-th block row of A.

BPNTRE 0 integer anay of length \(M B\) such that BPN TRE (I) BPNTRB (1) points to location in B \(\mathbb{N}\) D X of the lastblock entry of the I-th block row of A.

B 0 rectangular array w ith firstdin ension LD B .

LD B leading dim ension ofB

BETA Scalarparam eter
C 0 rectangular anray w ith firstdim ension LD C .

LD C leading dim ension ofC
W ORK () scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is not referenced in the cumentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at: http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1.For a generalm atrix (DESCRA (1)=0), array CPN TR can be different from RPNTR. Forallotherm atrix types, RPNTR \(m\) ustequalC CN TR and a single array can be passed forboth argum ents.
2. It is know \(n\) that there exists another representation of the variable block sparse row form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of six anay instead of the seven used in the current im plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each block row in the array \(B \mathbb{N} D X\) is used instead of two arrays BPNTRB and BPNTRE.To use the routine w th this kind of variable block sparse row form at the follow ing calling sequence should be used

SUBROUTINE SVBRMM (TRANSA,MB,N,KB,ALPHA,DESCRA,
* \(\quad V A L, \mathbb{N} D X, B \mathbb{N} D X, R P N T R, C P N T R, \mathbb{A}, \mathbb{A}(2)\),
* B,LDB,BETA, C,LDC,W ORK,LW ORK )

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

svbrsm -variable block sparse row form attriangular solve

```

\section*{SYNOPSIS}
```

SUBROUTINE SVBRSM(TRANSA,M B,N,UNITD,DV,ALPHA,DESCRA,

* VAL,\mathbb{NDX,BINDX,RPNTR,CPNTR,BPNTRB,BPNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
\mathbb{NTEGER TRANSA,MB,N,UNITD,DESCRA (5),LDB,LDC,LW ORK}
INTEGER \mathbb{NDX(*),BINDX(*),RPNTR M B+1),CPNTR M B+1),}
* BPNTRBMB),BPNTREMB)
REAL ALPHA,BETA
REAL DV (*),VAL (*),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINE SVBRSM_64(TRANSA,M B,N,UN ITD,DV,ALPHA,DESCRA,
* VAL,INDX,B\mathbb{NDX,RPNTR,CPNTR,BPNTRB,BPNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK )
INTEGER*8 TRANSA,M B,N,UNTTD,DESCRA (5),LDB,LDC,LW ORK
\mathbb{NTEGER*8 INDX (*),B\mathbb{NDX (*),RPNTR M B+1),CPNTR M B+1),}}\mathbf{~}\mathrm{ , (})
* BPNTRB MB),BPNTRE MB)
REAL ALPHA,BETA
REAL DV (*),VAL (*),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NEVBRSM (TRANSA,M B, N ],UNITD,DV,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,B}\mathbb{N}DX,RPNTR,CPNTR,BPNTRB,BPNTRE,
* B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])
INTEGER TRANSA,MB,UNITD

```

```

NNTEGER,D IM ENSION (:) :: RPNTR,CPNTR,BPNTRB,BPNTRE
REAL ALPHA,BETA
REAL,DIM ENSION (:) ::VAL,DV
REAL,D IM ENSION (:,:):: B,C

```

SUBROUTINEVBRSM_64 (TRANSA, MB, \(\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A\),
* \(\quad \mathrm{B},[\mathrm{LDB}], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N T E G E R *} 8\) TRANSA, MB,UNITD
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O}(:):: \operatorname{DESCRA}, \mathbb{N} D \mathrm{X}, \mathrm{B} \mathbb{N} D \mathrm{X}\)
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: R P N T R, C P N T R, B P N T R B, B P N T R E\)
REAL ALPHA,BETA
REAL, D \(\mathbb{M}\) ENSION (:) ::VAL, DV
REAL,D \(\mathbb{I M}\) ENSION (:, :) :: B , C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA C C <-ALPHA D op (A) B + BETA C } \\
& C<-A L P H A \text { op (A) D B + BETA C } \\
& \text { where A LPH A and BETA are scalar, C and B arem by n densem atrices, } \\
& D \text { is a block diagonalm atrix, A is a unit, ornon-unit, upper or } \\
& \text { low ertriangularm atrix represented in variable block sparse row } \\
& \text { form atand op (A ) is one of }
\end{aligned}
\]
op (A) \()=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}\left(\infty n \dot{g}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(\sin m\) atrix \(A\)

N \(\quad\) Num berof colum ns in matrix C

UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 : Scale on left (row block scaling)
3 : Scale on right (colum \(n\) block scaling)

DV () A rray containing the block entries of the block diagonalm atrix D. The size of the Jth block is RPN TR ( \(\mathrm{J}+1\) )-RPN TR (J) and each block containsm atrix entries stored colum n-m ajor. The total length of aray DV is given by the form ula:
sum over J from 1 to M B:

A LPH A Scalarparam eter

DESCRA ( D escriptor argum ent. Five elem ent integer amay
DESCRA (1) m atrix structure
0 : general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm Aian ( \(\mathrm{A}=\mathrm{CONJG}\) (A))
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJ}\) ( A ) )
N ote:For the routine, D ESCRA (1)=3 is only supported.
D ESCRA (2) upper/low er triangular indicator
1: low er
2 : upper
DESCRA (3) m ain diagonal type
0 : non-identity blocks on the \(m\) ain diagonal
1 : identily diagonalblock
2 : diagonalblocks are dense \(m\) atrices
DESCRA (4) A ray base (NOT IM PLEM ENTED)
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL ( scalar array of length NN Z consisting of the block entries ofA where each block entry is a dense rectangularm atrix stored colum \(n\) by colum \(n\).
NNZ is the total num berof pointentries in allnonzero block entries of am atrix A.
\(\operatorname{INDX}\) () integer array of length BNN Z+1 where BNN Z is the num ber block entries of a \(m\) atrix A such that the I-th elem ent of \(\mathbb{N}\) D X [] points to the location in V A L of the ( 1,1 ) elem ent of the I-th block entry.

B \(\mathbb{N}\) DX () integer array of length BNN Z consisting of the block colum \(n\) indices of the block entries of \(A\) where BNN \(Z\) is the num berblock entries of a m atrix A. B lock colum \(n\) indices M U ST be sorted in increasing order foreach block row .

RPN TR () integer array of length M B +1 such thatRPN TR (I) RPN TR (1) +1 is the row index of the firstpoint row in the I-th block row .
RPNTR \(M B+1\) ) is set to \(M+R P N T R(1)\) where \(M\) is the num ber
of row s in square triangularm atrix A. Thus, the num ber of point row s in the I-th block row is RPN TR (I+1)RPN TR (I).

NOTE: For the current version CPN TR m ustequalRPNTR and a single array can be passed forboth argum ents

CPN TR 0) integer array of length M B+1 such that CPN TR (J)-CPN TR (1)+1 is the colum \(n\) index of the firstpoint colum \(n\) in the \(J\) th block colum n. CPN TR M B+1) is set to M +CPN TR (1). Thus, the num ber of point \(c o l u m\) ns in the \(J\) th block \(\propto 0\) lum \(n\) is CPN TR ( \(\mathrm{J}+1\) )-CPNTR ( \(J\) ).

NOTE:For the cumentversion CPNTR m ustequal RPN TR and a single array can be passed forboth argum ents
BPNTRB 0 integer aray of length \(M B\) such thatBPNTRB (I)-BPNTRB (1)+1
points to location in B IND X of the firstblock entry of the I-th block row of A.

BPNTRE () integer array of length \(M B\) such thatBPN TRE (I) BPNTRB (1) points to location in B \(\mathbb{N}\) D X of the lastblock entry of the I-th block row of A.

B 0 rectangular anay w th first dim ension LD B .

LD B leading dim ension ofB
BETA Scalarparam eter
C \(0 \quad\) rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC
W ORK 0 scratch array of length LW ORK. On exit, ifLW ORK \(=-1, W\) ORK (1) retums the optim um size ofLW ORK.

LW ORK length ofW ORK array. LW ORK should be at least \(\mathrm{M}=\operatorname{RPNTR} \mathrm{M} \operatorname{B+1})\) RPNTR (1).

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=M\) *N_CPUS where N_CPUS is the m axim um num berof processors available to the program.

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.

IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK
array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FORTRAN Sparse B las U ser's G uide available at:
http://m ath nistgov/n csd/Staffk Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.
2. If \(D E S C R A\) (3)=0, the low er or upper triangularpart of each diagonalblock is used by the routine depending on DESCRA (2).
3. If \(D E S C R A\) (3)=1, the unitdiagonalblocksm ightorm ight notbe referenced in the \(V\) BR representation of a sparse \(m\) atrix. They are notused anyw ay .
4. If DESCRA (3)=2, diagonalblocks are considered as dense \(m\) atrices and the LU factorization \(w\) ith partial pivoting is used by the routine. WORK (1)=0 on retum if the factorization foralldiagonalblocks has been com pleted successfully, otherw ise WORK (1) = -iw here i is the block num ber forw hich the LU factorization could notbe com puted.
5.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse \(m\) atrix A is used. H ow erverDESCRA (1) m ustbe equal to 3 .
6. It is know \(n\) that there exists another representation of the variable block sparse row form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem s",W PS, 1996). Its data structure consists of six aray instead of the seven used in the current im plem entation. The \(m\) ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning ofeach block row in the amay \(B \mathbb{N} D X\) is used instead of two arrays BPN TRB and BPNTRE.To use the routine w th this kind of variable block sparse row

SU BROUTINE SVBRSM (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,
* \(\quad \mathrm{VAL}, \mathbb{N} D \mathrm{X}, \mathrm{B} \mathbb{N} D \mathrm{X}, \mathrm{RPNTR}, \mathrm{CPNTR}, \mathbb{A}, \mathbb{A}(2)\),
* B,LDB,BETA, C,LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}

Sw iener-perform W ienerdeconvolution oftw o signals

\section*{SYNOPSIS}

```

INTEGER N_PO INTS,ISW,\mathbb{ERR}
REALACOR (*),XCOR (*),FLTR (*),EROP (*)
SU BROUT\mathbb{NE SW ENER_64 N_PO INTS,ACOR,XCOR,FLTR,EROP,ISW, ERRR)}

```

```

REALACOR(*),XCOR(*),FLTR (*),EROP (*)

```

\section*{F95 INTERFACE}

SU BROUTINEW \(\mathbb{E N E R} \mathbb{N} \_P O \mathbb{N} T S, A C O R, X C O R, F L T R, E R O P, I S W, \mathbb{E R R}\) )
\(\mathbb{N} T E G E R:: N \_P O \mathbb{N} T S, \operatorname{ISW}, \mathbb{E R R}\)
REAL,D \(\mathbb{M}\) ENSION (:) ::ACOR,XCOR,FLTR,EROP

SU BROUTINEW \(\left.\mathbb{E N} E R \_64 \mathbb{N} \_P O \mathbb{I N} T S, A C O R, X C O R, F L T R, E R O P, I S W, \mathbb{E R R}\right)\)
\(\mathbb{N} T E G E R(8):: N \_P O \mathbb{N} T S, \operatorname{ISW}, \mathbb{E R R}\)
REAL,D \(\mathbb{I M}\) ENSION (:) ::ACOR,XCOR,FLTR,EROP

\section*{C INTERFACE}
\#include <sunperfh>
void Sw iener(intn_points, float*acor, float *xcor, float
* fltr, float *erop, int * isw , int * ienc);
void sw iener_64 (long n_points, float *acor, float *xcor,

\section*{PURPOSE}
sw ienerperform sW ienerdeconvolution oftw o signals.

\section*{ARGUMENTS}

N_PO \(\mathbb{I N} T S\) (input)
O n entry, the num berofpoints in the inputcomelations. U nchanged on exit.

ACOR (input)
On entry, autocorrelation coefficients. U nchanged on exit.

XCOR (input)
On entry, cross-comelation coefficients.
U nchanged on exit.

FLTR (output)
On exit, filter coefficients.

EROP (output)
On exit, the prediction error.

ISW (input)
On entry, if ISW EQ. O then perform spiking
deconvolution, otherw ise perform generaldeconvohution. U nchanged on exit.

ERR (output)
O n exit, the deconvolution \(w\) as successfuliff \(\mathbb{E R R}\)
EQ.0, otherw ise there w as an error.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE

\section*{NAME}
use_threads - set the upperbound on the num ber of threads that the calling thread w ants used

\section*{SYNOPSIS}

SUBROUTINEUSE_THREADS NTHREADS)
\(\mathbb{N}\) TEGER NTHREADS

SU BROUTINE USE_THREADS_64 (NTHREADS)
\(\mathbb{N}\) TEGER*8 NTHREAD S

\section*{F95 INTERFACE}

SU BROUTINE USE_THREADS \(\mathbb{N} T H R E A D S)\)
\(\mathbb{N} T E G E R:: N T H R E A D S\)

SU BROUTINE USE_THREADS_64 NTHREADS)
\(\mathbb{N} T E G E R(8):: N T H R E A D S\)
C INTERFACE
\#include <sunperfh>
void use_threads (intnthreads);
void use_threads_64 (long nthreads);

\section*{PURPOSE}
use_threads THREAD S sets an upperbound on the num ber of threads that the calling thread wants used. Subsequent calls to this routine resultin replacem ent of the previous U se num ber for the calling thread. This counts all threads w orking on the callers behalf, so if it passes 2 for NTHREADS and then calls som e subroutine, there w illbe at \(m\) ostl additionalthread started to do the computation. There is no restriction that the sum of allNTHREADS from U SE_THREAD S callsm ay notexceed the num ber of CPU \(s\) in a system.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE

\section*{NAME}
using_threads - retums the current U se num ber set by the USE_THREADS subroutine

\section*{SYNOPSIS}
\(\mathbb{N} T E G E R F U N C T I D N U S \mathbb{N} G \_T H R E A D S()\)
\(\mathbb{N}\) TEGER*8 FUNCTION USTNG_THREADS_64 ()
F95 INTERFACE
\(\mathbb{N} T E G E R F U N C T I D N U S \mathbb{N} G \_T H R E A D S 0\)
\(\mathbb{N}\) TEGER (8) FUNCTION USING_THREADS_64 ()
C INTERFACE
\#include <sunperfh>
intusing_threads();
long using_threads_64 ();

\section*{PURPOSE}
using_threads THREADS w ill retum the current Use num ber from the USE_THREADS subroutine forthe calling thread.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vcfftb - com pute a periodic sequence from its Fourier coefficients. The V CFFT operations are norm alized, so a callof V CFFTF follow ed by a callofV CFFTB w ill retum the original sequence.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE VCFFTB M,N,X,XT,MD IM X,ROW COL,W SAVE)}
CHARACTER * 1 ROW COL
COM PLEX X MDIMX,*),XTMD\mathbb{M X,*),W SAVE (*)}
INTEGER M,N,MD IM X
SUBROUT\mathbb{NEVCFFTB_64M,N,X,XT,MD IM X,ROW COL,W SAVE)}
CHARACTER * 1 ROW COL
COM PLEX X MDIMX,*),XTMDIM X,*),W SAVE (*)
\mathbb{NTEGER*8M,N,MD IM X}

```
F95 INTERFACE
    SU BROUTINE FFTB ( \(\mathbb{M}\) ], \(\mathbb{N}], X, X T, \mathbb{M} D \mathbb{I} X], R O W C O L, W\) SAVE)
    CHARACTER (LEN=1) ::ROW COL
    COM PLEX,D \(\mathbb{M}\) ENSION (:) ::W SAVE
    COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : X, XT
    \(\mathbb{N} T E G E R:: M, N, M D \mathbb{I} X\)
    SUBROUTINE FFTB_64( \(\mathbb{M}], \mathbb{N}], X, X T, \mathbb{M D} \mathbb{M} X], R O W C O L, W\) SAVE)
    CHARACTER (LEN=1) ::ROW COL
    COMPLEX,D \(\mathbb{M}\) ENSION (:) ::W SAVE
    COM PLEX,D \(\mathbb{I M}\) ENSION (: : : : : X, XT
    \(\mathbb{N} T E G E R(8):: M, N, M D \mathbb{I} X\)

\section*{C INTERFACE}
\#include < sunperfh>
void vcfftb (intm, int \(n\), com plex *x, com plex *xt, int \(m\) dim \(x\), char row col, com plex *w save);
void vcfftb_64 (long m, long n, com plex *x, com plex *xt, long \(m\) dim \(x\), char row col, com plex *w save);

\section*{ARGUMENTS}
\(M\) (input) IfROW COL = R' or 'r', \(M\) is the num ber of sequences to be transform ed. O therw ise, \(M\) is the length of the sequences to be transform ed. M >= 0 .

N (input) IfROW COL = R 'or \(\mathrm{r}^{\prime}\), N is the length of the sequences to be transform ed. O therw ise, N is the num ber of sequences to be transform ed. \(\mathrm{N}>=0\).

X (input) On entry, ifROW COL = R 'or 'r'X \(M D \mathbb{I} X, N\) ) is an aray whose firstM row s contain the sequences to be transform ed. O therw ise, X M D \(\mathbb{I} \mathrm{X}, \mathbb{N}\) ) contains data sequences of length \(M\) stored in \(N\) colum ns of X .

XT (input)
A w ork aray. The size of this w orkspace depends on the num ber of threads that are used to execute this routine. There are various functions that can be used to determ ine the num ber of threads available (get env, available_threads, etc). The appropriate am ount, which is (num berof threads * length of data sequences), can then be dynam ically allocated for \(X T\) from the driver routine. IfX \(T\) can only be allocated statically, then the size of X T should be (length of data sequences * num berof sequences).

MDIMX (input)
Leading dim ension of the arrays X and X T as specified in a dim ension or type statem ent. MD \(\mathbb{M} X>=\) M.

ROW COL (input)
Indicates w hether to transform row ( \(\mathrm{R}^{\prime}\) or ' I ') orcolum ns (C 'or と').

W SAVE (input/output)
On entry, an array of dimension \((L 2+15)\) or greater, where L2 \(=2 \star \mathrm{M}\) ifROW COL \(=(\mathrm{R}\) 'or 'r'). O therw ise, L2 \(=2 \star \mathrm{~N}\). W SAVE is initialized by VCFFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vcfffecom pute the Fourier coefficients of a periodic sequence. The V CFFT operations are norm alized, so a call of V CFFTF follow ed by a callofV CFFTB w ill retum the original sequence.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE VCFFTF M,N,X,XT,MD IM X,ROW COL,W SAVE)}
CHARACTER * 1 ROW COL
COM PLEX X MDIMX,*),XTMD\mathbb{M X,*),W SAVE (*)}
INTEGER M,N,MD IM X
SUBROUT\mathbb{NEVCFFTF_64M,N,X,XT,MD IM X,ROW COL,W SAVE)}
CHARACTER * 1 ROW COL
COM PLEX X MDIMX,*),XTMD MM X,*),W SAVE (*)
\mathbb{NTEGER*8M,N,M D IM X}

```

\section*{F95 INTERFACE}

SU BROUTINE FFTF ( \(\mathbb{M}], \mathbb{N}], X, X T, \mathbb{M} D \mathbb{I} X], R O W C O L, W\) SAVE)
CHARACTER (LEN=1) ::ROW COL
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::W SAVE
COM PLEX,D \(\mathbb{M}\) ENSION (: : : : : X, XT
\(\mathbb{N} T E G E R:: M, N, M D \mathbb{I} X\)
SU BROUTINE FFTF_64 (M ], \(\mathbb{N}], X, X T, \mathbb{M D} \mathbb{M} X], R O W C O L, W\) SAVE)
CHARACTER (LEN=1) ::ROW COL
COMPLEX,D \(\mathbb{M}\) ENSION (:) ::W SAVE
COM PLEX,D \(\mathbb{I M}\) ENSION (: : : : : X, XT
\(\mathbb{N} T E G E R(8):: M, N, M D \mathbb{I} X\)

\section*{C INTERFACE}
\#include <sunperfh>
void vcfflf(intm, int \(n\), com plex *x, com plex *xt, int \(m\) dim \(x\), char row col, com plex *w save);
void vcfflif 64 (long m, long n, com plex *x, com plex *xt, long \(m\) dim \(x\), char row col, com plex *w save);

\section*{ARGUMENTS}
\(M\) (input) IfROW COL = R' or ' \(r\) ', \(M\) is the num ber of sequences to be transform ed. O therw ise, \(M\) is the length of the sequences to be transform ed. M >= 0 .

N (input) IfROW COL = R 'or \(\mathrm{r}^{\prime}\), N is the length of the sequences to be transform ed. O therw ise, N is the num ber of sequences to be transform ed. \(N>=0\).

X (input) On entry, ifROW COL = R'or 'r'X MD \(\mathbb{M} \mathrm{X}, \mathbb{N}\) ) is an aray whose firstM row s contain the sequences to be transform ed. O therw ise, X M D \(\mathbb{I} \mathrm{X}, \mathbb{N}\) ) contains data sequences of length \(M\) stored in \(N\) colum ns of X .

XT (input)
A w ork aray. The size of this w orkspace depends on the num ber of threads that are used to execute this routine. There are various functions that can be used to determ ine the num ber of threads available (get env, available_threads, etc). The appropriate am ount, which is (num berof threads * length of data sequences), can then be dynam ically allocated for \(X T\) from the driver routine. IfX \(T\) can only be allocated statically, then the size of X T should be (length of data sequences * num berof sequences).

MDIM X (input)
Leading dim ension of the arrays \(\mathrm{X} . \mathrm{M} \operatorname{D} \mathbb{M} \mathrm{X}>=\mathrm{M}\).
ROW COL (input)
Indicates w hetherdata sequences in X are stored row wise ( \(\mathrm{R}^{\prime}\) 'or 'r') orcolum n-w ise (C 'or C ).

W SAVE (input/output)
On entry, an array of dimension ( \(L 2+15\) ) or
greater, where L2 \(=2 * \mathrm{M}\) ifROW COL \(=\) (R 'or 'r'). O therw ise, L2 \(=2 \star \mathrm{~N}\). W SAVE is initialized by VCFFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vcffti-initialize the array W SAVE, which is used in both VCFFTF and VCFFTB.

\section*{SYNOPSIS}

SUBROUTINE VCFFTIN,W SAVE)
COM PLEX W SAVE (*)
\(\mathbb{I N}\) TEGER N
SUBROUTINEVCFFTI_64 \(\mathbb{N}\),W SAVE)

COM PLEX W SAVE (*)
\(\mathbb{N}\) TEGER*8N

F95 INTERFACE
SUBROUTINE VFFTIN, W SAVE)

COM PLEX,D \(\mathbb{M}\) ENSION (:) ::W SAVE \(\mathbb{N} T E G E R:: N\)

SU BROUTINE VFFTI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
COM PLEX,D \(\mathbb{M}\) ENSION (:) ::W SAVE \(\mathbb{N}\) TEGER (8) :: N

\section*{C INTERFACE}
\#include <sunperfh>
void vcffti(intn, com plex *w save);
void vcffti_ 64 (long n, com plex *w save);

\section*{ARGUMENTS}

N (imput) Length of the sequence to be transform ed. \(\mathrm{N}>=0\).

W SAVE (input)
O n entry, an array of dim ension ( \(2{ }^{*} \mathrm{~N}+15\) ) or greater. V CFFTI needs to be called only once to initialize W SAVE before calling VCFFTF and/or VCFFTB if \(N\) and W SAVE rem ain unchanged betw een these calls. Thus, subsequent transform \(s\) or inverse transform s of sam e size can be obtained faster than the first since they do not require indialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vcosqb - synthesize a Fourier sequence from its representation in term s of a cosine series \(w\) th odd \(w\) ave num bers. The VCOSQ operations are nom alized, so a call of VCOSQF follow ed by a call of V C O SQ B w ill retum the original sequence.

\section*{SYNOPSIS}
\[
\text { SUBROUTINE VCOSQB } M, N, X, X T, M D \mathbb{I} X, W \text { SAVE) }
\]
\(\mathbb{N} T E G E R M, N, M D \mathbb{I} X\)
REALXMDIMX,*),XTMDIMX,*),WSAVE(*)
SU BROUTINE VCOSQB_64 \(M, N, X, X T, M D \mathbb{M} X, W\) SAVE)
\(\mathbb{N}\) TEGER*8 M , N, M D \(\mathbb{I}\) X
REALXMDIM X,*), XTMDIMX,*),W SAVE (*)

\section*{F95 INTERFACE}

SU BROUTINE COSQB (M ], \(\mathbb{N}], X, X T, \mathbb{M} D \mathbb{M} X], W\) SAVE)
\(\mathbb{N} T E G E R:: M, N, M D \mathbb{I} X\)
REAL,D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL,D IM ENSION (:,:): : X,XT

SU BROUTINECOSQB_64 (M) \(\mathbb{M}, \mathbb{N}], X, X T, \mathbb{M} D \mathbb{I} X], W\) SAVE)
\(\mathbb{N} T E G E R(8):: M, N, M D \mathbb{I} X\)
REAL,D \(\mathbb{I M}\) ENSION (:) ::W SAVE
REAL,D \(\mathbb{M}\) ENSION (:,:) ::X,XT

\section*{C INTERFACE}
\#include <sunperfh>
void vcosqb (intm , intn, float *x, float *xt, int m dim \(x\), float *w save);
void vcosqb_64 (long m, long n, float *x, float *xt, long m dim x, float *w save);

\section*{ARGUMENTS}

M (input) The num ber of sequences to be transform ed. \(\mathrm{M}>=\) 0.

N (input) Length of the sequence to be transform ed. These subroutines are \(m\) ostefficientw hen N is a product of sm allprim es. \(\mathrm{N}>=0\).

X (input/output)
O n entry, the row s contain the sequences to be transform ed. On exit, the quarterw ave cosine synthesis of the input.

XT (input)
A w ork aray.

M D \(\mathbb{M}\) X (input)
Leading dim ension of the anrays X and \(\mathrm{X} T\) as specified in a dim ension ortype statem ent. MD \(\mathbb{I}\) X >= M .

W SAVE (input)
O n entry, an aray ofdim ension \((2 * N+15)\) or greater initialized by V C O SQ I.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vcosqf-com pute the Fourier coefficients in a cosine series
representation w ith only odd w ave num bers. The V COSQ operations are norm alized, so a call of V C O SQ F follow ed by a call of VCOSQB w ill retum the original sequence.

\section*{SYNOPSIS}
\[
\text { SU BROUTINE VCOSQF M,N,X,XT,MD } \mathbb{M} X, W \text { SAVE) }
\]
\(\mathbb{N} T E G E R M, N, M D \mathbb{I} X\)
REALXMDIMX,*),XTMDIMX,*),W SAVE (*)
SU BROUTINE VCOSQF_64M,N,X,XT,MD \(\mathbb{M} X, W\) SAVE)
\(\mathbb{N}\) TEGER*8 M , N, M D \(\mathbb{I}\) X
REALX MD \(\mathbb{M} X, *), X T M D \mathbb{M} X, *), W \operatorname{SAVE}(*)\)

\section*{F95 INTERFACE}

SU BROUTINE COSQF (M \(\mathbb{M}, \mathbb{N}], X, X T, \mathbb{M D} \mathbb{M} X], W\) SAVE)
\(\mathbb{N} T E G E R:: M, N, M D \mathbb{I} X\)
REAL,D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL,D IM ENSION (:,:): : X,XT

SU BROUTINE COSQF_64 (M ], \(\mathbb{N}], X, X T, \mathbb{M} D \mathbb{M} X], W\) SAVE)
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{MD} \mathbb{I} \mathrm{X}\)
REAL,D \(\mathbb{I M}\) ENSION (:) ::W SAVE
REAL,D \(\mathbb{M}\) ENSION (:,:) ::X,XT

\section*{C INTERFACE}
\#include <sunperfh>
void vcosqf(intm, intn, float * \(x\), float *xt, int \(m\) dim \(x\), float *w save);
void vcosqf_64 (long m, long n, float *x, float *xt, long m dim x, float *w save);

\section*{ARGUMENTS}

M (input)
The num ber of sequences to be transform ed. M >= 0 .

N (input) Length of the sequence to be transform ed. These subroutines are \(m\) ostefficientw hen \(N\) is a product of sm allprim es. \(\mathrm{N}>=0\).

X (input/output)
On entry, an array of length N containing the sequence to be transform ed. ForVCOSQ F, a real tw o-dim ensional array \(w\) th dim ensions of \(M D \mathbb{M} X \times\) N) whose rows contain the sequences to be transform ed. On exit, the quarter-w ave cosine transform of the input.

X T (input)
A realtw o-dim ensionalw ork array w ith dim ensions of \(M D \mathbb{M} \times \mathrm{X})\).

M D IM X (input)
Leading dim ension of the arrays X and XT as specified in a dim ension ortype statem ent. MD \(\mathbb{M} X>=\) M.

W SAVE (input)
O n entry, an aray of dim ension ( \(2 * N+15\) ) or greater initialized by V CO STI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vcosqi-initialize the array W SA VE, which is used in both VCOSQF and VCOSQB.

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE VCOSQIN,W SAVE)}

```
\(\mathbb{N}\) TEGER N
REALW SAVE (*)
SUBROUTINEVCOSQI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER*8 N
REALW SAVE (*)

\section*{F95 INTERFACE}

SU BROUTINE VCOSQIN,W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL,D \(\mathbb{M}\) ENSION (:) ::W SAVE

SU BROUTINEVCOSQI_64 (N,W SAVE)
\(\mathbb{N}\) TEGER (8) :: N
REAL,D \(\mathbb{M}\) ENSION (:) ::W SAVE

\section*{C INTERFACE}
\#include <sunperfh>
void vcosqi(intn, float *w save);
void vcosqi_ 64 (long n, float *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. The m ethod is m ost efficientw hen N is a product of sm allprim es.

W SAVE (input)
On entry, an array ofdim ension ( 2 * \(\mathrm{N}+15\) ) or greater. V C O SQ I needs to be called only once to intialize W SAVE before calling VCOSQF and/or VCOSQB if N and W SAVE rem ain unchanged betw een these calls. Thus, subsequent transform \(s\) or inverse transforms of sam e size can be obtained faster than the first since they do not require initialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vcost-com pute the discrete Fourier cosine transform of an even sequence. The V CO ST transform is norm alized, so a call ofV COST follow ed by a call ofV COST w ill retum the originalsequence.

\section*{SYNOPSIS}
```

    SUBROUT\mathbb{NE VCOSTM,N,X,XT,MD MM X,W SAVE)}
    \mathbb{NTEGERM,N,MD IM X}
    REALXMDIMX,*),XTMDIMX,*),W SAVE (*)
    SU BROUT\mathbb{NE VCOST_64M,N,X,XT,MD IM X,W SAVE)}
    INTEGER*8 M ,N,M D IM X
    REALX MDIM X,*),XTMDIM X,*),W SAVE (*)
    F95 INTERFACE
SUBROUT\mathbb{NE COST (\mathbb{M ], N ],X,XT, MD IM X ],W SAVE)}}\mathbf{M}\mathrm{ (N)}
\mathbb{NTEGER ::M ,N,M D IM X}
REAL,D\mathbb{M ENSION (:) ::W SAVE}
REAL,DIM ENSION (:,:)::X,XT
SU BROUTINE COST_64(\mathbb{M ], N ],X,XT, M D IM X],W SAVE)}
\mathbb{NTEGER (8)::M ,N,M D IM X}
REAL,D IM ENSION (:) ::W SAVE
REAL,D IM ENSION (:,:) ::X,XT
C INTERFACE
\#include <sunperfh>

```
void vcost(intm, int \(n\), float * x , float *xt, int m dim x , float*w save);
void voost 64 (long \(m\), long \(n\), float * x , float *xt, long m dim x, float *w save);

\section*{ARGUMENTS}

M (input)
The num ber of sequences to be transform ed. \(\mathrm{M}>=\) 0 .

N (input) Length of the sequence to be transform ed. These subroutines are m ost efficient \(w\) hen \(N-1\) is a productofsm allprim es. \(\mathrm{N}>=2\).

X (input/output)
On entry, an array of length N containing the sequence to be transform ed. ForVCOST, a real tw o-dim ensional array w ith dim ensions of \(M D \mathbb{M} \times \times\) \((\mathbb{N}+1)\) ) whose rows contain the sequences to be transform ed. On exit, the cosine transform of the input.

X T (input)
A realtw o-dim ensionalw ork array w ith dim ensions of \(M D \mathbb{M} \times \times(\mathbb{N}-1)\) ).

M D \(\mathbb{I M} X\) (input)
Leading dim ension of the arrays X and XT as specified in a dim ension ortype statem ent. MD \(\mathbb{M}\) X >= M.

W SAVE (input)
O n entry, an aray ofdim ension \((2 * N+15)\) or greater initialized by V CO STI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vcosti-initialize the anay W SAVE, which is used in VCOST .

\section*{SYNOPSIS}
SU BROUTINE VCOSTIN,W SAVE)
\(\mathbb{N}\) TEGER N
REALW SAVE (*)
SU BROUTINEVCOSTI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER*8N
REALW SAVE (*)
F95 INTERFACE
SU BROUTINE VCOSTIN,W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL,D \(\mathbb{I M}\) ENSION (:) ::W SAVE
SU BROUTINEVCOSTI_64N,W SAVE)
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}\)
REAL,D \(\mathbb{M}\) ENSION (:) ::W SAVE

\section*{C INTERFACE}
\#include <sunperfh>
void vcosti(intn, float *w save);
void vcosti_ 64 (long n, float *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. The \(m\) ethod is \(m\) ostefficientw hen \(N-1\) is a product of sm allprim es. \(\mathrm{N}>=2\).

W SAVE (input)
On entry, an array ofdim ension ( 2 * \(\mathrm{N}+15\) ) or greater. VCOSTI is called once to initialize W SAVE before calling VCOST and need notbe called again betw een calls to VCOST ifN and W SAVE rem ain unchanged. Thus, subsequent transform s of sam e size can be obtained faster than the first since they do not require initialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vdcosqb -synthesize a Fourier sequence from its representation in term s of a cosine series \(w\) th odd \(w\) ave num bers. The VCOSQ operations are norm alized, so a call of VCOSQF follow ed by a call of V C O SQ B w ill retum the original sequence.

\section*{SYNOPSIS}

SU BROUTINE VDCOSQB \(M, N, X, X T, M D \mathbb{M} X, W\) SAVE)
\(\mathbb{N}\) TEGER \(M, N, M D \mathbb{I} X\) DOUBLE PRECISIONXMD \(\mathbb{M} X, \star)\), XTMD \(\mathbb{M} X, *), W \operatorname{SAVE}\left({ }^{*}\right)\)

SU BROUTINE VDCOSQB_64 \(M, N, X, X T, M D \mathbb{I} X, W\) SAVE)
\(\mathbb{N}\) TEGER*8 M , N , M D \(\mathbb{I}\) X
DOUBLE PRECISIONXMD \(\operatorname{M} X, *), X T M D \mathbb{M} X, *), W\) SAVE (*)

\section*{F95 INTERFACE}

SU BROUTINE COSQB (M ], \(\mathbb{N}], X, X T, \mathbb{M} D \mathbb{M} X], W\) SAVE)
\(\mathbb{N} T E G E R:: M, N, M D \mathbb{I} X\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL (8), D IM ENSION (:,:) :: X,XT

SU BROUTINECOSQB_64 (M) \(\mathbb{M}, \mathbb{N}], X, X T, \mathbb{M} D \mathbb{I} X], W\) SAVE)
\(\mathbb{N} T E G E R(8):: M, N, M D \mathbb{I}\) X
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL (8),D IM ENSION (:,:) :: X,XT

\section*{C INTERFACE}
\#include < sunperfh>
void vdcosqb (intm , intn, double *x, double *xt, intm dim \(x\), double *w save);
void vdcosqib_64 (long m, long n, double *x, double *xt, long \(m\) dim \(x\), double *w save);

\section*{ARGUMENTS}

M (input) The num ber of sequences to be transform ed. \(\mathrm{M}>=\) 0.

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product ofsm allprim es. \(\mathrm{N}>=0\).

X (input/output)
On entry, the row s contain the sequences to be transform ed. On exit, the quarterw ave cosine synthesis of the input.

XT (input)
A w ork array.

M D \(\mathbb{I M} X\) (input)
Leading dim ension of the arays X and XT as specified in a dim ension ortype statem ent. MD \(\mathbb{M} X>=\) M.

W SAVE (input)
O n entry, an aray ofdim ension \((2 * N+15)\) or greater initialized by V CO SQ I.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vdcosqf-com pute the Fourier coefficients in a cosine series representation \(w\) th only odd \(w\) ave num bers. The V C O SQ operations are norm alized, so a callofV C O SQ F follow ed by a call of V C O SQ B w ill retum the original sequence.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE VDCOSQFM,N,X,XT,MD IMX,W SAVE)}
\mathbb{NTEGERM,N,MD IM X}
DOUBLE PRECISION X MD IM X,*),XTMD MM X,*),W SAVE (*)
SUBROUT\mathbb{NE VDCOSQF_64M,N,X,XT,MD MM,W SAVE)}
INTEGER*8 M ,N,M D IM X
DOUBLE PRECISION X M D IM X,*),XTMD MM X,*),W SAVE (*)

```

\section*{F95 INTERFACE}
```

    SUBROUT\mathbb{NE COSQF(M ], N ],X,XT, MD IM X ],W SAVE)}
    \mathbb{NTEGER::M,N,MD IM X}
    REAL (8),D IM ENSION (:) ::W SAVE
    REAL (8),D IM ENSION (:,:) ::X,XT
    SU BROUT\mathbb{NE COSQF_64(M ], N ],X,XT, M D IM X ],W SAVE)}
    \mathbb{NTEGER (8)::M ,N,M D IM X}
    REAL (8),D IM ENSION (:) ::W SAVE
    REAL (8),D IM ENSION (:,:) ::X,XT
    ```

\section*{C INTERFACE}
```

\#include <sunperfh>

```
void vdcosqf(intm , intn, double *x, double *xt, intm dim \(x\), double *W save);
void vdcosqf_64 (long m, long n, double *x, double *xt, long m dim x , double *w save);

\section*{ARGUMENTS}

M (input)
The num ber of sequences to be transform ed. M >= 0 .

N (input) Length of the sequence to be transform ed. These subroutines are \(m\) ostefficientw hen \(N\) is a product of sm allprim es. \(\mathrm{N}>=0\).

X (input/output)
On entry, an array of length \(N\) containing the sequence to be transform ed. ForVCOSQ F, a real tw o-dim ensional array w ith dim ensions of \(M \mathrm{D} \mathbb{I}\) X x N) whose rows contain the sequences to be transform ed. On exit, the quarter-w ave cosine transform of the input.

X T (input)
A realtw o-dim ensionalw ork array w ith dim ensions of \(M D \mathbb{M} \times \times N)\).

M D IM X (input)
Leading dim ension of the arrays X and XT as specified in a dim ension ortype statem ent. MD \(\mathbb{M} X>=\) M.

W SAVE (input)
O n entry, an array ofdim ension \((2 * N+15)\) or greater initialized by V CO SQ I.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vdcosqi-initialize the array W SA VE, which is used in both VCOSQF and VCOSQB.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE VDCOSQIN,W SAVE)}

```

\section*{\(\mathbb{N}\) TEGER N}

DOUBLE PRECISION W SAVE (*)
SUBROUTINEVDCOSQI_64 \(\mathbb{N}, W\) SAVE)
\(\mathbb{N}\) TEGER*8 N
DOUBLE PRECISION W SAVE (*)

\section*{F95 INTERFACE}

SU BROUTINE VCOSQIN,W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W SAVE

SU BROUTINEVCOSQI_64 (N,W SAVE)
\(\mathbb{N} T E G E R(8):: N\)
REAL (8),D \(\mathbb{I M}\) ENSION (:) ::W SAVE

\section*{C INTERFACE}
\#include < sunperfh>
void vdcosqi(intn, double *w save);
void vdcosqi_ 64 (long n, double *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. The m ethod is m ost efficientw hen N is a product of sm allprim es.

W SAVE (input)
On entry, an array ofdim ension ( 2 * \(\mathrm{N}+15\) ) or greater. VD C O SQ Ineeds to be called only once to intializeW SAVE before calling VDCOSQF and/or
VDCOSQB if N andW SAVE rem ain unchanged betw een
these calls. Thus, subsequent transform \(s\) or
inverse transforms of sam e size can be obtained
faster than the first since they do not require
initialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vdoost-com pute the discrete Fourier cosine transform of an even sequence. The VCO ST transform is norm alized, so a call ofV COST follow ed by a call ofV COST w ill retum the originalsequence.

\section*{SYNOPSIS}
\[
\text { SUBROUTINE VDCOST } M, N, X, X T, M D \mathbb{M} X, W \text { SAVE) }
\]
\(\mathbb{N}\) TEGER \(M, N, M D \mathbb{I} \mathrm{X}\) DOUBLE PRECISIONX MD \(\mathbb{M} X, *), X T M D \mathbb{M} X, *), W \operatorname{SAVE}\left({ }^{*}\right)\)

SU BROUTINE VDCOST_64 \(M, N, X, X T, M D \mathbb{M} X, W\) SAVE)
\(\mathbb{N}\) TEGER*8 M , N , M D \(\mathbb{I}\) X
DOUBLE PRECISIONX MD \(\mathbb{M} X, \star)\), XTMD \(\mathbb{M} X, \star), W \operatorname{SAVE}\left({ }^{*}\right)\)

\section*{F95 INTERFACE}

SU BROUTINE COST (M ], \(\mathbb{N}], X, X T, \mathbb{M} D \mathbb{I} X], W\) SAVE)
\(\mathbb{N} T E G E R:: M, N, M D \mathbb{I} X\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL (8), D IM ENSION (:,:) :: X,XT

SU BROUTINECOST_64(M) \(\mathbb{M}], X, X T, \mathbb{M} D \mathbb{M} X], W\) SAVE)
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{M}, \mathrm{N}, \mathrm{MD} \mathbb{I} \mathrm{X}\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL (8),D \(\mathbb{I M}\) ENSION (:,:) ::X,XT

\section*{C INTERFACE}
\#include <sunperfh>
void vdcost(intm , intn, double *x, double *xt, int m dim \(x\), double *W save);
void vdcost_64 (long m, long n, double *x, double *xt, long m dim x , double *w save);

\section*{ARGUMENTS}

M (input)
The num ber of sequences to be transform ed. M >= 0 .

N (input) Length of the sequence to be transform ed. These subroutines are m ost efficient w hen \(\mathrm{N}-1\) is a productofsm allprim es. \(\mathrm{N}>=2\).

X (input/output)
On entry, an array of length N containing the sequence to be transform ed. ForVCOST, a real tw o-dim ensional array w ith dim ensions of \(M D \mathbb{M} \times \times\) \((\mathbb{N}+1))\) whose row s contain the sequences to be transform ed. On exit, the cosine transform of the input.

X T (input)
A realtw o-dim ensionalw ork array w ith dim ensions of \(M D \mathbb{M} \times \times(\mathbb{N}-1)\) ).

M D \(\mathbb{I M} X\) (input)
Leading dim ension of the arrays X and XT as specified in a dim ension ortype statem ent. MD \(\mathbb{M} X>=\) M.

W SAVE (input)
O n entry, an array ofdim ension \((2 * N+15)\) or greater initialized by VD COSTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vdcosti-initialize the array W SAVE, which is used in VCOST.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE VDCOSTIN,W SAVE)}

```
\(\mathbb{N}\) TEGER N
DOUBLE PRECISION W SAVE (*)
SUBROUTINEVDCOSTI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER*8 N
DOUBLE PRECISION W SAVE (*)

\section*{F95 INTERFACE}

SUBROUTINE VCOSTIN,W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL (8),D IM ENSION (:) ::W SAVE

SU BROUTINE VCOSTI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER (8) ::N
REAL (8),D \(\mathbb{I M}\) ENSION (:) ::W SAVE
C INTERFACE
\#include <sunperfh>
void vdcosti(intn, double *w save);
void vdcosti_ 64 (long n, double *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. The m ethod is m ostefficientw hen \(\mathrm{N}-1\) is a product ofsm allprim es. \(\mathrm{N}>=2\).

W SAVE (input)
On entry, an array ofdim ension ( 2 * \(\mathrm{N}+15\) ) or greater. VDCOSTI is called once to initialize W SA VE before calling VDCOST and need notbe called again betw een calls to VDCOST if N and W SAVE rem ain unchanged. Thus, subsequent transform \(s\) of same size can be obtained fasterthan the first since they do not require initialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vdfftb -com pute a periodic sequence from its Fourier coefficients. The V RFFT operations are norm alized, so a call of VRFFTF follow ed by a callofV RFFTB w ill retum the original sequence.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE VDFFTB M,N,X,XT,MD IM X,W SAVE)}
\mathbb{NTEGERM,N,MD IM X}
DOUBLE PRECISION X MD IM X,*),XTMD MM X,*),W SAVE (*)
SUBROUT\mathbb{NE VDFFTB_64M,N,X,XT,MD IM X,W SAVE)}
\mathbb{NTEGER*8 M ,N,M D IM X}
DOUBLE PRECISION X M D IM X,*),XTMD MM X,*),W SAVE (*)

```

\section*{F95 INTERFACE}

SUBROUTINE FFTB (M ], \(\mathbb{N}], X, X T, \mathbb{M} D \mathbb{M} X], W\) SAVE)
\(\mathbb{N} T E G E R:: M, N, M D \mathbb{I} X\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL (8), D IM ENSION (:,:) :: X,XT
SUBROUTINE FFTB_64 ( \(\mathbb{M}], \mathbb{N}], X, X T, \mathbb{M} D \mathbb{I} X], W\) SAVE)
\(\mathbb{N} T E G E R(8):: M, N, M D \mathbb{I} X\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL (8),D \(\mathbb{I}\) ENSION (:,:) ::X,XT

\section*{C INTERFACE}
\#include < sunperfh>
void vdfftb (intm , intn, double \({ }_{x}\), double *xt, int \(m\) dim \(x\), double *W save);
void vdfftb_64 (long m, long \(n\), double *x, double *xt, long \(m\) dim \(x\), double *w save);

\section*{ARGUMENTS}

M (input) The num ber of sequences to be transform ed. \(\mathrm{M}>=\) 0.

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product ofsm allprim es. \(\mathrm{N}>=0\).

X (input) \(O n\) entry, an aray of length N containing the sequence to be transform ed. ForVRFFTF, a real tw o-dim ensionalamay X \((M, N)\) whose rows contain the sequences to be transform ed.

X T (input)
A realtw o-dim ensionalw ork aray w ith dim ensions of \(M D \mathbb{M} \times \mathrm{X})\).

M D \(\mathbb{M}\) X (input)
Leading dim ension of the arrays X and X T as specified in a dim ension ortype statem ent. MD \(\mathbb{I M} X>=\) M.

W SAVE (input)
O n entry, an array of dim ension \((\mathbb{N}+15)\) or greater initialized by V RFFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vdfflf-com pute the Fourier coefficients of a periodic sequence. The V RFFT operations are norm alized, so a callof VRFFTF follow ed by a callofV RFFTB w ill retum the original sequence.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE VDFFTFM,N,X,XT,MDIMX,W SAVE)}
\mathbb{NTEGERM,N,MD IM X}
DOUBLE PRECISION X MD IM X,*),XT MD IM X,*),W SAVE (*)
SUBROUT\mathbb{NE VDFFTF_64M,N,X,XT,MD IM X,W SAVE)}
INTEGER*8 M ,N,M D IM X
DOUBLE PRECISION X M D IM X,*),XTMD MM X,*),W SAVE (*)

```

\section*{F95 INTERFACE}

SU BROUTINE FFTF (M ], \(\mathbb{N}], X, X T,\left[\begin{array}{l}\text { D } \mathbb{M} X], W \\ \text { SAVE) }\end{array}\right.\)
\(\mathbb{N} T E G E R:: M, N, M D \mathbb{I} X\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL (8), D IM ENSION (:,:) :: X,XT

SU BROUTINE FFTF_64 (M ], \(\mathbb{N}], X, X T, \mathbb{M} D \mathbb{I} X], W\) SAVE)
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{M}, \mathrm{N}, \mathrm{MD} \mathbb{I} \mathrm{X}\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL (8),D \(\mathbb{I}\) ENSION (:,:) ::X,XT

\section*{C INTERFACE}
\#include < sunperfh>
void vdffff(intm, intn, double \({ }^{*} x\), double \({ }^{*} x\), int \(m\) dim \(x\), double *W save);
void vdfflif 64 (long m, long n, double *x, double *xt, long \(m\) dim \(x\), double *w save);

\section*{ARGUMENTS}

M (input)
The num ber of sequences to be transform ed. \(\mathrm{M}>=\) 0 .

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product of sm allprim es. \(\mathrm{N}>=0\).
\(X\) (input) On entry, an array of length \(N\) containing the sequence to be transform ed. ForVRFFTF, a real tw o-dim ensional anray \(\mathrm{X}(\mathrm{M}, \mathrm{N})\) whose rows contain the sequences to be transform ed.

X T (input)
A realtw o-dim ensionalw ork amay w ith dim ensions of \(M D \mathbb{M} \times \times N)\).

MDIMX (input)
Leading dim ension of the amrays X and X T as specified in a dim ension ortype statem ent. M D \(\mathbb{M}\) X >= M.

W SAVE (input)
O n entry, an array ofdim ension \((\mathbb{N}+15)\) or greater initialized by V RFFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vdffti-initialize the anray W SA VE, which is used in both VRFFTF andVRFFTB.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE VDFFTIN,W SAVE)}

```
\(\mathbb{N}\) TEGER N
DOUBLE PRECISION W SAVE (*)
SUBROUTINEVDFFTI_64 \(\mathbb{N}\),W SAVE)
\(\mathbb{N}\) TEGER*8 N
DOUBLE PRECISION W SAVE (*)

\section*{F95 INTERFACE}

SUBROUTINE VFFTIN,W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL (8),D IM ENSION (:) ::W SAVE
SU BROUTINE VFFTI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
\(\mathbb{N} T E G E R(8):: N\)
REAL (8),D \(\mathbb{I M}\) ENSION (:) ::W SAVE

\section*{C INTERFACE}
\#include <sunperfh>
void vdffti(intn, double *w save);
void vdffti_ 64 (long \(n\), double *w save);

\section*{ARGUMENTS}

N (imput) Length of the sequence to be transform ed. \(\mathrm{N}>=0\).

W SAVE (input)
On entry, an array of dimension \((\mathbb{N}+15\) ) or greater. VRFFTI needs to be called only once to in itialize W SAVE before calling VRFFTF and/or VRFFTB if \(N\) and W SAVE rem ain unchanged betw een these calls. Thus, subsequent transform \(s\) or inverse transform s of sam e size can be obtained faster than the first since they do not require indialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vdsinqb - synthesize a Fourier sequence from its representation in term sofa sine series \(w\) th odd \(w\) ave num bers. The \(V S \mathbb{N} Q\) operations are norm alized, so a call of \(V S \mathbb{N} Q F\) fol low ed by a callof \(V S \mathbb{N} Q B\) w ill retum the original sequence.

\section*{SYNOPSIS}
\[
\text { SU BROUTINE VDSINQB } M, N, X, X T, M D \mathbb{M} X, W \text { SAVE) }
\]
\(\mathbb{N}\) TEGER \(M, N, M D \mathbb{I} \mathrm{X}\) DOUBLE PRECISION X MDIM X,*), XTMD \(\mathbb{M} X, \star), \mathrm{W} \operatorname{SAVE}\left({ }^{*}\right)\)

SU BROUTINEVDSINQB_64 \(\mathbb{M}, N, X, X T, M D \mathbb{M} X, W\) SAVE)
\(\mathbb{N}\) TEGER*8 M , N , M D \(\mathbb{I}\) X
DOUBLE PRECISIONXMD \(\operatorname{M} X, *), X T M D \mathbb{M} X, *), W\) SAVE (*)

\section*{F95 INTERFACE}

SUBROUTINE SINQB(M) \(\mathbb{N}], X, X T, \mathbb{M} D \mathbb{M} X], W\) SAVE)
\(\mathbb{N} T E G E R:: M, N, M D \mathbb{I} X\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL (8), D IM ENSION (:,:) ::X,XT

SU BROUTINE SINQB_64( \(\mathbb{M}], \mathbb{N}], X, X T, \mathbb{M} D \mathbb{M} X], W\) SAVE)
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{M}, \mathrm{N}, \mathrm{MD} \mathbb{I} \mathrm{X}\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL (8),D \(\mathbb{I M}\) ENSION (:,:) ::X,XT

\section*{C INTERFACE}
\#include < sunperfh>
void vdsinqb (intm , intn, double *x, double *xt, intm dim \(x\), double *w save);
void vdsinql_64 (long m, long n, double *x, double *xt, long m dim \(x\), double *w save);

\section*{ARGUMENTS}

M (input)
The num ber of sequences to be transform ed. \(\mathrm{M}>=\) 0 .

N (input) Length of the sequence to be transform ed. These subroutines are \(m\) ostefficientw hen \(N\) is a product of sm allprim es. \(\mathrm{N}>=0\).

X (input/output)
On entry, a real two-dimensional anay with dimensions of \((M D \mathbb{M} X \times N)\) whose row scontain the sequences to be transform ed. On exit, the quarterw ave sine synthesis of the input.

X T (input)
A realtw o-dim ensionalw ork array w ith dim ensions of \(M D \mathbb{M} \times \mathrm{X})\).

M D \(\mathbb{I M} X\) (input)
Leading dim ension of the arrays X and XT as specified in a dim ension ortype statem ent. M D \(\mathbb{M}\) X >= M.

\section*{W SAVE (input)}

O n entry, an array \(w\) ith dim ension of at least ( 2 *
\(\mathrm{N}+15\) ) for vectorsubroutines, initialized by
V SIN Q I.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vdsingf - com pute the Fourier coefficients in a sine series
representation w ith only odd w ave num bers. The VSINQ operations are norm alized, so a callofV SIN Q F follow ed by a call of \(\operatorname{SIN} Q B\) w ill retum the original sequence.

\section*{SYNOPSIS}

SUBROUTINEVDS \(\mathbb{N} Q F M, N, X, X T, M D \mathbb{M} X, W \operatorname{SAVE})\)
\(\mathbb{N}\) TEGER \(M, N, M D \mathbb{I} X\) DOUBLE PRECISION X MDIM X,*), XTMDIM \(\left.\mathrm{X},{ }^{\star}\right)\), W SAVE (*)

SU BROUTINEVDSINQF_64 M,N,X,XT,MD \(\mathbb{M} X, W\) SAVE)
\(\mathbb{N}\) TEGER*8 M , N , M D \(\mathbb{I}\) X


\section*{F95 INTERFACE}

SU BROUTINESINQF (M ], \(\mathbb{N}], X, X T, \mathbb{M} D \mathbb{M} X], W\) SAVE)
\(\mathbb{N} T E G E R:: M, N, M D \mathbb{I} X\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL (8), D IM ENSION (:,:) ::X,XT

SUBROUTINESNQF_64( \(\mathbb{M}], \mathbb{N}], X, X T, \mathbb{M} D \mathbb{M} X], W\) SAVE)
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{MD} \operatorname{IM} \mathrm{X}\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::X,XT

\section*{C INTERFACE}
\#include < sunperfh>
void vdsingf(intm , intn, double *\({ }_{x}\), double *xt, intm dim \(x\), double *W save);
void vdsingf_64 (long m, long n, double *x, double *xt, long \(m\) dim \(x\), double *w save);

\section*{ARGUMENTS}

M (input)
The num ber of sequences to be transform ed. M >= 0 .

N (input) Length of the sequence to be transform ed. These subroutines are \(m\) ostefficientw hen \(N\) is a product of sm allprim es. \(\mathrm{N}>=0\).

X (input/output)
On entry, an array of length \(N\) containing the sequence to be transform ed. ForV \(S \mathbb{N} Q F\), a real tw o-dim ensional array \(w\) ith dim ensions of \(M D \mathbb{I M} X \quad x\) N) whose rows contain the sequences to be transform ed. On exit, the quarterw ave sine transform of the input.

X T (input)
A realtw o-dim ensionalw ork array w ith dim ensions of \(M D \mathbb{M} \times \mathrm{X})\).

M D \(\mathbb{I M} X\) (input)
Leading dim ension of the arrays X and XT as specified in a dim ension ortype statem ent. MD \(\mathbb{M}\) X >= M.

W SAVE (input)
O n entry, an array w ith dim ension of at least (2 * \(N+15)\), initialized by \(V \operatorname{SINQ}\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vdsinqi-initialize the array \(W\) SA VE, which is used in both \(V S \mathbb{N} Q F\) and \(V S \mathbb{N} Q B\).

\section*{SYNOPSIS}

> SU BROUTINE VDSNQIN,W SAVE)

\section*{\(\mathbb{N}\) TEGER N}

DOUBLE PRECISION W SAVE (*)
SUBROUTINEVDSINQI_64N,W SAVE)
\(\mathbb{N}\) TEGER*8 N
DOUBLE PRECISION W SAVE (*)

F95 INTERFACE
SU BROUTINEVSINQIN, W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W SAVE

SU BROUTINE VSINQI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
\(\mathbb{N} T E G E R(8):: N\)
REAL (8), D IM ENSION (:) ::W SAVE

\section*{C INTERFACE}
\#include <sunperfh>
void vdsinqi(intn, double *w save);
void vdsinqi_ 64 (long n, double *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. The \(m\) ethod is \(m\) ost efficientw hen \(N\) is a product of sm allprim es.

W SAVE (input)
O \(n\) entry, an array \(w\) ith a dim ension of at least (2 * \(\mathrm{N}+15\) ). The sam ew ork array can be used for both \(V S \mathbb{N} Q F\) and \(V S \mathbb{N} Q B\) as long as \(N\) rem ains unchanged. D ifferent W SAVE arrays are required fordifferentvalues of N . This initialization
does not have to be repeated betw een calls to \(V S \mathbb{N} Q F \operatorname{orV} \operatorname{SIN} Q B\) as long as \(N\) and \(W\) SAVE rem ain unchanged, thus subsequent transform s can be obtained faster than the first.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vdsint-com pute the discrete Fourier sine transform of an odd sequence. The VSINT transform s are unnorm alized inverses of them selves, so a call of VSINT follow ed by another callofV SIN T w illm ultiply the input sequence by 2 * \((\mathbb{N}+1)\). The V SIN \(T\) transform s are norm alized, so a call of VSINT follow ed by a callofV SIN T w ill retum the original sequence.

\section*{SYNOPSIS}

SU BROUTINEVDSINTM,N,X,XT,MD \(\mathbb{M} X, W\) SAVE)
\(\mathbb{N}\) TEGER \(\mathrm{M}, \mathrm{N}, \mathrm{MD} \mathbb{I} \mathrm{X}\)
DOUBLE PRECISIONXMD \(\mathbb{M} X, \star)\), XTMD \(\left.\mathbb{M} X,{ }^{\star}\right), \mathrm{W} \operatorname{SAVE}\left({ }^{*}\right)\)

SU BROUTINE VDSINT_64 M,N,X,XT,MD \(\mathbb{M} X, W\) SAVE)
\(\mathbb{N}\) TEGER*8 \(\mathrm{M}, \mathrm{N}, \mathrm{M} \operatorname{D} \mathbb{I} \mathrm{X}\)


\section*{F95 INTERFACE}

SU BROUTINE SNT (M ], \(\mathbb{N}], X, X T, \mathbb{M} D \mathbb{M} X], W\) SAVE)
\(\mathbb{N} T E G E R:: M, N, M D \mathbb{M} X\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL (8), D IM ENSION (:,:) ::X,XT
SU BROUTINESINT_64 ( \(\mathbb{M}], \mathbb{N}], X, X T, \mathbb{M} D \mathbb{M} X], W\) SAVE)
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{M}, \mathrm{N}, \mathrm{MD} \mathbb{I} \mathrm{X}\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL (8),D \(\mathbb{M}\) ENSION (:,:) ::X,XT

\section*{C INTERFACE}
\#include <sunperfh>
void vdsint(intm, intn, double *x, double *xt, int m dim \(x\), double *w save);
void vdsint_64 (long m, long \(n\), double *x, double *xt, long \(m\) dim \(x\), double *w save);

\section*{ARGUMENTS}

M (input)
The num ber of sequences to be transform ed. \(\mathrm{M}>=\) 0 .
N (input) Length of the sequence to be transform ed. These subroutines are mostefficientw hen \(\mathrm{N}+1\) is a product of sm allprim es. \(\mathrm{N}>=0\).

X (input/output)
O n entry, a realtw o-dim ensional array \(w\) th dim ensions of \(\mathbb{M D} \mathbb{I} \times \times(\mathbb{N}+1))\) whose row s contain the sequences to be transform ed. On exit, the sine transform of the input.

X T (input/output)
A realtw o-dim ensionalw ork array w ith dim ensions of \(M D \mathbb{M} \times \times(\mathbb{N}+1)\) ).

M D \(\mathbb{M}\) X (input)
Leading dim ension of the arrays X and XT as specified in a dim ension ortype statem ent. MD \(\mathbb{M} X>=\) M.

W SAVE (input)
On entry, an array w ith dim ension of at least \(\operatorname{int}(2.5 * N+15)\) initialized by \(V S \mathbb{N} T\) I.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vdsinti-initialize the array W SAVE, which is used in subroutine VSINT.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEVDSINTIN,W SAVE)}

```
\(\mathbb{N}\) TEGER N
DOUBLE PRECISION W SAVE (*)
SU BROUTINEVDSINTI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER*8 N
DOUBLE PRECISION W SAVE (*)
F95 INTERFACE
    SU BROUTINEVSINTIN,W SAVE)
    \(\mathbb{N} T E G E R:: N\)
    REAL (8), D \(\mathbb{M}\) ENSION (:) ::W SAVE
    SUBROUTINEVSINTI_64 \(\mathbb{N}, W\) SAVE)
    \(\mathbb{N} T E G E R(8):: N\)
    REAL (8),D IM ENSION (:) ::W SAVE
C INTERFACE
    \#include <sunperfh>
    void vdsinti(intn, double *w save);
    void vdsinti_ 64 (long n, double *w save);

\section*{ARGUMENTS}

N (imput) Length of the sequence to be transform ed. \(\mathrm{N}>=0\).

W SAVE (input)
On entry, an array of dim ension ( \(2 \mathrm{~N}+\mathrm{N} / 2+15\) ) or greater. V SIN T I is called once to initialize W SA V E before calling \(V S \mathbb{N} T\) and need notbe called again between calls to VSINT if N andW SAVE rem ain unchanged. Thus, subsequent transform s of sam e size can be obtained faster than the first since they do not require initialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vrfftb -com pute a periodic sequence from its Fourier coefficients. The V RFFT operations are norm alized, so a callof VRFFTF follow ed by a callofV RFFTB w ill retum the original sequence.

\section*{SYNOPSIS}
```

    SU BROUT\mathbb{NE VRFFTB M,N,X,XT,MD IM X,W SAVE)}
    \mathbb{NTEGERM,N,MD IM X}
    REALXMDIMX,*),XTMDIMX,*),W SAVE (*)
    SUBROUT\mathbb{NE VRFFTB_64M,N,X,XT,MD IM X,W SAVE)}
    INTEGER*8 M ,N,M D IM X
    REALX MD M X,*),XTMDIMX,*),W SAVE (*)
    F95 INTERFACE
SUBROUT\mathbb{NE FFTB (M ], N ],X,XT, MD IM X ],W SAVE)}
\mathbb{NTEGER::M,N,MD IM X}
REAL,D\mathbb{M ENSION (:) ::W SAVE}
REAL,DIM ENSION (:,:)::X,XT

```

```

    \mathbb{NTEGER (8)::M ,N,M D IM X}
    REAL,D IM ENSION (:) ::W SAVE
    REAL,D IM ENSION (:,:) ::X,XT
    C INTERFACE
\#include <sunperfh>

```
void vrfftb (intm, intn, float *x, float *xt, int \(m\) dim \(x\), float *w save);
void vrfftb_64 (long m , long n, float *x, float *xt, long m dim x , float * w save);

\section*{ARGUMENTS}

M (input) The num ber of sequences to be transform ed. M >= 0.

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product ofsm allprim es. \(\mathrm{N}>=0\).

X (input) On entry, an array of length N containing the sequence to be transform ed. ForVRFFTF, a real tw o-dim ensionalamay \(X(M, N\) ) whose row s contain the sequenœes to be transform ed.

XT (input)
A realtw o-dim ensionalw ork array w ith dim ensions of \(M D \mathbb{M} \times \mathrm{N}\) ).

M D IM X (input)
Leading dim ension of the anays X and \(\mathrm{X} T\) as specified in a dim ension or type statem ent. M D \(\mathbb{M} X>=\) M.

W SAVE (input)
O n entry, an array ofdim ension \((\mathbb{N}+15\) ) or greater initialized by V RFFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vrffle-com pute the Fourier coefficients of a periodic sequence. The V RFFT operations are nom alized, so a call of VRFFTF follow ed by a callofV RFFTB w ill retum the original sequence.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE VRFFTFM,N,X,XT,MD IM X,W SAVE)}
\mathbb{NTEGERM,N,MD IM X}
REALXMDIMX,*),XTMDIMX,*),W SAVE (*)
SUBROUT\mathbb{NE VRFFTF_64 M,N,X,XT,MD IM X,W SAVE)}
INTEGER*8 M ,N,M D IM X
REALX MDIM X,*),XTMDIM X,*),W SAVE (*)
F95 INTERFACE
SUBROUT\mathbb{NE FFTF (M ], N ],X,XT, MD IM X],W SAVE)}
\mathbb{NTEGER::M,N,MD IM X}
REAL,D\mathbb{M ENSION (:) ::W SAVE}
REAL,DIM ENSION (:,:)::X,XT
SU BROUT\mathbb{NE FFTF_64(\mathbb{M ], N ],X,XT, M D IM X ],W SAVE)}}\mathbf{~}\mathrm{ ( }
\mathbb{NTEGER (8)::M ,N,M D IM X}
REAL,D IM ENSION (:) ::W SAVE
REAL,D IM ENSION (:,:) ::X,XT
C INTERFACE
\#include <sunperfh>

```
void vrffff(intm , intn, float *x, float *xt, int m dim \(x\), float*w save);
void vrffff_64 (long m, long n, float *x, float *xt, long m dim x, float *w save);

\section*{ARGUMENTS}

M (input)
The num ber of sequences to be transform ed. \(\mathrm{M}>=\) 0 .

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product of sm allprim es. \(\mathrm{N}>=0\).
\(X\) (input) On entry, an array of length \(N\) containing the sequence to be transform ed. ForVRFFTF, a real tw o-dim ensional anray \(\mathrm{X}(\mathrm{M}, \mathrm{N})\) whose rows contain the sequences to be transform ed.

X T (input)
A realtw o-dim ensionalw ork amay w ith dim ensions of \(M D \mathbb{M} \times \times N)\).

MDIMX (input)
Leading dim ension of the amrays X and X T as specified in a dim ension ortype statem ent. M D \(\mathbb{M}\) X >= M.

W SAVE (input)
O n entry, an array ofdim ension \((\mathbb{N}+15)\) or greater initialized by V RFFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vrffti-initialize the array W SAVE, which is used in both VRFFTF andVRFFTB.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE VRFFTIN,W SAVE)}

```
INTEGER N
REALW SAVE (*)
SUBROUTINEVRFFTI_64 \(\mathbb{N}\),W SAVE)
\(\mathbb{N}\) TEGER*8 N
REALW SAVE (*)

F95 INTERFACE
SU BROUTINE VFFTIN, W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL,D \(\mathbb{M}\) ENSION (:) ::W SAVE

SU BROUTINE VFFTI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER (8) :: N
REAL,D \(\mathbb{M}\) ENSION (:) ::W SAVE

\section*{C INTERFACE}
\#include <sunperfh>
void vrffti(intn, float *W save);
void vrffti_ 64 (long n, float *w save);

\section*{ARGUMENTS}

N (imput) Length of the sequence to be transform ed. \(\mathrm{N}>=0\).

W SAVE (input)
On entry, an array of dimension \((\mathbb{N}+15\) ) or greater. VRFFTI needs to be called only once to in itialize W SAVE before calling VRFFTF and/or VRFFTB if \(N\) and W SAVE rem ain unchanged betw een these calls. Thus, subsequent transform \(s\) or inverse transform s of sam e size can be obtained faster than the first since they do not require indialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vsinqb - synthesize a Fourier sequence from its representation in term s of a sine series \(w\) th odd \(w\) ave num bers. The \(V S \mathbb{N} Q\) operations are norm alized, so a call of \(V S \mathbb{N} Q F\) fol low ed by a callof \(V S \mathbb{N} Q B\) w ill retum the original sequence.

\section*{SYNOPSIS}
```

    SUBROUT\mathbb{NEVSNNQBM,N,X,XT,MDIMX,W SAVE)}
    \mathbb{NTEGERM,N,MD IM X}
    REALXMDIMX,*),XTMDIMX,*),W SAVE (*)
    SUBROUT\mathbb{NEVSINQB_64 M,N,X,XT,MD IM X,W SAVE)}
    INTEGER*8 M ,N,M D IM X
    REALX MD M X,*),XTMDIMX,*),W SAVE (*)
    F95 INTERFACE
SUBROUT\mathbb{NE SINQB(M ], N ],X,XT, MD IM X],W SAVE)}
\mathbb{NTEGER::M,N,MD IM X}
REAL,D\mathbb{M ENSION (:) ::W SAVE}
REAL,D IM ENSION (:,:) ::X,XT

```

```

    \mathbb{NTEGER (8)::M ,N,M D IM X}
    REAL,D IM ENSION (:) ::W SAVE
    REAL,D IM ENSION (:,:) ::X,XT
    C INTERFACE
\#include <sunperfh>

```
void vsinqio (intm, intn, float *x, float *xt, int \(m\) dim \(x\), float *w save);
void vsinqib_64 (long m, long n, float *x, float *xt, long m dim x, float *w save);

\section*{ARGUMENTS}

M (input)
The num ber of sequences to be transform ed. M >= 0 .

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product of sm allprim es. \(\mathrm{N}>=0\).

X (input/output)
On entry, a real two-dimensional aray with dim ensions of \((\mathrm{MD} \mathbb{M} \times \times \mathrm{N})\) whose row scontain the sequences to be transform ed. On exit, the quarterw ave sine synthesis of the input.

X T (input)
A realtw o-dim ensionalw ork array \(w\) ith dim ensions of \(M D \mathbb{M} \times \mathrm{X})\).

M D \(\mathbb{I M} X\) (input)
Leading dim ension of the arrays X and XT as specified in a dim ension ortype statem ent. M D \(\mathbb{M}\) X >= M.

W SAVE (input)
O n entry, an array w ith dim ension of at least (2 *
\(\mathrm{N}+15\) ) for vectorsubroutines, initialized by
VSN Q I.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vsinqf-com pute the Fouriercoefficients in a sine series representation \(w\) th only odd \(w\) ave num bers. The VSINQ operations are norm alized, so a callofV SIN Q F follow ed by a call of \(\operatorname{SIN} Q B\) w ill retum the original sequence.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NEVSNNQFM,N,X,XT,MDIMX,W SAVE)}
\mathbb{NTEGERM,N,MD IM X}
REALXMDIMX,*),XTMDIMX,*),W SAVE (*)
SUBROUT\mathbb{NE VSINQF_64M,N,X,XT,MD IM X,W SAVE)}
INTEGER*8 M ,N,M D IM X
REALX MDIM X,*),XTMDIM X,*),W SAVE (*)

```

\section*{F95 INTERFACE}
```

SUBROUTINESTNQF(M) $\mathbb{N}], X, X T, \mathbb{M} D \mathbb{M} X], W$ SAVE)
$\mathbb{N} T E G E R:: M, N, M D \mathbb{I} X$
REAL,D $\mathbb{M}$ ENSION (:) ::W SAVE
REAL,D IM ENSION (:,:): : X,XT
SUBROUTINESNQF_64( $\mathbb{M}], \mathbb{N}], X, X T, \mathbb{M} D \mathbb{M} X], W$ SAVE)
$\mathbb{N} T E G E R(8):: M, N, M D \mathbb{I}$ X
REAL,D $\mathbb{I}$ ENSION (:) ::W SAVE
REAL,D $\mathbb{M}$ ENSION (:,:) ::X,XT

```

\section*{C INTERFACE}
```

\#include < sunperfh>

```
void vsingf(intm, intn, float * \(x\), float * \(x\), int \(m\) dim \(x\), float *w save);
void vsinqf_64 (long m, long \(n\), float *x, float *xt, long m dim x , float *w save);

\section*{ARGUMENTS}

M (input)
The num ber of sequences to be transform ed. M >= 0 .

N (input) Length of the sequence to be transform ed. These subroutines are \(m\) ostefficientw hen \(N\) is a product of sm allprim es. \(\mathrm{N}>=0\).

X (input/output)
On entry, an array of length N containing the sequence to be transform ed. ForV SIN QF, a real tw o-dim ensional array with dim ensions of MD \(\mathbb{M}\) X x N) whose rows contain the sequences to be transformed. On exit, the quarterw ave sine transform of the input.

XT (input)
A realtw o-dim ensionalw ork array w ith dim ensions of \(M D \mathbb{M} \times \mathrm{X})\).

MDIM X (input)
Leading dim ension of the anays X and X T as specified in a dim ension or type statem ent. M D \(\mathbb{M}\) X >= M.

W SAVE (input)
O n entry, an array with dim ension of at least (2 * \(\mathrm{N}+15\) ), initialized by \(V S \mathbb{N} Q\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vsinqi-initialize the array W SA VE, which is used in both \(V S \mathbb{N} Q F\) and \(V S \mathbb{N} Q B\).

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE VSINQIN,W SAVE)}

```
\(\mathbb{N}\) TEGER N
REALW SAVE (*)
SUBROUTINEVSINQI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER*8 N
REALW SAVE (*)

F95 INTERFACE
SU BROUTINEVSINQIN,W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL,D \(\mathbb{M}\) ENSION (:) ::W SAVE

SU BROUTINEVSINQI_64 \(\mathbb{N}, W\) SAVE)
\(\mathbb{N}\) TEGER (8) :: N
REAL,D \(\mathbb{M}\) ENSION (:) ::W SAVE

\section*{C INTERFACE}
\#include <sunperfh>
void vsinqi(intn, float *w save);
void vsinqi_ 64 (long n, float *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. The \(m\) ethod is \(m\) ost efficientw hen \(N\) is a product of sm allprim es.

W SAVE (input)
O \(n\) entry, an array \(w\) ith a dim ension of at least (2 * \(\mathrm{N}+15\) ). The sam ew ork array can be used for both \(V S \mathbb{N} Q F\) and \(V S \mathbb{N} Q B\) as long as \(N\) rem ains unchanged. D ifferent W SAVE arrays are required fordifferentvalues of N . This initialization
does not have to be repeated betw een calls to \(V S \mathbb{N} Q F \operatorname{orV} \operatorname{SIN} Q B\) as long as \(N\) and \(W\) SAVE rem ain unchanged, thus subsequent transform s can be obtained faster than the first.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vsint-com pute the discrete Fourier sine transform of an odd sequence. The VSINT transforms are unnorm alized inverses of them selves, so a call of VSINT follow ed by another callofV S \(\mathbb{N}\) T w illm ultiply the input sequence by 2 * \((\mathbb{N}+1)\). The V SIN \(T\) transform s are norm alized, so a call of VSINT follow ed by a callofV SIN T w ill retum the original sequence.

\section*{SYNOPSIS}

SUBROUTINEVSINTM,N,X,XT,MD \(\mathbb{M} X, W\) SAVE)
\(\mathbb{N}\) TEGER \(\mathrm{M}, \mathrm{N}, \mathrm{MD} \mathbb{I} \mathrm{X}\)
REALXMDIMX,*),XTMDIMX,*),W SAVE (*)
SU BROUTINEVSINT_64M,N,X,XT,MD \(\mathbb{M} X, W\) SAVE)
\(\mathbb{N}\) TEGER*8 M , N , M D IM X
REALX MD \(\operatorname{M}\) X,*), XTMDIMX,*),W SAVE (*)

\section*{F95 INTERFACE}

SU BROUTINE SNT (M ], \(\mathbb{N}], X, X T, \mathbb{M} D \mathbb{M} X], W\) SAVE)
\(\mathbb{N} T E G E R:: M, N, M D \mathbb{I} X\)
REAL,D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL,D \(\mathbb{M}\) ENSION (: : : : : X , XT
SU BROUTINESINT_64 ( \(\mathbb{M}], \mathbb{N}], X, X T, \mathbb{M} D \mathbb{M} X], W\) SAVE)
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{M}, \mathrm{N}, \mathrm{MD} \operatorname{IM} \mathrm{X}\)
REAL,D \(\mathbb{M}\) ENSION (:) ::W SAVE
REAL,D \(\mathbb{M}\) ENSION (:,:) :: X,XT

\section*{C INTERFACE}
\#include <sunperfh>
void vsint(intm, intn, float *x, float *xt, int m dim \(x\), float*W save);
void vsint_ 64 (long m, long n, float *x, float *xt, long m dim x, float *w save);

\section*{ARGUMENTS}

M (input)
The num ber of sequences to be transform ed. M >= 0.

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen \(\mathrm{N}+1\) is a productofsm all prim es. \(\mathrm{N}>=0\).

X (input/output)
O n entry, a realtw o-dim ensionalaray w ith dim ensions of \(M D \mathbb{I} X \times(N+1)\) ) whose row s contain the sequences to be transform ed. On exit, the sine transform of the input.

X T (input/output)
A realtw o-dim ensionalw ork aray w th dim ensions of \(M D \mathbb{M} \times(\mathbb{N}+1)\) ).

MDIM X (input)
Leading dim ension of the arrays X and X T as specified in a dim ension ortype statem ent. M D \(\mathbb{I}\) X >= M.

W SAVE (input)
On entry, an array with dim ension of at least \(\operatorname{int}(2.5 * N+15)\) initialized by VSINTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vsinti-in itialize the array W SAVE, which is used in subroutine VSINT.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE VSINTIN,W SAVE)}
INTEGER N
REALW SAVE (*)
SUBROUT\mathbb{NEVSINTI_64 N,W SAVE)}
INTEGER*8 N
REALW SAVE (*)
F95 INTERFACE
SU BROUT\mathbb{NE VSINTIN,W SAVE)}
\mathbb{NTEGER ::N}
REAL,DIM ENSION (:) ::W SAVE
SUBROUT\mathbb{NE VSNNTI_64 N,W SAVE)}
\mathbb{NTEGER (8) ::N}
REAL,D IM ENSION (:) ::W SAVE
C INTERFACE
\#include <sunperfh>
void vsinti(intn, float *w save);
void vsinti_64 (long n, float *w save);

```

\section*{ARGUMENTS}

N (imput) Length of the sequence to be transform ed. \(\mathrm{N}>=0\).

W SAVE (input)
On entry, an array of dim ension ( \(2 \mathrm{~N}+\mathrm{N} / 2+15\) ) or greater. V SIN T I is called once to initialize W SA V E before calling \(V S \mathbb{N} T\) and need notbe called again between calls to VSINT if N andW SAVE rem ain unchanged. Thus, subsequent transform s of sam e size can be obtained faster than the first since they do not require initialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vzfftb - com pute a periodic sequence from its Fourier coefficients. TheV ZFFT operations are norm alized, so a call of V ZFFTF follow ed by a callofV ZFFTB w ill retum the original sequence.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE VZFFTB M,N,X,XT,MD IM X,ROW COL,W SAVE)}
CHARACTER * 1 ROW COL
DOUBLE COM PLEXXMD\mathbb{M X,*),XTMD\mathbb{M X,*),W SAVE (*)}}\mathbf{*}\mathrm{ (*)}
INTEGER M,N,MD IM X
SUBROUT\mathbb{NEVZFFTB_64M,N,X,XT,MDIMX,ROW COL,W SAVE)}
CHARACTER * 1 ROW COL
D OUBLE COM PLEX X MD IM X,*),XTMDIM X,*),W SAVE (*)
INTEGER*8 M ,N,M D IM X

```
F95 INTERFACE
    SU BROUTINE FFTB ( \(\mathbb{M}], \mathbb{N}], X, X T, \mathbb{M} D \mathbb{I} X], R O W C O L, W\) SAVE)
    CHARACTER (LEN=1) ::ROW COL
    COM PLEX (8),D IM ENSION (:) ::W SAVE
    COM PLEX (8), D IM ENSION (:,:) ::X,XT
    \(\mathbb{N} T E G E R:: M, N, M D \mathbb{I} X\)
    SUBROUTINE FFTB_64( \(\mathbb{M}], \mathbb{N}], X, X T, \mathbb{M D} \mathbb{M} X], R O W C O L, W\) SAVE)
    CHARACTER (LEN=1) ::ROW COL
    COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W SAVE
    COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::X,XT
    \(\mathbb{N} T E G E R(8):: M, N, M D \mathbb{I} X\)

\section*{C INTERFACE}
\#include <sunperfh>
void vzfftb (intm, intn, doublecom plex *x, doublecom plex
\({ }^{*} \mathrm{xt}\), int mdim x , char row col, doublecom plex \({ }^{*}\) w save);
void vzfftb_64 (long m, long n, doublecom plex *x, doublecom plex *xt, long m dim \(x\), char row col, doublecom plex \({ }^{*}\) w save);

\section*{ARGUMENTS}

M (input) The num ber of sequences to be transform ed. \(\mathrm{M}>=\) 0.

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product of sm allprim es. \(\mathrm{N}>=0\).

X (input) On entry, the row s contain the sequences to be transform ed.

XT (input)
A work array.

M D IM X (input)
Leading dim ension of the arrays X and X T as specified in a dim ension ortype statem ent. MD \(\mathbb{M} X>=\) M .

ROW COL (input)
Indicates whether to transform row ( \(\mathrm{R}^{\prime}\) or 'r') orcolum ns (C'or ヒ').

W SAVE (input/output)
O n entry, an aray ofdim ension ( \((\mathbb{K}+15\) ) or greater, where \(K=M\) ifROW COL = ( R 'or 'r'). Otherw ise, \(\mathrm{K}=\mathrm{N} . \mathrm{W}\) SAVE is initialized by V ZFFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vzfflf-com pute the Fourier coefficients of a periodic sequence. The V ZFFT operations are norm alized, so a callof V ZFFTF follow ed by a callofV ZFFTB w ill retum the original sequence.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE VZFFTF M,N,X,XT,MD IM X,ROW COL,W SAVE)}
CHARACTER * 1 ROW COL
DOUBLE COM PLEXXMD\mathbb{M X,*),XTMD M X,*),W SAVE (*)}
INTEGERM,N,MD IM X
SUBROUT\mathbb{NEVZFFTF_64M,N,X,XT,MDIM X,ROW COL,W SAVE)}
CHARACTER * 1 ROW COL
D OUBLE COM PLEX X MD IM X,*),XTMDIM X,*),W SAVE (*)
INTEGER*8 M ,N,M D IM X

```

\section*{F95 INTERFACE}

SU BROUTINE FFTF ( \(\mathbb{M}], \mathbb{N}], X, X T, \mathbb{M} D \mathbb{I} X], R O W C O L, W\) SAVE)
CHARACTER (LEN=1) ::ROW COL
COM PLEX (8),D IM ENSION (:) ::W SAVE
COM PLEX (8), D IM ENSION (:,:) ::X,XT
\(\mathbb{N} T E G E R:: M, N, M D \mathbb{I} X\)
SU BROUTINE FFTF_64 (M ], \(\mathbb{N}], X, X T, \mathbb{M D} \mathbb{M} X], R O W C O L, W\) SAVE)
CHARACTER (LEN=1) ::ROW COL
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W SAVE
COM PLEX (8), D IM ENSION (:,:) ::X,XT
\(\mathbb{N}\) TEGER (8) ::M , N, M D IM X

\section*{C INTERFACE}
\#include < sunperfh>
void vzfflf(intm, intn, doublecom plex *x, doublecom plex
*xt, int mdim \(x\), char row col, doublecom plex \({ }^{*}\) w save);
void vzfflf_ 64 (long m, long n, doublecom plex *x, doublecom plex *xt, long m dim \(x\), char row col, doublecom plex \({ }^{*}\) w save);

\section*{ARGUMENTS}

M (input) The num ber of sequences to be transform ed. M >= 0.

N (input) Length of the sequence to be transform ed. These subroutines are m ostefficientw hen N is a product of sm allprim es. \(\mathrm{N}>=0\).
\(X\) (input) \(O n\) entry, an aray \(X(M, N)\) whose row s contain the sequences to be transform ed.

XT (input)
A work array.

M D \(\mathbb{M} \mathrm{X}\) (input)
Leading dim ension of the arrays X and X T as specified in a dim ension ortype statem ent. M D \(\mathbb{M}\) X >= M.

ROW COL (input)
Indicates whether to transform row ( \(\mathrm{R}^{\prime}\) or 'r') orcolum ns (C 'or ヒ').

W SAVE (input/output)
O n entry, an array ofdim ension ( \((\mathbb{K}+15\) ) or greater, where \(K=M\) ifROW COL = ( R 'or 'r'). O therw ise, \(\mathrm{K}=\mathrm{N} . \mathrm{W}\) SAVE is initialized by V ZFFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
vzffti-initialize the array W SAVE, which is used in both VZFFTF and V ZFFTB.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE VZFFTIN,W SAVE)}

```
DOUBLE COM PLEX W SAVE (*)
\(\mathbb{I N}\) TEGER N
SUBROUTINEVZFFTI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
DOUBLE COM PLEX W SAVE (*)
\(\mathbb{N}\) TEGER*8N

\section*{F95 INTERFACE}

SUBROUTINE VFFTIN, W SAVE)

COM PLEX (8),D \(\mathbb{I}\) ENSION (:) ::W SAVE \(\mathbb{N} T E G E R:: N\)

SU BROUTINE VFFTI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
COM PLEX (8), D IM ENSION (:) ::W SAVE \(\mathbb{I}\) TEGER (8) ::N

\section*{C INTERFACE}
\#include <sunperfh>
void vzffti(intn, doublecom plex *w save);
void vzffli_ 64 (long n, doublecom plex *w save);

\section*{ARGUMENTS}

N (imput) Length of the sequence to be transform ed. \(\mathrm{N}>=0\).

W SAVE (input)
On entry, an array of dimension \((\mathbb{N}+15\) ) or greater. V ZFFTI needs to be called only once to initialize W SAVE before calling VZFFTF and/or V ZFFTB if N and W SAVE rem ain unchanged betw een these calls. Thus, subsequent transform \(s\) or inverse transform s of sam e size can be obtained faster than the first since they do not require initialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

zaxpy - com pute y := alpha * x + y

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZAXPY N,ALPHA,X, \mathbb{NCX,Y, NNCY)}}\mathbf{N},\textrm{N}
DOUBLE COM PLEX ALPHA
DOUBLE COM PLEX X (*),Y (*)
INTEGERN,INCX,\mathbb{NCY}

```

```

DOUBLE COM PLEX ALPHA
DOUBLE COM PLEX X (*),Y (*)
INTEGER*8N,\mathbb{NCX,INCY}
F95 INTERFACE

```

```

    COM PLEX (8) ::A LPHA
    COM PLEX (8),D IM ENSION (:) ::X,Y
    \mathbb{NTEGER ::N,INCX,\mathbb{NCY}}\mathbf{N}=\mathbb{N}
    ```

```

    COM PLEX (8) ::ALPHA
    COM PLEX (8),D IM ENSION (:) ::X,Y
    \mathbb{NTEGER (8)::N,\mathbb{NCX,INCY}}\mathbf{N}={
    ```

\section*{C INTERFACE}
```

\#include <sunperfh>

```
void zaxpy (intn, doublecom plex *alpha, doublecom plex *x, intincx, doublecom plex *y, intincy);
void zaxpy_64 (long n, doublecom plex *ałpha, doublecom plex
*x, long incx, doublecom plex *y, long incy);

\section*{PURPOSE}
zaxpy com pute \(y:=\) alpha * \(\mathrm{x}+\mathrm{y}\) w here alpha is a scalar and \(x\) and \(y\) are \(n\)-vectors.

\section*{ARGUMENTS}
\(N\) (input)
O \(n\) entry, \(N\) specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.

\section*{ALPHA (input)}

On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
aray ofD \(\mathbb{I M}\) ENSION at least ( \(1+(n-1)\) *abs( \(\mathbb{N C X}\) ) ). Before entry, the increm ented amay \(X\) \(m\) ustcontain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X\) m ustnotbe zero. U nchanged on exit.

Y (input/output)
aray ofD \(\mathbb{I M}\) ENSION at least ( \(1+(\mathrm{n}-1\) )*abs( \(\mathbb{N} C Y\) )). On entry, the increm ented amay \(Y \mathrm{~m}\) ust contain the vectory. On exit, \(Y\) is overw rilten by the updated vectory.
\(\mathbb{N} C Y\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y . \mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

zaxpyi-C om pute y := alpha * x + y

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZAXPYINZ,A,X,INDX,Y)}
DOUBLE COM PLEX A
DOUBLE COM PLEX X (*),Y (*)
INTEGER NZ
INTEGER INDX(*)
SUBROUTINE ZAXPYI_64 NZ,A,X,INDX,Y)
DOUBLE COM PLEX A
DOUBLE COM PLEX X (*),Y (*)
INTEGER*8NZ
INTEGER*8 \mathbb{NDX (*)}
F95 IN TERFACE
SUBROUT\mathbb{NE AXPYI(NZ],[A],X,NNDX,Y)}
COM PLEX (8) ::A
COMPLEX (8),D\mathbb{M ENSION (:) ::X,Y}
\mathbb{NTEGER ::NZ}
\mathbb{NTEGER,D IM ENSION (:) :: \mathbb{NDX}}\mathbf{N}=\mp@code{N}
SUBROUTINE AXPYI_64(NZ],[A],X,\mathbb{NDX,Y)}
COM PLEX (8) ::A
COM PLEX (8),D IM ENSION (:) ::X,Y
INTEGER (8) ::N Z
\mathbb{NTEGER (8),D IM ENSION (:) :: \mathbb{NDX}}\mathbf{N}\mathrm{ \ (%)}

```

ZA XPY IC om pute \(\mathrm{y}:=\) alpha * \(\mathrm{x}+\mathrm{y}\) where alpha is a scalar, x is a sparse vector, and \(y\) is a vector in fullstorage form
```

do i=1,n
y (indx (i)) = alpha * x (i) + y (indx (i))
enddo

```

\section*{ARGUMENTS}

NZ (input) - \(\mathbb{N}\) TEGER
\(N\) um ber of elem ents in the com pressed form .
U nchanged on exit.

A (input)
On entry, A (LPH A ) specifies the scaling value.
Unchanged on exit. A is defaulted to ( \(1.0 \mathrm{D} 0,0.0 \mathrm{D} 0\) )
forF95 \(\mathbb{I N}\) TERFACE.
X (input)
V ector containing the values of the com pressed form .
U nchanged on exit.
\(\mathbb{N} D \mathrm{X}\) (input) - \(\mathbb{N}\) TEGER
\(V\) ector containing the indices of the com pressed form. It is assum ed that the elem ents in \(\mathbb{N} D \mathrm{X}\) are distinctand greater than zero. U nchanged on exit.

Y (output)
V ectoron inputw hich contains the vectory in full storage form. On exit, only the elem ents
comesponding to the indices in \(\mathbb{N} D \mathrm{X}\) have been
m odified.

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

zbcom m -block coordinatem atrix m atrix m ultiply

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZBCOMM (TRANSA,MB,N,KB,ALPHA,DESCRA,}

* VAL,B\mathbb{NDX,BJNDX,BNNZ,LB,}
* B,LDB,BETA,C,LDC,W ORK,LWORK)
INTEGER TRANSA,MB,N,KB,DESCRA (5),BNNZ,LB,
* LDB,LDC,LW ORK
NNTEGER BINDX (BNNZ),BJNDX (BNNZ)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (LB*LB*BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINE ZBCOMM_64(TRANSA,MB,N,KB,ALPHA,DESCRA,
* VAL,BINDX,BJNDX,BNNZ,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER*8 TRANSA,MB,N,KB,DESCRA (5),BNNZ,LB,
* LDB,LDC,LW ORK
INTEGER*8 B\mathbb{NDX (BNNZ),BJNDX (BNNZ)}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (LB*LB *BNNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}

SUBROUTINE BCOMM (TRANSA, MB,N, \(K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D, B J N D\), * BNNZ,LB,B,[LDB],BETA,C,[LDC], [W ORK], [LW ORK])
\(\mathbb{N} T E G E R\) TRANSA, MB,N,KB,BNNZ,LB
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}\) (:) :: DESCRA,BINDX,BJNX
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::VAL
DOUBLE COM PLEX,D \(\mathbb{I}\) ENSION (:, :) :: B, C
SUBROUTINE BCOMM_64 (TRANSA, MB,N,KB,ALPHA,DESCRA, VAL,BINDX,BJND,
* BNNZ,LB,B,[LDB],BETA,C,[LDC],[WORK],[LWORK])
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,N,KB,BNNZ,LB
\(\mathbb{N} T E G E R * 8, D \mathbb{I M} E N S \mathbb{O} N(:):: D E S C R A, B \mathbb{N} D X, B J N D\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D \(\mathbb{I}\) ENSION (:) ::VAL
DOUBLE COM PLEX,D \(\mathbb{I}\) ENSION (:, :) :: B, C

\section*{DESCRIPTION}
\[
C \text { <-alpha op (A) B + beta C }
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrioes, \(A\) is a m atrix represented in block coordinate form at and op(A) is one of
```

op(A)=A or op(A )= A' or op(A )= conjg(A').

```
( 'indicates m atrix transpose)

\section*{ARGUMENTS}
\begin{tabular}{|c|c|}
\hline \multirow[t]{4}{*}{TRANSA} & A Indicates how to operate w ith the sparse \(m\) atrix \\
\hline & 0 : operate w th m atrix \\
\hline & 1 : operate w ith transpose m atrix \\
\hline & 2 : operate \(w\) th the conjugate transpose ofm atrix. 2 is equivalent to 1 if the \(m\) atrix is real. \\
\hline M B & N um berofblock row s in m atrix A \\
\hline N & \(N\) um berof colum ns in m atrix C \\
\hline K B & \(N\) um ber ofblock colum ns in m atrix A \\
\hline A LPH A & Scalarparam eter \\
\hline \multirow[t]{13}{*}{DESCRA} & 0 D escriptor argum ent. Five elem ent integer array \\
\hline & DESCRA (1) m atrix structure \\
\hline & 0 : general \\
\hline & 1 : symmetric ( \(A=A\) ) \\
\hline & \(2: \mathrm{Herm}\) Itian ( \(\mathrm{A}=\mathrm{CONJ}\) ( A ) ) \\
\hline & 3 :Triangular \\
\hline & 4 : Skew (A nti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) ) \\
\hline & 5 :D iagonal \\
\hline & 6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJG}(\mathrm{A})\) ) \\
\hline & D ESCRA (2) upper/low er triangular indicator \\
\hline & 1 : low er \\
\hline & 2 :upper \\
\hline & DESCRA (3) m ain diagonal type \\
\hline
\end{tabular}

0 :non-unit
1 : unit
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED )
0 :C C++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL ( scalar array of length LB *LB *BNN Z consisting of the non-zero block entries of \(A\), in any order. Each block is stored in standard colum n-m ajor form .
\(B \mathbb{N} D X(\) integer array of length \(B N N Z\) consisting of the block row indiaes of the block entries of .

B JND X 0 integer anray of length BNNZ consisting of the block colum \(n\) indiges of the block entries of \(A\).

BNNZ num berofblock entries

LB dim ension of dense blocks com posing A.
B 0 rectangular array with first dim ension LD B .
LD B leading dim ension ofB
BETA Scalarparam eter

C 0 rectangular array w ith firstdim ension LD C .

LD C leading dim ension ofC
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the currentversion.

LW ORK length ofW ORK array.LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov \(\mathrm{m}_{\mathrm{c}}\) csd/Staffk Rem ington/tspoblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

zbdimm -block diagonal form atm atrix-m atrix m ultiply

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZBDIMM(TRANSA,M B,N,KB,ALPHA,DESCRA,}

* VAL,BLDA,\mathbb{BDIAG,NBDIAG,LB,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,MB,N,KB,DESCRA (5),BLDA,NBDIAG,LB,}
* LDB,LDC,LW ORK
INTEGER \mathbb{BDIAG NBDIAG)}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (LB*LB *BLDA*NBD IAG),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINE ZBD IMM_64(TRANSA,M B,N,KB,ALPHA,DESCRA,
* VAL,BLDA,\mathbb{BDIAG,NBDIAG,LB,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
NNTEGER*8 TRANSA,MB,N,KB,DESCRA (5),BLDA,NBDIAG,LB,
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 \mathbb{BD IAG NBDIAG)}}\mathbf{N}=()
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (LB*LB*BLDA*NBD IAG),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}

SUBROUTINEBD \(\mathbb{I} M\) (TRANSA, MB, \(\mathbb{N}], K B, A L P H A, D E S C R A, V A L, B L D A\), * \(\mathbb{B D} \mathbb{I} G, N B D \mathbb{I} G, L B, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N} T E G E R\) TRANSA, MB,KB,BLDA,NBD \(\mathbb{I A} G, L B\)
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \quad\) DESCRA, \(\mathbb{B D} \mathbb{I A}\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::VAL
DOUBLE COM PLEX,D IM ENSION (:, :) :: B, C
SUBROUTINEBD \(\mathbb{M} M \_64\) (TRANSA, MB, \(\left.\mathbb{N}\right], K B, A L P H A, D E S C R A, V A L, B L D A\),
* \(\mathbb{B D} \mathbb{I} G, N B D \mathbb{I} G, L B, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,KB,BLDA,NBD \(\mathbb{I A} G, L B\)
\(\mathbb{N} T E G E R * 8, D \mathbb{I} \operatorname{ENSION}(:):: \quad D E S C R A, \mathbb{B D} \mathbb{I} G\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D \(\mathbb{I}\) ENSION (:) ::VAL
DOUBLE COM PLEX,D \(\mathbb{I}\) ENSION (:, :) :: B, C

\section*{DESCRIPTION}
\[
C \text { <-alpha op (A) B + beta C }
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrioes, \(A\) is a m atrix represented in block diagonal form at and op(A) is one of \(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).
('indicatesm atrix transpose)

\section*{ARGUMENTS}
\begin{tabular}{|c|c|}
\hline \multirow[t]{4}{*}{TRANSA} & A Indicates how to operate w th the sparse \(m\) atrix \\
\hline & 0 : operate w th m atrix \\
\hline & 1 : operate w ith transpose m atrix \\
\hline & 2 : operate \(w\) th the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real. \\
\hline M B & N um berofblock row s in m atrix A \\
\hline N & \(N\) um berof colum \(n s\) in \(m\) atrix \(C\) \\
\hline K B & \(N\) um ber ofblock colum ns in m atrix A \\
\hline A LPH A & Scalar param eter \\
\hline \multirow[t]{13}{*}{DESCRA} & () D escriptor argum ent. Five elem ent integer amay \\
\hline & DESCRA (1) m atrix structure \\
\hline & 0 : general \\
\hline & 1 : symmetric ( \(\mathrm{A}=\mathrm{A}\) ) \\
\hline & 2: Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) ) \\
\hline & 3 :Triangular \\
\hline & 4 : Skew (A nti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) ) \\
\hline & 5 :D iagonal \\
\hline & 6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJG}(\mathrm{A})\) ) \\
\hline & D ESCRA (2) upper/low er triangular indicator \\
\hline & 1 : low er \\
\hline & 2 : upper \\
\hline & D ESCRA (3) m ain diagonaltype \\
\hline
\end{tabular}

0 : non-unit
1 : unit
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL \(0 \quad\) tw o-dim ensionalLB *LB *BLD A -by-N BD IA G scalar anay consisting of the NBD IA G nonzero block diagonal in any order. Each dense block is stored in standard colum n.m ajor form .

BLD A leading block dim ension ofV A L ( ).
IBD IA G 0 integer amay of length N BD IA G consisting of the corresponding diagonaloffsets of the non-zero block diagonals ofA in VA L. Low ertriangular block diagonals have negative offsets, the \(m\) ain block diagonal has offset 0, and uppertriangular block diagonals have positive offset.

NBD IA G the num berofnon-zero block diagonals in A.
LB dim ension of dense blocks com posing A.
B 0 rectangular array w ith first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w th first dim ension LD C.

LD C leading dim ension ofC
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib org/utk/papers/sparse .ps

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

zbdism - block diagonal form at triangular solve

```

\section*{SYNOPSIS}
```

SUBROUTINE ZBD ISM (TRANSA, M B,N,UNITD,DV,ALPHA,DESCRA,

* VAL,BLDA, $\mathbb{B D} \mathbb{A} G, N B D \mathbb{I A}, \mathrm{LB}$,
* B,LDB,BETA, C,LDC,WORK,LWORK)
$\mathbb{I N} T E G E R$ TRANSA,MB,N,UNITD,DESCRA (5), BLDA,NBD $\mathbb{I A} G, L B$,
* LDB,LDC,LW ORK
$\mathbb{N} T E G E R \quad \mathbb{B D} \mathbb{I} G \mathbb{N} B \mathbb{I} G)$
D OUBLE COM PLEX ALPHA,BETA
D OUBLE COM PLEX DV M B *LB *LB) ,VAL (LB*LB*BLDA,NBD IAG), B (LDB,*), C (LD C , *),
* WORK (LW ORK)
SUBROUTINE ZBD ISM _64(TRANSA, M B,N,UNITD,DV,ALPHA,DESCRA,
* VAL,BLDA, $\mathbb{B D} \mathbb{I} G, N B D \mathbb{I A}, L B$,
* B,LDB,BETA, C,LDC,WORK,LW ORK)
$\mathbb{N} T E G E R * 8$ TRANSA,MB,N,UNITD,DESCRA (5), BLDA,NBD IA G,LB,
* LDB,LDC,LWORK
$\mathbb{N} T E G E R * 8 \mathbb{B D} \mathbb{I} G \mathbb{N} B \mathbb{A} G)$
D OUBLE COM PLEX ALPHA,BETA

```

```

* WORK (LW ORK)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE BD ISM (TRANSA,M B,N,UNITD,DV,ALPHA,DESCRA,VAL,BLDA,}

* \mathbb{BDIAG,NBD IAG,LB,B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])}
INTEGER TRANSA,MB,N,UNITD,BLDA,NBDIAG,LB
\mathbb{NTEGER,D IM ENSION (:) :: DESCRA, \mathbb{BD IAG}}\mathbf{|}=\mp@code{M}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D IM ENSION (:) ::VAL,DV
DOUBLE COM PLEX,D IM ENSION (:,:):: B,C

```

SUBROUTINE BD ISM _64 (TRANSA , M B , N , UNITD , DV, ALPHA, DESCRA, VAL, BLDA,
* \(\mathbb{B D} \operatorname{IAG}, N B D I A G, L B, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,N,UNITD, BLDA,NBD IAG, LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O}(:):: \quad \mathrm{DESCRA}, \mathbb{B D} \mathbb{I} G\)
DOUBLE COMPLEX ALPHA,BETA
DOUBLE COM PLEX,D \(\mathbb{I}\) ENSION (:) ::VAL, DV
DOUBLE COM PLEX ,D \(\mathbb{M}\) ENSION (:, :) :: B , C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { op (A)B+BETA C } \\
& C<-A L P H A \text { OP (A)D B + BETA } C
\end{aligned}
\]
where ALPHA and BETA are scalar, \(C\) and \(B\) are m by \(n\) dense \(m\) atrices, \(D\) is a block diagonalm atrix, \(A\) is a unit, ornon-unit, upperor low ertriangularm atrix represented in block diagonal form at and op (A) is one of \(\operatorname{op}(A)=\operatorname{inv}(A)\) or op \((A)=\operatorname{inv}(A) \operatorname{or} \operatorname{op}(A)=\operatorname{inv}\left(\operatorname{conjg}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRAN SA Indicates how to operate \(w\) th the sparse \(m\) atrix 0 : operate \(w\) ith \(m\) atrix 1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(s\) in \(m\) atrix \(A\)
\(N \quad N\) um berof colum \(n s\) in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity matrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)

DV () A rray of length M B *LB *LB containing the elem ents of the diagonalblocks of them atrix \(D\). The size of each square block is LB \(-b y-4 B\) and each block is stored in standard colum n-m ajor form .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay DESCRA (1) m atrix structure
            0 : general
            1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
            2 : Herm itian ( \(A=\operatorname{CONJG}(A))\)
            3 :Triangular
            4 : Skew (A nti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) )
            5 :D iagonal
            6 : Skew Herm itian ( \(A=-C O N J(A)\) )
                            N ote: For the routine, D ESCRA (1)=3 is only supported.
                            D ESCRA (2) upper/low er triangular indicator
            1 : low er
            2 :upper
DESCRA (3) m ain diagonaltype
            0 : non-identity blocks on the \(m\) ain diagonal
            1 : identity diagonalblocks
            2 : diagonalblocks are dense \(m\) atrices
            DESCRA (4) A may base \(\mathbb{N O T} \mathbb{I}\) PLEM ENTED)
            0 :C C ++ com patible
            1 :Fortran com patible
                    DESCRA (5) repeated indices? \(\mathbb{N O T} \mathbb{I}\) PLEM ENTED)
                    0 : unknown
                            1 : no repeated indices
VAL () Two-dim ensionalLB *LB *B LD A -by-N BD IA G scalaranay
consisting of the N BD IA G non-zero block diagonal.
Each dense block is stored in standard colum n-m ajor form .
B LD A Leading block dim ension ofV A L (). Should be greater
    than orequal to M B .
IBD IA G 0 integer amay of length NBD IA G consisting of the corresponding diagonal offsets of the non-zero block diagonals ofA in VA L. Low ertriangularblock diagonals have negative offsets, them ain block diagonalhas offset 0 , and upper triangularblock diagonals have positive offset. Elem ents of IBD IA G M UST be sorted in increasing order.
NBD IA G The num berofnon-zero block diagonals in A.
LB D im ension of dense blocks com posing A.
B 0 Rectangular aray with firstdim ension LD B .
LD B Leading dim ension of B .
BETA Scalarparam eter.
C 0 Rectangular array w ith first dim ension LD C .
LD C Leading dim ension of C .

W ORK 0 scratch array of length LW ORK. On exit, ifLW ORK=-1,W ORK (1) retums the optim um size ofLW ORK.

LW ORK length ofW ORK array. LW ORK should be at least M B *LB.

Forgood penform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=M B * L B * N \_C P U S\) where \(N\) _CPU \(S\) is the \(m\) axim um num berof processors available to the program .

IfLW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no enrorm essage related to LW ORK is issued by X ERBLA.

\section*{SEE ALSO}

N IST FORTRA N Sparse B las U sers G uide available at: http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such tests \(m\) ustbe perform ed before calling this routine.
2. If \(D E S C R A\) (3)=0, the low erorupper triangularpart of each diagonalblock is used by the routine depending on DESCRA (2).
3. If \(D E S C R A\) (3)=1, the unit diagonalblocksm ightorm ight notbe referenced in the BD I representation of a sparse \(m\) atrix. They are notused anyw ay .
4. If \(D E S C R A(3)=2\), diagonalblocks are considered as dense \(m\) atrices and the LU factorization \(w\) ith partialpivoting is used by the routine. WORK (1)=0 on retum if the factorization for alldiagonalblocks has been com pleted successfully, otherw ise WORK (1) = -iw here is the block
num ber forw hich the LU factorization could notbe com puted.
5. The routine can be applied for solving triangular system \(s\) w hen the upper or low er triangle of the general.sparse m atrix \(A\) is used. H ow erver \(D E S C R A\) (1) m ustbe equal to 3 in this case.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zbdsqr-com pute the singularvalue decom position (SVD ) of a realN -by -N (upper or low er) bidiagonalm atrix B.

\section*{SYNOPSIS}
```

SU BROUTINE ZBD SQR (UPLO,N,NCVT,NRU,NCC,D,E,VT,LDVT,U,LDU,C,
LDC,WORK,\mathbb{NFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX VT (LDVT,*),U (LDU,*),C (LDC ,*)
INTEGERN,NCVT,NRU,NCC,LDVT,LDU,LDC,INFO
DOUBLE PRECISION D (*),E (*),W ORK (*)
SU BROUTINE ZBDSQR_64 (UPLO,N,NCVT,NRU,NCC,D ,E,VT,LDVT,U,LDU,
C,LDC,WORK,\mathbb{NFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX VT (LDVT,*),U (LDU ,*),C (LDC ,*)
INTEGER*8N,NCVT,NRU,NCC,LDVT,LDU,LDC,INFO
DOUBLE PRECISION D (*),E (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE BDSQR (UPLO, \(\mathbb{N}], \mathbb{N C V T ] , ~} \mathbb{N} R U], \mathbb{N C C}], D, E, V T,[L D V T]\), U, [LD U ], C, [LD C ], [W ORK ], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D IM ENSION (:,:) ::VT,U,C
\(\mathbb{N} T E G E R:: N, N C V T, N R U, N C C, L D V T, L D U, L D C, \mathbb{N F O}\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK
SU BROUTINE BDSQR_64 (UPLO, \(\mathbb{N}], \mathbb{N} C V T], \mathbb{N R U}], \mathbb{N C C}], D, E, V T,[L D V T]\), \(\mathrm{U},[\mathrm{LD} \mathrm{U}], \mathrm{C},[\mathrm{LD} \mathrm{C}],[\mathrm{W}\) ORK ], [ \(\mathbb{N} F \mathrm{O}])\)

CHARACTER (LEN=1) :: UPLO
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : : VT, U , C
\(\mathbb{N}\) TEGER (8) :: N , NCVT, NRU, NCC, LDVT, LDU, LD C , \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zbdsqr(charuplo, intn, intncvt, int nru, int ncc, double *d, double *e, doublecom plex *vt, intldvt, doublecom plex *u, int ldu, doublecom plex \({ }^{*} \mathrm{C}\), int ldc, int*info);
void zbdsqr_64 (charuplo, long n, long ncvt, long nnu, long ncc, double *d, double *e, doublecom plex *vt, long ldvt, doublecom plex *u, long ldu, doublecom plex \({ }^{*}\) c, long ldc, long *info);

\section*{PURPOSE}
zbdsqr com putes the singularvalue decom position (SV D ) of a realN Hoy-N (upper or low er) bidiagonalm atrix B: B = Q * S * \(P^{\prime}\left(P^{\prime}\right.\) denotes the transpose of \(\left.P\right)\), w here \(S\) is a diagonal \(m\) atrix \(w\) ith non-negative diagonal elem ents the singular values of \(B\) ), and \(Q\) and \(P\) are orthogonalm atrices.

The routine com putes \(S\), and optionally com putes \(U * Q, P^{\prime} \star\) \(\mathrm{V} T\), or \(Q^{\prime *} \mathrm{C}\), forgiven com plex inputm atrices \(\mathrm{U}, \mathrm{V}\) T , and C.

See "C om puting Sm allSingularV ahues ofB idiagonalM atrioes W ith G uaranteed H igh R elative A ccuracy," by J. Dem m eland W . K ahan, LAPACK W orking N ote \#3 (orSIAM J. Sci. Statist. C om put.vol.11, no.5, pp. 873-912, Sept1990) and
"A ccurate singular values and differential qd algorithm \(s\),"
by B. Parlett and V.Femando, TechnicalReportCPAM -554, \(M\) athem atics \(D\) epartm ent, U niversity of \(C\) alifomia at Berkeley, July 1992 for a detailed description of the algorithm .

\section*{ARGUMENTS}
```

UPLO (input)
$=U$ ': B is upperbidiagonal;
$=1 \mathrm{~L}$ ': B is low erbidiagonal.

```

N (input) The order of the m atrix \(\mathrm{B} . \mathrm{N}>=0\).

NCVT (input)
The num berof colum ns of the m atrix V T.NCV T >=0.

NRU (input)
The num berof row s of the \(m\) atrix \(U . N R U>=0\).

NCC (input)
The num ber of colum ns of the m atrix C . N CC \(>=0\).

D (input/output)
O n entry, the n diagonalelem ents of the bidiagonal matrix B. On exit, if \(\mathbb{N F O}=0\), the singular values ofB in decreasing order.

E (input/output)
On entry, the elem ents of \(E\) contain the offdiagonalelem ents of of the bidiagonalm atrix w hose SV D is desired. On nom alexit ( \(\mathbb{N F O}=0\) ), E is destroyed. If the algorithm does not converge ( \(\mathbb{N}\) FO \(>0\) ), D and E w ill contain the diagonal and superdiagonal elem ents of a bidiagonalm atrix orthogonally equivalent to the one given as input. E (N) is used forw orkspace.

VT (input/output)
On entry, an N -by-N CV T m atrix VT. On exit, VT is overw rilten by \(\mathrm{P}^{\prime *} \mathrm{VT} . \mathrm{VT}\) is notreferenced if \(\mathrm{NCVT}=0\).

LDVT (input)
The leading dim ension of the array VT. LDV T >= \(\max (1, N)\) if NCV T > 0;LDVT >= 1 ifNCVT = 0 .

U (input/output)
On entry, an NRU -by N m atrix U. On exit, U is overw rilten by \(U * Q . U\) is not referenced ifNRU \(=0\).

LD U (input)
The leading dim ension of the array \(U\). LD U >= max (1,NRU).

C (input/output)
On entry, an N-by -NCC matrix C. On exit, C is overw ritten by \(Q\) '* C . C is not referenced if N C C \(=0\).

LD C (input)
The leading dim ension of the aray C. LD C >= \(m a x(1, N)\) if \(N C C>0 ; L D C>=1\) ifNCC \(=0\).

W ORK (w orkspace)
dim ension \((4 * N)\)
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) If \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue
>0: the algorithm did notconverge; D and E contain the elem ents of a bidiagonalm atrix which is orthogonally sim ilar to the input matrix B; if \(\mathbb{N F O}=\) i, ielem ents ofE have not converged to zero.

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
zbelm m -block Ellpack form atm atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZBELMM (TRANSA,M B,N,KB,A LPHA,DESCRA,}

* VAL,BINDX,BLDA,MAXBNZ,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER TRANSA,MB,N,KB,DESCRA (5),BLDA,MAXBNZ,LB,
* LDB,LDC,LWORK
INTEGER BINDX (BLDA,MAXBNZ)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (LB*LB*BLDA*M AXBNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUTINE ZBELMM_64(TRANSA,M B,N,KB,ALPHA,DESCRA,
* VAL,BINDX,BLDA,MAXBNZ,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER*8 TRANSA,MB,N,KB,DESCRA (5),BLDA,MAXBNZ,LB,
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 B\mathbb{NDX (BLDA,MAXBNZ)}}\mathbf{M}\mathrm{ (BAM}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (LB*LB *BLDA*M AXBNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE BELMM (TRANSA,MB,N ],KB,ALPHA,DESCRA,VAL,B INDX,}

* BLDA,MAXBNZ,LB,B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])
\mathbb{NTEGER TRANSA,MB,KB,BLDA,MAXBNZ,LB}

```

```

DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D IM ENSION (:) ::VAL
DOUBLE COM PLEX,D IM ENSION (:,:) :: B,C

```
SUBROUTINEBELMM_64(TRANSA, MB, \(\mathbb{N}], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\),

BLDA,MAXBNZ,LB,B,[LDB],BETA,C,[LDC], \(\mathbb{W}\) ORK], [LW ORK])
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,KB,BLDA,MAXBNZ,LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: \quad \mathrm{DESCRA}, \mathrm{B} \mathbb{N} D \mathrm{X}\)
DOUBLE COM PLEX ALPHA,BETA
D OUBLE COM PLEX ,D \(\mathbb{M}\) ENSION (:) ::VAL
DOUBLE COM PLEX ,D \(\mathbb{M}\) ENSION (:, :) :: B, C

\section*{DESCRIPTION}
C <-ałha op (A ) B + beta C
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, \(A\) is a m atrix represented in block Elloack form at and \(o p(A)\) is one of
```

op(A) =A or op(A )=A' or op (A ) = conjg(A').

```
( 'indicates m atrix transpose)

\section*{ARGUMENTS}
\begin{tabular}{|c|c|}
\hline \multirow[t]{4}{*}{TRANSA} & A Indicates how to operate w ith the sparse m atrix \\
\hline & 0 : operate w ith m atrix \\
\hline & 1 : operate w ith transpose \(m\) atrix \\
\hline & 2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real. \\
\hline M B & N um berofblock row s in m atrix A \\
\hline N & \(N\) um berof colum ns in m atrix C \\
\hline KB & \(N\) um ber ofblock 00 lum ns in m atrix A \\
\hline A LPH A & Scalar param eter \\
\hline \multirow[t]{13}{*}{DESCRA} & () D escriptor argum ent. Five elem ent integer amay \\
\hline & DESCRA (1) m atrix structure \\
\hline & 0 : general \\
\hline & 1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) ) \\
\hline & \(2:\) Herm itian ( \(\mathrm{A}=\mathrm{CONJ}\) ( A ) ) \\
\hline & 3 :Triangular \\
\hline & 4 : Skew (Anti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) ) \\
\hline & 5 :D iagonal \\
\hline & 6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJ}\) ( A ) ) \\
\hline & D ESCRA (2) upper/low er triangular indicator \\
\hline & 1 : low er \\
\hline & 2 :upper \\
\hline & D ESCRA (3) m ain diagonal type \\
\hline
\end{tabular}

0 : non-unit
1 : unit
DESCRA (4) A ray base NOT IM PLEM ENTED )
0 :C C++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL 0 scalar array of length LB *LB *BLDA *M AXBN Z containing \(m\) atrix entries, stored 00 lum \(n-m\) ajorw thin each dense block.
\(B \mathbb{N} D X_{0} \quad\) tw o-dim ensional integerBLD A -by \(-M A X B N Z\) aray such B IND X (i,:) consists of the block colum \(n\) indices of the nonzero blocks in block row i, padded by the integer value i if the num ber of nonzero blocks is less than MAXBNZ.

BLDA leading dim ension of \(\operatorname{INDX(:,:).}\)

M A X BN Z max num berof nonzerosblocks per row .
LB row and colum \(n\) dim ension of the dense blocks com posing VAL.

B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension ofB
BETA Scalarparam eter

C 0 rectangular array w th first dim ension LD C .
LD C leading dim ension of \(C\)
W ORK () scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK aray. LW ORK is not referenced in the cumentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U sers G uide available at:
http://m ath nist.gov/n csd/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)

Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

zbelsm -block Ellpack form at triangular solve

```

\section*{SYNOPSIS}
```

SUBROUTINE ZBELSM (TRANSA, M B,N,UNITD,DV,ALPHA,DESCRA,

* VAL,BINDX,BLDA,MAXBNZ,LB,
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
$\mathbb{I N} T E G E R$ TRANSA,MB,N,UNITD,DESCRA (5),BLDA,MAXBNZ,LB,
* LDB,LDC,LW ORK
$\mathbb{N} T E G E R \quad B \mathbb{N} D X(B L D A, M A X B N Z)$
DOUBLE COM PLEX ALPHA,BETA
D OUBLE COM PLEX DV MB*LB*LB),VAL (LB*LB*BLDA*MAXBNZ),B(LDB,*),C(LDC,*),
* WORK (LW ORK)
SUBROUTINE ZBELSM_64(TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,
* VAL,BINDX,BLDA,MAXBNZ,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
$\mathbb{N} T E G E R * 8$ TRANSA, M B,N,UNITD,DESCRA (5),BLDA,MAXBNZ,LB,
* LDB,LDC,LWORK
$\mathbb{N} T E G E R * 8$ B $\mathbb{N} D X(B L D A, M A X B N Z)$
DOUBLE COM PLEX ALPHA,BETA
D OUBLE COM PLEX DV M B *LB *LB) ,VAL (LB*LB*BLDA*MAXBNZ),B(LDB,*),C(LDC,*),
* $\quad$ WORK (LW ORK)

```

\section*{F95 INTERFACE}

SUBROUTINE BELSM (TRANSA, MB, \(\mathbb{N}], U N T T D, D V, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\),
* BLDA,MAXBNZ,LB,B,[LDB],BETA,C,[LDC], [W ORK], [LW ORK])
\(\mathbb{N} T E G E R\) TRANSA, MB,UNITD, BLDA,MAXBNZ,LB
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \quad D E S C R A, B \mathbb{N} D X\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D \(\mathbb{I}\) ENSION (:) ::VAL,DV
DOUBLE COM PLEX,D \(\mathbb{I M} \operatorname{ENSION}(:,:\) : : B, C

SUBROUT \(\mathbb{N} E \operatorname{BELSM}\) _64 (TRANSA, MB, \(\mathbb{N}], U N T D, D V, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\),
* \(B L D A, M A X B N Z, L B, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,UNITD, BLDA, MAXBNZ,LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: D E S C R A, B \mathbb{N} D X\)
DOUBLE COMPLEX ALPHA,BETA
DOUBLE COM PLEX,D \(\mathbb{I}\) ENSION (:) ::VAL, DV
DOUBLE COM PLEX ,D \(\mathbb{M} E N S I O N(:,:: \quad B, C\)

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { op (A)B+BETA C } \\
& C<-A L P H A \text { OP }(A) D B+B E T A C
\end{aligned}
\]
where ALPHA and BETA are scalar, \(C\) and \(B\) are m by \(n\) dense matrices, \(D\) is ablock diagonalm atrix, \(A\) is a unit, ornon-unit, upperor low ertriangularm atrix represented in block Elhoack form at and op (A ) is one of \(\operatorname{op}(A)=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}\left(o n \operatorname{jg}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRAN SA Indicates how to operate \(w\) th the sparse \(m\) atrix 0 : operate \(w\) ith \(m\) atrix 1 : operate \(w\) th transpose \(m\) atrix 2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalentto 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(s\) in \(m\) atrix \(A\)

N \(\quad\) Num berof colum ns in matrix C

UN ITD Type of scaling:
1 : Identity \(m\) atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)

DV () A may of the length M B *LB *LB consisting of the block entries ofblock diagonalm atrix D w here each block is stored in standard colum n-m ajor form .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general

1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2: Herm itian ( \(A=\operatorname{CONJ}(A)\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(\mathrm{A}=-\mathrm{A}\) )
5 :D iagonal
6 : Skew Herm titian ( \(A=-\operatorname{CON}\) J ( \(A\) ) )
N ote: For the routine, D ESCRA (1)=3 is only supported.
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 : upper
DESCRA (3) m ain diagonal type
0 : non-identity blocks on the \(m\) ain diagonal
1 : identity diagonalblocks
2 : diagonalblocks are dense \(m\) atrices
DESCRA (4) A ray base \(\mathbb{N O T} \mathbb{M}\) PLEM ENTED )
0 : C C C+ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? \(\mathbb{N} O T \mathbb{M}\) PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () scalar array of length LB *LB *BLD A *M A X BN Z containing \(m\) atrix entries, stored colum \(n-m\) ajorw thin each dense block.

B \(\mathbb{N}\) D X () tw o-dim ensionalintegerB LD A boy-M A X BN Z array such B IND X ( \(i\), : ) consists of the block colum \(n\) indices of the nonzero blocks in block row i, padded by the integer value iif the num ber ofnonzero blocks is less than M A X BN Z. The block colum \(n\) indioesM U ST be sorted in increasing order foreach block row.

BLDA leading dim ension ofB INDX (:,:).

M AXBNZ max num berofnonzerosblocks per row .
LB row and colum \(n\) dim ension of the dense blocks com posing A.

B 0 rectangular array w ith first dim ension LD B .

LD B leading dim ension of \(B\)

BETA Scalarparam eter

C 0 rectangular aray w ith first dim ension LD C .

LD C leading dim ension of C

W ORK () scratch array of length LW ORK.
On exit, ifLW ORK \(=-1, W\) ORK (1) retums the minim um
size ofLW ORK.

LW ORK length ofW ORK anay.LW ORK should be at least M B *LB.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=\mathrm{M} \mathrm{B} * \mathrm{LB} * \mathrm{~N}\) _CPU \(S\) where \(\mathrm{N} \_\)CPUS is the \(m\) axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W O RK array, and no errorm essage related to LW ORK is issued by XERBLA .

\section*{SEE ALSO}

\section*{N IST FO RTRA N Sparse B las U ser's G uide available at:} http:/m ath nist.gov/m cso/Staff/K Rem ington/Espblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib org/utk/papers/sparse _ps

\section*{NOTES /BUGS}
1.N o test for singularity ornear-singularity is included in this routine. Such tests m ust.be perform ed before calling this routine.
2. If \(D E S C R A(3)=0\), the low er or upper triangularpart of each diagonalblock is used by the routine depending on DESCRA (2) .
3. If \(D E S C R A(3)=1\), the unitdiagonalblocksm ightorm ight notbe referenced in the B EL representation of a sparse \(m\) atrix. They are notused anyw ay .
4. If \(D E S C R A(3)=2\), diagonalblocks are considered as dense m atrices and the LU factorization w ith partialpivoting is used by the routine. WORK (1)=0 on retum if the factorization for alldiagonalblocks has been com pleted successfully, otherw ise WORK (1) = -iw here i is the block num ber forw hich the LU factorization could notbe com puted.
5.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse
m atrix \(A\) is used. H ow erver DESCRA (1) m ust.be equalto 3 in this case.

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

zbsom m -block sparse colum n m atrix-m atrix m ultiply

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZBSCMM (TRANSA,M B,N,KB,ALPHA,DESCRA,}

* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,MB,N,KB,DESCRA (5),LB,}
* LDB,LDC,LW ORK
\mathbb{NTEGER BINDX (BNNZ),BPNTRB (KB),BPNTRE (KB)}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (LB *LB*BNNZ),B (LDB,*),C (LDC ,*),W ORK (LW ORK)

```
SUBROUTINE ZBSCMM_64(TRANSA, MB,N,KB,ALPHA,DESCRA,
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,N,KB,DESCRA (5),LB,
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R * 8\) B \(\mathbb{N} D X(B N N Z)\), BPNTRB (KB), BPNTRE (KB)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (LB*LB*BNNZ), B (LDB,*), C (LDC,*),WORK (LW ORK)
where: \(\operatorname{BNNZ}=\operatorname{BPNTRE}(\mathbb{K} B)\) BPNTRB (1)

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE BSCMM (TRANSA,M B, N ],KB,ALPHA,DESCRA,VAL,B INDX,}

* BPNTRB,BPNTRE,LB,B,[LD B ],BETA ,C,[LDC], [W ORK ], [LW ORK ])
INTEGER TRANSA,MB,KB,LB
\mathbb{NTEGER,D IM ENSION (:) :: DESCRA,B NDDX,BPNTRB,BPNTRE}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D IM ENSION (:) ::VAL
DOUBLE COM PLEX,D IM ENSION (:,:) :: B,C

```

SUBROUT \(\mathbb{N} E \operatorname{BSCM} M \_64(T R A N S A, M B, \mathbb{N}], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\),
* BPNTRB,BPNTRE,LB,B, [LDB],BETA,C, [LDC], [WORK], [LWORK])
\(\mathbb{N}\) TEGER*8 TRANSA, MB, KB,LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: \quad \mathrm{DESCRA}, \mathrm{B} \mathbb{N} D X, B P N T R B, B P N T R E\)
DOUBLE COMPLEX ALPHA,BETA
DOUBLE COMPLEX ,D \(\mathbb{M}\) ENSION (:) ::VAL
DOUBLE COM PLEX ,D \(\mathbb{M}\) ENSION (:, :) :: B , C

\section*{DESCRIPTION}
C <-alpha op (A ) B + beta C
where A LPHA andBETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, \(A\) is a m atrix represented in block sparse colum n form at and op (A ) is one of \(o p(A)=A \quad\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRA N SA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix.
2 is equivalentto 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(s\) in matrix A
\(N \quad N\) um berof colum \(n s\) in \(m\) atrix \(C\)

K B \(\quad\) Number ofblock colum ns in m atrix A

ALPHA Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(A=A\) )
2: Herm Itian ( \(A=C O N J(A)\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(A=-C O N J\) ( \(A\) ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper

DESCRA (3) m ain diagonaltype
0 : non-unit
1 :unit
DESCRA (4) A ray base NOT \(\mathbb{M}\) PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 :unknown
1 : no repeated indices
VAL () scalar array of length \(\mathrm{LB} * \mathrm{LB} * \mathrm{BNN} Z\) consisting of the block entries stored collm n-m ajorw thin each dense block .
\(B \operatorname{IND}\) X (integer array of length BNNZ consisting of the block row indioes of the block entries ofA .

BPN TRB 0 integer aray of length \(K B\) such that BPN TRB (J) BPN TRB (1)+1 points to location in B IND X of the firstblock entry of the J-th block colum n of A.
BPNTRE ( integeramay of length \(K B\) such that BPN TRE (J) BPN TRB (1) points to location in B IN D X of the last.block entry of the J-th block colum n of A.

LB dim ension of dense blocks com posing A.
B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension of \(B\)
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension ofC
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the current version.

LW ORK length ofW ORK array. LW ORK is notreferenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U sers G uide available at:
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse .ps

\section*{NOTES/BUGS}

It is know \(n\) that there exists another representation of the block sparse colum \(n\) form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each block colum \(n\) in the arrays VAL and B INDX is used instead of tw o arrays BPN TRB and BPN TRE. To use the routine \(w\) ith this kind ofblock sparse colum \(n\) form at the follow ing calling sequence should be used

CALL ZBSCMM (TRANSA,MB,N,KB,ALPHA,DESCRA, * \(\quad V A L, B \mathbb{N D}, \mathbb{A}, \mathbb{A}(2), L B\), * B,LDB,BETA, C,LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
zbscsm -block sparse colum \(n\) form at triangular solve

\section*{SYNOPSIS}

SUBROUTINE ZBSCSM (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* \(\quad\) B,LDB,BETA, C,LDC,W ORK,LW ORK)
\(\mathbb{N} T E G E R\) TRANSA, MB,N,UNITD,DESCRA (5), LB,
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R \quad B \mathbb{N} D X(B N N Z)\), BPNTRB \(M B)\), BPNTRE \(M\) B)
DOUBLE COM PLEX ALPHA,BETA

(LW ORK)
SUBROUTINE ZBSCSM_64(TRANSA,MB,N,UNITD,DV,ALPHA,DESCRA,
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA, C,LDC,WORK,LWORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,N,UNITD,DESCRA (5), LB, * LDB,LDC,LWORK
 D OUBLE COM PLEX ALPHA,BETA
 (LW ORK)
where: BNNZ = BPNTRE M B)-BPNTRB (1)

\section*{F95 INTERFACE}

SUBROUTINEBSCSM (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,VAL,B \(\mathbb{N} D X\), * BPNTRB,BPNTRE,LB,B,[LDB],BETA,C,[LDC],[WORK],[LWORK])
\(\mathbb{N} T E G E R\) TRANSA,MB,N,UNITD,LB
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: D E S C R A, B \mathbb{N} D X, B P N T R B, B P N T R E\)
DOUBLE COMPLEX ALPHA,BETA

SU BROUTINE BSCSM_64 (TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,VAL,BINDX,
* BPNTRB,BPNTRE,LB,B, [LDB],BETA,C, [LDC], [W ORK], [LW ORK])
\(\mathbb{N} T E G E R * 8\) TRANSA, \(\mathrm{M} B, N, \mathrm{UN}\) ITD , LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: D E S C R A, B \mathbb{N D} X, B P N T R B, B P N T R E\)
DOUBLE COMPLEX ALPHA,BETA
DOUBLE COM PLEX ,D \(\mathbb{M} E N S I O N(:):: V A L, D V\)
DOUBLE COM PLEX ,D \(\mathbb{M} E N S I O N(:,:):: B, C\)

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { op (A)B+BETA } C \quad C<-A L P H A D \text { op }(A) B+B E T A C \\
& C<-A L P H A \text { op }(A) D B+B E T A C
\end{aligned}
\]
where ALPHA andBETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense \(m\) atrices, \(D\) is a block diagonalm atrix, \(A\) is a unit, ornon-unit, upper or low er triangularm atrix represented in block sparse colum n form at and op (A ) is one of op (A) \()=\operatorname{inv}(A)\) or op (A \()=\operatorname{inv}(A)\) or op (A) \()=\operatorname{inv}\left(\operatorname{cong}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix
0 : operate w th m atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(s\) in \(m\) atrix \(A\)
\(N \quad N\) um berof colum ns in matrix C

UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 : Scale on left (row block scaling)
3 : Scale on right (colum \(n\) block scaling)

DV () A may of the length M B *LB *LB consisting of the block entries ofblock diagonalm atrix \(D\) where each block is stored in standard colum n-m ajor form .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Fíve elem ent integer anay

DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
\(2:\) Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) )
3 :Triangular
4 : Skew (Anti)-Symm etric ( \(A=-A\) )
5 : D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\operatorname{CONJ}\) (A))
N ote:For the routine, DESCRA \((1)=3\) is only supported.

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 : non-identity blocks on the \(m\) ain diagonal
1 : identity diagonalblocks
2 : diagonalblocks are densem atrices
DESCRA (4) A ray base (NOT IM PLEM ENTED)
0 : C C + + com patible
1 :Fortran com patible
D ESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL 0 scalar array of length LB *LB *BNN Z consisting of the block entries stored colum \(n-m\) ajorw thin each dense block.
\(B \mathbb{N} D X 0 \quad\) integer amray of length BNNZ consisting of the block row indices of the block entries ofA.
The block row indicesM U ST be sorted
in increasing order foreach block colum \(n\).
BPNTRB () integer array of length \(M B\) such that BPN TRB (J) BPN TRB (1)+1 points to location in B IND X of the first.block entry of the J-th block colum n of A.

BPNTRE ( integer anay of length M B such that
BPN TRE (J)-BPN TRB (1) points to location in B IND X of the last.block entry of the Jth block colum n of A .

LB dim ension of dense blocks com posing A.
B 0 rectangular array w ith firstdin ension LD B.

LD B leading din ension ofB
BETA Scalarparam eter
C 0 rectangular array with first dim ension LD C .

LD C leading dim ension of \(C\)

W ORK () scratch array of length LW ORK.
On exit, if LW ORK \(=-1\), W ORK (1) retums the optim um size ofLW ORK.

LW ORK length ofW ORK array.LW ORK should be at least M B*LB.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK >=M B *LB*N_CPU \(S\) where N_CPU \(S\) is the \(m\) axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/n cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.
2. If \(D E S C R A\) (3)=0, the low erorupper triangularpart of each diagonalblock is used by the routine depending on DESCRA (2).
3. If \(D E S C R A\) (3)=1, the unitdiagonalblocksm ightorm ight not.be referenced in the BSC representation of a sparse \(m\) atrix. They are notused anyw ay.
4. If \(D E S C R A\) (3)=2, diagonalblocks are considered as dense \(m\) atrices and the LU factorization \(w\) ith partialpivoting is
used by the routine. WORK (1)=0 on retum if the factorization foralldiagonalblocks has been com pleted successfully, otherw ise WORK (1) = -iw here i is the block num ber forw hich the LU factorization could notbe com puted.
5.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse m atrix \(A\) is used. H ow erver \(\operatorname{DESCRA}\) (1) m ustbe equalto 3 in this case.

6 . It is know \(n\) that there exists another representation of the block sparse colum n form at (see for exam ple Y Saad, "Tterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each block colum \(n\) in the arrays VAL and B \(\mathbb{N D} D\) is used instead of tw o arrays BPN TRB and BPN TRE. To use the routine \(w\) ith this kind ofblock sparse colum \(n\) form at the follow ing calling sequence should be used

CALL ZBSCSM (TRANSA,MB,N,UNITD,DV,ALPHA,DESCRA, * \(\quad V A L, B \mathbb{N} D X, \mathbb{A}, \mathbb{A}(2), L B\),
* B,LDB,BETA, C,LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
zbsmm -block sparse row form atm atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZBSRMM (TRANSA,M B,N,KB,ALPHA,DESCRA,}

* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,MB,N,KB,DESCRA (5),LB,}
* LDB,LDC,LW ORK
INTEGER BINDX (BNNZ),BPNTRB MB),BPNTRE MB)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (LB *LB*BNNZ),B (LDB,*),C (LDC ,*),W ORK (LW ORK)

```
SUBROUTINE ZBSRMM_64(TRANSA, MB,N,KB,ALPHA,DESCRA,
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,N,KB,DESCRA (5),LB,
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R * 8 \operatorname{B} \mathbb{N} D X(B N N Z), B P N T R B(M)\), BPNTREMB)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (LB*LB*BNNZ), B (LDB,*), C (LDC,*),WORK (LW ORK)
where: \(\operatorname{BNNZ}=\mathrm{BPNTRE}(\mathrm{M})\)-BPNTRB (1)

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE BSRMM (TRANSA,M B, N ],KB,ALPHA,DESCRA,VAL,B INDX,}

* BPNTRB,BPNTRE,LB,B,[LD B ],BETA,C,[LDC], [W ORK ], [LW ORK ])
INTEGER TRANSA,MB,KB,LB
\mathbb{NTEGER,D IM ENSION (:) :: DESCRA,B NDDX,BPNTRB,BPNTRE}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D IM ENSION (:) ::VAL
DOUBLE COM PLEX,D IM ENSION (:,:) :: B,C

```

SUBROUT \(\mathbb{N} E \operatorname{BSRM} M \_64(T R A N S A, M B, \mathbb{N}], K B, A L P H A, D E S C R A, V A L, B \mathbb{N} D X\),
* BPNTRB,BPNTRE,LB,B, [LDB],BETA,C,[LDC], [WORK], [LWORK])
\(\mathbb{N}\) TEGER*8 TRANSA, MB, KB,LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: \quad \mathrm{DESCRA}, \mathrm{B} \mathbb{N} D X, B P N T R B, B P N T R E\)
DOUBLE COMPLEX ALPHA,BETA
DOUBLE COMPLEX ,D \(\mathbb{M}\) ENSION (:) ::VAL
DOUBLE COM PLEX ,D \(\mathbb{M}\) ENSION (:, :) :: B , C

\section*{DESCRIPTION}
C <-alpha op (A ) B + beta C
where A LPHA andBETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, \(A\) is a m atrix represented in block sparse row form at and op (A ) is one of \(o p(A)=A \quad\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix \(A\) is real.

M B \(\quad\) Num ber ofblock row \(s\) in matrix A

N \(\quad\) um berof colum ns in \(m\) atrix \(C\)

K B \(\quad\) Number ofblock colum ns in m atrix A

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(A=A\) )
2: Herm Itian ( \(A=C O N J(A)\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(A=-C O N J\) ( \(A\) ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper

DESCRA (3) main diagonal type
0 : non-unit
1 :unit
DESCRA (4) A ray base NOT \(\mathbb{M}\) PLEM ENTED)
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () scalar array of length LB *LB*BNNZ consisting of the block entries stored colum n-m ajorw thin each dense block .
\(B \operatorname{IND}\) X (integer array of length BNNZ consisting of the block colum n indices of the block entries of A.

BPN TRB () integeramay of length \(M B\) such that BPN TRB (J) BPN TRB (1)+1 points to location in B IND X of the first.block entry of the \(J\)-th block row of A.
BPN TRE () integer array of length \(M B\) such that BPN TRE (J)-BPN TRB (1) points to location in B IND X of the lastblock entry of the \(J\) th block row ofA.

LB dim ension of dense blocks com posing A.
B 0 rectangular array w th first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w th first dim ension LD C.
LD C leading dim ension of \(C\)
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the currentversion.

LW ORK length ofW ORK array. LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B lasU ser's G uide available at:
htep://m ath nist.gov/m csd/Staff/k Rem ington/Aspblas/
"D ocum ent forthe B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse дps

\section*{NOTES /BUGS}

It is know \(n\) that there exists another representation of the block sparse row form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem s",W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. The \(m\) ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each block row in the amays \(V A L\) and \(B \mathbb{N D X}\) is used instead of tw o arays BPN TRB and BPN TRE .To use the routine w ith this kind ofblock sparse row form at the follow ing calling sequence should be used

CALL ZBSRMM (TRANSA, MB,N,KB,ALPHA,DESCRA, * \(V A L, B \mathbb{N} D X, \mathbb{A}, \mathbb{I A}(2), L B\),
* \(\quad \mathrm{B}, \mathrm{LD} B, B E T A, C, L D C, W\) ORK,LWORK )

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

zbsrsm -block sparse row form at triangular solve

```

\section*{SYNOPSIS}
```

SUBROUTINE ZBSRSM(TRANSA,M B,N,UNITD,DV,ALPHA,DESCRA,

* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER TRANSA,MB,N,UNITD,DESCRA (5),LB,
* LDB,LDC,LW ORK
\mathbb{NTEGER BINDX (BNNZ),BPNTRB MB),BPNTRE MB)}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX DV M B *LB*LB),VAL (LB*LB*BNNZ),B (LD B,*),C (LDC,*),W ORK

```
(LW ORK)
SUBROUTINE ZBSRSM_64(TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,
* VAL,BINDX,BPNTRB,BPNTRE,LB,
* B,LDB,BETA, C,LDC,WORK,LW ORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, MB,N,UNITD,DESCRA (5), LB,
* LDB,LDC,LWORK
\(\mathbb{N} T E G E R * 8 \operatorname{B} \mathbb{N} D X(B N N Z), B P N T R B(M)\), BPNTREMB)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX DV M B *LB*LB),VAL (LB*LB*BNNZ),B(LDB,*),C(LDC,*),WORK
(LW ORK)
where: \(\operatorname{BNN} Z=B P N T R E M B)-B P N T R B(1)\)

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NEBSRSM (TRANSA,MB,N,UNITD,DV,ALPHA,DESCRA,VAL,B}\mathbb{N}DX,

* BPNTRB,BPNTRE,LB,B,[LDB],BETA,C,[LDC], [W ORK], [LW ORK])
\mathbb{NTEGER TRANSA,MB,N,UNTTD,LB}
\mathbb{NTEGER,D IM ENSION (:) :: DESCRA,BINDX,BPNTRB,BPNTRE}\
DOUBLE COMPLEX ALPHA,BETA

```

SU BROUTINE BSRSM _64 (TRANSA, MB,N, UN ITD, DV, ALPHA,DESCRA, VAL, B \(\mathbb{N} D X\),
* BPNTRB,BPNTRE,LB,B, [LDB],BETA,C, [LDC], [W ORK], [LW ORK])
\(\mathbb{N} T E G E R * 8\) TRANSA, \(\mathrm{M} B, N, \mathrm{UN}\) ITD , LB
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: D E S C R A, B \mathbb{N D} X, B P N T R B, B P N T R E\)
DOUBLE COMPLEX ALPHA,BETA
DOUBLE COM PLEX ,D \(\mathbb{M} E N S I O N(:):: V A L, D V\)
DOUBLE COM PLEX ,D \(\mathbb{M} E N S I O N(:,:):: B, C\)

\section*{DESCRIPTION}
```

C<-ALPHA Op(A)B + BETA C C <-ALPHA D Op (A)B + BETA C
C<-ALPHA Op(A)D B + BETA C

```
where ALPHA andBETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense \(m\) atrices, \(D\) is a block diagonalm atrix, \(A\) is a unit, ornon-unit, upperor low er triangularm atrix represented in block sparse row form at form atand op (A ) is one of
op (A) \()=\operatorname{inv}(A)\) or op (A \()=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}\left(\operatorname{cong}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalentto 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(s\) in \(m\) atrix \(A\)
\(N \quad N\) um berof colum ns in matrix C

UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 : Scale on left (row block scaling)
3 : Scale on right (colum \(n\) block scaling)

DV () A may of the length M B *LB *LB consisting of the block entries ofblock diagonalm atrix \(D\) where each block is stored in standard colum n-m ajor form .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Fíve elem ent integer anay

DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm tian ( \(\mathrm{A}=\mathrm{CONJG}\) (A))
3 :Triangular
4 : Skew (Anti)-Symm etric ( \(A=-A\) )
5 : D iagonal
6 : Skew Herm Hian ( \(\mathrm{A}=-\mathrm{CONJG}(\mathrm{A})\) )
N ote:For the routine, DESCRA \((1)=3\) is only supported.

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 : non-identity blocks on the \(m\) ain diagonal
1 : identity diagonalblocks
2 : diagonalblocks are densem atrices
DESCRA (4) A ray base (NOT IM PLEM ENTED)
0 : C C + + com patible
1 :Fortran com patible
D ESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL 0 scalar array of length LB *LB *BNN Z consisting of the block entries stored colum \(n-m\) ajorw thin each dense block.
\(B \mathbb{N} D X 0 \quad\) integer amray of length BNNZ consisting of the block collum \(n\) indiges of the block entries of A.
The block colum \(n\) indices M U ST be sorted in increasing order foreach block row .

BPNTRB () integeramay of length \(M B\) such that BPN TRB (J) BPN TRB (1)+1 points to location in B IND X of the firstblock entry of the J-th block row of A.

BPN TRE () integer aray of length \(M B\) such that BPN TRE (J)-BPN TRB (1) points to location in B IND X of the lastblock entry of the \(J\) th block row of A.

LB dim ension of dense blocks com posing A.
B 0 rectangular anray with firstdim ension LD B .

LD B leading din ension ofB
BETA Scalarparam eter
C 0 rectangular array with first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch array of length LW ORK.
On exit, if LW ORK \(=-1\), W ORK (1) retums the optim um size ofLW ORK.

LW ORK length ofW ORK array.LW ORK should be at least M B*LB.

Forgood perform ance, LW O RK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK >=M B *LB*N_CPU \(S\) where N_CPU \(S\) is the \(m\) axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/n cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.
2. If \(D E S C R A\) (3)=0, the low erorupper triangularpart of each diagonalblock is used by the routine depending on DESCRA (2).
3. If \(D E S C R A\) (3)=1, the unitdiagonalblocksm ightorm ight not.be referenced in the BSC representation of a sparse \(m\) atrix. They are notused anyw ay.
4. If \(D E S C R A\) (3)=2, diagonalblocks are considered as dense \(m\) atrices and the LU factorization \(w\) ith partialpivoting is
used by the routine. WORK (1)=0 on retum if the factorization foralldiagonalblocks has been com pleted successfully, otherw ise WORK (1) = -iw here i is the block num ber forw hich the LU factorization could notbe com puted.
5.The routine can be applied for solving triangular system \(s\) w hen the upper or low er triangle of the general sparse \(m\) atrix A is used. H ow erverDESCRA (1) m ustbe equalto 3 in this case.

6 . It is know \(n\) that there exists another representation of the block sparse row form at (see forexam ple Y Saad, "Tterative M ethods forSparse LinearSystem s",W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each block row in the arrays VAL and B \(\mathbb{N} D X\) is used instead oftw o arrays BPN TRB and BPN TRE. To use the routine \(w\) ith this kind ofblock sparse row form at the follow ing calling sequence should be used

CALL ZBSRSM (TRANSA,MB,N,UNITD,DV,ALPHA,DESCRA, * \(\quad V A L, B \mathbb{N} D X, \mathbb{A}, \mathbb{A}(2), L B\),
* B,LDB,BETA, C,LDC,W ORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zcnvoor-com pute the convolution or comelation of com plex vectors

\section*{SYNOPSIS}

SU BROUTINE ZCNVCOR (CNVCOR,FOUR,NX,X, \(\mathbb{F X}, \mathbb{N} C X, N Y, N P R E, M, Y\), \(\mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N} C 2 Y, N Z, K, Z, \mathbb{F Z}, \mathbb{N} C 1 Z, \mathbb{N} C 2 Z, W\) ORK,LW ORK)

CHARACTER * 1 CNVCOR,FOUR
DOUBLE COM PLEX X (*), Y (*), Z (*), W ORK (*)
\(\mathbb{N} T E G E R N X, \mathbb{F X}, \mathbb{N} C X, N Y, N P R E, M, \mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N C} 2 Y, N Z\), \(\mathrm{K}, \mathbb{F Z}, \mathbb{N} \mathrm{C} 1 \mathrm{Z}, \mathbb{N} \mathrm{C} 2 \mathrm{Z}, \mathrm{LW}\) ORK

SU BROUTINE ZCNVCOR_64 (CNVCOR,FOUR,NX,X, \(\mathbb{F X}, \mathbb{N} C X, N Y, N P R E, M, Y\), \(\mathbb{F} Y, \mathbb{N C} 1 Y, \mathbb{N} C 2 Y, N Z, K, Z, \mathbb{F} Z, \mathbb{N} C 1 Z, \mathbb{N} C 2 Z, W\) ORK,LW ORK)

CHARACTER * 1 CNVCOR,FOUR
DOUBLE COM PLEXX (*), Y (*), Z (*), W ORK (*)
\(\mathbb{N} T E G E R * 8 N X, \mathbb{F} X, \mathbb{N} C X, N Y, N P R E, M, \mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N} C 2 Y, N Z\),
\(\mathrm{K}, \mathbb{F Z}, \mathbb{N} \mathrm{C} 1 \mathrm{Z}, \mathbb{N} \mathrm{C} 2 \mathrm{Z}, \mathrm{LW}\) ORK

\section*{F95 INTERFACE}

SU BROUTINE CNVCOR (CNVCOR,FOUR, \(\mathbb{N} X], X, \mathbb{F X},[\mathbb{N C X}], N Y, N P R E, M, Y\), \(\mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N} C 2 Y, N Z, K, Z, \mathbb{F} Z, \mathbb{N C} 1 Z, \mathbb{N C} 2 Z, W\) ORK, [LW ORK ])

CHARACTER (LEN=1): :CNVCOR,FOUR
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::X,Y,Z,W ORK
\(\mathbb{N} T E G E R:: N X, \mathbb{F} X, \mathbb{N} C X, N Y, N P R E, M, \mathbb{F} Y, \mathbb{N C} 1 Y, \mathbb{N} C 2 Y\), \(\mathrm{NZ}, \mathrm{K}, \mathbb{F} \mathrm{Z}, \mathbb{N} \mathrm{C} 1 \mathrm{Z}, \mathbb{N} \mathrm{C} 2 \mathrm{Z}, \mathrm{LW}\) ORK

SU BROUTINE CNVCOR_64 (CNVCOR,FOUR, \(\mathbb{N} X], X, \mathbb{F} X,[\mathbb{N} C X], N Y, N P R E, M\), \(\mathrm{Y}, \mathbb{F} \mathrm{Y}, \mathbb{N} \mathrm{C} 1 \mathrm{Y}, \mathbb{N} \mathrm{C} 2 \mathrm{Y}, \mathrm{N} \mathrm{Z}, \mathrm{K}, \mathrm{Z}, \mathbb{F} \mathrm{Z}, \mathbb{N} \mathrm{C} 1 \mathrm{Z}, \mathbb{N} \mathrm{C} 2 \mathrm{Z}, \mathrm{W}\) ORK, [LW ORK ])

CHARACTER (LEN=1) ::CNVCOR,FOUR
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) :: X , Y, Z,W ORK
\(\mathbb{N} \operatorname{TEGER}(8):: N X, \mathbb{F} X, \mathbb{N} C X, N Y, N P R E, M, \mathbb{F} Y, \mathbb{N} C 1 Y, \mathbb{N} C 2 Y\), NZ,K, \(\mathbb{F} Z, \mathbb{N} C 1 Z, \mathbb{N} C 2 Z, L W O R K\)

\section*{C INTERFACE}
\#include <sunperfh>
void zcnvcor(char cnvoor, char four, int nx, doublecom plex
\({ }^{*} x\), int ifx, intincx, intny, intnpre, intm, doublecom plex *y, int ify, int incly, int inc2y, int \(n z\), int \(k\), doublecom plex \({ }^{*} z\), int ifz, int incl \(z\), int inc \(2 z\), doublecom plex *w ork, int lw ork); void zenvcor_64 (charcnvcor, char four, long nx, doublecom plex *x, long ifx, long incx, long ny, long npre, long \(m\), doublecom plex *y, long ify, long incly, long inc2y, long \(n z\), long \(k\), doublecom plex * \(z\), long ify, long inc1z, long inc2z, doublecom plex *W Ork, long lw ork);

\section*{PURPOSE}
zanvoor com putes the convolution or correlation of com plex vectors.

\section*{ARGUMENTS}

CNVCOR (input)
\(V\) 'or \(\mathrm{V}^{\prime}\) if convolution is desired, \(\mathrm{R}^{\prime}\) or \(\mathrm{r}^{\prime}\) if comelation is desired.

FOUR (input)
\(T\) 'or t'if the Founier transform \(m\) ethod is to be used, D 'or d'ifthe com putation should be done directly from the definition. The Fourier transform m ethod is generally faster, but itm ay introduce noticeable errors into certain results, notably w hen both the realand im aginary parts of the filter and data vectors consist entirely of integers or vectors w here elem ents of either the filtervector or a given data vectordiffer significantly in \(m\) agnitude from the 1 -norm of the vector.

NX (input)
Length of the filtervector. NX >= 0. ZCNVCOR
w ill retum im m ediately if \(\mathrm{X}=0\).
X (input) dim ension (*)
Filtervector.

FFX (input)
Index of the firstelem entofX. \(\mathrm{NX}>=\mathbb{F X}>=1\).
\(\mathbb{N C X}\) (input)
Stride betw een elem ents of the filtervector in \(X\).
\(\mathbb{N} C X>0\).

NY (input)
Length of the inputvectors. NY >=0. ZCNVCOR w ill retum im m ediately if \(\mathrm{N} Y=0\).

NPRE (input)
The num ber of im plicit zeros prepended to the \(Y\) vectors. NPRE >=0.

M (input)
Num berof inputvectors. M >= 0. ZCNVCOR will retum imm ediately if \(M=0\).

Y (input) dim ension ( \({ }^{*}\) )
Inputvectors.
IFY (input)
Index of the firstelem entof Y . \(\mathrm{N} Y>=\mathbb{F Y}>=1\).
\(\mathbb{N} C 1 Y\) (input)
Stride betw een elem ents of the inputvectors in \(Y\). \(\mathbb{N} C 1 \mathrm{Y}>0\).
\(\mathbb{N} C 2 Y\) (input)
Stride betw een the inputvectors in \(Y . \mathbb{N} C 2 Y>0\).

NZ (input)
Length of the output vectors. NZ >= 0. ZCNVCOR w ill retum im m ediately if \(\mathrm{N}=0\). See the N otes section below for inform ation abouthow this argu\(m\) ent interactsw ith NX and NY to control circular versus end-off shifting.

K (input)
\(N\) um berof \(Z\) vectors. \(K>=0\). If \(K=0\) then ZCNVCOR will retum immediately. If \(K<M\) then only the firstK inputvectors \(w\) ill be processed. If \(K>M\) then \(M\) inputvectors \(w\) illbe processed.

Z (output)
dim ension (*)
Resultvectors.

FZ (input)
Index of the firstelem entof \(Z . N Z>=\mathbb{F Z}>=1\).
\(\mathbb{N C} 1 Z\) (input)
Stride betw een elem ents of the output vectors in Z. \(\mathbb{N} C 1 Z>0\).
\(\mathbb{N C} 2 \mathrm{Z}\) (input)
Stride betw een the output vectors in Z. \(\mathbb{N C}\) C 2 Z > 0 .

W ORK (input/output)
(input/scratch) dim ension (LW ORK)
Scratch space. Before the first call to ZCNVCOR
w ith particular values of the integer argum ents the firstelem entofW ORK mustbe set to zero. If \(\mathrm{W} O R K\) is w rilten betw een calls to ZCNVCOR or if ZCNVCOR is called \(w\) ith different values of the integer argum ents then the firstelem entofW ORK m ustagain be set to zero before each call. If W ORK has notbeen w rilten and the sam e values of the integer argum ents are used then the firstelem entofW ORK to zero. This can avoid certain initializations that store their results into \(\mathrm{W} O R K\), and avoiding the initialization can \(m\) ake ZCNVCOR nun faster.

LW ORK (input)
Length of ORK . LW ORK \(>=2 \star \mathrm{MAX} \mathbb{N} X, N Y, N Z)+8\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zcnvcor2 - com pute the convolution orcomelation of com plex m atrices

\section*{SYNOPSIS}

SU BROUTINE ZCNVCOR2 CNVCOR,METHOD,TRANSX, SCRATCHX,TRANSY, SCRATCHY,MX,NX,X,LDX,MY,NY,MPRE,NPRE,Y,LDY,MZ,NZ,Z, LD \(\mathrm{Z}, \mathrm{W}\) ORK \(\mathbb{N}\), LW ORK)

CHARACTER * 1 CNVCOR, METHOD, TRANSX, SCRATCHX, TRANSY, SCRATCHY
DOUBLE COM PLEX X (LDX,*), Y (LDY,*), Z (LD Z, \(\left.{ }^{\star}\right), \mathrm{W} O R K \mathbb{N}\) ( \({ }^{*}\) )
\(\mathbb{N}\) TEGER M X,NX,LDX,MY,NY,MPRE,NPRE,LDY,M Z, NZ, LD Z, LW ORK

SUBROUTINE ZCNVCOR2_64 (CNVCOR,M ETHOD,TRANSX,SCRATCHX,TRANSY, SCRATCHY,MX,NX,X,LDX,MY,NY,MPRE,NPRE,Y,LDY,MZ,NZ,Z, LD Z, W ORK \(\mathbb{N}\),LW ORK)

CHARACTER * 1 CNVCOR, METHOD, TRANSX, SCRATCHX, TRANSY, SCRATCHY

\(\mathbb{N} T E G E R * 8\) M X,NX,LDX,MY,NY,MPRE,NPRE,LDY,M Z,NZ,LDZ, LW ORK

\section*{F95 INTERFACE}

SU BROUTINE CNVCOR2 (CNVCOR,M ETHOD,TRANSX,SCRATCHX,TRANSY, SCRATCHY, \(\mathbb{M} X], \mathbb{N} X], X,[L D X], \mathbb{M} Y], \mathbb{N} Y], M \operatorname{PRE}, N P R E, Y,[L D Y]\), \(\mathbb{M} Z], \mathbb{N} Z], Z,[L D Z], W\) ORK \(\mathbb{N},[L W O R K])\)

CHARACTER (LEN=1) :: CNVCOR, METHOD, TRANSX, SCRATCHX, TRANSY,SCRATCHY
    SCRATCHY, \(\mathbb{M} X], \mathbb{N X}], X,[L D X], \mathbb{M} Y], \mathbb{N} Y], M P R E, N P R E, Y,[L D Y]\),
    \(\mathbb{M} Z], \mathbb{N} Z], Z,[L D Z], W\) ORK \(\mathbb{N},[\) LW ORK ])

CHARACTER (โEN=1) :: CNVCOR, METHOD, TRANSX, SCRATCHX, TRANSY, SCRATCHY
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK \(\mathbb{N}\)
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) :: X , Y , Z
\(\mathbb{N}\) TEGER (8) :: M X , NX,LDX, MY,NY,MPRE,NPRE,LDY,MZ,NZ,
LD Z , LW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zcnvcor2 (char cnvcor, charm ethod, char transx, char scratchx, chartransy, charscratchy, intm \(x\), int nx , doublecom plex *x, int ldx, intm y , int \(n y\), int \(m\) pre, intnpre, doublecom plex \({ }^{*} y\), int ldy, intm \(z\), intnz, doublecom plex *z, int ldz, doublecom plex *W orkin, intlw ork);
void zenvcor2_64 (charcnvcor, charm ethod, chartransx, char scratchx, char transy, char scratchy, long \(m x\), long \(n x\), doublecom plex *x, long ldx, long m y, long ny, long m pre, long npre, doublecom plex *y, long ldy, long m z, long nz, doublecom plex * \(z\), long ldz, doublecom plex *w orkin, long lw ork);

\section*{PURPOSE}
zcnvoor2 com putes the convolution or correlation of com plex m atrices.

\section*{ARGUMENTS}

CNVCOR (input)
V 'or も'to com pute convolution, R 'or 'r' to com pute comelation.

METHOD (input)
T 'or t'if the Fourier transform \(m\) ethod is to be used, D 'or d'to com pute directly from the definition.

TRANSX (input)
\(N\) 'or \(h\) 'if \(X\) is the filterm atrix, \(T\) ' or \(t^{\prime}\)
if transpose \((X)\) is the filterm atrix.

SCRATCHX (input)
N 'or h'ifX m ustbe preserved, S'or s 'if X can be used as scratch space. The contents ofX are undefined after retuming from a callin which X is allow ed to be used for scratch.

TRANSY (input)
N 'or h'ify is the inputm atrix, \(T\) 'or \(\mathrm{t}^{\prime}\) if transpose \((Y)\) is the inputm atrix.
SCRATCHY (input)
N 'or h'ifY m ustbe preserved, S'or s 'ify can be used as scratch space. The contents of \(Y\) are undefined after retuming from a callin which Y is allow ed to be used for scratch.

M X (input)
N um ber of row s in the filterm atrix. M X >=0.
NX (input)
Num ber of colum ns in the filterm atrix. NX \(>=0\).
\(X\) (input) dim ension (LD X,\(N X\) )
O n entry, the filterm atrix. U nchanged on exitif SCRATCHX is N' or h', undefined on exitif SCRATCHX is S'or \(\mathrm{s}^{\prime}\).

LD X (input)
Leading dim ension of the amay that contains the filterm atrix.

M Y (input)
\(N\) um ber of row \(s\) in the inputm atrix. \(M Y>=0\).
NY (input)
\(N\) um berof \(\infty 0\) lum \(n s\) in the inputm atrix. \(N Y>=0\).

M PRE (input)
N um ber of im plicit zeros to prepend to each row of the inputm atrix. M PRE \(>=0\).

NPRE (input)
\(N\) um berof im plicit zeros to prepend to each colum n of the inputm atrix. NPRE \(>=0\).

Y (input) din ension (LD Y ,*)

Inputm atrix. U nchanged on exit if SCRATCHY is \(N^{\prime}\) or \(h^{\prime}\), undefined on exitifSCRATCHY is \(S^{\prime}\) or \(s^{\prime}\).

LD Y (input)
Leading dim ension of the array that contains the inputm atrix.

M Z (input)
N um ber of row s in the output m atrix. \(\mathrm{M} \mathrm{Z}>=0\). ZCNVCOR2 will retum im m ediately if M \(\mathrm{Z}=0\).

N Z (input)
N um ber of colum ns in the outputm atrix. \(\mathrm{NZ}>=0\). ZCNVCOR2 w ill retum imm ediately if \(\mathrm{NZ}=0\).

Z (output)
dim ension ( \(\ddagger \mathrm{LD} Z, \star\) )
Resultm atrix.

LD Z (input)
Leading dim ension of the array that contains the resultm atrix. LD Z >= M AX ( \(1, \mathrm{M} \mathrm{Z}\) ).

W ORK \(\mathbb{N}\) (input/output)
(input/scratch) dim ension (LW ORK)
O n entry for the first call to ZCNVCOR2,W ORK \(\mathbb{N}(1)\)
\(m\) ust contain CM PLX ( \(0.0,0.0\) ). A fter the first call, W ORK \(\mathbb{N}\) (1) m ustbe set to CM PLX (0.0,0.0) iff
W ORK \(\mathbb{N}\) has been altered since the last call to this subroutine or if the sizes of the arrays have
changed.

LW ORK (input)
Length of the w ork vector. If the FFT is to be used then forbestperform ance LW ORK should be at least 30 w ords longer than the am ount of \(m\) em ory needed to hold the trig tables. If the FFT is not used, the value ofLW ORK is unim portant.

\section*{Contents}
- NAME
- SYNOPSIS

\title{
- F95 INTERFACE
}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

zcoom m -coordinatem atrix-m atrix m ultiply

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZCOOMM(TRANSA,M,N,K,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,JNDX,NNZ,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,M,N,K,DESCRA (5),NNZ}
* LDB,LDC,LW ORK
\mathbb{NTEGER INDX NNZ),JNDX NNZ)}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NE ZCOOMM_64(TRANSA,M,N,K,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,JNDX,NNZ,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER*8 TRANSA,M,N,K,DESCRA (5),NNZ
* LDB,LDC,LW ORK
\mathbb{NTEGER*8 INDX NNZ),NNDX NNZ)}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NECOOMM(TRANSA,M, N ],K,ALPHA,DESCRA,}

* VAL, NNDX, JNDX,NNZ,B,[LDB],BETA,C,[LDC],
* [W ORK], [LW ORK])
INTEGER TRANSA,M,K,NNZ
\mathbb{NTEGER,D\mathbb{M ENSION (:) :: DESCRA,INDX,UNDX}}\mathbf{~}=\mathbb{N}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D IM ENSION (:) ::VAL
DOUBLE COM PLEX,D IM ENSION (:,:) :: B,C

```

SUBROUTINECOOMM_64(TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A\),
* VAL, \(\mathbb{N} D X, \operatorname{JNDX}, N N Z, B,[L D B], B E T A, C,[L D C]\),
* [W ORK], [LW ORK])
\(\mathbb{I N T E G E R *}\) TRANSA, M, K, NNZ
\(\mathbb{N} T E G E R * 8, D \mathbb{I M} E N S \mathbb{I O N}(:):: D E S C R A, \mathbb{N} D X, J N D X\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D IM ENSION (:) ::VAL
DOUBLE COM PLEX,D \(\mathbb{M}\) ENSION (:, :) :: B,C

\section*{DESCRIPTION}
C <-alpha op (A) B + beta C
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrioes, \(A\) is a \(m\) atrix represented in coordinate form at and op(A) is one of \(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{con} \dot{g}\left(A^{\prime}\right)\).
( 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRAN SA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate w ith m atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M \(\quad\) um berof row \(s\) in \(m\) atrix A
N \(\quad\) Num berof \(C o l u m n s\) in \(m\) atrix \(C\)

K \(\quad \mathrm{Num}\) berof colum ns in \(m\) atrix \(A\)

A LPH A Scalarparam eter
DESCRA () D escriptor argum ent. Five elem ent integer array
DESCRA (1) m atrix structure
0 :general
1 : symmetric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) )
3 :Triangular
4 : Skew (Anti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJ}\) ( A ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper

DESCRA (3) main diagonal type
0 : non-unit
1 : unit
DESCRA (4) A ray base NOT IM PLEM ENTED)
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 :no repeated indices
VAL () scalar array of length NNZ consisting of the non-zero entries of \(A\), in any order.
\(\mathbb{I N D X}\) () integer array of length NNZ consisting of the comesponding row indices of the entries of A.

JND X () integer amray of length NNZ consisting of the corresponding colum \(n\) indioes of the entries of A.

NN Z number of non-zero elem ents in A.
B 0 rectangular array w th first dim ension LD B.
LD B leading din ension ofB

BETA Scalarparam eter
C 0 rectangular anray with firstdim ension LD C.

LD C leading dim ension of \(C\)
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is notreferenced in the current version.

\section*{SEE ALSO}

N IST FORTRA N Sparse B las U sers G uide available at:
http://m ath nistgov/m csd/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib.org/utk/papers/sparse.ps

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

zoopy -C opy x to y

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZCOPY (N,X,\mathbb{NCX,Y,\mathbb{NCY)}}\mathbf{N}=(})={
DOUBLE COM PLEX X (*),Y (*)
INTEGERN,\mathbb{NCX,INCY}
SUBROUT\mathbb{NE ZCOPY_64 N,X,\mathbb{NCX,Y,INCY)}}\mathbf{N}={
DOUBLE COM PLEX X (*),Y (*)
INTEGER*8N,\mathbb{NCX,INCY}

```
F95 INTERFACE
    SU BROUTINE COPY ( \(\mathbb{N}], X,[\mathbb{N C X}], Y,[\mathbb{N} C Y])\)
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::X,Y
    \(\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y\)
    SU BROUTINE COPY_64 (N ],X, [ \(\mathbb{N} C X], Y,[\mathbb{N} C Y])\)
    COM PLEX (8),D \(\mathbb{I}\) ENSION (:) ::X,Y
    \(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{I N C X}, \mathbb{N} C Y\)
C INTERFACE
    \#include <sunperfh>
    void zcopy (intn, doublecom plex *x, intincx, doublecom plex
        *y, int incy);
    void zcopy_64 (long n, doublecom plex *x, long incx, doub-

\section*{PURPOSE}
zoopy C opy \(x\) to \(y\) where \(x\) and \(y\) are \(n\)-vectors.

\section*{ARGUMENTS}

N (input)
On entry, \(N\) specifies the num ber of elem ents in the vector. N m ustbe at leastone for the subroutine to have any visible effect. U nchanged on exit.
X (input)
ofD \(\mathbb{I M} E N S I O N\) at least ( \(1+(\mathrm{n}-1) * a b s(\mathbb{N} C X)\)
). Before entry, the increm ented array \(\mathrm{X} m\) ust contain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X\) m ustnotbe zero. U nchanged on exit.

Y (output)
ofD \(\mathbb{I M}\) ENSIO N at least ( \(1+(\mathrm{m}-1) * a b s(\mathbb{N} C Y)\)
). On entry, the increm ented array \(Y\) m ustcontain the vectory. On exit, \(Y\) is overw ritten by the vectorx.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y\). \(\mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
zcscm m - com pressed sparse colum \(n\) form atm atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZCSCMM(TRANSA,M,N,K,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER TRANSA,M,N,K,DESCRA (5),
* LDB,LDC,LW ORK
INTEGER INDX NNZ),PNTRB(K),PNTRE (K)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NE ZCSCMM_64(TRANSA,M,N,K,A LPHA,DESCRA,}
* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER*8 TRANSA,M,N,K,DESCRA (5),}
* LDB,LDC,LW ORK

```

```

DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
where NN Z = PN TRE (K)-PN TRB (1)

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NECSCMM (TRANSA,M, N ],K,ALPHA,DESCRA,VAL, INDX,}

* PNTRB,PNTRE,B,[LDB],BETA,C, [LDC],[W ORK],[LW ORK])
INTEGER TRANSA,M,K
NNTEGER,D IM ENSION (:) :: DESCRA, NNDX,PN TRB,PNTRE
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D IM ENSION (:) ::VAL

```

DOUBLE COM PLEX,D \(\mathbb{M}\) ENSION (:, :) :: B,C

SUBROUTINE CSCMM_64(TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N D X}\),
* PNTRB, PNTRE, B, [LDB],BETA, C, [LDC], [WORK], [LW ORK])
\(\mathbb{N}\) TEGER*8 TRANSA, M, K
\(\mathbb{N} T E G E R * 8, D \mathbb{M}\) ENSION (:) :: DESCRA, \(\mathbb{N} D \mathrm{X}, \mathrm{PNTRB}, \mathrm{PNTRE}\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D IM ENSION (:) ::VAL
DOUBLE COM PLEX,D \(\mathbb{I}\) ENSION (:, :) :: B,C

\section*{DESCRIPTION}
\[
C \text { <-alpha op (A) B + beta C }
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, \(A\) is a \(m\) atrix represented in com pressed sparse colum \(n\) form at and op (A) is one of \(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=c o n j\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate w ith m atrix
1 : operate w th transpose m atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in \(m\) atrix A

N \(\quad N\) um berof colum ns in matrix C
K \(\quad\) Num berof colum ns in matrix A

A LPH A Scalarparam eter

DESCRA ( D escriptor argum ent. Fíve elem ent integer anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(A=A\) )
2 : Herm Aian ( \(\mathrm{A}=\mathrm{CONJG}\) (A))
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJ}\) ( A ) )
D ESCRA (2) upper/low er triangular indicator
1: low er

2 :upper
DESCRA (3) main diagonal type
0 :non-unit
1 : unit
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED)
0 :C C ++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM EN TED )
0 : unknown
1 : no repeated indices

VAL () scalar array of length NN Z consisting of nonzero entries ofA.

IND X \(0 \quad\) integer array of length NN Z consisting of the row indices of nonzero entries ofA .

PN TRB 0 integer amray of length \(K\) such thatPN TRB (J)-PN TRB (1)+1 points to location in VA L of the firstnonzero elem ent in colum n J .
PN TRE 0 integer array of length \(K\) such thatPN TRE (J)-PN TRB (1) points to location in V A L of the lastnonzero elem ent in colum n J .

B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension ofB

BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C.
LD C leading dim ension of \(C\)

W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the currentversion.

LW ORK length ofW ORK array.LW ORK is notreferenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov fin csd/Staffk Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee,
http://w w w netlib .org/utk/papers/sparse .ps

\section*{NOTES/BUGS}

It is know \(n\) that there exists another representation of the com pressed sparse colum n form at (see forexam ple Y Saad, "IterativeM ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each colum \(n\) in the arrays VA L and \(\mathbb{N} D \mathrm{X}\) is used instead oftw o arraysPN TRB and PN TRE.To use the routine \(w\) th this kind of sparse colum \(n\) form at the follow ing calling sequence should be used

SUBROUTINE SCSCMM (TRANSA, M,N,K,ALPHA,DESCRA,
* \(V A L, \mathbb{N} D X, \mathbb{A}, \mathbb{I}(2), B, L D B, B E T A\),
* C,LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
zcscsm -com pressed sparse colum n form at triangular solve

\section*{SYNOPSIS}
```

SUBROUTINE ZCSCSM(TRANSA,M ,N,UNITD,DV,ALPHA,DESCRA,

* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,M,N,UNITD,DESCRA (5),}
* LDB,LDC,LWORK
\mathbb{NTEGER INDX NNZ),PNTRB(M),PNTREM)}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX DV M ),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
SUBROUTINE ZCSCSM_64(TRANSA, M,N,UNITD,DV,ALPHA,DESCRA,
* VAL, \(\mathbb{N} D \mathrm{X}, \mathrm{PNTRB}, \mathrm{PN}\) TRE,
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, M,N,UNITD,DESCRA (5),
* LDB,LDC,LWORK
\(\mathbb{N} T E G E R * 8 \mathbb{N} D X(N N Z)\), PNTRB \((M)\), PNTRE \(M\) )
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX DV M),VAL NNZ), B (LDB,*), C (LDC ,*), W ORK (LW ORK)
where \(N N Z=P N T R E M)-P N T R B(1)\)

\section*{F95 INTERFACE}

SUBROUT \(\mathbb{N} E \operatorname{CSCSM}(T R A N S A, M, \mathbb{N}], U N \mathbb{T} D, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X\), * PNTRB, PNTRE, B, [LDB],BETA,C, [LDC], [WORK], [LW ORK]) \(\mathbb{N}\) TEGER TRANSA, M, UN ITD \(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: D E S C R A, \mathbb{N} D X, P N T R B, P N T R E\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COMPLEX,D \(\mathbb{I}\) ENSION (:) ::VAL,DV
DOUBLE COM PLEX,D \(\mathbb{M}\) ENSION (:,:) :: B,C

SUBROUT \(\mathbb{N} E \operatorname{CSC} M\) _ 64 (TRANSA \(, ~ M, ~ \mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X\),
* PNTRB, PNTRE, B, [ LDB],BETA, C, [LDC], [W ORK], [LW ORK])
\(\mathbb{I N T E G E R * 8 T R A N S A , M , U N I T D ~}\)
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: D E S C R A, \mathbb{N D} X, \operatorname{PNTRB}, \operatorname{PNTRE}\)
DOUBLE COMPLEX ALPHA,BETA
DOUBLE COM PLEX ,D \(\mathbb{M} E N S I O N(:):: V A L, D V\)
DOUBLE COM PLEX,D \(\mathbb{M} E N S I O N(:,:):\) B, C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \quad \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { Op (A) B + BETA C } \\
& C<-A L P H A \text { OP (A)D B + BETA } C
\end{aligned}
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense matrices, \(D\) is a diagonalscaling \(m\) atrix, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix represented in com pressed sparse colum n form atand op (A ) is one of \(\operatorname{op}(A)=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}(A)\) or op (A) \()=\operatorname{inv}\left(\operatorname{cong}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix 0 : operate w th m atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in \(m\) atrix \(A\)

N \(\quad\) umberof colum \(n s\) in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum \(n\) scaling)
4 :A utom atic colum n scaling (see section N OTES for furtherdetails)

DV () A rray of length M containing the diagonalentries of the scaling diagonalm atrix D.

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem ent integer anay DESCRA (1) m atrix structure
\[
\begin{aligned}
& 0 \text { : general } \\
& 1 \text { : sym m etric ( } A=A \text { ) } \\
& 2: \text { H erm itian ( } A=\operatorname{CON} \text { JG (A }) \text { ) } \\
& 3 \text { : Triangular } \\
& 4 \text { : Skew (A nti)-Sym m etric ( } A=-A \text { ) } \\
& 5 \text { : D iagonal } \\
& 6 \text { : Skew Herm itian ( } A=-\operatorname{CON} \text { JG (A ) ) }
\end{aligned}
\]

N ote: For the routine, D ESCRA (1)=3 is only supported.

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonaltype
0 : non-unit
1 :unit
DESCRA (4) A may base \(\mathbb{N O T} \mathbb{M}\) PLEM ENTED)
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? \(N\) OT \(\mathbb{M}\) PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL () scalar array of length N N Z consisting of nonzero entries ofA.
\(\mathbb{N} D \mathrm{X}\) () integer array of length N N Z consisting of the row indices of nonzero entries of . (R ow indigesM UST be sorted in increasing order for each colum n).

PNTRB () integer amay of length \(M\) such thatPN TRB (J) PN TRB (1)+1 points to location in VA L of the firstnonzero elem ent in colum \(n \mathrm{~J}\).

PN TRE () integer array of length \(M\) such thatPN TRE (J)-PN TRB (1) points to location in VA L of the lastnonzero elem ent in colum \(n \mathrm{~J}\).

B 0 rectangular array w ith first dim ension LD B .

LD B leading dim ension ofB

BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch array of length LW ORK.

On exit, if LW ORK \(=-1, W\) ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK array. LW ORK should be at leastM .
Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=M\) *N_CPUS where N_CPUS is the m axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
htep://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such tests \(m\) ust.be perform ed before calling this routine.
2. If UN ITD \(=4\), the routine scales the colum ns of A such that their 2 -norm s are one. The scaling \(m\) ay im prove the accuracy of the com puted solution. C orresponding entries of VA L are changed only in the particular case. On retum D V \(m\) atrix stored as a vector contains the diagonalm atrix by which the colum ns have been scaled. UN ITD = 3 should be used for the next calls to the routine \(w\) ith overw ritten VA L and DV.

WORK (1)=0 on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here i is the colum n num berw hich 2 -norm is exactly zero.
3. If \(D E S C R A(3)=1\) and \(U N\) ITD < 4, the unitdiagonalelem ents m ightorm ightnotbe referenced in the C SC representation
of a sparse \(m\) atrix. They are notused anyw ay in these cases. ButifU N ITD = 4, the unitdiagonalelem ents M U ST be referenced in the CSC representation.
4.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general.sparse \(m\) atrix A is used. H ow ever DESCRA (1) m ustbe equal to 3 in this case.
5. It is know \(n\) that there exists another representation of the com pressed sparse colum n form at (see for exam ple Y Saad, "Iterative M ethods forSparse LinearSystem \(s\) ", W PS, 1996). Its data structure consists of three anray instead of the fourused in the cumentim plem entation. Them ain difference is thatonly one amray, IA , containing the pointers to the beginning ofeach colum \(n\) in the arrays VA L and \(\mathbb{I N D X}\) is used instead of tw o amaysPNTRB and PN TRE.To use the routine \(w\) th this kind of sparse colum \(n\) form at the follow ing calling sequence should be used

SUBROUTINESCSCSM (TRANSA, M,N,UNITD,DV,ALPHA,DESCRA,
* \(V A L, \mathbb{N} D X, \mathbb{A}, \mathbb{A}(2), B, L D B, B E T A\),
* \(\quad\), LDC,\(W\) ORK,LW ORK )

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

zcsmm -com pressed sparse row form atm atrix-m atrix m ultiply

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZCSRMM (TRANSA,M,N,K,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,M,N,K,DESCRA (5),}
* LDB,LDC,LW ORK
\mathbb{NTEGER INDX NNZ),PNTRB (M),PNTREM)}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
SUBROUTINE ZCSRMM_64(TRANSA, M,N,K,ALPHA,DESCRA,
* VAL, \(\mathbb{N} D \mathrm{X}, \mathrm{PN} T \mathrm{RB}, \mathrm{PN}\) TRE,
* B,LDB,BETA, C,LDC,W ORK,LWORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, \(M, N, K, D E S C R A(5)\),
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R * 8 \mathbb{N} D X(N N Z)\), PNTRB \((M)\), PNTRE \(M\) )
DOUBLE COM PLEX ALPHA, BETA
DOUBLE COM PLEX VAL (NNZ), B (LDB,*), C (LDC,*),W ORK (LW ORK)
where \(N \mathrm{~N} Z=\operatorname{PNTRE} \mathrm{M})\) PNTRB(1)

\section*{F95 INTERFACE}
```

SUBROUTINE CSRMM (TRANSA,M, N ],K,ALPHA,DESCRA,VAL, INDX,

* PNTRB,PNTRE,B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])
INTEGER TRANSA,M,K
\mathbb{NTEGER,D IM ENSION (:) :: DESCRA, INDX,PNTRB,PNTRE}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D IM ENSION (:) ::VAL
DOUBLE COM PLEX,D IM ENSION (:,:) :: B,C

```

SUBROUTINE CSRMM_64 (TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N D X}\),
* PNTRB, PNTRE, B, [LDB],BETA, C , [LDC], [W ORK], [LWORK])
\(\mathbb{N}\) TEGER*8 TRANSA, M, K
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: \operatorname{DESCRA}, \mathbb{N} D X, \operatorname{PNTRB}, \operatorname{PNTRE}\)
DOUBLE COMPLEX ALPHA,BETA
DOUBLE COM PLEX ,D \(\mathbb{M}\) ENSION (:) ::VAL
DOUBLE COM PLEX ,D \(\mathbb{M}\) ENSION (: : : : : B , C

\section*{DESCRIPTION}
\[
C<- \text { alpha op (A) B + beta C }
\]
where A LPHA andBETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, A is am atrix represented in com pressed sparse row form at and \(o p(A)\) is one of \(o p(A)=A \quad\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRA N SA Indicates how to operate \(w\) th the sparse \(m\) atrix
0 : operate w ith m atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um ber of row \(s\) in \(m\) atrix A

N \(\quad\) Num berof colum ns in \(m\) atrix \(C\)

K \(\quad\) umberof colum ns in matrix A

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(A=A\) )
2: Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(A=-C O N J\) ( \(A\) ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper

DESCRA (3) main diagonal type
0 : non-unit
1 : unit
DESCRA (4) A ray base NOT \(\mathbb{M}\) PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 :no repeated indices
VAL ( scalar array of length NN Z consisting of nonzero entries ofA.
\(\mathbb{I N D X} 0 \quad\) integer array of length NN Z consisting of the colum \(n\) indioes of nonzero entries of \(A\).

PN TRB () integer array of length \(M\) such thatPN TRB (J) PN TRB (1)+1
points to location in VA L of the firstnonzero elem ent in row J .
PNTRE ( integerarray of length \(M\) such thatPNTRE (J)-PNTRB (1) points to location in V A L of the lastnonzero elem ent in row J .

B 0 rectangular array w th first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C \(0 \quad\) rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC
W ORK () scratch array of length LW ORK.W ORK is not referenced in the curment version.

LW ORK length ofW ORK aray. LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/n cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:

\section*{NOTES/BUGS}

It is know \(n\) that there exists another representation of the com pressed sparse row form at (see forexam ple Y Saad, "Herative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three anray instead of the fourused in the currentim plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each row in the arrays VA L and \(\mathbb{N D} \mathrm{X}\) is used instead of tw o arrays PN TRB and PN TRE. To use the routine w th this kind of com pressed sparse row form at the follow ing calling sequence should be used

SUBROUTINESCSRMM (TRANSA, M, N, K,ALPHA,DESCRA,
* \(\quad V A L, \mathbb{N} D X, \mathbb{A}, \mathbb{I A}(2), B, L D B, B E T A\),
* C,LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
zcsrsm - com pressed sparse row form at triangular solve

\section*{SYNOPSIS}
```

SUBROUTINE ZCSRSM(TRANSA,M ,N,UNITD,DV,ALPHA,DESCRA,

* VAL,\mathbb{NDX,PNTRB,PNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER TRANSA,M,N,UNITD,DESCRA (5),
* LDB,LDC,LW ORK
\mathbb{NTEGER INDX NNZ),PNTRB (M),PNTREM)}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX DV M ),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
SUBROUTINE ZCSRSM_64(TRANSA, M,N,UNTD,DV,ALPHA,DESCRA,
* VAL, \(\mathbb{N} D \mathrm{X}, \mathrm{PNTRB}, \mathrm{PNTRE}\),
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, M,N,UNITD,DESCRA (5),
* LDB,LDC,LWORK
\(\mathbb{N} T E G E R * 8 \mathbb{N} D X(N N Z)\), PNTRB \((M)\), PNTRE \(M\) )
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX DV M),VAL NNZ), B (LDB,*), C (LDC ,*), W ORK (LW ORK)
where \(N N Z=P N T R E M)-P N T R B(1)\)

\section*{F95 INTERFACE}

SUBROUT \(\mathbb{N} E C S R S M\) (TRANSA, \(M, \mathbb{N}], U N \mathbb{T} D, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X\),
* PNTRB, PNTRE, B, [LDB],BETA,C, [LDC], [WORK], [LW ORK])
\(\mathbb{N}\) TEGER TRANSA, M, UN ITD
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: D E S C R A, \mathbb{N} D X, P N T R B, P N T R E\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D \(\mathbb{M} E N S I O N(:):: V A L, D V\)
DOUBLE COM PLEX,D \(\mathbb{M}\) ENSION (:,:) :: B,C

SUBROUTINECSRSM_64 (TRANSA, M, \(\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X\),
* PNTRB,PNTRE, B, [LDB],BETA, C , [LDC], [W ORK], [LW ORK])
\(\mathbb{I N T E G E R * 8 T R A N S A , M , U N I T D ~}\)
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: D E S C R A, \mathbb{N D} X, \operatorname{PNTRB}, \operatorname{PNTRE}\)
DOUBLE COMPLEX ALPHA,BETA
DOUBLE COM PLEX,D \(\mathbb{M}\) ENSION (:) ::VAL, DV
DOUBLE COM PLEX ,D \(\mathbb{M} E N S I O N(:,:):\) B, C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { op }(A) B+B E T A C \\
& C<-A L P H A \text { OP }(A) D B+B E T A C
\end{aligned}
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense matrices, \(D\) is a diagonalscaling \(m\) atrix, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix represented in com pressed sparse row form atand op (A ) is one of op (A) \(=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}\left(\operatorname{conjg}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix 0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) th transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in \(m\) atrix \(A\)

N \(\quad\) um berof colum \(n s\) in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum \(n\) scaling)
4 : A utom atic row scaling (see section N O TES for furtherdetails)

DV () A ray of length M containing the diagonalentries of the scaling diagonalm atrix D.

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem ent integer anay DESCRA (1) m atrix structure
\[
\begin{aligned}
& 0 \text { : general } \\
& 1 \text { : sym m etric ( } A=A \text { ) } \\
& 2: \text { H erm itian ( } A=\operatorname{CON} \text { JG (A }) \text { ) } \\
& 3 \text { : Triangular } \\
& 4 \text { : Skew (A nti)-Sym m etric ( } A=-A \text { ) } \\
& 5 \text { : D iagonal } \\
& 6 \text { : Skew Herm itian ( } A=-\operatorname{CON} \text { JG (A ) ) }
\end{aligned}
\]

N ote: For the routine, only DESCRA (1)=3 is supported.

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 : non-unit
1 :unit
DESCRA (4) A ray base \(\mathbb{N O T} \mathbb{M}\) PLEM ENTED )
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? \(\mathbb{N O T} \mathbb{I}\) PLEM ENTED)
0 : unknown
1 : no repeated indices
VAL () scalar array of length N N Z consisting of nonzero entries ofA.
\(\mathbb{I N D X ( ) \quad i n t e g e r a m a y ~ o f ~ l e n g t h ~ N ~ N ~ Z ~ c o n s i s t i n g ~ o f ~ t h e ~ c o l u m ~ n ~}\) indices of nonzero entries ofA (colum n indices M U ST be sorted in increasing order for each row )

PNTRB () integer amay of length \(M\) such thatPN TRB (J) PN TRB (1)+1 points to location in VA L of the firstnonzero elem ent in row J.

PN TRE () integer amay of length M such thatPNTRE (J)-PN TRB (1) points to location in VA L of the lastnonzero elem ent in row J.

B 0 rectangular array w ith first dim ension LD B .

LD B leading dim ension ofB

BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch array of length LW ORK.

On exit, ifLW ORK \(=-1, W\) ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK array. LW ORK should be at leastM .
Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=M\) *N_CPUS where N_CPUS is the m axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
htep://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such tests \(m\) ustbe perform ed before calling this routine.
2. IfUN ITD \(=4\), the routine scales the row s of \(A\) such that their 2 -nom s are one. The scaling \(m\) ay im prove the accuracy of the com puted solution. C orresponding entries ofV A L are changed only in the particular case. On retum D V m atrix stored as a vector contains the diagonalm atrix by which the row s have been scaled. U N ITD = 2 should be used for the next calls to the routine w ith overw ritten VAL and DV.

WORK (1)=0 on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here i is the row num berw hich 2 -norm is exactly zero.
3. If \(\operatorname{DESCRA}(3)=1\) and UN ITD < 4, the unitdiagonalelem ents \(m\) ightorm ightnotbe referenced in the CSR representation of a sparse m atrix. They are notused anyw ay in these cases.

ButifUN ITD \(=4\), the unit diagonalelem ents M U ST be referenced in the CSR representation.
4.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse m atrix \(A\) is used. H ow ever \(\operatorname{DESCRA}\) (1) m ustbe equal to 3 in this case.
5. It is know \(n\) that there exists another representation of the com pressed sparse row form at (see for exam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of three array instead of the fourused in the current im plem entation. Them ain difference is that only one array, IA , containing the pointers to the beginning ofeach row in the amaysVA L and \(\mathbb{N} D \mathrm{X}\) is used instead of tw o arrays PN TRB and PN TRE. To use the routine \(w\) ith this kind of com pressed sparse row form at the follow ing calling sequence should be used

SUBROUTINESCSRSM (TRANSA,M,N,UNTID,DV,ALPHA,DESCRA,
* \(\quad V A L, \mathbb{N} D X, \mathbb{A}, \mathbb{I A}(2), B, L D B, B E T A, C\),
* LDC,WORK,LWORK)

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

zdiam m -diagonal form atm atrix-m atrix m ultiply

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZDIAMM(TRANSA,M,N,K,ALPHA,DESCRA,}

* VAL,LDA,\mathbb{D IAG,NDIAG,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,M,N,K,DESCRA (5),LDA,NDIAG,}
* LDB,LDC,LWORK

```

```

DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (LDA,ND IAG),B (LDB,*),C (LDC **),W ORK (LW ORK)

```
SUBROUTINE ZD IAMM_64(TRANSA, M,N,K,ALPHA,DESCRA,
* VAL,LDA, \(\mathbb{D} \mathbb{A} G, N D \mathbb{A} G\),
* B,LDB,BETA, C,LDC,WORK,LWORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, \(M, N, K, D E S C R A(5), L D A, N D \mathbb{I A}\),
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R * 8 \mathbb{D} \mathbb{A} G \mathbb{N} \mathbb{I} G)\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (LDA,ND IAG),B (LDB,*),C (LDC,*),W ORK (LW ORK)

\section*{F95 INTERFACE}

SUBROUTINED \(\operatorname{IAM}\) M (TRANSA, \(M, \mathbb{N}], K, A L P H A, D E S C R A, V A L,[L D A]\), * \(\mathbb{D} \mathbb{I} G, N D \mathbb{A}, \mathrm{~B},[\mathrm{LDB}], \mathrm{BETA}, \mathrm{C},[\mathrm{LDC}],[\mathbb{W} O R K],[\mathrm{LW} O R K])\)
\(\mathbb{N}\) TEGER TRANSA, M, K, NDIAG
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}\) (:) :: DESCRA, \(\mathbb{D} \mathbb{I A}\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COMPLEX,D \(\mathbb{I M} \operatorname{ENSION(:,:)::VAL,B,C~}\)
SUBROUTINEDIAMM_64 (TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L,[L D A]\), * \(\mathbb{D} \mathbb{I} G, N D \mathbb{A} G, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)

\section*{DESCRIPTION}

C <-alpha op (A) B + beta C
where A LPH A and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, \(A\) is a \(m\) atrix represented in diagonal form at and op (A) is one of
\[
o p(A)=A \quad \text { or } \operatorname{op}(A)=A^{\prime} \text { or op }(A)=\operatorname{conjg}\left(A^{\prime}\right) .
\]
( 'indicatesm atrix transpose)
TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate w ith m atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate w ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M \(\quad\) Num berof row \(s\) in \(m\) atrix A
N \(\quad\) Num berof colum ns in \(m\) atrix \(C\)
K \(\quad\) um berof colum ns in \(m\) atrix A

A LPH A Scalarparam eter
DESCRA () D escriptor argum ent. Five elem ent integer array
0 : general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm Aian ( \(\mathrm{A}=\mathrm{CONJG}\) (A))
3 :Triangular
4 : Skew (Anti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\operatorname{CONJ}\) ( A ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) main diagonal type
0 :non-unit
1 : unit
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
D ESCRA (5) repeated indices? NOT IM PLEM EN TED )
0 : unknown
1 : no repeated indices

VAL ( ) tw o-dim ensionalLD A boy ND IA G array such thatV A L (:I) consists of non-zero elem ents on diagonal ID IA G (I) of A. D iagonals in the low er triangularpart of A are padded from the top, and those in the upper triangularpart are padded from the bottom.

LDA leading dim ension ofVAL,m ustbe GE.M \(\mathbb{N} \mathbb{M}, K\) )
ID IA G () integer array of length ND IA G consisting of the comesponding diagonal offsets of the non-zero diagonals ofA in VA L. Low ertriangular diagonals have negative offsets, them ain diagonal has offset 0 , and upper triangular diagonals have positive offset.

ND IA G num berof non-zero diagonals in A.
B 0 rectangular array with first dim ension LD B .
LD B leading dim ension ofB

BETA Scalarparam eter
C 0 rectangular array w th first dim ension LD C .

LD C leading dim ension ofC
W ORK () scratch array of length LW ORK.W ORK is not referenced in the current version.

LW ORK length ofW ORK array. LW ORK is not referenced in the current version.

\section*{SEE ALSO}

N IST FO RTRAN Sparse B las U ser's G uide available at:
http://m ath nistgov/m csd/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib.org/utk/papers/sparse.ps

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

zdiasm -diagonal form at triangular solve

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZDIASM(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,}

* VAL,LDA,\mathbb{D IAG,NDIAG,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,M,N,UNITD,DESCRA (5),LDA,ND IAG,}
* LDB,LDC,LWORK
\mathbb{NTEGER IDIAG NDIAG)}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX DV M ),VAL (LDA,ND IAG),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
SUBROUTINE ZD IA SM_64 (TRANSA, M, N, UN ITD, DV,ALPHA, DESCRA,
* VAL,LDA, \(\mathbb{D} \mathbb{I} G, N D \mathbb{A} G\),
* B,LDB,BETA, C,LDC,WORK,LWORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, M,N,UNITD,DESCRA (5), LDA, ND IA G,
* LDB,LDC,LWORK
\(\mathbb{N} T E G E R * 8 \mathbb{D} \mathbb{I} G \mathbb{N} \mathbb{I A}\) )
DOUBLE COM PLEX ALPHA,BETA
D OUBLE COM PLEX DV M),VAL (LDA,ND IAG), B (LDB,*), C (LDC, \(\left.{ }^{\star}\right), W\) ORK (LW ORK)

\section*{F95 INTERFACE}
```

SUBROUTINEDIASM (TRANSA,M, N ],UNITD,DV,A LPHA,DESCRA,VAL,

* [LDA],\mathbb{D IAG,NDIAG,B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])}
INTEGER TRANSA,M,ND IAG

```

```

DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D IM ENSION (:) :: DV
DOUBLE COM PLEX,D IM ENSION (:,:) :: VAL,B,C

```
SUBROUTINEDIASM_64(TRANSA,M, N ],UNITD,DV,ALPHA,DESCRA,VAL,
* [LDA], \(\mathbb{D} \mathbb{I A G}, N D \mathbb{I A G}, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N} T E G E R * 8\) TRANSA, M,NDIAG
\(\mathbb{N} T E G E R * 8, D \mathbb{M}\) ENSION (:) :: DESCRA, \(\mathbb{D} \mathbb{I A}\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D \(\mathbb{M}\) ENSION (:) :: DV
DOUBLE COM PLEX,D \(\mathbb{M}\) ENSION (:, :) :: VAL,B,C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { op(A)B + BETA C } \quad C<-A L P H A D \text { op(A)B+BETA C } \\
& C<-A L P H A \text { op(A)D B + BETA } C
\end{aligned}
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense \(m\) atrices, \(D\) is a diagonalscaling \(m\) atrix, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix represented in diagonal form at and op (A ) is one of
\(\mathrm{op}(\mathrm{A})=\operatorname{inv}(\mathrm{A})\) or op (A) \(=\operatorname{inv}(A)\) or op (A) \()=\operatorname{inv}\left(\right.\) (oonjg ( \(\left.\left.A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate w th transpose m atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in \(m\) atrix A

N \(\quad N\) um berof colum ns in matrix C
UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 :Scale on left (row scaling)
3 : Scale on right (colum \(n\) scaling)
4 :A utom atic row scaling (see section NOTES for furtherdetails)

DV 0 A ray of length M containing the diagonalentries of the scaling diagonalm atrix D .

A LPH A Scalarparam eter
DESCRA 0 D escriptor argum ent. Five elem ent integer aray
DESCRA (1) m atrix structure
0 : general

1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm itian ( \(A=C O N J(A)\) )
3 :Triangular
4 : Skew (A nti)-Symm etric ( \((A=-A\) )
5 :D iagonal
6 : Skew Herm titian ( \(A=-\operatorname{CON}\) J ( \(A\) ) )
N ote: For the routine, only D ESCRA (1)=3 is supported.

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 :non-unit
1 :unit
DESCRA (4) A ray base \(\mathbb{N O T} \mathbb{M}\) PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? (NOT \(\mathbb{M}\) PLEM ENTED)
0 : unknown
1 : no repeated indices
VAL () tw o-dim ensionalLD A boy-ND IA G aray such thatVAL(:,I) consists of non-zero elem ents on diagonal ID IA G (I) of A. D iagonals in the low er triangularpart of A are padded from the top, and those in the upper triangularpart are padded from the bottom.

LDA leading dim ension ofVAL, m ustbe GE.M \(\mathbb{N}(M, K)\)

ID IA G () integer anay of length ND IA G consisting of the corresponding diagonaloffsets of the non-zero diagonals ofA in VAL. Low ertriangular diagonals have negative offsets, them ain diagonalhas offset 0 , and uppertriangular diagonals have positive offset. Elem ents of \(\mathbb{D}\) IA G ofM UST be sorted in increasing order.

ND IA G num berofnon-zero diagonals in A.

B 0 rectangular array w ith firstdim ension LD B .

LD B leading dim ension of \(B\)

BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch amay of length LW ORK. On exit, if LW ORK = -1,W ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK aray. LW ORK should be at leastM.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=M\) *N_CPU \(S\) where \(N\) _CPU \(S\) is the \(m\) axim um num berof processors available to the program .

IfLW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK anray, retums this value as the firstentry of theW ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov fn csd/Staff/k Rem ington/tspoblas/
"D ocum ent for the Basic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. No test for singularity ornear-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.
2. If U N ITD \(=4\), the routine scales the row sofA such that their 2 -norm are one. The scaling \(m\) ay im prove the accuracy of the com puted solution. C orresponding entries ofV A L are changed only in the particular case. On retum DV m atrix stored as a vector contains the diagonalm atrix by w hich the row s have been scaled. U N ITD \(=2\) should be used for the next calls to the routine w ith overw ritten VA L and DV .

WORK ( 1 )=0 on retum if the scaling has been com pleted successfully, otherw ise \(W O R K(1)=-i w\) here \(i\) is the row num berw hich 2 -norm is exactly zero.
3. If \(D E S C R A(3)=1\) and \(U N\) ITD \(<4\), the unitdiagonalelem ents m ightorm ightnotbe referenced in the D IA representation of a sparse m atrix. They are notused anyw ay in these cases. ButifUN ITD = 4, the unit diagonalelem ents M U ST be referenced in the \(D \mathbb{I A}\) representation.
4.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse
m atrix \(A\) is used. H ow ever DESCRA (1) m ustbe equal to 3 in this case.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zdotc -com pute the dotproduct oftw o vectors con \(\dot{g}(x)\) and \(y\).

\section*{SYNOPSIS}

DOUBLE COM PLEX FUNCTION ZDOTC \(\mathbb{N}, \mathrm{X}, \mathbb{N} C X, Y, \mathbb{N C Y})\)
D OUBLE COM PLEX X (*), Y (*)
\(\mathbb{N} T E G E R N, \mathbb{N C X}, \mathbb{N} C Y\)
DOUBLE COM PLEX FUNCTION ZDOTC_64 \(\mathbb{N}, \mathrm{X}, \mathbb{N} C X, Y, \mathbb{N} C Y\) )
DOUBLE COM PLEX X (*) , Y (*)
\(\mathbb{N} T E G E R * 8 N, \mathbb{N} C X, \mathbb{N} C Y\)
F95 INTERFACE
COM PLEX (8) FUNCTION DOTC ( \(\mathbb{N}], \mathrm{X},[\mathbb{N} C X], Y,[\mathbb{N C Y}])\)
COMPLEX (8), D \(\mathbb{I M} \operatorname{ENSION}(:):\) : \(\mathrm{X}, \mathrm{Y}\)
\(\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N C Y}\)
COM PLEX (8) FUNCTION DOTC_64 ( \(\mathbb{N}], X,[\mathbb{N} C X], Y,[\mathbb{N C Y}])\)
COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::X,Y
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{I N C X}, \mathbb{N} C Y\)

\section*{C INTERFACE}
\#include <sunperfh>
doublecom plex zdotc (intn, doublecom plex *x, int incx, doublecom plex *y, int incy);
doublecom plex zdotc_64 (long n, doublecom plex *x, long incx, doublecom plex *y, long incy);

\section*{PURPOSE}
zdotc com pute the dot product of con \(g(x)\) and \(y\) where \(x\) and y are \(n\)-vectors.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. IfN is notpositive then the function retums the value 0.0. U nchanged on exit.
X (input)
OfD \(\mathbb{M}\) ENSION at least ( \(1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C X)\)
). On entry, the increm ented amay \(X\) m ust contain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X\) m ustnotbe zero. U nchanged on exit.

Y (input)
OfD \(\mathbb{M}\) ENSION at least ( \(1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C Y)\) ). On entry, the increm ented amay \(Y\) m ust contain the vectory. U nchanged on exit.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y . \mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS

\section*{NAME}
zdotci-C om pute the com plex conjugated indexed dotproduct.

\section*{SYNOPSIS}

DOUBLE COM PLEX FUNCTION ZDOTCINZ,X, \(\mathbb{N} D X, Y)\)
DOUBLE COM PLEX X (*), Y (*)
\(\mathbb{N}\) TEGER NZ
\(\mathbb{N} T E G E R \mathbb{N} D X(*)\)
DOUBLE COM PLEX FUNCTION ZDOTCI_64 \(\mathbb{N} Z, X, \mathbb{N} D X, Y)\)
DOUBLE COM PLEX X (*), Y (*)
\(\mathbb{N} T E G E R * 8 N Z\)
\(\mathbb{I N} T E G E R * 8 \mathbb{N} D X(*)\)
F95 \(\mathbb{I N}\) TERFACE
DOUBLE COM PLEX FUNCTION DOTCI(NZ],X, \(\mathbb{N} D X, Y)\)
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::X,Y
\(\mathbb{I N T E G E R}::\) NZ
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{N D X}\)
DOUBLE COM PLEX FUNCTION DOTCI_64 ( \(\mathbb{N} Z], X, \mathbb{N} D X, Y\) )
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::X,Y
\(\mathbb{N}\) TEGER (8) ::NZ
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{N D X}\)

\section*{PURPOSE}
com plex vectory in fullstorage form .
```

dot $=0$
do $i=1, n$
dot $=\operatorname{dot}+\operatorname{con}$ gg $(x$ (i)) * $y$ (indx (i))
enddo

```

\section*{ARGUMENTS}

N Z (input)
\(N\) um ber of elem ents in the com pressed form .
U nchanged on exit.
\(X\) (input)
V ector in com pressed form . U nchanged on exit.
\(\mathbb{N} D X\) (input)
\(V\) ector containing the indioes of the com pressed form. It is assum ed that the elem ents in \(\mathbb{N} D \mathrm{X}\) are distinctand greater than zero. U nchanged on exit.

Y (input)
V ector in fullstorage form. O nly the elem ents comesponding to the indices in \(\mathbb{N}\) D X w illbe accessed.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zdotu - com pute the dotproductoftw o vectors x and y .

\section*{SYNOPSIS}
```

DOUBLE COM PLEX FUNCTION ZDOTU N,X,\mathbb{NCX,Y,INCY)}
DOUBLE COM PLEX X (*),Y (*)
\mathbb{NTEGERN,}\mathbb{NCX,\mathbb{NCY}}\mathbf{}\mathrm{ \}
DOUBLE COM PLEX FUNCTION ZDOTU_64 N,X,\mathbb{NCX,Y,\mathbb{NCY)}}\mathbf{N}\mathrm{ ( }
DOUBLE COM PLEX X (*),Y (*)
INTEGER*8N,\mathbb{NCX,INCY}
F95 INTERFACE

```

```

    COM PLEX (8),D IM ENSION (:) ::X,Y
    \mathbb{NTEGER ::N,\mathbb{NCX,}\mathbb{NCY}}\mathbf{N}=\mp@code{N}
    ```

```

    COM PLEX (8),D IM ENSION (:) ::X,Y
    ```

```

C INTERFACE
\#include <sunperfh>
doublecom plex zdotu(intn, doublecom plex *x, int incx, doub-
lecom plex *y, int incy);
doublecom plex zdotu_64 (long n, doublecom plex *x, long incx,

```

\section*{PURPOSE}
zdotu com pute the dotproduct of \(x\) and \(y ~ w h e r e ~ x a n d ~ y ~ a r e ~\) n-vectors.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. If N is notpositive then the function retums the value 0.0. U nchanged on exit.
X (input)
ofD \(\mathbb{I M}\) ENSION at least ( \(1+(\mathrm{n}-1) * a b s(\mathbb{N} C X)\)
). On entry, the increm ented array \(X\) m ustcontain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of X. \(\mathbb{N} C X\) m ustnotbe zero. U nchanged on exit.

Y (input)
ofD \(\mathbb{I M}\) ENSION at least ( \(1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C Y)\) ). On entry, the increm ented amay \(Y \mathrm{~m}\) ust contain the vectory. U nchanged on exit.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y . \mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS

\section*{NAME}
zdotui-C om pute the com plex unconjugated indexed dot product.

\section*{SYNOPSIS}
```

DOUBLE COM PLEX FUNCTION CDOTCINZ,X,INDX,Y)
DOUBLE COM PLEX X (*),Y (*)
INTEGER NZ
INTEGER \mathbb{NDX (*)}
DOUBLE COM PLEX FUNCTION CDOTCI_64 NZ,X,INDX,Y)
DOUBLE COM PLEX X (*),Y (*)
INTEGER*8NZ
INTEGER*8 \mathbb{NDX (*)}
F95 INTERFACE
DOUBLE COM PLEX FUNCTION DOTCI(NZ],X,\mathbb{NDX,Y)}
COM PLEX (8),D IM ENSION (:) ::X,Y
\mathbb{NTEGER ::NZ}
NNTEGER,D IM ENSION (:) ::\mathbb{NDX}
DOUBLE COM PLEX FUNCTION DOTCI_64(NZ],X,\mathbb{NDX,Y)}
COM PLEX (8),D IM ENSION (:) ::X,Y
INTEGER (8)::N Z
\mathbb{NTEGER (8),D IM ENSION (:)::\mathbb{NDX}}\mathbf{N}=\mp@code{N}

```

\section*{PURPOSE}
of a com plex sparse vectorx stored in com pressed form w ith a com plex vectory in fullstorage form .
```

dot=0
do i= 1,n
dot= dot+x (i) * y (indx (i))
enddo

```

\section*{ARGUMENTS}

NZ (input)
N um ber of elem ents in the com pressed form .
U nchanged on exit.

X (input)
V ector in com pressed form. U nchanged on exit.
\(\mathbb{I N D X}\) (input)
V ector containing the indioes of the com pressed
form. It is assum ed that the elem ents in \(\mathbb{N} D X\) are
distinctand greater than zero. U nchanged on exit.
\(Y\) (input)
V ector in fullstorage form. O nly the elem ents corresponding to the indices in \(\mathbb{I N D ~ X ~ w ~ i l l b e ~}\) accessed.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

zdrot-A pply a plane rotation.

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZDROT N,CX, INCX,CY,\mathbb{NCY,C,S)}}\mathbf{N}=()
DOUBLE PRECISION C,S
DOUBLE COM PLEX CX (*),CY (*)

```


```

DOUBLE PRECISION C,S
DOUBLE COM PLEX CX (*),CY (*)
INTEGER*8N,\mathbb{NCX,INCY}
F95 INTERFACE
SUBROUT\mathbb{NE ROT (N ],CX,[\mathbb{NCX ],CY , [NNCY ],C ,S)}}\mathbf{~}\mathrm{ ( }
REAL (8) ::C,S
COM PLEX (8),D IM ENSION (:) ::CX,CY
\mathbb{NTEGER ::N,\mathbb{NCX,INCY}}\mathbf{N}={

```

```

    REAL (8) ::C,S
    COM PLEX (8),D IM ENSION (:) ::CX,CY
    \mathbb{NTEGER (8)::N,INCX,INCY}
    ```

\section*{C INTERFACE}
```

\#include <sunperfh>

```
void zdrot(intn, doublecom plex *cx, int incx, doublecom plex
*cy, int incy, double c, double s);
void zdrot 64 (long n, doublecom plex *cx, long incx, doublecom plex *cy, long incy, double c, double s);

\section*{PURPOSE}
zdrotA pply a plane rotation, w here the cos and sin (c and s) are real and the vectors \(x\) and \(y\) are com plex.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N m ustbe at leastone for the subroutine to have any visible effect. U nchanged on exit.

CX (input)
Before entry, the increm ented array CX must contain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elements of \(C X . \mathbb{N} C X\) must not be zero. U nchanged on exit.

CY (output)
On entry, the increm ented array CY m ustcontain the vector \(y\). On exit, \(C Y\) is overw rilten by the updated vectory.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elements of CY. \(\mathbb{N C Y} \mathrm{must}\) not be zero. U nchanged on exit.

C (input)
O n entry, the cosine. U nchanged on exit.
\(S\) (input)
On entry, the sin. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

zdscal-C om pute y := alpha * y

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZDSCAL N,ALPHA,Y,INCY)}
DOUBLE COM PLEX Y (*)

```

```

DOUBLE PRECISION ALPHA
SUBROUTINE ZDSCAL_64 N,A LPHA,Y,\mathbb{N CY)}
DOUBLE COM PLEX Y (*)
INTEGER*8N,\mathbb{NCY}
DOUBLE PRECISION ALPHA
F95 INTERFACE
SUBROUT\mathbb{NE SCAL (N ],ALPHA,Y,[\mathbb{NCY ])}}\mathbf{~}\mathrm{ , (N)}
COM PLEX (8),D IM ENSION (:) ::Y
\mathbb{NTEGER ::N,\mathbb{NCY}}\mathbf{N}=\mp@code{N}
REAL (8) ::ALPHA
SU BROUTINE SCAL_64 (N ],ALPHA,Y,[\mathbb{N CY ])}
COM PLEX (8),D IM ENSION (:) ::Y
\mathbb{NTEGER (8) ::N,\mathbb{NCY}}\mathbf{}=1
REAL (8) ::ALPHA

```

\section*{C INTERFACE}
```

\#include <sunperfh>

```
void zdscal(int \(n\), double alpha, doublecom plex *y, int incy);
void zdscal_64 (long n, double alpha, doublecom plex *y, long incy);

\section*{PURPOSE}
zdscalC om pute y := alpha * y w here alpha is a scalar and y is an \(n\)-vector.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

Y (input/output)
( \(1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C Y)\) ). On entry, the increm ented array \(Y\) m ust contain the vectory. On exit, \(Y\) is overw ritten by the updated vectory.
\(\mathbb{N} C Y\) (input)
O n entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of Y. \(\mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

zelm m -Ellpack form atm atrix-m atrix m ultiply

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZELLMM (TRANSA,M,N,K,ALPHA,DESCRA,}

* VAL,INDX,LDA,MAXNZ,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER TRANSA,M,N,K,DESCRA (5),LDA,MAXNZ,
* LDB,LDC,LWORK
INTEGER INDX(LDA,MAXNZ)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (LDA,M AXNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
SUBROUTINE ZELLMM_64(TRANSA, M,N,K,ALPHA,DESCRA,
* VAL, \(\mathbb{N} D \mathrm{X}, \mathrm{LDA}, \mathrm{MAXNZ}\),
* B,LDB,BETA,C,LDC,WORK,LWORK)
\(\mathbb{N}\) TEGER*8 TRANSA, M, N, K, DESCRA (5), LDA, MAXNZ,
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R * 8 \mathbb{N} D X(L D A, M A X N Z)\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (LDA, MAXNZ), B (LDB,*), C (LDC, \(\left.{ }^{\star}\right), W\) ORK (LW ORK)

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE ELLMM (TRANSA,M, N ],K,ALPHA,DESCRA,VAL, INDX,}

* [LDA],MAXNZ,B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])
\mathbb{NTEGER TRANSA,M,K,MAXNZ}
INTEGER,D\mathbb{M ENSION (:) :: DESCRA}
INTEGER,D\mathbb{M ENSION (:,:):: \mathbb{NDX}}\mathbf{N}=\mp@code{N}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D IM ENSION (:,:) :: VAL,B,C

```
SUBROUTINE ELLMM _64(TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N} D X\),

\section*{DESCRIPTION}
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, A is a m atrix represented in Ellpack form at form at and op (A) is one of
\(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{conj}\left(A^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate w ith m atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate w ith the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M \(\quad\) um berof row \(s\) in \(m\) atrix A

N \(\quad\) Num berof colum ns in \(m\) atrix \(C\)

K \(\quad N\) um berof \(c o l u m n s\) in \(m\) atrix A

A LPH A Scalarparam eter
DESCRA () D escriptor argum ent. Five elem ent integer array
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(A=A\) )
2 : Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) )
3 :Triangular
4 : Skew (Anti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJ}\) ( A ))
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonaltype
0 : non-unit
1 :unit

DESCRA (4) A ray base NOT IM PLEM ENTED)
0 : C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? (NOT IM PLEM ENTED)
0 : unknown
1 : no repeated indices
VAL ( 0 tw o-dim ensionallD A -by M A XNZ array such thatVA L ( \(\mathrm{I}, \mathrm{:}\) ) consists of non-zero elem ents in row IofA, padded by zero values if the row contains less than M AXN Z .
\(\mathbb{I N D X} 0 \quad\) tw o-dim ensional integer LD A by -M A XN Z aray such \(\mathbb{N} D \mathrm{X}\) ( \(I\), :) consists of the colum n indices of the nonzero elem ents in row \(I\), padded by the integer value I if the num berof nonzeros is less than M AXNZ.

LD A leading dim ension ofVAL and \(\mathbb{N D} X\).

MAXNZ max num berofnonzeros elem ents per row .
B 0 rectangular array w th first dim ension LD B.
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w th first dim ension LD C .

LD C leading dim ension of \(C\)
W ORK ( scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK anay. LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

zellsm -Ellpack form attriangular solve

```

\section*{SYNOPSIS}
```

SUBROUTINE ZELLSM (TRANSA, M,N,UNITD,DV,ALPHA,DESCRA,

* VAL, $\mathbb{N} D X, L D A, M A X N Z$,
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
$\mathbb{N} T E G E R$ TRANSA, M,N,UNITD,DESCRA (5), LDA, MAXNZ,
* LDB,LDC,LW ORK
$\mathbb{N} T E G E R \quad \mathbb{N} D X(L D A, M A X N Z)$
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX DV M),VAL (LDA,MAXNZ),B( $\mathbb{L D} B, \star), C(\mathbb{C D}, \star), W$ ORK (LW ORK)
SUBROUTINE ZELLSM_64(TRANSA, M,N,UNITD,DV,ALPHA,DESCRA,
* VAL, $\mathbb{N} D \mathrm{X}, \mathrm{LDA}, \mathrm{MAXNZ}$,
* B,LDB,BETA,C,LDC,WORK,LWORK)
$\mathbb{N} T E G E R * 8$ TRANSA, M,N,UNITD,DESCRA (5), LDA, MAXNZ,
* LDB,LDC,LW ORK
$\mathbb{N} T E G E R * 8 \mathbb{N} D X(L D A, M A X N Z)$
DOUBLE COM PLEX ALPHA, BETA
DOUBLE COM PLEX DV M),VAL (LDA, MAXNZ), B (LDB,*), C (LDC,*),WORK (LW ORK)

```

\section*{F95 INTERFACE}

SUBROUTINE ELLSM (TRANSA,M, \(\mathbb{N}], U N \mathbb{I T}, D V, A L P H A, D E S C R A, V A L\), * \(\mathbb{N D D},[\) [DA \(], M A X N Z, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{I N T E G E R}\) TRANSA, M, MAXNZ
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: DESCRA
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENS} \operatorname{ION}(:,:):: \mathbb{N D X}\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D \(\mathbb{I M}\) ENSION (:) :: DV
DOUBLE COM PLEX,D \(\mathbb{I}\) ENSION (: :) :: VAL,B,C

SUBROUTINE ELLSM _64(TRANSA, M, \(\mathbb{N}], U N \mathbb{T} D, D V, A L P H A, D E S C R A, V A L\), * \(\mathbb{N} D X,[L D A], M A X N Z, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{I N T E G E R * 8 ~ T R A N S A , ~ M , ~ M A X N Z ~}\)
\(\mathbb{N} T E G E R * 8, D \mathbb{M}\) ENSION (:) :: DESCRA
\(\mathbb{N} T E G E R * 8, D \mathbb{M}\) ENSION (: : : : : \(\mathbb{N} D X\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D \(\mathbb{M}\) ENSION (:) :: DV
DOUBLE COMPLEX,D \(\mathbb{M}\) ENSION (: : : :: VAL,B,C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { op(A)B + BETA C } \quad C<-A L P H A D \text { op (A) B + BETA C } \\
& C<-A L P H A \text { op(A)D B + BETA C }
\end{aligned}
\]
where A LPHA and BETA are scalar, C and B are m by n dense \(m\) atrices, \(D\) is a diagonalscaling \(m\) atrix, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix represented in Ellpack form at and op (A) is one of \(\mathrm{op}(\mathrm{A})=\operatorname{inv}(\mathrm{A})\) or \(\mathrm{op}(\mathrm{A})=\operatorname{inv}\left(\mathrm{A}^{\prime}\right)\) or \(\mathrm{op}(\mathrm{A})=\operatorname{inv}\left(\mathrm{conjg}\left(\mathrm{A}^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix 0 : operate with m atrix 1 : operate \(w\) th transpose m atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M \(\quad\) um berof row \(s\) in \(m\) atrix A
N \(\quad\) Num berof colum ns in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity \(m\) atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)
4 :A utom atic row scaling (see section N O TES for furtherdetails)

DV 0 A ray of length M containing the diagonalentries of the scaling diagonalm atrix D .

A LPH A Scalarparam eter
DESCRA 0 D escriptor argum ent. Five elem ent integer aray DESCRA (1) m atrix structure
```

    0 :general
    1 : symm etric ( }A=A\mathrm{ ) )
    2:H erm Itian (A = CON JG (A ))
    3:Triangular
    4 :Skew (A nti)-Symm etric (A=-A )
    5 :D iagonal
    6:Skew Herm titian (A= CON JG (A ) )
    N ote:For the routine, only D ESCRA (1)=3 is supported.
    D ESCRA (2) upper/low er triangular indicator
        1 : low er
        2 :upper
    DESCRA (3) m ain diagonaltype
        0:non-unit
        1 :unit
    DESCRA (4) A ray base NOT IM PLEM ENTED )
        0 : C C ++ com patible
        1 :Fortran com patible
    DESCRA (5) repeated indices? (NOT IM PLEM ENTED)
        0 :unknow n
        1: no repeated indices
    VAL () tw o-dim ensionalLD A foy M A X N Z array such thatV A L (I,:)
consists of non-zero elem ents in row IofA, padded by
zero values if the row contains less than M AXN Z .
INDX () tw o-dim ensionalintegerLD A boy-M A XN Z array such
\mathbb{N D X (I,:) consists of the colum n indiges of the}
nonzero elem ents in row I, padded by the integer
value I if the num berofnonzeros is less than M A XN Z .
The colum n indices M U ST be sorted in increasing order
foreach row .
LDA leading dim ension ofV A L and \mathbb{ND X .}
M A X N Z m ax num ber ofnonzeros elem ents per row .
B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension ofC

```
    W ORK () scratch amay of length LW ORK.
        On exit, ifLW ORK \(=-1, W\) ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK array. LW ORK should be at leastM .
Forgood penform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=\mathrm{M}\) *N_CPUS where N_CPUS is the m axim um num berof processors available to the program .

IfLW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of theW ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
htep://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such tests \(m\) ustbe perform ed before calling this routine.
2. IfUN ITD \(=4\), the routine scales the row s of \(A\) such that their 2 -nom s are one. The scaling \(m\) ay im prove the accuracy of the com puted solution. C orresponding entries ofV A L are changed only in the particular case. On retum DV m atrix stored as a vector contains the diagonalm atrix by w hich the row s have been scaled. U N ITD = 2 should be used for the next calls to the routine w ith overw ritten VAL and DV.

WORK ( 1 )=0 on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here i is the row num berw hich 2 -norm is exactly zero.
3. If \(\operatorname{DESCRA}(3)=1\) and U N ITD < 4, the unitdiagonalelem ents \(m\) ightorm ightnotbe referenced in the ELL representation of a sparse m atrix. They are notused anyw ay in these cases.

ButifUN ITD \(=4\), the unit diagonalelem ents M U ST be referenced in the ELL representation.
4.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse m atrix A is used. H ow ever DESCRA (1) m ustbe equal to 3 in this case.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
zfft2b - com pute a periodic sequence from its Fourier coefficients. The FFT operations are unnorm alized, so a call of ZFFT2F follow ed by a callof ZFFT2B w illm ultiply the input sequence by \(M\) * N .

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZFFT2B M,N,A,LDA,W ORK,LW ORK)}
D OUBLE COM PLEX A (LDA,*)
INTEGERM,N,LDA,LW ORK
DOUBLE PRECISION W ORK (*)
SUBROUTINE ZFFT2B_64M,N,A,LDA,W ORK,LW ORK)
DOUBLE COM PLEX A (LDA,*)
INTEGER*8M,N,LDA,LW ORK
DOUBLE PRECISION W ORK (*)

```
F95 INTERFACE
    SUBROUTINE FFT2B (M ], \(\mathbb{N}], A,[L D A], W\) ORK,LW ORK)
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N} T E G E R:: M, N, L D A, L W\) ORK
    REAL (8),D \(\mathbb{I}\) ENSION (:) ::W ORK
    SU BROUTINE FFT2B_64 (M ], \(\mathbb{N}], A,[L D A], W\) ORK,LW ORK)
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N}\) TEGER ( 8 ) :: M , N,LDA ,LW ORK
    REAL (8),D IM ENSION (:) ::W ORK
\#include <sunperfh>
void zfft2b (intm, intn, doublecom plex *a, intlda, double *W ork, intlw ork);
void zffi2b_64 (long m, long n, doublecom plex *a, long lda, double *W ork, long lw ork);

\section*{ARGUMENTS}
\(M\) (input) N um ber of row s to be transform ed. These subroutines are \(m\) ost efficientw hen \(M\) is a product of sm allprim es. \(\mathrm{M}>=0\).
N (input) N um ber of colum ns to be transform ed. These subroutines are \(m\) ostefficientw hen \(N\) is a product of sm allprim es. \(\mathrm{N}>=0\).

A (input/output)
O n entry, a tw o-dim ensionalanray A \((\mathrm{M}, \mathrm{N})\) that contains the sequences to be transform ed.

LD A (input)
Leading dim ension of the array containing the data to be transform ed. LD A >=M .

W ORK (input)
On entry, an aray w ith dimension of at least LW ORK. W ORK m usthave been initialized by ZFFT2I.

LW ORK (input)
The dim ension of the array W ORK. LW ORK >= (4* M \(+\mathrm{N})+30\) )

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
zfft2f-com pute the Fourier coefficients of a periodic
sequence. The FFT operations are unnorm alized, so a callof ZFFT2F follow ed by a callof ZFFT2B w illm ultiply the input sequence by \(M * N\).

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZFFT2F M,N,A,LDA,W ORK,LW ORK)}
D OUBLE COM PLEX A (LDA,*)
INTEGERM,N,LDA,LW ORK
DOUBLE PRECISION W ORK (*)
SUBROUT\mathbb{NE ZFFT2F_64M,N,A,LDA,W ORK,LW ORK)}
D OUBLE COM PLEX A (LDA,*)
INTEGER*8M,N,LDA,LW ORK
DOUBLE PRECISION W ORK (*)

```
F95 INTERFACE
    SU BROUTINE FFT2F (M ], \(\mathbb{N}], A,[L D A], W\) ORK,LW ORK)
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N} T E G E R:: M, N, L D A, L W\) ORK
    REAL (8),D \(\mathbb{I}\) ENSION (:) ::W ORK
    SU BROUTINE FFT2F_64 (M ], \(\mathbb{N}], A,[L D A], W\) ORK,LW ORK)
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N}\) TEGER ( 8 ) :: M , N,LDA ,LW ORK
    REAL (8),D IM ENSION (:) ::W ORK
\#include <sunperfh>
void zfft2f(intm, intn, doublecom plex *a, intlda, double *W ork, intlw ork);
void zfft2f_64 (long m, long n, doublecom plex *a, long lda, double *W Ork, long lw ork);

\section*{ARGUMENTS}
\(M\) (input) N um ber of row s to be transform ed. These subroutines are \(m\) ost efficientw hen \(M\) is a product of sm allprim es. \(\mathrm{M}>=0\).
N (input) N um ber of colum ns to be transform ed. These subroutines are \(m\) ostefficientw hen \(N\) is a product of sm allprim es. \(\mathrm{N}>=0\).

A (input/output)
O n entry, a tw o-dim ensional array \(A(M, N)\) thatcontains the sequences to be transform ed.

LD A (input)
Leading dim ension of the array containing the data to be transform ed. LD A \(>=M\).

W ORK (input)
On input, w orkspace W ORK m usthave been initialized by ZFFT 2 I.

LW ORK (input)
The dim ension of the array W ORK. LW ORK >= (4* M \(+\mathrm{N})+30\) )

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
zfft2i-initialize the array W SAVE, which is used in both the forw ard and backw ard transform s.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZFFT2IM ,N,W ORK)}

```
\(\mathbb{I N}\) TEGERM,N
DOUBLE PRECISION W ORK (*)
SUBROUTINE ZFFT2I_64 M ,N,W ORK)
\(\mathbb{N}\) TEGER*8 M , N
DOUBLE PRECISION WORK (*)
F95 INTERFACE
    SU BROUTINE ZFFT2IM, \(N\),W ORK)
\(\mathbb{N} T E G E R:: M, N\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
SU BROUTINE ZFFT2I_64 (M,N W ORK)
\(\mathbb{N} T E G E R(8):: M, N\)
REAL (8),D IM ENSION (:) ::W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zfft2i(intm , intn, double *w ork);
void zfft2i_64 (long m, long n, double *w ork);

\section*{ARGUMENTS}

M (input) N um ber of row s to be transform ed. \(\mathrm{M}>=0\).

N (input) N um ber of colum ns to be transform ed. \(\mathrm{N}>=0\).

W ORK (input/output)
On entry, an array of dim ension ( 4 * \(M+N\) ) + 30) or greater. ZFFT2I needs to be called only once to initialize amay \(W\) ORK before calling ZFFT2F and/or ZFFT2B if \(\mathrm{M}, \mathrm{N}\) and W ORK rem ain unchanged betw een these calls. Thus, subsequent transform s or inverse transform \(s\) of sam e size can be obtained faster than the first since they do not require initialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
zffl3b - com pute a periodic sequence from its Fourier coefficients. The FFT operations are unnorm alized, so a callof ZFFT 3F follow ed by a callof ZFFT 3B w illm ultiply the input sequence by \(M * N * K\).

\section*{SYNOPSIS}
```

SUBROUTINE ZFFT3B M ,N,K,A,LDA,LD 2A,W ORK,LW ORK)
DOUBLE COM PLEX A (LDA,LD 2A,*)
INTEGERM,N,K,LDA,LD 2A,LW ORK
DOUBLE PRECISION W ORK (*)
SUBROUT\mathbb{NE ZFFT3B_64M,N,K,A,LDA,LD 2A,W ORK,LW ORK)}
DOU BLE COM PLEX A (LDA,LD 2A,*)
INTEGER*8M,N,K,LDA,LD 2A,LW ORK
DOUBLE PRECISION W ORK (*)
F95 INTERFACE
SUBROUT\mathbb{NE FFT3B (M ], N ],[K ],A,[LDA ],LD 2A,W ORK,LW ORK)}
COM PLEX (8),D IM ENSION (:r:%)::A
INTEGER ::M,N,K,LDA,LD 2A,LW ORK
REAL (8),D IM ENSION (:) ::W ORK

```

```

    COM PLEX (8),D IM ENSION (:,:,:) ::A
    INTEGER (8)::M ,N,K,LDA,LD 2A,LW ORK
    REAL (8),D IM ENSION (:) ::W ORK
    ```
void zfft3b (intm , intn, int k, doublecom plex *a, int lda, intld2a, double *w ork, int lw ork);
void zffl3b_64 (long m, long n, long k, doublecom plex *a, long lda, long ld2a, double *w ork, long lw ork);

\section*{ARGUMENTS}

M (input) \(N\) um ber of row s to be transform ed. These subroutines are most efficientw hen \(M\) is a productof sm allprim es. M >=0.
N (input) \(N\) um ber of \(\propto 0\) lum ns to be transform ed. These subroutines are \(m\) ostefficientw hen \(N\) is a product of sm allprim es. \(\mathrm{N}>=0\).

K (input) \(N\) um ber of planes to be transform ed. These subroutines are m ost efficientw hen K is a product of sm allprim es. \(K>=0\).

A (input/output)
On entry, a three-dim ensionalamay A (LD A , LD 2A , K ) that contains the sequences to be transform ed.

LD A (input)
Leading dim ension of the amay containing the data to be transform ed. LD A >=M.

LD 2A (input)
Second dim ension of the array containing the data to be transform ed. LD 2A >= N .

W ORK (input)
On input, w orkspace W ORK musthave been initialized by ZFFT 3I.

LW ORK (input)
The dim ension of the array \(W\) ORK. LW ORK >= \((4 * M+\) \(\mathrm{N}+\mathrm{K})+45\) ).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
zfft3f-com pute the Fourier coefficients of a periodic
sequence. The FFT operations are unnorm alized, so a callof
ZFFT 3F follow ed by a callof ZFFT 3B w illm ultiply the input sequence by M * N *K.

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZFFT3F M ,N,K,A,LDA,LD 2A,W ORK,LW ORK)}
DOUBLE COM PLEX A (LDA,LD 2A,*)
INTEGERM,N,K,LDA,LD 2A,LW ORK
DOUBLE PRECISION W ORK (*)
SUBROUTINE ZFFT3F_64M,N,K,A,LDA,LD 2A,W ORK,LW ORK)
DOUBLE COM PLEX A (LDA,LD 2A,*)
INTEGER*8M,N,K,LDA,LD 2A,LW ORK
DOUBLE PRECISION W ORK (*)

```
F95 INTERFACE
    SU BROUTINE FFT3F ( \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], L D 2 A, W\) ORK, LW ORK )
    COM PLEX (8), D IM ENSION (:,:r:) ::A
    \(\mathbb{N} T E G E R:: M, N, K, L D A, L D 2 A, L W\) ORK
    REAL (8),D IM ENSION (:) ::W ORK
    SU BROUTINE FFT3F_64 ( \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], L D 2 A, W\) ORK,LW ORK)
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:,:) ::A
    \(\mathbb{N} T E G E R(8):: M, N, K, L D A, L D 2 A, L W\) ORK
    REAL (8), D IM ENSION (:) ::W ORK
void zffl3f(intm, intn, int k, doublecom plex *a, int lda, int ld2a, double *w ork, int lw ork);
void zffl3£ 64 (long m, long \(n\), long \(k\), doublecom plex *a, long lda, long ld2a, double *w ork, long lw ork);

\section*{ARGUMENTS}

M (input) \(N\) um ber of row s to be transform ed. These subroutines are most efficientw hen \(M\) is a productof sm allprim es. M >=0.
N (input) N um ber of C lum ns to be transform ed. These subroutines are \(m\) ostefficientw hen \(N\) is a product of sm allprim es. \(\mathrm{N}>=0\).
\(K\) (input) \(N\) um ber of planes to be transform ed. These subroutines are m ost efficientw hen K is a product of sm allprim es. \(K>=0\).

A (input/output)
O n entry, a three-dim ensionalarray A \(M, N, K\) ) that contains the sequences to be transform ed.

LDA (input)
Leading dim ension of the amay containing the data to be transform ed. LD A >=M.

LD 2A (input)
Second dim ension of the array containing the data to be transform ed. LD 2A >=N .

W ORK (input)
On input, w orkspace W ORK musthave been initialized by ZFFT 3I.

LW ORK (input)
The dim ension of the anay W ORK. LW ORK >= (4* M + \(\mathrm{N}+\mathrm{K})+45\) ).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
zfflini-initialize the array W SAVE, which is used in both ZFFT 3F and ZFFT 3B.

\section*{SYNOPSIS}
```

SUBROUTINE ZFFT3IM,N,K,W ORK)

```
\(\mathbb{N}\) TEGER \(\mathrm{M}, \mathrm{N}\), K
DOUBLE PRECISION W ORK (*)
SU BROUTINE ZFFT3I_64 M,N,K,WORK)
\(\mathbb{N}\) TEGER*8M,N,K
DOUBLE PRECISION W ORK ( \({ }^{\star}\) )
F95 INTERFACE
SU BROUTINE ZFFT3IM, N, K, W ORK)
\(\mathbb{N} T E G E R:: M, N, K\)
REAL (8), D IM ENSION (:) ::W ORK

SU BROUTINE ZFFT3I_64 M,N,K,WORK)
\(\mathbb{N} T E G E R(8):: M, N, K\)
REAL (8),D IM ENSION (:) ::W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zffl3i(intm , intn, int , double *w ork);
void zfft3i_64 (long m, long n, long k, double *w ork);

\section*{ARGUMENTS}

M (input) N um ber of row s to be transform ed. \(\mathrm{M}>=0\).

N (input) N um ber of colum ns to be transform ed. \(\mathrm{N}>=0\).

K (input) N um ber of planes to be transform ed. \(\mathrm{K}>=0\).

W ORK (input/output)
On entry, an aray ofdim ension (4* \(M+N+K\) ) +
45) or greater. ZFFT3Ineeds to be called only once to initialize array \(W\) ORK before calling ZFFT3F and/or ZFFT3B ifM, N, K andW ORK rem ain unchanged betw een these calls. Thus, subsequent transform s or inverse transform s of sam e size can be obtained fasterthan the first since they do not require initialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
zfftb -com pute a periodic sequence from its Fouriercoefficients. The FFT operations are unnorm alized, so a call of ZFFTF follow ed by a callofZFFTB w ill multiply the input sequence by N .

\section*{SYNOPSIS}
```

SUBROUTINE ZFFTB N,X,W SAVE)
DOUBLE COM PLEX X (*)
INTEGERN
DOUBLE PRECISION W SAVE (*)
SUBROUT\mathbb{NE ZFFTB_64 N,X,W SAVE)}
DOUBLE COM PLEX X (*)
INTEGER*8N
DOUBLE PRECISION W SAVE (*)

```
F95 INTERFACE
    SU BROUTINE FFTB ( \(\mathbb{N}\) ],X,W SAVE)
    COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::X
    \(\mathbb{N} T E G E R:: N\)
    REAL (8), D \(\mathbb{M}\) ENSION (:) ::W SAVE
    SU BROUTINE FFTB_64 (N ],X,W SAVE)
    COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::X
    \(\mathbb{N}\) TEGER (8) ::N
    REAL (8), D IM ENSION (:) ::W SAVE
\#include <sunperfh>
void zfflb (intn, doublecom plex *x, double *w save);
void zfflb_64 (long n, doublecom plex *x, double *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. These subroutines are \(m\) ostefficientw hen \(N\) is a product ofsm allprim es. \(\mathrm{N}>=0\).

X (input) On entry, an amay of length N containing the sequence to be transform ed.

W SAVE (input/output)
O n entry, W SAVE m ustbe an aray ofdim ension (4* \(\mathrm{N}+15\) ) orgreater and m usthave been initialized by ZFFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO

\section*{NAME}
zffld - initialize the trigonom etric w eight and factor tables or com pute the inverse FastFourier T ransform of a double com plex sequence.

\section*{SYNOPSIS}

\(\mathbb{N}\) TEGER \(\mathbb{I O P T}, \mathrm{N}, \mathbb{F} A C(*)\), LW ORK, \(\mathbb{E R R}\)
DOUBLE COM PLEXX (*)
DOUBLE PRECISION SCALE, Y (*), TRIGS (*),WORK (*)
SU BROUTINE ZFFTD_64 (IOPT,N,SCALE,X,Y,TRIGS, FAC,W ORK,LW ORK, ERR)
\(\mathbb{N} T E G E R * 8 \mathbb{I} P T, N, \mathbb{F A C}(*), L W O R K, \mathbb{E R R}\)
DOUBLE PRECISION SCALE, Y (*), TRIGS (*),WORK (*)
DOUBLE COM PLEX X (*)

\section*{F95 INTERFACE}

SU BROUTINE FFT (IOPT,N, [SCALE],X,Y,TRIGS, \(\mathbb{F A C}, \mathrm{W}\) ORK, [LW ORK ], \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER, \(\mathbb{N}\) TENT \((\mathbb{N}):: \mathbb{I O P T}, N\)
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N})\),OPTIONAL ::LW ORK
REAL (8), \(\mathbb{N}\) TENT \((\mathbb{N})\),OPTIONAL ::SCALE
\(\operatorname{COM} \operatorname{PLEX}(8), \mathbb{N}\) TENT ( \(\mathbb{N}\) ), D \(\mathbb{M}\) ENSION (:) ::X
REAL (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:) :: Y
REAL (8), \(\mathbb{N}\) TENT ( \(\mathbb{N} O U T\) ), D \(\mathbb{M}\) ENSION (:) ::TRIGS
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{I M}\) ENSION (:) :: FAC
REAL (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:) ::W ORK
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(O U T):: \mathbb{E R R}\)

SU BROUTINE FFT_64 (IOPT,N, [SCALE],X,Y,TRIGS, IFAC,W ORK, [LW ORK], ERR)
\(\mathbb{N}\) TEGER (8), \(\mathbb{N}\) TENT \((\mathbb{N}):: \mathbb{I O P T}, \mathrm{N}\)
\(\mathbb{N}\) TEGER (8), \(\mathbb{N} \operatorname{TENT}(\mathbb{N})\), OPTIONAL ::LW ORK
REAL (8), \(\mathbb{N}\) TENT \((\mathbb{N})\), OPTIONAL :: SCALE
COM PLEX (8), \(\mathbb{N} \operatorname{TENT}(\mathbb{N}), D \mathbb{M} E N S I O N(:):: X\)
REAL (8), \(\mathbb{N} T E N T(O U T), D \mathbb{M} E N S I O N(:):: Y\)
REAL (8), \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} E N S I O N(:):: T R I G S\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N} O U T), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{F A C}\)
REAL (8), \(\mathbb{N} T E N T\) (OUT),D \(\mathbb{I}\) ENSION (:) ::W ORK
\(\mathbb{N}\) TEGER (8), \(\mathbb{N}\) TENT (OUT) :: \(\mathbb{E R R}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zffftd (int*iopt, int*n, double *scale, doublecom plex
*x, double *y, double *trigs, int*ifac, double
*W ork, int *lw ork, int *ien);
void zfftd_64_ (long *iopt, long *n, double *scale, doublecom plex *x, double *y, double *trigs, long *ifac, double *W ork, long *lw ork, long *ienc);

\section*{PURPOSE}
zfftd initializes the trigonom etric \(w\) eight and factor tables or com putes the inverse FastF ourier T ransform of a double com plex sequence as follow s:

\section*{N-1}
\(Y(k)=\) scale * SUM W *X (i)
\(=0\)
where
k ranges from 0 to \(\mathrm{N}-1\)
\(i=\operatorname{sqnt}(-1)\)
isign \(=1\) for inverse transform or -1 for forw ard transform
\(W=\exp \left(i s i g n \star i^{\star} j^{\star} k * 2 * p i N N\right)\)
In com plex-to-realtransform of length \(N\), the \((\mathbb{N} / 2+1)\) com plex input data points stored are the positive-frequency half of the spectrum of the \(D\) iscrete Fourier T ransform. The other half can be obtained through com plex conjugation and therefore is notstored. Furtherm ore, due to sym \(m\) etries the im aginary of the com ponentof \(\mathrm{X}(0)\) and \(\mathrm{X} \mathbb{N} / 2\) ) (if N is even in the latter) is assum ed to be zero and is notreferenced.

\section*{ARGUMENTS}

IOPT (input)
Integer specifying the operation to be perform ed:
IO PT \(=0\) com putes the trigonom etric weight table and factor table
IO PT = 1 com putes inverse FFT
N (input)
Integer specifying length of the input sequence \(X\).
N ism ostefficientw hen it is a product of sm all prim es. \(\mathrm{N}>=0\). U nchanged on exit.

SCALE (input)
D ouble precision scalariby which transform results are scaled. Unchanged on exit. SCALE is defaulted to 1.0 D 0 forF \(95 \mathbb{N}\) TERFACE.
\(X\) (input) \(O n\) entry, \(X\) is a double com plex anay whose first \((N / 2+1)\) elem ents are the input sequence to be transform ed.
Y (output)
D ouble precision array of dim ension at least \(N\) that contains the transform results. X and Y m ay be the sam e array starting at the sam e mem ory location. O therw ise, it is assum ed that there is no overlap betw een \(X\) and \(Y\) in \(m\) em ory.

\section*{TRIGS (input/output)}

D ouble precision amay of length \(2 \star \mathrm{~N}\) that contains the trigonom etric weights. The weights are com puted w hen the routine is called w ith \(\mathbb{I D} P T=0\) and they are used in subsequent calls w hen IOPT \(=1\). U nchanged on exit.

FAC (input/output)
Integer array of dim ension at least 128 that contains the factors of \(N\). The factors are com puted when the routine is called \(w\) ith \(\mathbb{I O}\) PT \(=0\) and they are used in subsequent calls where IOPT \(=1\). U nchanged on exit.

W ORK (w orkspace)
D ouble precision array ofdim ension at least N . The user can also choose to have the routine allocate its ow n w orkspace (see LW ORK).

LW ORK (input)
Integer specifying workspace size. If LW ORK \(=0\), the routine w illallocate its ow n w orkspace.
\(\mathbb{E R R}\) (output)

On exit, integer \(\mathbb{E R R}\) has one of the follow ing
values:
0 = norm alretum
\(-1=10\) PT is not 0 or 1
\(-2=\mathrm{N}<0\)
\(-3=(\mathbb{L W}\) ORK is not 0 ) and (LW ORK is less than \(N\) )
\(-4=m\) em ory allocation forw orkspace failed

\section*{SEE ALSO}
ff

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO
- CAUTIONS

\section*{NAME}
zfftd2 -initialize the trigonom etric weight and factor
tables or com pute the tw o-dim ensional inverse FastFourier
\(T\) ransform of a tw o-dim ensional double com plex aray.

\section*{SYNOPSIS}

SU BROUTINE ZFFTD 2 ( \(\mathbb{O P T}, \mathrm{N} 1, N 2\), SCALE, \(\mathrm{X}, \mathrm{LD} \mathrm{X}, \mathrm{Y}, \mathrm{LD} \mathrm{Y}, \mathrm{TR}\) IGS, \(\mathbb{F} A C, W\) ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER \(\mathbb{I O P T}, N 1, N 2, L D X, L D Y, \mathbb{F A C}(*), L W O R K, \mathbb{E R R}\)
DOUBLE COM PLEXX (LDX,*)
D OUBLE PRECISION SCALE,Y (LDY, *), TRIGS (*),WORK (*)
SU BROUTINE ZFFTD 2_64 (IOPT,N1,N2,SCALE, X,LDX,Y,LDY,TRIGS, FAC,W ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N} T E G E R * 8 \mathbb{I} P T, N 1, N 2, L D X, L D Y, \mathbb{F} A C(*), L W O R K, \mathbb{E R R}\)
DOUBLE COM PLEXX (LDX,*)
D OUBLE PRECISION SCALE, Y (LDY ,*),TRIGS (*),WORK (*)

\section*{F95 INTERFACE}

SU BROUTINE FFT2 (TOPT,N1, N2], [SCALE],X, [LDX],Y, [LDY],TRIGS, FAC, W ORK, [LWORK], ERR)
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N}):: \operatorname{IOPT}, N 1\)
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N})\),OPTIONAL ::N2,LDX,LDY,LW ORK
REAL (8), \(\mathbb{N}\) TENT ( \(\mathbb{N}\) ), OPTIONAL :: SCALE
COM PLEX (8), \(\mathbb{N}\) TENT ( \(\mathbb{N}\) ), D \(\mathbb{M}\) ENSION (: : : : : : X
REAL (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:,:) :: Y
REAL (8), \(\mathbb{N}\) TENT ( \(\mathbb{N} O U T\) ), D \(\mathbb{I M}\) ENSION (:) ::TRIGS
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{I M}\) ENSION (:) :: \(\mathbb{F A C}\)

REAL (8), \(\mathbb{N} T E N T\) (OUT),D \(\mathbb{M}\) ENSION (:) ::W ORK
\(\mathbb{N} T E G E R, \mathbb{I N} T E N T(O U T):: \mathbb{E R R}\)
SU BROUTINE FFT2_64 (TOPT,N1, \(\mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S, \mathbb{F A C}\), W ORK, [LW ORK], ERR)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} T E N T(\mathbb{N}):: \mathbb{I O P T}, \mathrm{N} 1\)
\(\mathbb{N}\) TEGER (8), \(\mathbb{N}\) TENT \((\mathbb{N})\) ), OPT IONAL ::N 2, LD X ,LDY,LW ORK
REAL (8), \(\mathbb{N} T E N T(\mathbb{N})\),OPTIONAL ::SCALE
COM PLEX (8), \(\mathbb{N}\) TENT ( \(\mathbb{N}\) ), D \(\mathbb{M}\) ENSION (:,:) ::X
REAL (8), \(\mathbb{I N T E N T}(O U T), D \mathbb{I M} E N S I O N(:,:):: Y\)
REAL (8), \(\mathbb{N}\) TENT ( \(\mathbb{N}\) OUT), D \(\mathbb{M}\) ENSION (:) ::TRIGS
\(\mathbb{N} T E G E R(8), \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M}\) ENSION (:) :: \(\mathbb{F}\) AC
REAL (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:) ::W ORK
\(\mathbb{N}\) TEGER (8), \(\mathbb{N}\) TENT (OUT) :: \(\mathbb{E R R}\)

\section*{C INTERFACE}
\#include < sunperfh>
void zfftd2_ (int*iopt, int*n1, int *n2, double *scale, doublecom plex *x, int*ldx, double *y, int*ldy, double *trigs, int *ifac, double *work, int *lw ork, int*ien);
void zfftd2_64_ (long *iopt, long *n1, long *n2, double *scale, doublecom plex *x, long *ldx, double *y, long *ldy, double *trigs, long *ifac, double *w ork, long *lw ork, long *ien);

\section*{PURPOSE}
zfftd2 initializes the trigonom etric weight and factor tables or com putes the tw o-dim ensional inverse FastF ourier \(T\) ransform of a tw o-dim ensional double com plex anay. In com puting the tw o-dim ensionalFFT, one-dim ensionalFFT s are com puted along the row s of the input array. O ne-dim ensional FFT s are then com puted along the colum ns of the interm ediate results.

> N1-1 N 2-1

Y (k1,k2) = scale * SUM SUM W 2*W 1*X (1, 1, 2)
\(j=0 \quad \mathfrak{l}=0\)
where
k 1 ranges from 0 to \(\mathrm{N} 1-1\) and k 2 ranges from 0 to \(\mathrm{N} 2-1\)
\(i=\operatorname{sqrt}(-1)\)
isign \(=1\) for inverse transform
W \(1=\exp \left(\right.\) isign*i* \(\left.{ }^{1} * k 1 * 2 * p i / N 1\right)\)
W \(2=\exp (\) isign*i* \(2 * k 2 * 2 * p i N 2)\)
In com plex-to-real transform of length \(N 1\), the \((\mathbb{N} 1 / 2+1\) ) com -
plex input data points stored are the positive-frequency half of the spectrum of the D iscrete Fourier T ransform. The other half can be obtained through com plex conjugation and therefore is not stored.

\section*{ARGUMENTS}

IO PT (input)
Integer specifying the operation to be perform ed:
IO PT \(=0\) com putes the trigonom etric weight table and factor table
IO \(\mathrm{PT}=1\) com putes inverse FFT

N 1 (input)
Integer specifying length of the transform in the first dim ension. N 1 is m ostefficientw hen it is a productofsm allprim es. N \(1>=0\). U nchanged on exit.

N 2 (input)
Integer specifying length of the transform in the second dim ension. N 2 is m ostefficientw hen it is a productofsm allprim es. N \(2>=0\). U nchanged on exit.

SCALE (input)
D ouble precision scalarby which transform results are scaled. Unchanged on exit. SCALE is defaulted to 1.0 D 0 forF \(95 \mathbb{N}\) TERFACE .

X (input) X is a double com plex aray of din ensions (LD X , N 2 ) that contains inputdata to be transform ed.

LD X (input)
Leading dim ension of \(\mathrm{X} . \operatorname{LDX}>=\mathbb{N} 1 / 2+1)\) U nchanged on exit.

Y (output)
Y is a double precision anay of dim ensions (LD Y, N 2 ) that contains the transform results. X and Y can be the sam e array starting at the sam e \(m\) em ory location, in which case the input data are overw rilten by their transform results. O therw ise, it is assum ed that there is no overlap betw een \(X\) and \(Y\) in \(m\) em ory.

LD Y (input)
Leading dimension of \(Y\). If \(X\) and \(Y\) are the same
aray, LD \(=2 *\) LD \(X\) Else LD \(Y>=2 * L D X\) and LD \(Y\) must be even. U nchanged on exit.

TR IG S (input/output)
D ouble precision array of length 2* (N1+N2) that contains the trigonom etric w eights. The w eights are com puted when the routine is called w ith IO PT = 0 and they are used in subsequent calls w hen IO PT \(=1\). U nchanged on exit.

IFAC (input/output)
Integer anray ofdim ension at least \(2 * 128\) that contains the factors of 1 and N2. The factors are com puted when the routine is called w ith IO PT
\(=0\) and they are used in subsequent calls w hen
IO PT \(=1\). U nchanged on exit.
W ORK (w orkspace)
D ouble precision array of dim ension at least
\(\operatorname{MAX}(\mathbb{N} 1,2 \star \mathrm{~N} 2)\) where \(\mathrm{NCPU} S\) is the num berof threads
used to execute the routine. The user can also
choose to have the routine allocate its own w orkspace (see LW ORK).

LW ORK (input)
Integer specifying w orkspace size. If LW ORK = 0, the routine w illallocate its ow n w orkspace.

\section*{\(\mathbb{E R R}\) (output)}

On exit, integer \(\mathbb{E R R}\) has one of the follow ing
values:
0 = norm alretum
\(-1=\mathbb{O P T}\) is not 0,1
\(-2=\mathrm{N} 1<0\)
\(-3=N 2<0\)
\(-4=(\operatorname{LD~X}<\mathrm{N} 1 / 2+1)\)
\(-5=\) LD \(Y\) notequal \(2 *\) LD \(X\) when \(X\) and \(Y\) are same array
\(-6=(\mathbb{L D} Y<2 * L D X\) orLD \(Y\) odd) when \(X\) and \(Y\) are
sam e array
\(-7=(L W\) ORK not equal 0) and (LWORK <
MAX (N1,2*N2))
\(-8=m\) em ory allocation failed

\section*{SEE ALSO}
fft

\section*{CAUTIONS}

On exit, output subanay \(Y(1: L D Y, 1 \mathbb{N} 2)\) is overw ritten.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO
- CAUTIONS

\section*{NAME}
zfftd3-initialize the trigonom etric weight and factor
tables or com pute the three-dim ensional inverse FastFourier
Transform of a three-dim ensional double com plex array.

\section*{SYNOPSIS}

SU BROUTINE ZFFTD 3 (IOPT,N1,N2,N3,SCALE,X,LD X 1, LD X 2, Y, LD Y 1, LD Y 2, TRIGS, \(\mathbb{F A C}, \mathrm{W}\) ORK,LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER IOPT,N1,N2,N3,LDX1,LDX2,LDY1, LDY2, \(\mathbb{F A C}\) (*) \(^{*}\),
LW ORK, \(\mathbb{E R R}\)
DOUBLE COM PLEXX (LDX1,LDX2,*)
DOUBLE PRECISION SCALE,TRIGS (*),WORK (*), Y (LDY1,LDY2,*)
SU BROUTINE ZFFTD 3_64 (IOPT,N 1,N 2,N 3, SCALE,X,LD X 1, LD X 2, Y, LD Y 1, LD Y 2, TRIGS, \(\mathbb{F} A C, W\) ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER*8 \(\mathbb{I O}\) PT,N1,N2,N3,LD X 1, LD X 2, LD Y 1, LD Y 2, \(\mathbb{F} A C(\star)\), LW ORK, \(\mathbb{E R R}\)
D OUBLE COM PLEX X (LDX1,LDX2,*)
DOUBLE PRECISION SCALE,TRIGS (*),W ORK (*), Y (LDY1,LDY2,*)

\section*{F95 INTERFACE}

SU BROUTINE FFT3 (IOPT ,N 1, \(\mathbb{N} 2], \mathbb{N} 3]\) [SCALE],X, [LD X 1], LD X 2, Y, [LD Y 1], LD Y 2, TRIG \(\mathrm{S}, \mathbb{F} A C, W\) ORK, (LW ORK], \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER, \(\mathbb{N}\) TENT \((\mathbb{N}):: \operatorname{IOPT}, N 1, L D X 2, L D Y 2\)
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N})\),OPTIONAL ::N 2,N3,LDX1,LDY1,LW ORK
REAL (8), \(\mathbb{N}\) TENT \((\mathbb{N})\),OPTIONAL ::SCALE
COM PLEX (8), \(\mathbb{N}\) TENT ( \(\mathbb{N}\) ), D \(\mathbb{M}\) ENSION (:,:) ::X
REAL (8), \(\mathbb{I N T E N T}(\mathrm{OUT}), \mathrm{D} \mathbb{I M}\) ENSION (:,:) :: Y

REAL（8）， \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} E N S I O N(:):: T R I G S\) \(\mathbb{N} T E G E R, \mathbb{N}\) TENT（ \(\mathbb{N} O U T\) ），D \(\mathbb{M} E N S I O N(:):: \mathbb{F A C}\) REAL（8）， \(\mathbb{N}\) TENT（OUT），D \(\mathbb{M} E N S I O N(:):\) W ORK \(\mathbb{N} T E G E R, \mathbb{N} T E N T(O U T):: \mathbb{E R R}\)

SU BROUTINE FFT3＿64（TOPT，N1， \(\mathbb{N} 2\) ］， \(\mathbb{N} 3],[S C A L E], X,[L D X 1], L D X 2, Y\) ， ［LDY1］，LDY 2，TRIGS，正AC，WORK，［LWORK］，正RR）
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N}\) TENT（ \(\mathbb{N}):: \operatorname{IOPT}, N 1, L D X 2, L D Y 2\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} T E N T(\mathbb{N})\) ，OPTIONAL \(:: \mathrm{N} 2, \mathrm{~N} 3, L D X 1, L D Y 1\) ， LW ORK
REAL（8）， \(\mathbb{N} T E N T(\mathbb{N})\) ，OPTIONAL ：：SCALE
COM PLEX（8）， \(\mathbb{N}\) TENT（ \(\mathbb{N}\) ），D \(\mathbb{I M}\) ENSION（：\(:\) ：）：：X
REAL（8）， \(\mathbb{N}\) TENT（OUT），D \(\mathbb{M} \operatorname{ENSION(:,:)::Y~}\)
REAL（8）， \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} E N S I O N(:):: T R I G S\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N} O U T), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{F} A C\)
REAL（8）， \(\mathbb{N} T E N T(O U T), D \mathbb{M} E N S I O N(:):\) W ORK
\(\mathbb{N}\) TEGER（8）， \(\mathbb{N}\) TENT（OUT）：： \(\mathbb{E R R}\)

\section*{C INTERFACE}
\＃include＜sunperfh＞
void zfftd3＿（int＊iopt，int＊n1，int＊n2，int＊n3，double ＊scale，doublecom plex＊x，int＊ldx1，int＊ldx2， double＊y，int＊ldy1，int＊ldy2，double＊trigs， int＊ifac，double＊W ork，int＊lw ork，int＊ierr）；
void zfftd3＿64＿（long＊iopt，long＊n1，long＊n2，long＊n3， double＊scale，doublecom plex＊x，long＊ldx1，long ＊ldx2，double＊y，long＊ldy1，long＊ldy2，double ＊trigs，long＊ifac，double＊w ork，long＊lw ork， long＊ien）；

\section*{PURPOSE}
zffld3 initializes the trigonom etric weight and factor tables or computes the three－dim ensional inverse Fast Fourier T ransform of a three－dim ensional double com plex aray．

> N 3-1 N 2-1 N 1-1
\(Y(k 1, k 2, k 3)=\) scale＊SUM SUM SUM W 3＊W 2＊W 1＊X（1，飞，³）
\[
\mathfrak{j}=0 \quad \mathfrak{2}=0 \quad \mathfrak{j}=0
\]
w here
k 1 ranges from 0 to \(\mathrm{N} 1-1 ; k 2\) ranges from 0 to \(\mathrm{N} 2-1\) and \(k 3\)
ranges from 0 to \(\mathrm{N} 3-1\)
\(i=\operatorname{sqnt}(-1)\)
isign \(=1\) for inverse transform
W \(1=\exp \left(i \operatorname{sign} * i^{\star} j 1 * k 1 * 2 * p i N 1\right)\)

W \(2=\exp \left(\right.\) isign*i* \(\mathbf{2}^{*}\) k2*2*piN 2)
W \(3=\exp (\) isign*i* \(3 * k 3 * 2 * p i N 3\) )

\section*{ARGUMENTS}

IO PT (input)
Integer specifying the operation to be perform ed:
IO PT \(=0\) com putes the trigonom etric \(w\) eight table
and factor table
IO \(\mathrm{PT}=+1\) com putes inverse FFT
N 1 (input)
Integer specifying length of the transform in the first dim ension. N 1 ism ostefficientw hen it is a productofsm allprim es. N \(1>=0\). U nchanged on exit.

N 2 (input)
Integer specifying length of the transform in the second dim ension. N 2 is m ostefficientw hen it is a productofsm allprim es. N \(2>=0\). U nchanged on exit.

N 3 (input)
Integer specifying length of the transform in the third dim ension. N 3 ism ostefficientw hen it is a productofsm allprim es. N \(3>=0\). U nchanged on exit.

SCALE (input)
D ouble precision scalarby which transform results are scaled. Unchanged on exit. SCA LE is defaulted to 1.0 D 0 forF \(95 \mathbb{I N}\) TERFACE.

X (input) X is a double com plex anray of dim ensions (LD X 1, LDX2, N3) that contains input data to be transform ed.

LD X 1 (input)
firstdim ension of . LD X \(1>=\) N 1/2+1 U nchanged on exit.

LD X 2 (input)
second dim ension of X . LD X \(2>=\mathrm{N} 2\) U nchanged on exit.

Y (output)
\(Y\) is a double com plex array of dim ensions (LD Y 1,

LD Y 2, N 3) that contains the transform results. X and \(Y\) can be the sam e array starting at the sam e \(m\) em ory location, in which case the input data are overw ritten by their transform results. O therw ise, it is assum ed that there is no overlap betw een \(X\) and \(Y\) in \(m\) em ory.

LD Y 1 (input)
firstdim ension of \(Y\). If \(X\) and \(Y\) are the same aray, LD Y \(1=2 *\) LD X 1 Else LD Y \(1>=2 *\) LD X 1 and LD Y 1
is even U nchanged on exit.

LD Y 2 (input)
second dim ension of \(Y\). If \(X\) and \(Y\) are the sam \(e\) array, LD Y 2 = LD X 2 Else LD Y 2 >= N 2 U nchanged on exit.

TR IG S (input/output)
D ouble precision array of length \(2 *(\mathbb{N} 1+N 2+N 3)\) that contains the trigonom etric w eights. The w eights are com puted w hen the routine is called w ith IO PT
= 0 and they are used in subsequent calls w hen
IO PT \(=1\). U nchanged on exit.

FAC (input/output)
Integer array ofdim ension at least \(3 * 128\) that
contains the factors of \(1, \mathrm{~N} 2\) and N 3 . The factors are com puted w hen the routine is called w ith IO PT \(=0\) and they are used in subsequent calls when IO PT = 1. Unchanged on exit.

W ORK (w orkspace)
D ouble precision array of dim ension at least MAX \(\left.\left.\mathbb{N}, 2{ }^{\star} \mathrm{N} 2,2{ }^{\star} \mathrm{N} 3\right)+16 \star \mathrm{~N} 3\right)\) * \(\mathrm{NCPU} S\) where NCPUS is the num ber of threads used to execute the routine. The user can also choose to have the routine allocate its ow n w orkspace (see LW ORK).

LW ORK (input)
Integer specifying w orkspace size. If LW ORK \(=0\), the routine w illallocate its ow n w orkspace.
\(\mathbb{E R R}\) (output)
On exit, integer \(\mathbb{E R R}\) has one of the follow ing
values:
0 = norm alretum
\(-1=10 \mathrm{PT}\) is not 0 or 1
\(-2=N 1<0\)
\(-3=\mathrm{N} 2<0\)
\(-4=\mathrm{N} 3<0\)
\(-5=(\operatorname{LLD} 1<\mathrm{N} 1 / 2+1)\)
\(-6=(\operatorname{LDX} 2<\mathrm{N} 2)\)
-7 = LD Y 1 notequal \(2 \star\) LD X 1 when \(X\) and \(Y\) are same aray
\(-8=(\mathrm{LD}\) Y \(1<2 * \mathrm{LD} \mathrm{X} 1)\) or (LD Y 1 is odd) w hen X and Y are not sam e array
\(-9=(\mathbb{L D} Y 2<N 2)\) or (LD Y 2 notequalLD X 2) when X and \(Y\) are same array
\(-10=(L W\) ORK not equal 0) and ( \((L W\) ORK <
M AX (N, \(2 *\) N \(2,2 * N 3)+16 * N 3) * N C P U S)\)
\(-11=m\) em ory allocation failed

\section*{SEE ALSO}
ff

\section*{CAUTIONS}

On exit, outputsubarray Y ( \(1: L D Y 1,1 \mathbb{N} 2,1 \mathbb{N} 3\) ) is overw ritten.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO

\section*{NAME}
zfftdm -initialize the trigonom etric weight and factor
tables or com pute the one-dim ensional inverse FastFourier
\(T\) ransform of a set of double com plex data sequences stored in a tw o-dim ensional array.

\section*{SYNOPSIS}

SUBROUTINE ZFFTDM (IOPT,N1,N2,SCALE,X,LDX,Y,LDY,TRIGS, \(\mathbb{F A C}, \mathrm{W} O R K\), LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N} \operatorname{TEGER} \mathbb{I} P \mathrm{P}, \mathrm{N} 1, \mathrm{~N} 2, \operatorname{LD}, \mathrm{LD} \mathrm{Y}, \mathbb{F} A C(*), L W\) ORK, \(\mathbb{E R R}\)
DOUBLE COM PLEX X (LDX,*)
D OUBLE PREC ISION SCALE,Y (LDY,*), TRIGS (*),W ORK (*)
SU BROUTINE ZFFTDM_64(TOPT,N1,N2,SCALE,X,LDX,Y,LDY,TRIGS, FFAC,WORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N} T E G E R * 8 \mathbb{I} P T, N 1, N 2, L D X, L D Y, \mathbb{F} A C(*), L W O R K, \mathbb{E R R}\)
DOUBLE COM PLEXX (LDX,*)
DOUBLE PRECISION SCALE,Y (LDY,*),TRIGS (*),W ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE FFTM (IOPT,N1, N2], [SCALE],X, [LDX],Y, [LDY],TRIGS, تAC,W ORK, [LW ORK], ERR)
\(\mathbb{N}\) TEGER, \(\mathbb{N}\) TENT \((\mathbb{N})::\) IOPT,N1
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N})\),OPTIONAL ::N2,LDX,LDY,LW ORK
REAL (8), \(\mathbb{N} T E N T(\mathbb{N})\),OPTIONAL :: SCALE
COM PLEX (8), \(\mathbb{N}\) TENT ( \(\mathbb{N}\) ), D \(\mathbb{M}\) ENSION (: : : : : : X
REAL (8), \(\mathbb{I N T E N T}(O U T), D \mathbb{M}\) ENSION (: \(:\) :) :: Y
REAL (8), \(\mathbb{N}\) TENT ( \(\mathbb{N} O U T\) ), D \(\mathbb{M}\) ENSION (:) ::TRIGS
\(\mathbb{N}\) TEGER, \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M}\) ENSION (:) :: \(\mathbb{F A C}\)

SUBROUTINE FFTM _64 (IOPT,N1, \(\mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S, \mathbb{F} A C\), W ORK, [LW ORK], \(\mathbb{E R R}\) )


```

REAL (8), \mathbb{NTENT (\mathbb{N}),OPTIONAL ::SCALE}
COM PLEX (8),\mathbb{NTENT (\mathbb{N}),D\mathbb{M ENSION (:,:) ::X}}\mathbf{~}={
REAL (8), INTENT (OUT),D IM ENSION (:,:)::Y
REAL (8), \mathbb{NTENT (NNOUT),D IM ENSION (:) ::TRIGS}

```

```

REAL (8), INTENT (OUT),D IM ENSION (:) ::W ORK

```


\section*{C INTERFACE}
\#include < sunperfh>
void zfftdm _ (int*iopt, int *m, int *n, double *scale, doublecom plex *x, int*ldx, double *y, int*ldy, double *trigs, int *ifac, double *work, int *lw ork, int *ien);
void zfftdm _64_ (long *iopt, long *m, long *n, double *scale, doublecom plex *x, long *ldx, double *y, long *ldy, double *trigs, long *ifac, double *w ork, long *lw ork, long *ien);

\section*{PURPOSE}
zfftdm initializes the trigonom etric weight and factor tables orcom putes the one-dim ensional inverse FastFourier Transform of a setofdouble com plex data sequences stored in a tw o-dim ensionalarray:

> N 1-1
\(Y(k, l)=\) scale * SUM W *X (jl)
\[
\dot{j} 0
\]
where
\(k\) ranges from 0 to N 1-1 and lranges from 0 to N 2-1
\(i=\operatorname{sqrt}(-1)\)
isign \(=1\) for inverse transform
\(W=\exp \left(i s i g n * i^{\star}{ }^{j} \star k * 2 \star \operatorname{piN} 1\right)\)
In com plex-to-real transform of length \(N 1\), the \((N 1 / 2+1)\) com plex input data points stored are the positive-frequency half of the spectrum of the \(D\) iscrete Fourier \(T\) ransform. The other half can be obtained through com plex conjugation and therefore is not stored. Furthem ore, due to sym \(m\) etries the
im aginary of the com ponent of \(\mathrm{X}(0,0 \mathbb{N} 2-1)\) and \(\mathrm{X}(\mathbb{N} 1 / 2,0\) N 2-1) (ifN 1 is even in the latter) is assum ed to be zero and is not referenced.

\section*{ARGUMENTS}

IO PT (input)
Integer specifying the operation to be perform ed:
IO PT \(=0\) com putes the trigonom etric weight table and factortable
IO PT = 1 com putes inverse FFT

N 1 (input)
Integer specifying length of the input sequences.
N 1 ism ostefficientw hen it is a product ofsm all prim es. N \(1>=0\). U nchanged on exit.

N 2 (input)
Integer specifying num ber of input sequences. N 2 \(>=0\). U nchanged on exit.
SCALE (input)
D ouble precision scalarby which transform results are scaled. Unchanged on exit. SCALE is defaulted to 1.0 D 0 forF \(95 \mathbb{N}\) TERFACE .
\(X\) (input) \(X\) is a double com plex aray of din ensions (LD \(X\), \(N\) 2) that contains the sequences to be transform ed stored in its colum ns in \(X(0 \mathbb{N} 1 / 2,0\) N 2-1).

LD X (input)
Leading dim ension ofX . LD X >= \(\mathbb{N} 1 / 2+1\) ) U nchanged on exit.

Y (output)
Y is a double precision array ofdim ensions (LD Y,
N 2 ) that contains the transform results of the input sequences in \(Y(0 \mathbb{N} 1-1,0\) N \(2-1\) ). \(X\) and \(Y\) can be the same array starting at the sam emem ory location, in which case the input sequences are overw ritten by their transform results. O therw ise, it is assum ed that there is no overlap betw een \(X\) and \(Y\) in \(m\) em ory.

LD Y (input)
Leading dim ension of \(Y\). If \(X\) and \(Y\) are the same array, LDY \(=2 * L D X\) Else LD Y >=N 1 U nchanged on exit.

TR IG S (input/output)
double precision array of length \(2 \star \mathrm{~N} 1\) that contains the trigonom etric w eights. The w eights are com puted w hen the routine is called w ith IO PT \(=0\) and they are used in subsequent calls when IO PT = 1. U nchanged on exit.

IFAC (input/output)
Integer array ofdim ension at least 128 that contains the factors of N 1 . The factors are com puted when the routine is called w ith IO PT \(=0\) and they are used in subsequent calls when IOPT \(=1\). U nchanged on exit.

W ORK (w orkspace)
double precision array of dim ension at least N 1.
The user can also choose to have the routine allocate its ownw orkspace (see LW ORK).

LW ORK (input)
Integer specifying workspace size. IfLW ORK \(=0\), the routine w illallocate its ow n w orkspace.
\(\mathbb{E R R}\) (output)
On exit, integer \(\mathbb{E R R}\) has one of the follow ing
values:
0 = norm alretum
\(-1=10 \mathrm{PT}\) is not 0 or 1
\(-2=\mathrm{N} 1<0\)
\(-3=N 2<0\)
\(-4=(\operatorname{LDX}<\mathrm{N} 1 / 2+1)\)
\(-5=(\mathrm{LD} \mathrm{Y}<\mathrm{N} 1)\) or (LD Y notequal2*LD X when X and
Y are sam e array)
\(-6=(\mathbb{L W}\) ORK notequal0) and (LW ORK < N 1)
\(-7=m\) em ory allocation failed

\section*{SEE ALSO}
fft

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
zfflf-com pute the Fourier coefficients of a periodic
sequence. The FFT operations are unnorm alized, so a callof ZFFTF follow ed by a callof ZFFTB w ill multiply the input sequence by N .

\section*{SYNOPSIS}
```

SUBROUTINE ZFFTF N,X,W SAVE)
DOUBLE COM PLEX X (*)
INTEGERN
DOUBLE PRECISION W SAVE (*)
SUBROUT\mathbb{NE ZFFTF_64N,X,W SAVE)}
DOUBLE COM PLEX X (*)
INTEGER*8N
DOUBLE PRECISION W SAVE (*)

```
F95 INTERFACE
    SUBROUTINE FFTF ( \(\mathbb{N}\) ], X,W SAVE)
    COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::X
    \(\mathbb{N} T E G E R:: N\)
    REAL (8),D \(\mathbb{M}\) ENSION (:) ::W SAVE
    SU BROUTINE FFTF_64 ( \(\mathbb{N}\) ],X,W SAVE)
    COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::X
    \(\mathbb{N} T E G E R(8):: N\)
    REAL (8), D IM ENSION (:) ::W SAVE
\#include <sunperfh>
void zffff(intn, doublecom plex *x, double *w save);
void zfflf_64 (long n, doublecom plex *x, double *w save);

\section*{ARGUMENTS}

N (input) Length of the sequence to be transform ed. These subroutines are \(m\) ostefficientw hen \(N\) is a product of sm allprim es. \(\mathrm{N}>=0\).

X (input) On entry, an amay of length N containing the sequence to be transform ed.

W SAVE (input)
O n entry, W SAVE m ustbe an array ofdim ension (4* \(\mathrm{N}+15\) ) orgreater and m usthave been initialized by ZFFTI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- ARGUMENTS

\section*{NAME}
zffti-initialize the array \(W\) SAVE, which is used in both ZFFTF and ZFFTB.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZFFTIN,W SAVE)}

```
\(\mathbb{N}\) TEGER N
DOUBLE PRECISION W SAVE (*)
SUBROUTINE ZFFTI_64 \(\mathbb{N}, \mathrm{W}\) SAVE)
\(\mathbb{N}\) TEGER*8 N
DOUBLE PRECISION W SAVE (*)

\section*{F95 INTERFACE}

SU BROUTINE ZFFTIN,W SAVE)
\(\mathbb{N} T E G E R:: N\)
REAL (8),D IM ENSION (:) ::W SAVE

SUBROUTINE ZFFTI_64 (N,W SAVE)
\(\mathbb{N} T E G E R(8):: N\)
REAL (8),D \(\mathbb{I M}\) ENSION (:) ::W SAVE

\section*{C INTERFACE}
\#include <sunperfh>
void zfffid(intn, double *w save);
void zffti_ 64 (long n, double *w save);

\section*{ARGUMENTS}

N (imput) Length of the sequence to be transform ed. \(\mathrm{N}>=0\).

W SAVE (input/output)
O n entry, an array ofdim ension ( \(4 * N+15\) ) or greater. ZFFTI needs to be called only once to initialize array \(W\) ORK before calling ZFFTF and/or ZFFTB if N and W SAVE rem ain unchanged betw een these calls. Thus, subsequent transform \(s\) or inverse transform s of sam e size can be obtained faster than the first since they do not require initialization of the w orkspace.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE

\section*{NAME} zfftopt-com pute the length of the closest fastFFT

\section*{SYNOPSIS}

INTEGER FUNCTION ZFFTOPT (LEN)
\(\mathbb{I N T E G E R L E N}\)
\(\mathbb{I N}\) TEGER*8 FUNCTION ZFFTOPT_64 (LEN)
\(\mathbb{N}\) TEGER *8 LEN

F95 INTERFACE
INTEGER FUNCTION ZFFTOPT (LEN)
\(\mathbb{N}\) TEGER ::LEN

INTEGER (8) FUNCTION ZFFTOPT_64 (LEN)
\(\mathbb{N}\) TEGER (8) :: LEN

C INTERFACE
\#include <sunperfh>
int zfflopt(int len);
long zfftopt_64 (long len);

\section*{PURPOSE}

Fourier transform algorithm s , including those used in Perform ance Library, w ork bestw ith vector lengths that are products of sm all prim es. Forexam ple, an FFT of length \(32=2 * * 5 \mathrm{w}\) ill nun fasterthan an FFT of prime length 31 because 32 is a productofsm allprim es and 31 is not. If your application is such that you can taper or zero pad your vector to a larger length then this function \(m\) ay help you select a better length and run your FFT faster.

ZFFTOPT w ill retum an integerno sm aller than the input argum entN that is the closestnum ber that is the product of sm allprim es. ZFFTOPT w ill retum 16 foran input of \(\mathrm{N}=16\) and retum 18=2*3*3 foran inputof \(\mathrm{N}=17\).

N ote that the length com puted here is not guaranteed to be optim al, only to be a
product of sm allprim es. A lso, the value retumed \(m\) ay change as the underlying
FFT sbecom e capable of handling larger prim es. For exam ple, passing in \(N=51\) to day \(w\) ill retum \(52=2 \star 2 \star 13\) rather than \(51=3 * 17\) because the FFT s in Perform ance Library do not have fast radix 17 code. In the future, radix 17 code m ay be added
and then \(\mathrm{N}=51 \mathrm{w}\) ill retum 51 .

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO

\section*{NAME}
zfftz - initialize the trigonom etric weight and factor tables or com pute the FastFouriertransform (forw ard or inverse) of a double com plex sequence.

\section*{SYNOPSIS}

\(\mathbb{N} \operatorname{TEGER} \mathbb{I} P \mathrm{P}, \mathrm{N}, \mathbb{F} A C(*), L W\) ORK, \(\mathbb{E R R}\)
DOUBLE COM PLEX X (*) , Y (*)
DOUBLE PRECISION SCALE,TRIGS (*), W ORK (*)
SU BROUTINE ZFFTZ_64 (IOPT,N,SCALE,X,Y,TRIGS, IFAC,W ORK,LW ORK, ERR)
\(\mathbb{N} T E G E R * 8 \mathbb{I} P T, N, \mathbb{F A C}(*), L W O R K, \mathbb{E R R}\)
DOUBLE PRECISION SCALE, TRIGS (*), W ORK (*) DOUBLE COM PLEX X (*), Y (*)

\section*{F95 INTERFACE}

SU BROUTINE FFT (TOPT, \(\mathbb{N}\) ], [SCALE],X,Y,TRIGS, \(\mathbb{F A C}, W\) ORK, [LW ORK ], \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER, \(\mathbb{N} T E N T(\mathbb{N})::\) IOPT
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N})\),OPTIONAL ::N,LW ORK REAL (8), \(\mathbb{N}\) TENT \((\mathbb{N})\), OPTIONAL ::SCALE
\(\operatorname{COM} \operatorname{PLEX}(8), \mathbb{N}\) TENT ( \(\mathbb{N}\) ), D \(\mathbb{M}\) ENSION (:) ::X
COM PLEX (8), \(\mathbb{I N T E N T}\) (OUT),D \(\mathbb{I M}\) ENSION (:) ::Y
REAL (8), \(\mathbb{N} T E N T(\mathbb{N} O U T)\), D \(\mathbb{M}\) ENSION (:) ::TRIGS
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{I M}\) ENSION (:) :: \(\mathbb{F A C}\)
REAL (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:) ::W ORK
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(O U T):: \mathbb{E R R}\)

SU BROUTINE FFT_64 (IOPT, \(\mathbb{N}],[S C A L E], X, Y, T R I G S, \mathbb{F A C}, W\) ORK, [LW ORK ], \(\mathbb{E R R}\) )
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N}):: \operatorname{IOPT}\)
\(\mathbb{N} T E G E R(8), \mathbb{N} T E N T(\mathbb{N})\), OPTIONAL :: N , LW ORK
REAL (8), \(\mathbb{N} T E N T(\mathbb{N})\), OPTIONAL :: SCALE
COM PLEX (8), \(\mathbb{N} \operatorname{TENT}(\mathbb{N})\), D \(\mathbb{M} \operatorname{ENSION}(:):: X\)
COM PLEX (8), \(\mathbb{I N}\) TENT (OUT), D \(\mathbb{M} E N S I O N(:):: Y\)
REAL (8), \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} E N S I O N(:):: T R I G S\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N} O U T), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{F A C}\)
REAL (8), \(\mathbb{N} T E N T\) (OUT),D \(\mathbb{I}\) ENSION (:) ::W ORK
\(\mathbb{N}\) TEGER (8), \(\mathbb{N}\) TENT (OUT) :: \(\mathbb{E R R}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zfftz_ (int*iopt, int *n, double *scale, doublecom plex *x, doublecom plex *y, double *trigs, int*ifac, double *w ork, int *lw ork, int *ien);
void zfflz_64_ (long *iopt, long *n, double *scale, doublecom plex \({ }^{2}\), doublecom plex \({ }^{*} y\), double *trigs, long *ifac, double *w ork, long *lw ork, long *ienc);

\section*{PURPOSE}
zfftz initializes the trigonom etric \(w\) eight and factor tables or com putes the FastFourier transform (forw ard or inverse)
of a double com plex sequence as follow s:
\[
\begin{aligned}
& \mathrm{N}-1 \\
& \text { Y (k) = scale * SUM W *X ( }) \\
& \ddagger
\end{aligned}
\]
where
k ranges from 0 to \(\mathrm{N}-1\)
\(i=\operatorname{sqnt}(-1)\)
isign \(=1\) for inverse transform or -1 for forw ard transform
\(W=\exp \left(i s i g n \star i^{\star} j^{\star} k \star 2 \star \mathrm{pi} \mathrm{N}\right)\)

\section*{ARGUMENTS}

IO PT (input)
Integer specifying the operation to be perform ed:
IO P T \(=0\) com putes the trigonom etric \(w\) eight table
and factor table
IO PT \(=-1\) com putes forw ard FFT
IO PT \(=+1\) com putes inverse FFT

N (input)
Integer specifying length of the input sequence X . N ism ostefficientw hen it is a product of \(s m\) all prim es. \(\mathrm{N}>=0\). U nchanged on exit.

SCALE (input)
D ouble precision scalarby w hich transform results are scaled. Unchanged on exit. SCALE is defaulted to 1 .OD 0 forF \(95 \mathbb{I N}\) TERFACE.
\(X\) (input) O \(n\) entry, \(X\) is a double com plex array of dim ension at least N that contains the sequence to be transform ed.

Y (output)
D ouble com plex aray of dim ension at least \(N\) that contains the transform results. \(X\) and \(Y m\) ay be the sam e array starting at the sam e mem ory location. O therw ise, it is assum ed that there is no overlap betw een \(X\) and \(Y\) in \(m\) em ory.

TRIGS (input/output)
D ouble precision anray of length \(2 * \mathrm{~N}\) that contains the trigonom etric \(w\) eights. The w eights are com puted w hen the routine is called w ith IO PT \(=0\) and they are used in subsequent calls w hen \(\mathbb{I D P T}=1\) or IO PT \(=-1\). U nchanged on exit.

IFAC (input/output)
Integer array ofdim ension at least 128 that contains the factors of \(N\). The factors are com puted w hen the routine is called w ith IO PT \(=0\) and they are used in subsequent calls w here IO PT \(=1\) or IO PT = -1. U nchanged on exit.

W ORK (w orkspace)
D ouble precision array ofdim ension at least \(2 * \mathrm{~N}\).
The user can also choose to have the routine مcate its ow n w orkspace (see LW ORK).

LW ORK (input)
Integer specifying w orkspace size. If LW ORK \(=0\), the routine w illallocate its ow n w orkspace.

正RR (output)
O n exit, integer \(\mathbb{E} R R\) has one of the follow ing
values:
0 = norm alretum
\(-1=10\) PT is not 0,1 or -1
\(-2=\mathrm{N}<0\)
\(-3=(\llbracket W O R K\) is not0) and ( \(\ddagger W\) ORK is less than \(2 * N\) )
\(-4=m\) em ory allocation forw orkspace failed

\section*{SEE ALSO}
fflt

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO
- CAUTIONS

\section*{NAME}
zfflz2-inilialize the trigonom etric weight and factor
tables or com pute the tw o-dim ensionalFastFourier T ransform
(forw ard or inverse) of a tw o-dim ensional double com plex array.

\section*{SYNOPSIS}

SU BROUTINE ZFFTZ2 (IOPT,N1,N2,SCALE,X,LDX,Y,LDY,TRIGS, FAC,W ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER \(\mathbb{I O P T}, N 1, N 2, L D X, L D Y, \mathbb{F A C}(*), L W O R K, \mathbb{E R R}\)
DOUBLE COM PLEXX (LDX,*), Y (LDY,*)
DOUBLE PRECISION SCALE,TRIGS (*), W ORK (*)
SU BROUTINE ZFFTZ2_64 (IOPT,N1,N2,SCALE,X,LDX,Y,LDY,TRIGS, \(\mathbb{F} A C, W\) ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER*8 \(\mathbb{I O P T}, \mathrm{N} 1, N 2, L D \mathrm{X}, \mathrm{LD} \mathrm{Y}, \mathbb{F} A C(*), L W\) ORK, \(\mathbb{E R R}\)
DOUBLE PRECISION SCALE,TRIGS (*), W ORK (*)
DOUBLE COM PLEX X (LDX,*), Y (LDY,*)

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E \operatorname{FFT} 2\) (IOPT, \(\mathbb{N} 1], \mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S\), 펻, W ORK, [LW ORK], \(\mathbb{E R R}\) )
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N}):: \operatorname{IOPT}\)
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N})\),OPTIONAL ::N1,N2,LDX,LDY,LW ORK
REAL (8), \(\mathbb{N} T E N T(\mathbb{N})\),OPTIONAL ::SCALE
\(\operatorname{COM} \operatorname{PLEX}(8), \mathbb{N}\) TENT ( \(\mathbb{N}\) ), D \(\mathbb{M}\) ENSION \((:,:\) : : : X
COM PLEX (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:,:) ::Y
REAL (8), \(\mathbb{N} T E N T(\mathbb{N} O U T), D \mathbb{I M} E N S I O N(:):: T R I G S\)
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} E N S I O N(:):: \mathbb{F A C}\) REAL (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:) ::W ORK
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(O U T):: \mathbb{E R R}\)

SU BROU T \(\mathbb{N} E \operatorname{FFT} 2 \_64\) ( \(\mathbb{O P}\) PT, \(\left.\left.\mathbb{N} 1\right], \mathbb{N} 2\right],[S C A L E], X,[L D X], Y,[L D Y], T R I G S\), تAC,W ORK, [LW ORK], 芭RR)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N}):: \mathbb{I O P T}\)
\(\mathbb{N}\) TEGER (8), \(\mathbb{N}\) TENT ( \(\mathbb{N}\) ), OPTIONAL ::N1,N2,LDX,LDY,LW ORK
REAL (8), \(\mathbb{N}\) TENT ( \(\mathbb{N}\) ), OPTIONAL :: SCALE
COM PLEX (8), \(\mathbb{N} T E N T(\mathbb{N}), D \mathbb{M} E N S I O N(:,:):: X\)
COM PLEX (8), \(\mathbb{I N T E N T}(O U T), D \mathbb{M} E N S I O N(:,:):: Y\)
REAL (8), \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} E N S I O N(:):: T R I G S\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N O U T}), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{F A C}\)
REAL (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M} E N S I O N(:):: W O R K\)
\(\mathbb{N}\) TEGER (8), \(\mathbb{N}\) TENT (OUT) :: \(\mathbb{E R R}\)

\section*{C INTERFACE}
\#include < sunperfh>
void zfftz2_ (int*iopt, int*n1, int *n2, double *scale, doublecom plex \({ }^{*} x\), int *ldx, doublecom plex *y, int
* ldy, double *trigs, int *ifac, double *w ork, int
*lw ork, int *ient);
void zfftz2_64_ (long *iopt, long *n1, long *n2, double *scale, doublecom plex *x, long *ldx, doublecom plex
*y, long *ldy, double *trigs, long *ifac, double *W ork, long *lw ork, long *ien);

\section*{PURPOSE}
zfflz2 initializes the trigonom etric weight and factor tables or computes the tw o-dim ensional Fast Fourier \(T\) ransform (forw ard or inverse) of a tw o-dim ensional double complex array. In computing the two-dim ensional FFT, one-dim ensionalFFTs are com puted along the colum ns of the input anay. O ne-dim ensionalFFTs are then com puted along the row s of the interm ediate results.
\[
\text { N } 2-1 \quad \text { N 1-1 }
\]

Y \((k 1, k 2)=\) scale * SUM SUM W 2*W 1*X (1, 2)
\[
\mathfrak{2}=0 \quad j=0
\]
w here
k 1 ranges from 0 to \(\mathrm{N} 1-1\) and \(k 2\) ranges from 0 to \(\mathrm{N} 2-1\)
\(i=\operatorname{sqnt}(-1)\)
isign \(=1\) for inverse transform or -1 for forw ard transform
W \(1=\exp (\) isign *i* \(11 * k 1 * 2 * \mathrm{pi} N 1)\)
W \(2=\exp \left(i \operatorname{sign} * i^{\star} 2 * k 2 * 2 * \mathrm{pi}\right.\) N 2\()\)

\section*{ARGUMENTS}

IO PT (input)
Integer specifying the operation to be perform ed:
IO PT \(=0\) com putes the trigonom etric \(w\) eight table
and factor table
IO PT \(=-1\) com putes forw ard FFT
IO \(\mathrm{PT}=+1\) com putes inverse FFT

N 1 (input)
Integer specifying length of the transform in the first dim ension. N 1 is m ostefficientw hen it is a productofsm allprim es. N \(1>=0\). U nchanged on exit.

N 2 (input)
Integer specifying length of the transform in the second dim ension. N 2 is m ostefficientw hen it is a productofsm allprim es. N \(2>=0\). U nchanged on exit.
SCALE (input)
D ouble precision scalarby which transform results are scaled. Unchanged on exit. SCALE is defaulted to 1.0 D 0 forF \(95 \mathbb{I N}\) TERFACE .
\(X\) (input) \(X\) is a double com plex aray of dim ensions (LD \(X\), N 2) that contains inputdata to be transform ed.

LD X (input)
Leading dim ension of X . LD X >= N 1 U nchanged on exit.

Y (output)
\(Y\) is a double com plex array of dim ensions (LD Y, N 2 ) that contains the transform results. X and \(Y\) can be the sam e array starting at the sam \(e m\) em ory location, in which case the input data are overw ritten by their transform results. O therw ise, it is assum ed that there is no overlap betw een \(X\) and \(Y\) in \(m\) em ory.

LD Y (input)
Leading dim ension of \(Y\). If \(X\) and \(Y\) are the same aray, LD Y \(=\) LD X Else LD Y >=N 1 U nchanged on exit.

TR IG S (input/output)
D ouble precision array of length \(2 *(\mathbb{N} 1+\mathrm{N} 2)\) that
contains the trigonom etric w eights. The w eights are com puted when the routine is called w ith IO PT
= 0 and they are used in subsequent calls w hen
IO PT = 1 or \(\mathbb{I O} P T=-1\). Unchanged on exit.

IFAC (input/output)
Integer anray ofdim ension at least \(2 * 128\) that
contains the factors of N 1 and N 2 . The factors
are com puted when the routine is called w ith IO PT
= 0 and they are used in subsequent calls w hen
IO PT = 1 or IO PT =-1. Unchanged on exit.

W ORK (w orkspace)
D ouble precision array of dimension at least
\(2 * \operatorname{MAX}(\mathbb{N} 1, N 2) * N C P U S\) where NCPUS is the num berof
threads used to execute the routine. The usercan
also choose to have the routine allocate its ow \(n\)
w orkspace (see LW ORK).
LW ORK (input)
Integer specifying workspace size. IfLW ORK \(=0\),
the routine w illallocate its ow n w orkspace.
\(\mathbb{E R R}\) (output)
On exit, integer \(\mathbb{E R R}\) has one of the follow ing
values:
\(0=\) norm alretum
\(-1=\mathbb{O P P T}\) is not 0,1 or -1
\(-2=\mathrm{N} 1<0\)
\(-3=N 2<0\)
\(-4=(\mathbb{L D X}<\mathrm{N} 1)\)
\(-5=(\mathbb{L D} Y<N 1)\) or (LD Y notequalLD \(X\) when \(X\) and \(Y\)
are sam e aray)
\(-6=(L W\) ORK not equal 0) and (LWORK <
2*M AX (N1,N2)*NCPUS)
\(-7=m\) em ory allocation failed

\section*{SEE ALSO}
ff

\section*{CAUTIONS}

On exit, entire outputaray \(Y(1: L D Y, 1 \mathbb{N} 2)\) is overw ritten.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO
- CAUTIONS

\section*{NAME}
zfflz3-initialize the trigonom etric weight and factor tables or compute the three-dim ensional Fast Fourier \(T\) ransform (forw ard or inverse) of a three-dim ensional double com plex array.

\section*{SYNOPSIS}

SU BROUTINE ZFFTZ3 (IOPT,N1,N2,N3,SCALE,X,LDX1,LDX2,Y,LDY1,LDY2, TRIGS, \(\mathbb{F A C}, \mathrm{W}\) ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER IOPT,N1,N2,N 3,LD X 1, LD X 2, LD Y 1, LD Y 2, \(\mathbb{F} A C\) (*), LW ORK, ERR
DOUBLE COM PLEXX (LDX1,LDX2,*), Y (LDY 1,LDY2,*)
DOUBLE PRECISION SCALE,TRIGS (*), W ORK (*)
SU BROUTINE ZFFTZ3_64 (IOPT,N1,N2,N 3, SCALE,X,LDX1,LDX2, Y,LDY 1, LD Y 2, TRIGS, \(\mathbb{F A C}, \mathrm{W}\) ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER*8 IO PT,N1,N2,N3,LDX1,LDX2,LD Y 1,LD Y 2, \(\mathbb{F A C}\) (*), LW ORK, \(\mathbb{E R R}\)
DOUBLE COM PLEX X (LDX1,LDX2,*), Y (LDY1,LDY2,*)
DOUBLE PRECISION SCALE, TRIGS (*), W ORK (*)

\section*{F95 INTERFACE}

SUBROUTINE FFT3 (TOPT, \(\mathbb{N} 1], \mathbb{N} 2], \mathbb{N} 3],[S C A L E], X,[L D X 1], L D X 2, Y,[L D Y 1]\), LD Y \(2, T R I G S, \mathbb{F A C}, \mathrm{~W} O R K,[L W O R K], \mathbb{E R R})\)
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N})::\) IOPT,LDX \(2, L D Y 2\)
\(\mathbb{N}\) TEGER, \(\mathbb{N}\) TENT \((\mathbb{N})\), OPTIONAL ::N \(1, N 2, N 3, L D X 1, L D Y 1\), LW ORK
REAL (8), \(\mathbb{N}\) TENT \((\mathbb{N})\) ), OPTIONAL :: SCALE

COM PLEX (8), \(\mathbb{N} \operatorname{TENT}(\mathbb{N}), \operatorname{D} \mathbb{M} \operatorname{ENSION}(:,:):: X\)
COM PLEX (8), \(\mathbb{N}\) TENT (OUT), D \(\mathbb{M} E N S I O N(:,:):: Y\)
REAL (8), \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} E N S I O N(:):: T R I G S\)
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{F A C}\) REAL (8), \(\mathbb{N}\) TENT (OUT), D \(\mathbb{M}\) ENSION (:) ::W ORK \(\mathbb{N} T E G E R, \mathbb{N} T E N T(O U T):: \mathbb{E R R}\)

SU BROUTINE FFT3_64 (TOPT, \(\mathbb{N} 1], \mathbb{N} 2], \mathbb{N} 3],[S C A L E], X,[L D X 1], L D X 2, Y\), [LD Y 1], LD Y 2, TR IG \(S\), \(\mathbb{F A} C, W\) ORK, [LW ORK], \(\mathbb{E R R}\) )
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N})::\) IOPT,LDX \(2, L D Y 2\)
\(\mathbb{N} T E G E R(8), \mathbb{N} T E N T(\mathbb{N}), O P T \mathbb{I} \operatorname{NAL}:: N 1, N 2, N 3, L D X 1, L D Y 1\),
LW ORK
REAL (8), \(\mathbb{N} T E N T(\mathbb{N})\), OPTIONAL :: SCALE
COM PLEX (8), \(\mathbb{N} \operatorname{TENT}(\mathbb{N}), D \mathbb{M} \operatorname{ENSION}(:,:):: X\)
COM PLEX (8), \(\mathbb{N}\) TENT (OUT), D \(\mathbb{M} E N S I O N(:,:):: Y\)
REAL (8), \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} E N S I O N(:):: T R I G S\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N} O U T), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{F A C}\)
REAL (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:) ::W ORK
\(\mathbb{N}\) TEGER (8), \(\mathbb{N}\) TENT (OUT) :: \(\mathbb{E R R}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zfflz3_ (int*iopt, int*n1, int*n2, int *n3, double *scale, doublecom plex *x, int*ldx1, int*ldx2, doublecom plex *y, int *ldy1, int *ldy2, double *trigs, int *ifac, double *w ork, int *lw ork, int *ienc);
void zfflz3_64_ (long *iopt, long *n1, long *n2, long *n3, double *scale, doublecom plex *x, long *ldx1, long *ldx2, doublecom plex *y, long *ldy1, long *ldy2, double *trigs, long *ifac, double *w ork, long *lw ork, long *ien);

\section*{PURPOSE}
zffiz3 initializes the trigonom etric w eight and factor tables or com putes the three-dim ensional Fast Fourier T ransform (forw ard or inverse) of a three-dim ensionaldouble com plex array.
\[
\begin{gathered}
\text { N 3-1 N 2-1 N 1-1 } \\
Y(k 1, k 2, k 3)=\text { scale * SUM SUM SUM W 3*W 2*W 1*X }(\mathfrak{1}, \mathfrak{\imath}, \mathfrak{j}) ~ \\
j \mathfrak{j}=0 \quad \mathfrak{R}=0 \quad \mathfrak{j}=0
\end{gathered}
\]
where
k 1 ranges from 0 to \(\mathrm{N} 1-1\); k2 ranges from 0 to \(\mathrm{N} 2-1\) and \(k 3\) ranges from 0 to \(\mathrm{N} 3-1\)
\(i=\operatorname{sqrt}(-1)\)
isign \(=1\) for inverse transform or -1 for forw ard transform
W \(1=\exp (i s i g n * i * j * k 1 * 2 * p i N 1)\)
W \(2=\exp \left(i s i g n * i * \sum^{*} * 2 * 2 *\right.\) piN 2\()\)
W \(3=\exp (i s i g n * i * j 3 * k 3 * 2 * p i N 3)\)

\section*{ARGUMENTS}

IO PT (input)
Integer specifying the operation to be perform ed:
IO PT \(=0\) com putes the trigonom etric weight table and factor table
IO PT \(=-1\) com putes forw ard FFT
IO \(\mathrm{PT}=+1\) com putes inverse FFT
N 1 (input)
Integer specifying length of the transform in the first dim ension. N 1 is m ostefficientw hen it is a productofsm allprim es. N \(1>=0\). U nchanged on exit.

N 2 (input)
Integer specifying length of the transform in the second dim ension. N 2 is m ostefficientw hen it is a productofsm allprim es. N \(2>=0\). U nchanged on exit.

N 3 (input)
Integer specifying length of the transform in the third dim ension. N 3 ism ostefficientw hen it is a productofsm allprim es. N \(3>=0\). U nchanged on exit.

SCALE (input)
D ouble precision scalarby which transform results are scaled. Unchanged on exit. SCALE is defaulted to 1.0 D 0 forF \(95 \mathbb{I N}\) TERFACE.

X (input) X is a double com plex array of dim ensions (LD X 1 , LDX2, N3) that contains input data to be transform ed.

LD X 1 (input)
firstdim ension ofX. LD X 1 >= N1 U nchanged on exit.

LD X 2 (input)
second dim ension of X . LD X \(2>=\mathrm{N} 2\) U nchanged on
exit.

Y (output)
Y is a double com plex array of dim ensions (LD Y 1, LD Y 2, N 3) that contains the transform results. X and \(Y\) can be the sam e array starting at the sam e m em ory location, in which case the input data are overw ritten by their transform results. O therw ise, it is assum ed that there is no overlap betw een \(X\) and \(Y\) in \(m\) em ory.

LD Y 1 (input)
firstdin ension of \(Y\). If \(X\) and \(Y\) are the same aray, LD Y \(1=\) LD X 1 Else LD Y \(1>=\mathrm{N} 1\) Unchanged on exit.

LD Y 2 (input)
second dim ension of \(Y\). If \(X\) and \(Y\) are the sam e aray, LD Y \(2=\) LD X 2 Else LD Y \(2>=\mathrm{N} 2\) U nchanged on exit.

TRIGS (input/output)
D ouble precision anay of length \(2 *(\mathbb{N} 1+\mathrm{N} 2+\mathrm{N} 3)\) that contains the trigonom etric w eights. The w eights are com puted when the routine is called w ith IO PT \(=0\) and they are used in subsequent calls w hen IO PT = 1 or IO PT =-1. U nchanged on exit.
IFAC (input/output)
Integer array ofdim ension at least \(3 * 128\) that
contains the factors of \(1, N 2\) and N 3 . The factors are com puted w hen the routine is called \(w\) ith IOPT = 0 and they are used in subsequent calls when IO PT = 1 or IO PT =-1. U nchanged on exit.

W ORK (w orkspace)
D ouble precision anray of dim ension at least \((2 * \mathrm{M} A X(\mathbb{N}, N 2, N 3)+32 * N 3) * N C P U S\) where \(N C P U S\) is the num ber of threads used to execute the routine.
The user can also choose to have the routine allocate its ow n w orkspace (see LW ORK).

LW ORK (input)
Integer specifying workspace size. If LW ORK \(=0\), the routine w illallocate its ow n w orkspace.
\(\mathbb{E R R}\) (output)
On exit, integer \(\mathbb{E R R}\) has one of the follow ing
values:
0 = norm alretum
\(-1=\mathbb{I O P T}\) is not 0,1 or -1
\(-2=\mathrm{N} 1<0\)
\(-3=N 2<0\)
\(-4=N 3<0\)
\(-5=(\mathbb{L D X} 1<\mathrm{N} 1)\)
\(-6=(\mathbb{L D X} 2<\mathrm{N} 2)\)
\(-7=(\lfloor D Y 1<N 1)\) or (LDY 1 notequal LD X 1 when \(X\) and \(Y\) are sam e amay)
\(-8=(L D Y 2<N 2)\) or ( \(\mathbb{L D} Y 2\) notequal LD X 2 when \(X\) and \(Y\) are sam e anray)
\(-9=(L W\) ORK not equal 0) and (LW ORK < \((2 * M A X(N, N 2, N 3)+16 * N 3) * N C P U S)\)
\(-10=m\) em ory allocation failed

\section*{SEE ALSO}
fft

\section*{CAUTIONS}

On exit, outputsubarray \(Y(1: L D Y 1,1 \mathbb{N} 2,1 \mathbb{N} 3)\) is overw ritten.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- SEE ALSO

\section*{NAME}
zfflzm -initialize the trigonom etric weight and factor tables or com pute the one-dim ensionalF astF ourier \(T\) ransform (forw ard or inverse) of a set of data sequenœes stored in a tw o-dim ensionaldouble com plex anay.

\section*{SYNOPSIS}

SUBROUTINE ZFFTZM (IOPT,N1,N2,SCALE,X,LDX,Y,LDY,TRIGS, \(\mathbb{F} A C, W\) ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER \(\mathbb{I O P T}, N 1, N 2, \operatorname{LD}, L D Y, \mathbb{F} A C(*), L W O R K, \mathbb{E R R}\)
DOUBLE COM PLEX X (LDX,*), Y (LDY,*)
DOUBLE PRECISION SCALE,TRIGS (*),W ORK (*)
SU BROUTINE ZFFTZM_64 (IOPT,N1,N2,SCALE, X,LDX,Y,LDY,TRIGS, FFAC,W ORK, LW ORK, \(\mathbb{E R R}\) )
\(\mathbb{N} T E G E R * 8 \mathbb{I} P T, N 1, N 2, L D X, L D Y, \mathbb{F} A C(*), L W O R K, \mathbb{E R R}\)
DOUBLE PRECISION SCALE,TRIGS (*), W ORK (*)
DOUBLE COM PLEX X (LDX,*), Y (LDY,*)

\section*{F95 INTERFACE}

SU BROUTINE FFTM (IOPT, \(\mathbb{N} 1], \mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S\), تAC,W ORK, [LW ORK], \(\mathbb{E R R}\) )
\(\mathbb{N}\) TEGER, \(\mathbb{N}\) TENT \((\mathbb{N})\) :: IOPT
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N})\),OPTIONAL ::N1,N2,LDX,LDY,LW ORK
REAL (8), \(\mathbb{N} T E N T(\mathbb{N})\),OPTIONAL :: SCALE
COM PLEX (8), \(\mathbb{N}\) TENT ( \(\mathbb{N}\) ), D \(\mathbb{M}\) ENSION \((:,:\) : : : X
COM PLEX (8), \(\mathbb{I N T E N T}(\mathrm{OUT}), \mathrm{D} \mathbb{M}\) ENSION (: : : : : : Y
REAL (8), \(\mathbb{N}\) TENT ( \(\mathbb{N} O U T\) ), D \(\mathbb{M}\) ENSION (:) ::TRIGS
\(\mathbb{N} T E G E R, \mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{I M}\) ENSION (:) :: \(\mathbb{F A C}\)

SU BROUTINE FFTM _64 (IOPT, \(\mathbb{N} 1], \mathbb{N} 2],[S C A L E], X,[L D X], Y,[L D Y], T R I G S\), \(\mathbb{F} A C, W\) ORK, [LW ORK], \(\mathbb{E R R})\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N}):: \operatorname{IOPT}\)
\(\mathbb{N}\) TEGER (8), \(\mathbb{N}\) TENT \((\mathbb{N})\), OPTIONAL : \(:\) N 1, N 2 , LDX , LDY, LW ORK
REAL (8), \(\mathbb{N} T E N T(\mathbb{N})\), OPTIONAL :: SCALE
COM PLEX (8), \(\mathbb{N} T E N T(\mathbb{N}), D \mathbb{M} E N S I O N(:,:):: X\)
COM PLEX (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M} E N S I O N(:,:):: Y\)
REAL (8), \(\mathbb{N} T E N T(\mathbb{N O U T}), D \mathbb{M} E N S I O N(:):: T R I G S\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathbb{N} \operatorname{TENT}(\mathbb{N} O U T), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{F A C}\)
REAL (8), \(\mathbb{N}\) TENT (OUT),D \(\mathbb{M}\) ENSION (:) ::W ORK
\(\mathbb{N}\) TEGER (8), \(\mathbb{I N}\) TENT (OUT) :: \(\mathbb{E R R}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zfflzm _ (int*iopt, int *m, int *n, double *scale, doublecom plex \({ }^{*} x\), int *ldx, doublecom plex \({ }^{*} y\), int *ldy, double *trigs, int *ifac, double *w ork, int *Iw ork, int *ien);
void zfftzm _64_ (long *iopt, long *m, long *n, double *scale, doublecom plex \({ }^{*} x\), long *ldx, doublecom plex *y, long *ldy, double *trigs, long *ifac, double *W ork, long *lw ork, long *ien);

\section*{PURPOSE}
zffizm initializes the trigonom etric weight and factor
tables or computes the one-dim ensional Fast Fourier
T ransform (forw ard or inverse) of a set of data sequences
stored in a tw o-dim ensional double com plex array:

N 1-1
\(Y(k, l)=\operatorname{SUM} W * X(j)\)
\(\ddagger 0\)
where
k ranges from 0 to \(\mathrm{N} 1-1\) and 1 ranges from 0 to \(\mathrm{N} 2-1\)
\(i=\operatorname{sqrt}(-1)\)
isign \(=1\) for inverse transform or -1 for forw ard transform
\(W=\exp \left(i s i g n * i^{\star} j^{\star} k \star 2 * p i \neq 1\right)\)

\section*{ARGUMENTS}

IOPT (input)
Integer specifying the operation to be perform ed:
IO PT \(=0\) com putes the trigonom etric weight table
and factor table
IO PT \(=-1\) com putes forw ard FFT
IO \(\mathrm{PT}=+1\) com putes inverse FFT
N 1 (input)
Integer specifying length of the input sequences. N 1 is m ostefficientw hen it is a product of sm all prim es. N \(1>=0\). U nchanged on exit.

N 2 (input)
Integer specifying num ber of input sequences. N 2 \(>=0\). Unchanged on exit.

SCALE (input)
D ouble precision scalarby w hich transform results are scaled. Unchanged on exit. SCALE is defaulted to 1.0 D 0 forF95 \(\mathbb{I N}\) TERFACE.

X (input) X is a double com plex aray of dim ensions (LD X ,
N 2 ) that contains the sequences to be transform ed stored in its colum ns.

LD X (input)
Leading dim ension of X . LD X \(>=\) N 1 U nchanged on exit.

Y (output)
\(Y\) is a double com plex array of dim ensions (LD \(Y\), N 2 ) that contains the transform results of the inputsequences. \(X\) and \(Y\) can be the sam \(e\) array starting at the sam e \(m\) em ory location, in which case the input sequences are overw rilten by their transform results. O therw ise, it is assum ed that there is no overlap betw een \(X\) and \(Y\) in \(m\) em ory.

\section*{LD Y (input)}

Leading dimension of \(Y\). If \(X\) and \(Y\) are the same aray, LD Y \(=\) LD X Else LD Y >= N 1 U nchanged on exit.

\section*{TRIGS (input/output)}

D ouble precision array of length \(2 * N 1\) that contains the trigonom etric w eights. The w eights are com puted w hen the routine is called w ith IOPT \(=0\) and they are used in subsequent calls w hen ID PT = 1 or IO PT = -1. U nchanged on exit.

FAC (input/output)

Integer anay of dim ension at least 128 that contains the factors of N 1 . The factors are com puted w hen the routine is called w ith \(\mathbb{I O}\) PT \(=0\) and they are used in subsequent calls w hen IO PT \(=1\) or IO PT \(=-1\). U nchanged on exit.

W ORK (w orkspace)
D ouble precision anay of dimension at least
2*N 1*N CPUS where NCPUS is the num ber of threads
used to execute the routine. The user can also
choose to have the routine allocate its own
w orkspace (see LW ORK).

LW ORK (input)
Integer specifying w orkspace size. IfLW ORK \(=0\),
the routine w illallocate its ow n w ork.space.

ERR (output)
On exit, integer \(\mathbb{F} R \mathrm{R}\) has one of the follow ing
values:
0 = norm alretum
\(-1=10 P T\) is not 0,1 or -1
\(-2=\mathrm{N} 1<0\)
\(-3=N 2<0\)
\(-4=(\llbracket D X<N 1)\)
\(-5=(L D Y<N 1)\) or (LD Y notequalLD \(X w\) hen \(X\) and \(Y\)
are sam e array)
\(-6=(\mathbb{L}\) ORK notequal0) and (LW ORK \(<2 * \mathrm{~N} 1 * \mathrm{NCPUS})\)
\(-7=m\) em ory allocation failed

\section*{SEE ALSO}
ffl

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgbbrd - reduce a com plex generalm -by-n band \(m\) atrix \(A\) to real upperbidiagonal form \(B\) by a unitary transform ation

\section*{SYNOPSIS}
```

SUBROUTINE ZGBBRD NECT,M,N,NCC,KL,KU,AB,LDAB,D,E,Q,LDQ,
PT,LDPT,C,LDC,W ORK,RW ORK,INFO)
CHARACTER * 1VECT
DOUBLE COM PLEX AB (LDAB,*),Q (LDQ,*), PT (LDPT,*), C (LDC,*),
W ORK (*)
INTEGERM,N,NCC,KL,KU,LDAB,LDQ,LDPT,LDC, INFO
DOUBLE PRECISION D (*),E (*),RW ORK (*)
SUBROUT\mathbb{NE ZGBBRD_64NECT,M,N,NCC,KL,KU,AB,LDAB,D,E,Q,LDQ,}
PT,LDPT,C,LDC,W ORK,RW ORK,INFO)
CHARACTER * 1VECT
DOUBLE COM PLEX AB (LDAB,*),Q (LDQ ,*), PT (LDPT,*), C (LD C,*),
W ORK (*)
INTEGER*8M,N,NCC,KL,KU,LDAB,LDQ,LD PT,LD C, IN FO
DOUBLE PRECISION D (*),E (*),RW ORK (*)

```

\section*{F95 INTERFACE}

SUBROUTINE GBBRD \(\operatorname{NECT}, \mathrm{M}, \mathbb{N}], \mathbb{N C C}], K L, K U, A B,[L D A B], D, E, Q\),
[LDQ],PT, [LDPT],C, [LDC], [W ORK], [RWORK], [ \(\mathbb{N} F O]\) )
CHARACTER (LEN=1)::VECT
COMPLEX (8),D IM ENSION (:) ::W ORK
COMPLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::AB,Q,PT,C
\(\mathbb{N} T E G E R:: M, N, N C C, K L, K U, L D A B, L D Q, L D P T, L D C, \mathbb{N} F O\)
REAL (8), D IM ENSION (:) ::D , E, RW ORK

SUBROUTINE GBBRD_64 \(N E C T, M, \mathbb{N}], \mathbb{N C C}], K L, K U, A B,[L D A B], D, E\), Q , [LD Q ], PT, [LDPT], C , [LDC], [W ORK], [RW ORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::VECT
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{I M} \operatorname{ENSION}(:,:\) : : \(:\) AB, \(\mathrm{Q}, \mathrm{PT}, \mathrm{C}\)
\(\mathbb{N}\) TEGER (8) :: M , N, NCC, KL, KU, LDAB, LDQ, LDPT, LD C , \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D ,E,RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zgbbord (charvect, intm, intn, intncc, int kl, int ku, doublecom plex *ab, int ldab, double *d, double *e, doublecom plex *q, int ldq, doublecom plex *pt, int ldpt, doublecom plex * C , int ldc, int *info);
void zgbbrd_64 (charvect, long m, long n, long ncc, long kl, long ku, doublecom plex *ab, long ldab, double *d, double *e, doublecom plex *q, long ldq, doublecom plex *pt, long ldpt, doublecom plex *C, long ldc, long *info);

\section*{PURPOSE}
zgbbrd reduces a com plex generalm -by-n band \(m\) atrix \(A\) to realupperbidiagonal form \(B\) by a unitary transform ation: \(Q^{\prime}\)
* \(A * P=B\).

The routine com putes B, and optionally form s Q or P', or com putes \(Q\) *C for given \(m\) atrix \(C\).

\section*{ARGUMENTS}

\section*{VECT (input)}

Specifies w hether ornot the \(m\) atrices \(Q\) and \(P\) 'are
to be form ed. = N ': do not form Q orP';
\(=Q^{\prime}\) : form Q only;
\(=P^{\prime}:\) form \(P^{\prime}\) only;
\(=B^{\prime}\) : form both.

M (input) The num ber of row s of the \(m\) atrix \(A . M>=0\).

N (input) The num ber of colum ns of them atrix \(A . N>=0\).

NCC (input)

The num ber of colum ns of the \(m\) atrix \(\mathrm{C} . \mathrm{NCC}>=0\).
\(K L\) (input)
The num ber of subdiagonals of the m atrix A. \(\mathrm{KL}>=\) 0 .

KU (input)
The num ber of superdiagonals of the m atrix A. KU \(>=0\).

AB (input/output)
On entry, the \(m\)-by \(-n\) band \(m\) atrix \(A\), stored in row \(s\) 1 to \(\mathrm{KL}+\mathrm{KU}+1\). The \(j\) th colum n of A is stored in the \(j\) th colum \(n\) of the array \(A B\) as follows: \(A B(k u+1+i-j)=A(i, j)\) for \(\max (1, j\) \(\mathrm{ku})<=i<=\mathrm{m}\) in \((\mathrm{m}, \dot{\mathrm{j}} \mathrm{kl} \mathrm{l})\). On exit, A is overw ritten by values generated during the reduction.

LDAB (input)
The leading dim ension of the anay A. LD AB >= \(K L+K U+1\).

D (output)
The diagonalelem ents of the bidiagonalm atrix B.

E (output)
The superdiagonal elem ents of the bidiagonal \(m\) atrix B.

Q (output)
If VECT \(=Q\) 'or \(B\) ', the \(m\)-by \(m\) unitary \(m\) atrix \(Q\).
If VECT \(=N\) 'or \(P\) ', the array \(Q\) is not referenced.

LD Q (input)
The leading dim ension of the amay \(Q\). LDQ >= \(m \operatorname{ax}(1, M)\) ifVECT \(=Q\) 'or \(B ; L D Q>=1\) otherw ise.

PT (output)
IfVECT \(=P^{\prime}\) or \(B^{\prime}\), the n -by-n unitary \(m\) atrix \(P^{\prime}\). If VECT \(=N\) 'or \(Q\) ', the amay PT is not referenced.

LDPT (input)
The leading dim ension of the array PT. LD PT >= \(\mathrm{max}(1, \mathrm{~N})\) if VECT \(=\mathrm{P}\) 'or \(\mathrm{B} ; \mathrm{LDPT}>=1\) otherw ise.

C (input/output)
On entry, an \(m\)-by-nocm atrix C. On exit, \(C\) is
overw ritten by Q *C. C is notreferenced if \(\mathrm{NCC}=\) 0 .

LD C (input)
The leading dim ension of the aray \(C\). LD C >= \(m a x(1, M)\) if \(N C C>0 ; L D C>=1\) if \(N C C=0\).
W ORK (w orkspace)
dim ension ( \(\mathrm{M} A X(M, N))\)

RW ORK (w orkspace)
dim ension ( \(\mathrm{M} A X(\mathrm{M}, \mathrm{N})\) )
\(\mathbb{N} F O\) (output)
= 0: successfulexit.
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\) th argum enthad an illegalvałue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgbcon -estim ate the reciprocal of the condition num ber of a com plex generalband \(m\) atrix \(A\), in either the 1 -norm or the infinity-norm,

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGBCON NORM,N,NSUB,NSUPER,A,LDA, PPIVOT,ANORM,}
RCOND,W ORK,W ORK2,INFO)
CHARACTER * 1 NORM
DOUBLE COM PLEX A (LDA,*),W ORK (*)
INTEGERN,NSUB,NSUPER,LDA, INFO
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION W ORK2 (*)
SUBROUT\mathbb{NE ZGBCON_64 NORM,N,NSUB,NSUPER,A,LDA,\mathbb{PIVOT,ANORM,}}\mathbf{N},
RCOND,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 NORM
DOUBLE COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGER*8N,NSUB,NSUPER,LDA,NNFO}
INTEGER*8 \mathbb{PIVOT (*)}
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION W ORK2 (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE GBCON $\mathbb{N} O R M, \mathbb{N}], N S U B, N S U P E R, A,[L D A], \mathbb{P} \mathbb{I} O T, A N O R M$, RCOND, $[\mathbb{W}$ ORK], [WORK2], [ $\mathbb{N} F O$ ])

```

CHARACTER (LEN=1) ::NORM
COMPLEX (8),D IM ENSION (:) ::W ORK

COM PLEX (8), D \(\mathbb{I M} E N S I O N(:,:):: A\)
\(\mathbb{N}\) TEGER :: N , N SUB, N SUPER, LD A , \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{I} O T\)
REAL (8) ::ANORM,RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ORK2

SU BROUTINE GBCON_64 \(\mathbb{N} O R M,[N], N S U B, N S U P E R, A,[L D A], \mathbb{P} \mathbb{I} O T, A N O R M\), RCOND, \(\mathbb{W} O R K],[\mathbb{W} O R K 2],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::NORM
COMPLEX (8), D IM ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER (8) :: N , N SUB , N SUPER, LD A , \(\mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT}\)
REAL (8) ::ANORM,RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zgboon (charnorm, intn, intnsub, int nsuper, doublecom plex *a, int lda, int *ịivot, double anorm , double *rcond, int *info);
void zgbcon_64 (char norm , long n, long nsub, long nsuper, doublecom plex *a, long lda, long *ípivot, double anorm , double *rcond, long *info);

\section*{PURPOSE}
zgbcon estim ates the reciprocal of the condition num ber of a com plex general band m atrix \(A\), in either the 1 -norm or the infinity-norm, using the LU factorization computed by CGBTRF.

A \(n\) estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as
\(\operatorname{RCOND}=1 /(\operatorname{nom}(A) * \operatorname{norm}(\operatorname{inv}(A)))\).

\section*{ARGUMENTS}

\section*{NORM (input)}

Specifies w hether the 1 -norm condition num ber or the infinity-norm condition num ber is required:
= 1'or O': 1-norm;
\(=\mathrm{I}^{\prime}: \quad\) Infinity-norm.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

N SUB (input)
The num ber of subdiagonals w ithin the band of A. N SUB \(>=0\).

\section*{N SU PER (input)}

The num ber of superdiagonals w ithin the band of A. N SU PER \(>=0\).

A (input) D etails of the LU factorization of the band \(m\) atrix
A, as com puted by CGBTRF. U is stored as an upper triangularband \(m\) atrix \(w\) ith N SU B +N SU PER superdiagonals in rows 1 to NSUB+NSUPER+1, and them ultipliers used during the factorization are stored in row SN SU B + N SU PER +2 to \(2 *\) N SU B + N SU PER +1 .

\section*{LDA (input)}

The leading dim ension of the array A . LDA >= \(2 * N\) SU B + N SU PER +1 .

PIVOT (input)
The pivot indices; for \(1<=i<=N\), row \(i\) of the \(m\) atrix \(w\) as interchanged w ith row \(\mathbb{P} \mathbb{I V}\) T (i).

ANORM (input)
IfNORM = I' or \(\mathrm{D}^{\prime}\) ', the 1 -norm of the original \(m\) atrix \(A\). IfNORM = 'I', the infinity-norm of the originalm atrix A.

RCOND (output)
The reciprocal of the condition number of the \(m\) atrix \(A\), computed as RCOND \(=1 /(\) norm (A) * norm (inv (A))).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgbequ - com pute row and colum n scalings intended to equilibrate an M -by -N band matrix A and reduce its condition num ber

\section*{SYNOPSIS}
```

SUBROUTINE ZGBEQU M,N,KL,KU,A,LDA,R,C,ROW CN,
COLCN,AMAX,INFO)
DOUBLE COM PLEX A (LDA,*)
\mathbb{NTEGER M,N,KL,KU,LDA, IN FO}
DOUBLE PRECISION ROW CN,COLCN,AMAX
DOUBLE PRECISION R (*),C (*)
SUBROUTINE ZGBEQU_64M,N,KL,KU,A,LDA,R,C,ROW CN,
COLCN,AMAX,INFO)
D OUBLE COM PLEX A (LDA,*)
\mathbb{NTEGER*8M,N,KL,KU,LDA,}\mathbb{N}FO
DOUBLE PRECISION ROW CN,COLCN,AMAX
DOUBLE PRECISION R (*),C (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE GBEQU (M ], $\mathbb{N}], K L, K U, A,[L D A], R, C$, ROW CN, COLCN,AMAX, $\mathbb{N} F O]$ )
COM PLEX (8),D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R:: M, N, K L, K U, L D A, \mathbb{N} F O$
REAL (8) ::ROW CN, COLCN,AMAX
REAL (8),D $\mathbb{M}$ ENSION (:) ::R,C

```


COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{M}, \mathrm{N}, \mathrm{KL}, \mathrm{KU}, \mathrm{LD} A, \mathbb{N} F \mathrm{O}\)
REAL (8) ::ROW CN,COLCN,AMAX
REAL (8),D \(\mathbb{I}\) ENSION (:) ::R,C

\section*{C INTERFACE}
\#include <sunperfh>
void zgbequ (intm, intn, intkl, intku, doublecom plex *a, int lda, double *r, double *c, double *row cn, double *colcn, double *am ax, int*info);
void zgbequ_64 (long m, long n , long kl , long ku, doublecom plex *a, long lda, double *r, double *c, double *row cn, double *colen, double *am ax, long *info);

\section*{PURPOSE}
zgbequ com putes row and colum n scalings intended to equilibrate an M -by N band \(m\) atrix \(A\) and reduce its condition number. \(R\) retums the row scale factors and \(C\) the colum \(n\) scale factors, chosen to try to \(m\) ake the largestelem ent in each row and colum \(n\) of the \(m\) atrix \(B \quad w i t h\) elements B \((i, j)=R(i) \star A(i, 7) * C(i)\) have absolute value 1 .

R (i) and C (i) are restricted to be betw een SM LN UM = sm allest safe num ber and B IG N UM = largest safe num ber. U se of these scaling factors is notguaranteed to reduce the condition num berofA butw orksw ellin practice.

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of collm ns of the \(m\) atrix \(\mathrm{A} . \mathrm{N}>=0\).

KL (input)
The num ber of subdiagonals w ithin the band of A. \(\mathrm{KL}>=0\).

KU (input)
The num ber of superdiagonals w ithin the band of A.
K U >=0.

A (input) The band m atrix A, stored in row s 1 to \(K L+K U+1\).
The jth colum n of A is stored in the \(j\) th column
of the array A as follow s: A \((k u+1+i-j)=A(i, 7)\) form ax \((1, j \mathrm{jku})<=i<=m\) in \((m, j+k l)\).

LD A (input)
The leading dim ension of the aray A. LD A >= K L+KU +1.

R (output)
If \(\mathbb{N} F O=0\), or \(\mathbb{N} F O>M, R\) contains the row scale factors forA.
C (output)
If \(\mathbb{N} F O=0, C\) contains the colum \(n\) scale factors
forA.

ROW CN (output)
If \(\mathbb{I N F O}=0\) or \(\mathbb{N} F O>M, R O W C N\) contains the ratio
of the sm allest \(R(i)\) to the largest \(R\) (i). If
ROW CN >= 0.1 and \(A M A X\) is neither too large nortoo sm all, it is notw orth scaling by \(R\).

COLCN (output)
If \(\mathbb{N} F O=0, C O L C N\) contains the ratio of the sm allest C (i) to the largestC (i). IfC O LCN >=0.1, it is notw orth scaling by \(C\).

AM AX (output)
A bsolute value of largestm atrix elem ent. IfA M A X
is very close to overflow orvery close to under-
flow , the \(m\) atrix should be scaled.
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an ille-
galvalue
> 0 : if \(\mathbb{N} F O=i\), and \(i\) is
\(<=\mathrm{M}\) : the i-th row ofA is exactly zero
\(>M\) : the ( -H ) -th colum n of A is exactly zero

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zg.bm \(v\)-perform one of the \(m\) atrix-vectoroperations \(y:=\) alpha*A *x + beta*y, ory : alpha*A *x + beta*y, or \(y:=\) alpha*cong (A ')*x + beta*y

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGBMV (TRANSA,M,N,NSUB,NSUPER,ALPHA,A,LDA,X,INCX,}
BETA,Y,\mathbb{NCY)}

```
CHARACTER * 1 TRANSA
D OUBLE COM PLEX ALPHA,BETA

\(\mathbb{N}\) TEGERM,N,NSUB,NSUPER,LDA, \(\mathbb{N} C X, \mathbb{N} C Y\)
SUBROUTINE ZGBMV_64 (TRANSA, M,N,NSUB,NSUPER,ALPHA,A,LDA, X,
    \(\mathbb{N} C X, B E T A, Y, \mathbb{N} C Y)\)
CHARACTER * 1 TRANSA
DOUBLE COM PLEX ALPHA,BETA
D OUBLE COM PLEX A (LDA,*), X (*), Y ( \({ }^{\star}\) )
\(\mathbb{N}\) TEGER*8M,N,NSUB,NSUPER,LDA, \(\mathbb{N} C X, \mathbb{N} C Y\)

\section*{F95 INTERFACE}

SU BROUTINE GBM V ([TRANSA], \(\mathbb{M}], \mathbb{N}], N S U B, N S U P E R, A L P H A, A,[L D A], X\), \([\mathbb{N C X}], B E T A, Y,[\mathbb{N C Y}])\)

CHARACTER (LEN=1) ::TRANSA
COM PLEX (8) ::ALPHA,BETA
COMPLEX (8),D IM ENSION (:) ::X,Y
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, N S U B, N S U P E R, L D A, \mathbb{N} C X, \mathbb{N} C Y\)

SUBROUTINE GBMV_64 ([TRANSA], M ], \(\mathbb{N}], N \operatorname{SUB}, N \operatorname{SUPER}, A L P H A, A,[L D A]\), \(\mathrm{X},[\mathbb{N C X}], \mathrm{BETA}, \mathrm{Y},[\mathbb{N C Y}])\)

CHARACTER (LEN=1) ::TRANSA
COM PLEX (8) ::ALPHA,BETA
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::X,Y
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : A
\(\mathbb{N} T E G E R(8):: M, N, N S U B, N S U P E R, L D A, \mathbb{N} C X, \mathbb{N} C Y\)

\section*{C INTERFACE}
\#include < sunperfh>
void zgbm v (Chartransa, intm, intn, intnsub, int nsuper, doublecom plex *alpha, doublecom plex *a, int lda, doublecom plex *x, int incx, doublecom plex *beta, doublecom plex *y, int incy);
void zgbm v_64 (char transa, long m, long n, long nsub, long nsuper, doublecom plex *alpha, doublecom plex *a, long lda, doublecom plex *x, long incx, doublecom plex *beta, doublecom plex *y, long incy);

\section*{PURPOSE}
zgbm v perform s one of the \(m\) atrix-vector operations \(y:=\) alpha*A*x + beta*y, ory : alpha*A *x + beta*y, or \(y:=\) alpha*con'g (A')*x + beta*y where alpha and beta are scalars, \(x\) and \(y\) are vectors and \(A\) is an \(m\) by \(n\) band \(m\) atrix, w ith nsub sub-diagonals and nsuper super-diagonals.

\section*{ARGUMENTS}

TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s :
TRANSA \(=\mathrm{N}^{\prime}\) or \(\mathrm{h}^{\prime} \mathrm{y}:=\) alpha*A \({ }^{*} \mathrm{x}+\) beta y y .
TRANSA \(=\) ' 'or \(t^{\prime} y=a l p h a * A ~ * x+b e t a * y\).
TRANSA = C'ort' \(y=\) alpha*cong (A')*x + beta*y.
U nchanged on exit.
TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.

M (input)
O n entry, M specifies the num berof row s of the \(m\) atrix A. M mustbe at least zero. U nchanged on exit.

N (input)
On entry, N specifies the num ber of colum ns of the m atrix A. N m ustbe at least zero. U nchanged on exit.

NSUB (input)
On entry, NSUB specifies the num ber of subdiagonals of the matrix A.NSUB mustsatisfy 0 le. N SU B . U nchanged on exit.

N SU PER (input)
On entry, N SU PER specifies the num ber of superdiagonals of them atrix A.N SU PER m ust satisfy 0 le. N SU PER. U nchanged on exit.

\section*{A LPHA (input)}

On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
Before entry, the leading (nsub + nsuper+1) by \(n\) part of the array A m ust contain the matrix of coefficients, supplied colum \(n\) by colum \(n\), w ith the leading diagonal of the \(m\) atrix in row (nsuper +1 ) of the array, the first super-diagonal starting at position 2 in row nsuper, the firstsubdiagonal starting atposition 1 in row (nsuper + 2 ), and so on. Elem ents in the aray A that do not conespond to elem ents in the band matrix (such as the top leftnsuperby nsupertriangle) are not referenced. The follow ing program segm ent \(w\) illtransfer a band \(m\) atrix from conventional full \(m\) atrix storage to band storage:
\[
\begin{aligned}
& \text { DO } 20, \mathrm{~J}=1, \mathrm{~N} \\
& \mathrm{~K}=\mathrm{N} \text { SUPER }+1-\mathrm{J} \\
& \text { DO } 10, \mathrm{I}=\mathrm{MAX}(1, \mathrm{~J}-\mathrm{NSUPER}), \mathrm{M} \mathbb{N}(\mathrm{M}, \mathrm{~J}+ \\
& \text { NSUB }) \\
& \quad \text { A }(\mathrm{K}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(I, J) \\
& 10 \quad \text { CONTINUE } \\
& 20 \text { CONTINUE }
\end{aligned}
\]

U nchanged on exit.

LDA (input)
O \(n\) entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A m ust be at least (nsub + nsuper+1). U nchanged on exit.

X (input)
\((1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C X))\) when TRANSA = N 'or h' and at least ( \(1+(m-1) * a b s(\mathbb{N C X})\) )
otherw ise. Before entry, the increm ented anray \(X\) \(m\) ust contain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X\) m ustnotbe zero. U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then \(Y\) need not.be seton input. U nchanged on exit.

Y (input/output)
\((1+(m-1) * \operatorname{abs}(\mathbb{N} C Y))\) when TRANSA \(=\mathrm{N}\) 'or \(h^{\prime}\) and at least ( \(\left.1+(\mathrm{n}-1)^{*} \mathrm{abs}(\mathbb{N} C Y)\right)\)
otherw ise. B efore entry, the increm ented anay \(Y\) \(m\) ustcontain the vectory. On exit, \(Y\) is overw ritten by the updated vectory.
\(\mathbb{N} C Y\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y . \mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgbrfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is banded, and provides errorbounds and backw ard error estim ates for the solution

\section*{SYNOPSIS}
```

SUBROUTINE ZGBRFS (TRANSA,N,KL,KU,NRHS,A,LDA,AF,LDAF,
\mathbb{PIVOT,B,LDB,X,LDX,FERR,BERR,W ORK,W ORK 2, \mathbb{NFO)}}\mathbf{N},\textrm{L}

```
CHARACTER * 1 TRANSA
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK ( \({ }^{*}\) )
\(\mathbb{N}\) TEGER N,KL,KU,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{P} \mathbb{V} O T\left({ }^{( }\right)\)
D OUBLE PRECISION FERR (*), BERR (*) , W ORK 2 (*)
SUBROUTINE ZGBRFS_64 (TRANSA,N,KL,KU,NRHS,A,LDA,AF,LDAF,
    \(\mathbb{P} \mathbb{I V O T}, \mathrm{B}, \mathrm{LD} B, \mathrm{X}, \mathrm{LD} \mathrm{X}, \mathrm{FERR}, \mathrm{BERR}, \mathrm{W} O R \mathrm{~K}, \mathrm{~W} O R K 2, \mathbb{N} F O)\)
CHARACTER * 1 TRANSA
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
\(\mathbb{N}\) TEGER*8N,KL,KU,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{I V O T}\) ( \({ }^{*}\) )
DOUBLE PRECISION FERR (*), BERR (*),W ORK 2 (*)

\section*{F95 INTERFACE}

SU BROUTINE GBRFS ([TRANSA], \(\mathbb{N}], K L, K U, N R H S], A,[L D A], A F\), [LDAF], \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LD} \mathrm{B}], \mathrm{X},[\mathrm{LDX}], F E R R, B E R R,[\mathbb{W}\) ORK ], [W ORK 2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::TRANSA
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{I}\) ENSION (: : : : : A , AF, B, X
\(\mathbb{N}\) TEGER : \(: N, K L, K U, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL (8), D \(\mathbb{I M} E N S I O N(:):: F E R R, B E R R, W\) ORK 2

SU BROUT巩E GBRFS_64 ([TRANSA], \(\mathbb{N}], K L, K U, \mathbb{N} R H S], A,[L D A]\), \(A F,[L D A F], \mathbb{P} \mathbb{I} O T, B,[L D B], X,[L D X], F E R R, B E R R,[W O R K]\), [W ORK2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::TRANSA
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{I}\) M ENSION (:,:) ::A,AF,B,X


REAL (8), D \(\mathbb{I M}\) ENSION (:) ::FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void zgbrfs (char transa, intn, int kl, int ku, int nrhs, doublecom plex *a, intlda, doublecom plex *af, int ldaf, int *ipivot, doublecom plex *b, int ldb, doublecom plex *x, int ldx, double *ferr, double *berr, int *info);
void zgbrfs_64 (chartransa, long n, long kl, long ku, long nrhs, doublecom plex *a, long lda, doublecom plex *af, long ldaf, long *ipívot, doublecom plex *b, long ldb, doublecom plex *x, long ldx, double * ferrr, double *berr, long *info);

\section*{PURPOSE}
zgbrfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is banded, and provides errorbounds and backw ard error estim ates for the solution.

\section*{ARGUMENTS}

TRANSA (input)
Specifies the form of the system of equations:
\(=\mathrm{N}: A * X=B \quad\) ( 0 transpose)
\(=T\) ': \(A * * T \times=B \quad\) (Transpose)
\(=C\) ': \(A * * H * X=B \quad\) (C onjugate transpose)

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

KL (input)
The num berof subdiagonals w ithin the band of A. \(\mathrm{KL}>=0\).

KU (input)
The num ber of superdiagonals w thin the band of A. \(K U>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the \(m\) atrioes B and X. NRHS \(>=0\).
A (input) The originallband matrix A, stored in row s 1 to \(K L+K U+1\). The \(j\) th column of \(A\) is stored in the \(j\) th colum \(n\) of the array A as follow s: A (ku+1+i\(j, j)=A(i, j)\) form ax \((1, j k u)<=i<=m\) in \((n, j+k l)\).

LD A (input)
The leading dim ension of the aray A. LDA >= \(K L+K U+1\).

\section*{AF (input)}

D etails of the LU factorization of the band \(m\) atrix
A, as com puted by C GBTRF. U is stored as an upper triangularband \(m\) atrix \(w\) th \(K L+K U\) superdiagonals in row \(s 1\) to \(K L+K U+1\), and the \(m\) ultipliers used during the factorization are stored in row S \(K L+K U+2\) to \(2 * K L+K U+1\).

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(2 * K L * K U+1\).
\(\mathbb{P I V O T}\) (input)
The pinot indices from CGBTRF; for \(1<=i<=\mathrm{N}\), row i of the \(m\) atrix \(w\) as interchanged w ith row \(\mathbb{P}\) IV OT (i).
\(B\) (input) The righthand side \(m\) atrix \(B\).
LD B (input)
The leading dim ension of the amay \(B\). LD B >= \(\max (1, \mathbb{N})\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by CGBTRS. On exit, the im proved solution \(m\) atrix \(X\).

\section*{LD X (input)}

The leading dim ension of the array X . LD X >= \(\max (1, N)\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X()\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix \(X\) ). If \(X T R U E\) is the true solution corresponding to \(X(\mathcal{H})\), FERR ( \(\mathcal{1}\) ) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in \((X(\mathcal{O})-X\) TRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

BERR (output)
The com ponentw ise relative backw ard error of each solution vector \(X\) ( \(j\) ) (i.e., the sm allest relative change in any elem entofA orB thatm akes X (i) an exactsolution).

W ORK (w orkspace)
dim ension \((2 * N)\)

W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N N}\) FO \(=-i\), the \(i\)-th argum enthad an ille-
galvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zgbsv - com pute the solution to a com plex system of linear equations \(A * X=B\), where \(A\) is a band \(m\) atrix of order \(N\) \(w\) th \(K L\) subdiagonals and \(K U\) superdiagonals, and \(X\) and \(B\) are N -by-N R H S m atrices

\section*{SYNOPSIS}

DOUBLE COM PLEX A (LDA, *), B (LDB, \({ }^{\text {( }) ~}\)
\(\mathbb{N}\) TEGER N, KL,KU,NRHS,LDA,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{P} \mathbb{V} O T\left({ }^{*}\right)\)
SU BROUTINE ZGBSV_64 \(\mathbb{N}, K L, K U, N R H S, A, L D A, \mathbb{P} \mathbb{I V O T}, B, L D B\),
    \(\mathbb{N} F O\) )
DOUBLE COM PLEX A (LDA,*), B (LDB, \(\left.{ }^{*}\right)\)
\(\mathbb{N}\) TEGER* \(8 \mathrm{~N}, \mathrm{KL}, \mathrm{KU}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} A, L D B, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER * \(8 \mathbb{P} \mathbb{I V O T}\) ( \(\left.{ }^{( }\right)\)

\section*{F95 INTERFACE}

SU BROUTINE GBSV ( \(\mathbb{N}], K L, K U, \mathbb{N} R S], A,[L D A], \mathbb{P} \mathbb{I V} \operatorname{T}, \mathrm{B},[\mathrm{LDB}]\), [ \(\mathbb{N}\) FO ])

COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A,B
\(\mathbb{N} T E G E R:: N, K L, K U, N R H S, L D A, L D B, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
 [LDB], [ \(\mathbb{N} F O]\) )

\section*{C INTERFACE}
\#include <sunperfh>
void zgbsv (intn, intkl, intku, int nrhs, doublecom plex
*a, int lda, int *ịívot, doublecom plex *b, int ldb, int*info);
void zgbsv_64 (long n, long kl, long ku, long nihs, doublecom plex *a, long lda, long *ìpívot, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
zgbsv com putes the solution to a com plex system of linear equations \(A\) * \(\mathrm{X}=\mathrm{B}\), where A is a band m atrix of order N \(w\) ith \(K L\) subdiagonals and \(K U\) superdiagonals, and \(X\) and \(B\) are N -by-N R H S m atrices.

The LU decom position \(w\) ith partialpivoting and row interchanges is used to factorA as A \(=\mathrm{L} * \mathrm{U}\), where L is a product of perm utation and unit low er triangularm atrioes \(w\) ith \(K L\) subdiagonals, and \(U\) is upper triangularw ith \(K L+K U\) superdiagonals. The factored form ofA is then used to solve the system of equations \(A * X=B\).

\section*{ARGUMENTS}

N (input) The num ber of linearequations, i.e., the order of the matrix \(A . N>=0\).

KL (input)
The num berof subdiagonals w ithin the band of A. \(\mathrm{KL}>=0\).

KU (input)
The num ber of superdiagonals w ithin the band of A.
\(K U>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS >=0.

A (input/output)

On entry, them atrix A in band storage, in row \(s\) \(K L+1\) to \(2 * K L+K U+1\); row s 1 to \(K L\) of the anray need notbe set. The \(j\) th column ofA is stored in the \(j\) th column of the array \(A\) as follows:
\(A(K L+K U+1+i-j, j)=A(i, j)\) for \(\max (1, j\) K \(U\) ) \(<=i<=m\) in ( \(N\), \(\mathfrak{j}+\mathrm{K} L\) ) On exit, details of the factorization: \(U\) is stored as an upper triangular band \(m\) atrix w ith \(K L+K U\) superdiagonals in row \(s\) to \(\mathrm{K} L+\mathrm{KU}+1\), and the \(\mathrm{m} u\) ltipliers used during the factorization are stored in rows KL+KU+2 to \(2 \star K L+K U+1\). See below for further details.

LD A (input)
The leading dim ension of the array A. LDA >= \(2 \star K L+K U+1\).
\(\mathbb{P I V O T}\) (output)
The pivot indices that define the perm utation \(m\) atrix \(P\); row i of the \(m\) atrix \(w\) as interchanged w th row \(\mathbb{P} \mathbb{I V O T}\) (i).

B (input/output)
On entry, the \(\mathrm{N}-\) by -N RH S righthand side m atrix B . On exit, if \(\mathbb{N} F O=0\), the \(N\)-by-NRHS solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the anay \(B\). LD B \(>=\) \(\max (1, N)\).
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue \(>0:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{U}(\mathrm{i}, \mathrm{i})\) is exactly zero. The factorization has been com pleted, but the factor U is exactly singular, and the solution has notbeen com puted.

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, when \(M=N=6, K L=2, K U=1\) :

On entry: On exit:

> u36
* * + + + + * * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55 u66
a21 a32 a43 a54 a65 * m 21 m 32 m 43 m 54 m 65
*
a31 a42 a53 a64 * * m 31 m 42 m 53 m 64 *

A nay elem entsm arked * are notused by the routine; ele\(m\) entsm arked + need notbe seton entry, butare required by the routine to store elem ents of \(U\) because of fill-in resulting from the row interchanges.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgbsvx -use the LU factorization to com pute the solution to a complex system of linearequations \(\mathrm{A} * \mathrm{X}=\mathrm{B}, \mathrm{A} * * \mathrm{~T} * \mathrm{X}=\) B, orA **H * \(\mathrm{X}=\mathrm{B}\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGBSVX (FACT,TRANSA,N,KL,KU,NRHS,A,LDA,AF,}
LDAF,\mathbb{PIVOT,EQUED,R,C,B,LDB,X,LDX,RCOND,FERR,}
BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 FACT,TRANSA,EQUED
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
INTEGERN,KL,KU,NRHS,LDA,LDAF,LDB,LDX,INFO
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION RCOND
DOUBLE PRECISION R (*),C (*),FERR (*),BERR (*),W ORK 2 (*)
SU BROUTINE ZGBSVX_64(FACT,TRANSA,N,KL,KU,NRHS,A,LDA,AF,
LDAF,\mathbb{PIVOT,EQUED,R,C,B,LDB,X,LDX,RCOND,FERR,}
BERR,W ORK,W ORK2,\mathbb{NFO)}

```
CHARACTER * 1 FACT,TRANSA, EQUED
DOUBLE COM PLEX A (LDA ,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK ( \({ }^{*}\) )
\(\mathbb{N}\) TEGER*8N,KL,KU,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER*8 \(\mathbb{P} \mathbb{I V O T}\) ( \({ }^{*}\) )
DOUBLE PRECISION RCOND
DOUBLE PRECISIONR (*), C (*), FERR (*), BERR (*), WORK 2 (*)

\section*{F95 INTERFACE}

SUBROUTINE GBSVX (FACT, [TRANSA], \(\mathbb{N}], K L, K U, \mathbb{N R H S}], A,[L D A]\),

AF, [LDAF], \(\mathbb{P} \mathbb{I} O T, E Q U E D, R, C, B,[L D B], X,[L D X]\), RCOND,FERR,BERR, [W ORK ], [W ORK 2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::FACT,TRANSA,EQUED
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::A,AF,B,X
\(\mathbb{N} T E G E R:: N, K L, K U, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8) :: RCOND
REAL (8),D IM ENSION (:) ::R,C ,FERR,BERR,W ORK2

SU BROUTINE GBSVX_64 (FACT, [TRANSA], \(\mathbb{N}], K L, K U, N R H S], A\), [LDA],AF, [LDAF], \(\mathbb{P} \mathbb{I V O T}, E Q U E D, R, C, B,[L D B], X,[L D X]\), RCOND,FERR,BERR, [W ORK], [W ORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::FACT,TRANSA,EQUED
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : A, AF, B, X
\(\mathbb{N}\) TEGER (8) ::N,KL,KU,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL (8) ::RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::R,C,FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void zgbsvx (char fact, chartransa, intn, int \(k\) l, int ku, intnrhs, doublecom plex *a, int lda, doublecom plex
*af, int ldaf, int *ipivot, charequed, double *r, double \({ }^{*}\) c, doublecom plex *b, int ldb, doublecom plex *x, intldx, double *rcond, double *ferr, double *berr, int *info);
void zgbsvx_64 (char fact, char transa, long n, long kl, long ku, long nihs, doublecom plex *a, long lda, doublecom plex *af, long ldaf, long *ịíivot, char equed, double \({ }^{*}\) r, double \({ }^{*}\) c, doublecom plex \({ }^{*}\) b, long ldb, doublecomplex *x, long ldx, double *rcond, double * ferr, double *berr, long *info);

\section*{PURPOSE}
zgbsvx uses the LU factorization to com pute the solution to a complex system of linearequations \(A * X=B, A * * T\) * \(=\) \(B\), or \(A * * H * X=B\), where \(A\) is a band \(m\) atrix of order \(N\) w ith \(K L\) subdiagonals and \(K U\) superdiagonals, and \(X\) and \(B\) are \(N^{-}\) by-N RH S m atrices.

Enorbounds on the solution and a condition estim ate are
also provided.
The follow ing steps are perform ed by this subroutine:
1. IfFACT = E', real scaling factors are com puted to equilibrate
the system :
TRANS \(=N^{\prime}: \operatorname{diag}(R) * A * \operatorname{diag}(C) \quad * i n v(d i a g(C)) * X=\) \(\operatorname{diag}(\mathbb{R}) * B\)

TRANS \(=T:(\operatorname{diag}(R) * A * \operatorname{diag}(C)) * * T * \operatorname{inv}(\operatorname{diag}(R)) * X=\) diag (C ) *B

TRANS \(=C^{\prime}:(\operatorname{diag}(\mathbb{R}) * A * \operatorname{diag}(C)) * * H * \operatorname{inv}(\operatorname{diag}(\mathbb{R})) * X=\) diag (C) *B
W hether or not the system willbe equilibrated depends on the scaling of the m atrix A, but ifequilibration is used, A is overw rilten by \(\operatorname{diag}(\mathbb{R}) * A\) *diag \((C)\) and \(B\) by \(\operatorname{diag}(\mathbb{R}) * B\) (if TRANS \(=N\) )
ordiag (C)*B (if TRANS = T'or C).
2. IfFACT = N 'or E', the LU decom position is used to factor the \(m\) atrix A (afterequilibration ifFACT = E) as \(A=L * U\),
where \(L\) is a product of perm utation and unit low er triangular
m atrices w ith \(\mathrm{K} L\) subdiagonals, and U is upper triangular w ith
\(K L+K U\) superdiagonals.
3. If som e \(U(i, i)=0\), so that \(U\) is exactly singular, then the routine
retums w ith \(\mathbb{N} F O=\) i. O therw ise, the factored form of A is used
to estim ate the condition num ber of the \(m\) atrix \(A\). If the
reciprocal of the condition num ber is less than \(m\) achine precision,
\(\mathbb{N} F O=N+1\) is retumed as a waming, but the routine stillgoes on
to solve for X and com pute error bounds as described below .
4.The system of equations is solved for \(X\) using the factored form
of A.
5. Iterative refinem ent is applied to im prove the com puted solution
m atrix and calculate error bounds and backw ard error
6. If equilibration w as used, the m atrix \(X\) is prem ultiplied by diag (C) (if TRANS = N ) ordiag \((\mathbb{R})\) (ifTRANS = T' or C) so that itsolves the originalsystem before equilibration.

\section*{ARGUMENTS}

\section*{FACT (input)}

Specifies w hether ornotthe factored form of the \(m\) atrix \(A\) is supplied on entry, and ifnot, whether them atrix A should be equilibrated before it is factored. = F': On entry, AF and IPIVOT contain the factored form of \(A\). IfEQUED is not \(N\) ', the \(m\) atrix A has been equilibrated \(w\) ith scaling factors given by R and \(\mathrm{C} . \mathrm{A}, \mathrm{AF}\), and IP IV OT are not m odified. \(=\mathrm{N}\) ': The m atrix A w illbe copied to A F and factored.
\(=\mathrm{E}\) : The matrix A will be equilibrated if necessary, then copied to AF and factored.

TRANSA (input)
Specifies the form of the system of equations. = \(\mathrm{N}: A * X=B \quad\) N o transpose)
\(=T\) ': \(A * * T * X=B \quad\) (Transpose)
\(=C\) ': \(A * * H * X=B \quad\) (C onjugate transpose)

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

N (input) The num ber of linearequations, ie., the order of them atrix \(A . N>=0\).

K L (input)
The num ber of subdiagonals \(w\) thin the band of \(A\).
\(\mathrm{K} L>=0\) 。

KU (input)
The num ber of superdiagonals \(w\) ithin the band of \(A\).
\(K U>=0\) 。

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the \(m\) atriges B and X. NRH S \(>=0\).

A (input/output)
O n entry, the m atrix A in band storage, in row s 1
to \(K L+K U+1\). The \(j\) th colum \(n\) ofA is stored in the \(j\) th colum \(n\) of the array A as follow s: A ( \(K U+1+i-\) \(j, j)=A(i, j)\) form \(a x(1, j K U)<=i<=m\) in \((\mathbb{N}, j+k l)\)

IfFACT = F'andEQUED is not \(N\) ', then \(A\) must have been equilibrated by the scaling factors in \(R\) and/orC. A is notm odified ifFACT=F'or \(N^{\prime}\), or iffACT = E'and EQUED = N 'on exit.

On exit, ifEQ UED ne. \(N\) ', A is scaled as follow \(\mathrm{s}: \mathrm{EQUED}=\mathrm{R}\) : A \(:=\operatorname{diag}(\mathbb{R}) * A\)
EQUED = C ': A :=A * diag (C )
\(E Q U E D=B^{\prime}: A:=\operatorname{diag}(R) * A * \operatorname{diag}(C)\).

LD A (input)
The leading dim ension of the aray A. LDA >= \(K L+K U+1\).

AF (input/output)
IfFACT = \(F\) ', then \(A F\) is an inputargum entand on entry contains details of the LU factorization of the band \(m\) atrix \(A\), as com puted by CGBTRF. U is stored as an upper triangularband \(m\) atrix \(w\) ith \(K L+K U\) superdiagonals in row \(s 1\) to \(K L+K U+1\), and the m ultipliers used during the factorization are stored in row sKL+KU+2 to \(2 \star K L+K U+1\). If EQUED
ne. \(N\) ', then \(A F\) is the factored form of the equilibrated \(m\) atrix A.

IfFACT \(=N\) ', then \(A F\) is an output argum ent and on exit retums details of the LU factorization of A.

IfFACT = E', then AF is an output argum ent and on exit retums details of the LU factorization of the equilibrated \(m\) atrix \(A\) (see the description of A for the form of the equilibrated \(m\) atrix).

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(2 * K L+K U+1\).

IPIVOT (input)
IfFACT \(=\mathrm{F}^{\prime}\), then \(\mathbb{P} \mathbb{I V O T}\) is an input argum ent and on entry contains the pivotindiges from the factorization \(A=L * U\) as com puted by CGBTRF; row i of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P}\) IV OT (i).

If \(\mathrm{FACT}=\mathrm{N}\) ', then \(\mathbb{P} \mathbb{I V O T}\) is an output argum ent and on exit contains the pivotindioes from the
factorization \(\mathrm{A}=\mathrm{L} * \mathrm{U}\) of the originalm atrix A .

IfFACT = E', then \(\mathbb{P I V O T}\) is an output argum ent and on exit contains the pivotindioes from the factorization \(\mathrm{A}=\mathrm{L} * \mathrm{U}\) of the equilibrated m atrix A.

EQUED (input)
Specifies the form of equilibration thatw as done. \(=N^{\prime}: ~ N o\) equilibration (alw ays true iffACT = N).
\(=R\) ': Row equilibration, ie., A has been prem ultiplied by diag \((\mathbb{R})\). = C ': Column equilibration, ie., A has been postm ultiplied by diag (C ). = B': B oth row and colum \(n\) equilibration, ie., A has been replaced by diag \((\mathbb{R})\) * A * diag (C). EQUED is an inputargum entifFACT= F'; otherw ise, it is an outputargum ent.

R (input/output)
The row scale factors forA. IfEQUED = R' or \(B\) ', A is multiplied on the leftby diag \((\mathbb{R})\); if EQUED = N 'or C', R is notaccessed. \(R\) is an input argum ent if \(F A C T=F\) '; otherw ise, \(R\) is an output argum ent. IfFACT = F'and EQUED = R'or \(B\) ', each elem entofR m ustbe positive.

C (input/output)
The collm \(n\) scale factors for \(A\). If \(\mathrm{EQ} \mathrm{UED}=\mathrm{C}\) 'or
B', A is multiplied on the rightby diag (C ) ; if \(E Q U E D=N\) 'or \(R\) ', \(C\) is notaccessed. \(C\) is an input argum ent ifFACT = F '; otherw ise, C is an outputargum ent. IfFACT = \(\mathrm{F}^{\prime}\) and \(E Q U E D=C\) 'or \(B\) ', each elem entofC \(m\) ust.be positive.

B (input/output)
On entry, the righthand side m atrix B. On exit, ifEQUED \(=N^{\prime}\) ', \(B\) is notm odified; ifTRANSA \(=N^{\prime}\) and EQUED \(=R^{\prime}\) or \(B^{\prime}, B\) is overw rilten by \(\operatorname{diag}(\mathbb{R}) * B\); if TRANSA \(=T\) 'or \(C^{\prime}\) and \(E Q U E D=C^{\prime}\) or \(B\) ', \(B\) is overw ritten by diag ( \(C\) )*B .

LD B (input)
The leading dim ension of the array \(B\). LD B \(>=\) \(\max (1, \mathbb{N})\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=\mathrm{N}+1\), the \(\mathrm{N}-\) by -NRH S solution
\(m\) atrix \(X\) to the original system ofequations.
\(N\) ote that \(A\) and \(B\) arem odified on exit if EQUED
ne. \(N\) ', and the solution to the equilibrated
system is inv (diag (C))*X ifTRANSA \(=N\) 'and EQUED
\(=C\) 'or \(B^{\prime}\) 'orinv \((\operatorname{diag}(R)) * X\) ifTRANSA \(=T\) 'or
\(C^{\prime}\) and \(E Q U E D=R\) 'or \(B '\).

LD X (input)
The leading dim ension of the array X . LD X >= \(\max (1, N)\).
RCOND (output)
The estim ate of the reciprocal condition num ber of the matrix A after equilibration (if done). If RCOND is less than them achine precision (in particular, ifRCOND \(=0\) ), the \(m\) atrix is singular to w orking precision. This condition is indicated by a retum code of \(\mathbb{N} F O>0\).

\section*{FERR (output)}

The estim ated forw ard emrorbound for each solution vector \(X(\mathcal{)}\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{O}), \operatorname{FERR}(\mathcal{)})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{D})-X\) TRU \(E\) ) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector \(X(j)\) (i.e., the \(s m\) allest relative change in any elem entofA orB thatm akes \(X\) ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dimension (N) On exit, W ORK \(2(1)\) contains the reciprocal pivot grow th factornom (A)/norm (U). The " m ax absolute elem ent" norm is used. If W ORK \(2(1)\) is much less than 1 , then the stability of the LU factorization of the (equilibrated) \(m\) atrix A could be poor. This also \(m\) eans that the solution X, condition estim atorRC O ND, and forw ard error bound FERR could be unreliable. If factorization fails \(w\) ith \(0<\mathbb{N} F O<=N\), then \(W\) ORK 2 (1) contains the reciprocal pivot grow th factor for the leading \(\mathbb{N}\) FO colum ns of A.
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum ent had an illegalvalue
\(>0\) : if \(\mathbb{N F O}=i\), and \(i\) is
\(<=\mathrm{N}: \mathrm{U}(i, i)\) is exactly zero. The factorization has been com pleted, but the factorU is exactly singular, so the solution and emror bounds could not be com puted. RC OND \(=0\) is retumed. \(=\mathrm{N}+1: \mathrm{U}\) is nonsingular, butRCOND is less than machine precision, \(m\) eaning that the \(m\) atrix is singular to w orking precision. N evertheless, the solution and emror bounds are com puted because there are a num ber of situationsw here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{Contents}
－NAME
－SYNOPSIS
－F95 INTERFACE
－C INTERFACE
－PURPOSE
－ARGUMENTS
－FURTHER DETAILS

\section*{NAME}
zgbtf2－com pute an LU factorization of a com plex \(m\)－by－n band \(m\) atrix A using partialpivoting \(w\) th row interchanges

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGBTF2M,N,KL,KU,AB,LDAB,\mathbb{P}\mathbb{N},\mathbb{NFO)}}\mathbf{M}\mathrm{ (N,}
DOUBLE COM PLEX AB (LDAB,*)
INTEGERM,N,KL,KU,LDAB,INFO
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(})
SUBROUT\mathbb{NE ZGBTF2_64M,N,KL,KU,AB,LDAB,\mathbb{PIV,INFO)}}⿱㇒⿻二丿⿴囗⿱一一⿰亻
DOUBLE COM PLEX AB (LDAB,*)
\mathbb{NTEGER*8M,N,KL,KU,LDAB,INFO}
\mathbb{NTEGER*8 \mathbb{PIV (*)}}\mathbf{(})
F95 INTERFACE

```

```

COM PLEX (8),D IM ENSION (:,:)::AB
\mathbb{NTEGER ::M,N,KL,KU,LDAB,NNFO}
INTEGER,D IM ENSION (:) :: \mathbb{PIV}

```

```

COM PLEX (8),D IM ENSION (:,:)::AB
INTEGER (8)::M,N,KL,KU,LDAB,\mathbb{NFO}
INTEGER (8),D \mathbb{M ENSION (:) ::\mathbb{PIV}}\mathbf{N}=\mp@code{N}

```
\#include < sunperfh>
void zg.btf2 (intm, intn, int kl, int ku, doublecom plex *ab, int ldab, int *ipiv, int *info);
void zgbtfe_64 (long m , long n, long kl, long ku, doublecom plex *ab, long ldab, long *ịív, long *info );

\section*{PURPOSE}
zgbtf2 com putes an LU factorization of a com plex m -by-n band \(m\) atrix A using partialpivoting \(w\) ith row interchanges.

This is the unblocked version of the algorithm, calling Level2 BLAS.

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
KL (input)
The num ber of subdiagonals w ithin the band of A.
\(\mathrm{KL}>=0\).
KU (input)
The num ber of superdiagonals within the band of A. \(K U>=0\).

AB (input/output)
O n entry, them atrix \(A\) in band storage, in row \(s\) \(K L+1\) to \(2 * K L+K U+1\); row s 1 to \(K L\) of the anray need not.be set. The \(j\) th column ofA is stored in the \(j\) th column of the array AB as follows: \(A B(k l+k u+1+i-j, j)=A(i, j)\) for \(\max (1, j\) \(\mathrm{ku})<=i<=m\) in \((\mathrm{m}, \mathrm{j}+\mathrm{kl})\)

On exit, details of the factorization: U is stored as an upper triangular band \(m\) atrix \(w\) ith \(K L+K U\) superdiagonals in row s 1 to \(K L+K U+1\), and the \(m u l-\) tipliers used during the factorization are stored in row \(K L+K U+2\) to \(2 * K L+K U+1\). See below for furtherdetails.

The leading dim ension of the array AB. LDAB >=
\(2 * K L+K U+1\).

IPIV (output)
The pivot indices; for \(1<=i<=m\) in \(M, N\) ), row i of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P} \mathbb{V}\) (i).
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0\) : if \(\mathbb{N N F O}=-\) i, the \(i\)-th argum ent had an illegalvalue \(>0\) : if \(\mathbb{N} F O=+i, U(i, i)\) is exactly zero. The factorization has been com pleted, but the factor \(U\) is exactly singular, and division by zero will occur if it is used to solve a system ofequations.

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, \(w\) hen \(M=N=6, K L=2, K U=1\) :

On entry: On exit:
```

    * * * + + + * * * u14 u25
    u36
* * + + + + * * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55
u66
a21 a32 a43 a54 a65 * m 21 m 32 m 43 m 54 m 65
*
a31 a42 a53 a64 * * m 31 m 42 m 53 m 64 *

```
*

A ray elem entsm arked * are notused by the routine; ele\(m\) entsm arked + need notbe seton entry, but are required by the routine to store elem ents of \(U\), because of fill-in resulting from the row interchanges.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zgbtrf-com pute an LU factorization of a com plex \(m\)-by-n band \(m\) atrix A using partialpivoting \(w\) th row interchanges

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGBTRFM,N,KL,KU,AB,LDAB,\mathbb{PIVOT,INFO)}}\mathbf{M}\mathrm{ (N,N}
DOUBLE COM PLEX AB (LDAB,N)
INTEGERM,N,KL,KU,LDAB,INFO
INTEGER \mathbb{PIVOTM IN M NN)}

```

```

DOUBLE COM PLEX AB (LDAB,N)
\mathbb{NTEGER*8M,N,KL,KU,LDAB,INFO}
\mathbb{NTEGER*8 \mathbb{PIVOTM}\mathbb{N}M,N))}

```

\section*{F95 INTERFACE}
```

SU BROUTINE GBTRF $M, \mathbb{N}], K L, K U, A B,[L D A B], \mathbb{P} \mathbb{V} O T,[\mathbb{N F O}])$
COM PLEX (8), D IM ENSION (:,:) ::AB
$\mathbb{N}$ TEGER ::M,N,KL,KU,LDAB, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T$
SU BROUTINE GBTRF_64M, N ],KL,KU,AB, [LDAB], $\mathbb{P} \mathbb{I V O T},[\mathbb{N} F O])$
COM PLEX (8), D $\mathbb{I M}$ ENSION (:,:) ::AB
$\mathbb{N}$ TEGER (8) :: M , N, KL,KU,LDAB, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V} O T$

```
void zgbtrf(intm , intn, intkl, int ku, doublecom plex *ab, int ldab, int *ipivot, int *info);
void zgbtrf_64 (long m, long n, long kl, long ku, doublecom plex *ab, long ldab, long *ípivot, long *info);

\section*{PURPOSE}
zgbtrf com putes an LU factorization of a com plex \(m\)-by-n band \(m\) atrix A using partialpivoting \(w\) ith row interchanges.

This is the blocked version of the algorithm, calling Level 3 BLAS .

\section*{ARGUMENTS}

M (input) Integer
The num ber of row sof the m atrix A. M \(>=0\).
N (input) Integer
The num berof colum ns of the m atrix A. \(\mathrm{N}>=0\).

K L (input) Integer
The num ber of subdiagonals w ithin the band of A. \(K L>=0\).

KU (input) Integer
The num ber of superdiagonals \(w\) ithin the band of A.
\(K U>=0\).

AB (input/output) D ouble com plex array of dim ension (LD AB N ).
On entry, the matrix A in band storage, in row \(s\) \(K L+1\) to \(2 \star K L+K U+1\); row s 1 to \(K L\) of the array need not.be set. The \(j\) th colum n of A is stored in the \(j\) th column of the array \(A B\) as follows: AB \((\mathbb{K} L+K U+1+I J, J)=A(I, J)\) for MAX \((1, J\) \(K U)<=\mathbb{K}=M \mathbb{N}(M, J+K L)\)

O n exit, details of the factorization: U is stored as an upper triangular band \(m\) atrix \(w\) ith \(K L+K U\) superdiagonals in row s 1 to \(K L+K U+1\), and the \(m u l-\) tipliers used during the factorization are stored in row \(S K+K U+2\) to \(2 * K L+K U+1\). See below for further details.

LD AB (input) Integer
The leading dim ension of the anay A. LDA >= \(2 * K L+K U+1\).
\(\mathbb{P} \mathbb{V} O T\) (output) Integerarray ofdim ension \(M \mathbb{N} \mathrm{M}, \mathrm{N}\) )
The pívotindioes; for \(1<=I<=m\) in \(M, N\) ), row \(I\) of the \(m\) atrix \(w\) as interchanged \(w\) th row \(\mathbb{P} \mathbb{I V O T}\) (I).
\(\mathbb{I N} F O\) (output) Integer
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) I, the I-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=+\mathbb{I}, U(I, I)\) is exactly zero. The factorization has been com pleted, but the factor \(U\) is exactly singular, and division by zero will occur if it is used to solve a system ofequations.

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, when \(M=N=6, K L=2, K U=1\) :

On entry: On exit:
```

    * * * + + + * * * u14 u25
    u36
* * + + + + * * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55
u66
a21 a32 a43 a54 a65 * m 21 m 32 m 43 m 54 m 65
*
a31 a42 a53 a64 * * m31 m 42 m 53 m 64 *
*

```

A ray elem entsm arked * are notused by the routine; ele\(m\) entsm arked + need notbe seton entry, but are required by the routine to store elem ents of \(U\) because of fill-in resulting from the row interchanges.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

zgbtrs - solve a system of linear equations A * X = B , A **T

```
* \(\mathrm{X}=\mathrm{B}\), or \(\mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B}\) w ith a generalband m atrix A using the LU factorization com puted by CGBTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGBTRS (TRANSA,N,NSUB,NSUPER,NRHS,A,LDA, P\mathbb{IVOT,B,}}\mathbf{N},\textrm{N},\textrm{N}
LDB,\mathbb{NFO)}
CHARACTER * 1 TRANSA
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGERN,NSUB,NSUPER,NRHS,LDA,LDB, INFO
INTEGER \mathbb{PIVOT (*)}
SUBROUT\mathbb{NE ZGBTRS_64(TRANSA,N,NSUB,NSUPER,NRHS,A,LDA, \mathbb{PIVOT,}}\mathbf{N},\textrm{N},\textrm{N}
B,LDB,INFO)
CHARACTER * 1 TRANSA
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
NNTEGER*8 N,NSUB,NSUPER,NRHS,LDA,LDB,INFO
INTEGER*8 \mathbb{PIVOT (*)}

```

\section*{F95 INTERFACE}

SU BROUTINE GBTRS ([TRANSA], \(\mathbb{N}], N S U B, N S U P E R, ~ N R H S], A,[L D A]\), \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}],[\mathbb{N F O}])\)

CHARACTER (LEN=1) ::TRANSA
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, B
\(\mathbb{N}\) TEGER ::N,N SUB,NSUPER,NRHS,LDA,LDB, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{I}\) ENSION (:) :: \(\mathbb{P} \mathbb{V} O T\)
SU BROUTINE GBTRS_64 ([TRANSA], \(\mathbb{N}], N S U B, N S U P E R, ~ \mathbb{N} R H S], A,[L D A]\),


CHARACTER (LEN=1) ::TRANSA
COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : : : A, B
\(\mathbb{N}\) TEGER (8) ::N ,NSUB,NSUPER,NRHS,LDA,LD B, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8),D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V}\) OT

\section*{C INTERFACE}
\#include <sunperfh>
void zgbtrs (char transa, intn, intnsub, int nsuper, int nrhs, doublecom plex *a, int lda, int*ipivot, doublecom plex *b, int ldb, int*info);
void zgbtrs_64 (chartransa, long n, long nsub, long nsuper, long nihs, doublecomplex *a, long lda, long *ịíivot, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
zgbtres solves a system of linearequations
\(A * X=B, A * * T * X=B\), or \(A * * H * X=B\) with a general band \(m\) atrix A using the LU factorization com puted by CGBTRF.

\section*{ARGUMENTS}

TRANSA (input)
Specifies the form of the system ofequations. =
N : : A * X = B Notranspose)
\(=T: A * * T * X=B \quad(T\) ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose)
TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NSUB (input)
The num ber of subdiagonals w ithin the band of A. N SUB \(>=0\).

\section*{N SU PER (input)}

The num ber of superdiagonals \(w\) ithin the band of A.
N SU PER \(>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS >=0.

A (input) \(D\) etails of the \(L U\) factorization of the band \(m\) atrix
A, as com puted by C G B TRF. U is stored as an upper triangularband \(m\) atrix \(w\) ith \(N\) SU B +N SU PER superdiagonals in rows 1 to NSU B+N SU PER+1, and them ultipliers used during the factorization are stored in row SN SU B +N SU PER + 2 to \(2 \star\) N SU B + N SU PER +1 .

LD A (input)
The leading dim ension of the anay A. LD A \(>=\) 2*N SU B +N SU PER +1 .
\(\mathbb{P I V O T}\) (input)
The pívotindiges; for \(1<=i<=N\), row \(i\) of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P I V O T}\) (i).

B (input/output)
On entry, the right hand side matrix B. On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array \(B\). LD B \(>=\) \(\max (1, N)\).
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
<0: if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgebak - form the rightor lefteigenvectors of a com plex general \(m\) atrix by backw ard transform ation on the com puted eigenvectors of the balanced \(m\) atrix outputby C G EBA L

\section*{SYNOPSIS}

```

CHARACTER * 1 JOB,S\mathbb{DE}
DOUBLE COM PLEX V (LDV,*)

```

```

DOUBLE PRECISION SCALE (*)

```

```

CHARACTER * 1 JOB,SIDE
DOUBLE COM PLEX V (LDV,*)

```

```

DOUBLE PRECISION SCALE (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GEBAK (JOB,SDE, \(\mathbb{N}\) ], \(\mathbb{H} O, \mathbb{H} I, S C A L E, ~ M], V,[L D V]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1):: JOB,SDE
COM PLEX (8),D IM ENSION (:,:) ::V
\(\mathbb{N}\) TEGER :: N, \(\mathbb{L} O, \mathbb{H} \mathrm{I}, \mathrm{M}, \mathrm{LDV}, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::SCALE

SU BROUTINE GEBAK_64 (JOB,SIDE, N ], \(\mathbb{I} O, \mathbb{H} I, S C A L E, \mathbb{M}], V,[L D V]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) :: JB B,SDE
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::V
\(\mathbb{N}\) TEGER (8) ::N, \(\mathbb{N} O, \mathbb{H} \mathrm{I}, \mathrm{M}, \mathrm{LDV}, \mathbb{N} F \mathrm{O}\)
REAL (8),D \(\mathbb{I}\) ENSION (:) ::SCALE

\section*{C INTERFACE}
\#include <sunperfh>
void zgebak (char job, char side, intn, int ilo, int ini, double *scale, intm, doublecom plex *v, int ldv, int*info);
void zgebak_64 (char jंb, char side, long n, long ilo, long ihi, double *scale, long \(m\), doublecom plex *v, long ldv, long *info);

\section*{PURPOSE}
zgebak form sthe rightor left eigenvectors of a com plex general \(m\) atrix by backw ard transform ation on the com puted eigenvectors of the balanced \(m\) atrix outputby CGEBA L .

\section*{ARGUMENTS}
\(J O B\) (input)
Specifies the type of backw ard transform ation required: = N ', do nothing, retum im \(m\) ediately; = P ', do backw ard transform ation for perm utation only; = S', do backw ard transform ation for scaling only; = B ', do backw ard transform ations for both perm utation and scaling. JOB m ustbe the sam e as the argum ent \(J 0\) B supplied to CG EBA L .

SIDE (input)
= R : : V contains righteigenvectors;
\(=\mathrm{L}: \mathrm{V}\) contains lefteigenvectors.

N (input) The num ber of row s of the m atrix \(\mathrm{V} . \mathrm{N}>=0\).

ㅍO (input)
The integer ILO determ ined by CGEBAL. \(1<=\mathbb{L O}\) <= \(\mathbb{H} I<=N\), if \(N>0 ; \mathbb{H}=1\) and \(\mathbb{H} \mathrm{I}=0\), if \(\mathrm{N}=0\).

HH I (input)
The integer \(\mathbb{H}\) Ideterm ined by CGEBAL. \(1<=\mathbb{L O}\) <= \(\mathbb{H} I<=N\), if \(N>0 ; \mathbb{O}=1\) and \(\mathbb{H} \mathrm{I}=0\), if \(\mathrm{N}=0\).

SCALE (input)
D etails of the perm utation and scaling factors, as retumed by CGEBAL.

M (input) The num ber of collm ns of the \(m\) atrix \(V . M>=0\).
V (input/output)
O \(n\) entry, the \(m\) atrix of right or lefteigenvectors to be transform ed, as retumed by CHSE IN or CTREVC. On ex㝳, \(V\) is overw ritten by the transform ed eigenvectors.

LD V (input)
The leading dim ension of the array \(V\). LDV >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zgebal-balance a general com plex m atrix A

\section*{SYNOPSIS}

```

CHARACTER * 1 J B
DOUBLE COM PLEX A (LDA,*)

```

```

DOUBLE PRECISION SCALE (*)
SUBROUT\mathbb{NE ZGEBAL_64(JOB,N,A,LDA,\mathbb{LO},\mathbb{H}I,SCALE,NNFO)}
CHARACTER * 1 Job
DOUBLE COM PLEX A (LDA,*)
\mathbb{NTEGER*8N,LDA, IOO,\mathbb{HI, INFO}}\mathbf{N}=1
DOUBLE PRECISION SCALE (*)

```
F95 INTERFACE
    SU BROUTINE GEBAL (JOB, \(\mathbb{N}], A,[L D A], \mathbb{I} O, \mathbb{H} I, S C A L E,[\mathbb{N F O}])\)
    CHARACTER (LEN=1):: JOB
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathrm{LD} A, \mathbb{L} \mathrm{O}, \mathbb{H} \mathrm{I}, \mathbb{I N F O}\)
    REAL (8),D \(\mathbb{M}\) ENSION (:) ::SCALE
    SU BROUTINE GEBAL_64 (OOB, \(\mathbb{N}], A,[L D A], \mathbb{L O}, \mathbb{H} I, S C A L E,[\mathbb{N} F O])\)
    CHARACTER (LEN=1):: JOB
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N}\) TEGER (8) :: N,LDA, \(\mathbb{L O}, \mathbb{H} \mathrm{I}, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zgebal(char jंb, intn, doublecom plex *a, int lda, int *ilo, int *ihi, double *scale, int *info);
void zgebal 64 (char job, long n, doublecom plex *a, long lda, long *ilo, long *ihi, double *scale, long *info);

\section*{PURPOSE}
zgebalbalances a general com plex m atrix A. This involves, first, perm uting A by a sim ilarity transform ation to isolate eigenvalues in the first 1 to \(\mathbb{I L O - 1}\) and last \(\mathbb{H}\) I+1 to N elem ents on the diagonal; and second, applying a diagonal sm ilarity transform ation to row sand colum ns \(\mathbb{I} O\) to \(\mathbb{H}\) I to \(m\) ake the rows and colum ns as close in norm as possible. B oth steps are optional.

B alancing \(m\) ay reduce the 1 -norm of the \(m\) atrix, and im prove the accuracy of the com puted eigenvalues and/oreigenvectors.

\section*{ARGUMENTS}

JOB (input)
Specifies the operations to be perform ed on \(A\) :
= N ': none: simply set \(\mathbb{H}=1, \mathbb{H} \mathrm{I}=\mathrm{N}\), SCALE (I) = 1.0 fori=1,...N; \(=P^{\prime}:\) perm ute only;
= S ': scale only;
= B ': both perm ute and scale.
N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the inputm atrix A. On exit, A is overw rilten by the balanced \(m\) atrix. If \(\mathrm{JO}=\mathrm{N}\) ', \(A\) is not referenced. See FurtherD etails.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

IO and \(\mathbb{H}\) Iare set to integers such thaton exit A \((i, 7)=0\) if \(i>\) jand \(j=1, \ldots\), ILO-1 or \(I=\)
 \(=\mathrm{N}\).

HI I (output)
IO and \(\mathbb{H}\) Iare set to integers such thaton exit A \((i, j)=0\) if \(i>j a n d j=1, \ldots\), IL O-1 or \(I=\) \(\mathrm{HH} \mathrm{I}+1, \ldots, \mathrm{~N}\). If \(\mathrm{JOB}=\mathrm{N}\) 'or \(\mathrm{S}^{\prime}, \mathrm{LIO}=1\) and H I \(=\mathrm{N}\).

SCALE (output)
D etails of the perm utations and scaling factors applied to \(A\). IfP \((j)\) is the index of the row and colum \(n\) interchanged \(w\) th row and colum \(n\) jand \(D(1)\) is the scaling factor applied to row and column \(\mathfrak{j}\) then SCALE \((j)=P(j) \quad\) for \(j=1, \ldots, I L O-1=D(j)\) for \(j=\mathbb{L O}, \ldots, \mathbb{H} I=P(j) \quad\) for \(j=\mathbb{H} I+1, \ldots, N\). The order in which the interchanges are m ade is N to \(\mathbb{H} \mathrm{I}+1\), then 1 to \(\mathbb{L} O-1\).
\(\mathbb{N} F O\) (output)
= 0: successfulexit.
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue.

\section*{FURTHER DETAILS}

The perm utations consist of row and column interchanges which put the \(m\) atrix in the form
\[
\begin{gathered}
(\mathrm{T} 1 \mathrm{X} \\
\mathrm{PAP}=\left(\begin{array}{ll}
0 & \mathrm{~B}
\end{array}\right) \\
\left(\begin{array}{ll}
0 & \mathrm{O}
\end{array}\right)
\end{gathered}
\]
where T1 and T2 are uppertriangularm atrices whose eigenvalues lie along the diagonal. The colum \(n\) indioes \(\Pi 0\) and IH Im ark the starting and ending colum ns of the subm atrix B. Balancing consists of applying a diagonal sim ilarity transform ation inv (D) * \(B * D\) to \(m\) ake the 1 -norm \(s\) of each row of \(B\) and its comesponding colum n nearly equal. The outputm atrix is
\(\left(\begin{array}{lll}T 1 & X * D & Y\end{array}\right)\)
\(\left(\begin{array}{lll}0 & \operatorname{inv}(D) * B * D & \operatorname{inv}(D) * Z\end{array}\right)\).
\(\left(\begin{array}{lll}0 & 0 & T 2\end{array}\right)\)

Inform ation about the perm utations \(P\) and the diagonalm atrix D is retumed in the vectorSC A LE .

This subroutine is based on the E ISPA CK routine CBAL.
M odified by Tzu-Y iChen, C om puter Science D ìvision, U niversity of
C alifomia atB erkeley, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zgebrd - reduce a general com plex M -by -N m atrix A to upper or low erbidiagonal form B by a unitary transform ation

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGEBRD M,N,A,LDA,D,E,TAUQ,TAUP,W ORK,LW ORK,INFO)}
DOUBLE COM PLEX A (LDA,*),TAUQ (*),TAUP (*),W ORK (*)
INTEGERM,N,LDA,LW ORK,INFO
DOUBLE PRECISIOND (*),E (*)
SU BROUT\mathbb{NE ZGEBRD_64M,N,A,LDA,D,E,TAUQ,TAUP,W ORK,LW ORK,}
INFO)
DOUBLE COM PLEX A (LDA,*),TAUQ (*),TAUP (*),W ORK (*)
INTEGER*8M,N,LDA,LW ORK,NNFO
DOUBLE PRECISIOND (*),E (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GEBRD ( \(\mathbb{M}], \mathbb{N}], A,[L D A], D, E, T A U Q, T A U P,[\mathbb{W} O R K],[L W O R K]\), [ \(\mathbb{N} F O\) ])

COM PLEX (8),D \(\mathbb{I M} E N S I O N(:):: T A U Q, T A U P, W\) ORK
COM PLEX (8),D IM ENSION (:,:)::A
\(\mathbb{N}\) TEGER ::M ,N,LDA,LW ORK, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D ,E

SU BROUTINE GEBRD_64 (M ], \(\mathbb{N}], A,[L D A], D, E, T A U Q, T A U P,[W O R K]\), [LW ORK], [ \(\mathbb{N F O}\) ])

\section*{C INTERFACE}
\#include <sunperfh>
void zgebrd (intm, intn, doublecom plex *a, int lda, double
*d, double *e, doublecom plex *tauq, doublecom plex
*taup, int*info);
void zgebrd_64 (long m, long n, doublecom plex *a, long lda, double *d, double *e, doublecom plex *tauq, doublecom plex *taup, long *info);

\section*{PURPOSE}
zgebrd reduces a general com plex M -by-N m atrix A to upper or low er bidiagonal form B by a unitary transform ation: Q ** H * \(A * P=B\).

Ifm >= \(n, B\) is upperbidiagonal; ifm < \(n, B\) is low erbidiagonal

\section*{ARGUMENTS}

M (input) The num ber of row s in the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of 00 llm ns in them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the \(M-b y-N\) generalm atrix to be reduced. On exit, if \(m>=n\), the diagonal and the first superdiagonal are overw rilten w ith the upperbidiagonal m atrix B ; the elem ents below the diagonal, \(w\) ith the amay TAU \(Q\), represent the unitary \(m\) atrix \(Q\) as a product of elem entary reflectors, and the elem ents above the first superdiagonal, w ith the array TAUP, represent the unitary \(m\) atrix \(P\) as a product ofelem entary reflectors; ifm < n, the diagonal and the first subdiagonalare overw rilten w ith the low erbidiagonalm atrix \(B\); the elem ents below the first subdiagonal, w ith the amay TAU Q, represent the unitary \(m\) atrix \(Q\) as a product of elem entary reflectors, and the elem ents above the diagonal, w ith the array TA UP, represent the unitary \(m\) atrix \(P\) as a productofelem entary reflec-
tors. See FurtherD etails.

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, M)\).

D (output)
The diagonalelem ents of the bidiagonalm atrix B :
\(D(i)=A(i, i)\).

E (output)
The off-diagonalelem ents of the bidiagonalm atrix
\(B\) : ifm \(>=n, E(i)=A(i, i+1)\) for \(i=1,2, \ldots, n-\)
1 ; ifm \(<n, E(i)=A(i+1, i)\) for \(i=1,2, \ldots, m-1\).
TAUQ (output)
The scalar factors of the elem entary reflectors which represent the unitary \(m\) atrix \(Q\). See Further D etails.

TAUP (output)
The scalar factors of the elem entary reflectors w hich represent the unitary m atrix P. See Further D etails.

W ORK (w orkspace)
On exiv, if \(\mathbb{N F O}=0, W\) ORK ( 1 ) retums the optim al LW ORK.

LW ORK (input)
The length of the amay \(W\) ORK. LW ORK >= \(\max (1, M, N)\). For optim um perform ance LW ORK >= \((M+N) \star N B\), w here N B is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{I N F O}\) (output)
= 0: successfillexit.
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the i-th argum enthad an illegalvalue.

\section*{FURTHER DETAILS}

Them atrices \(Q\) and \(P\) are represented as products of elem entary reflectors:

Ifm \(>=n\),
\[
Q=H(1) H(2) \ldots H(n) \text { and } P=G(1) G(2) \ldots G(n-1)
\]

Each H (i) and G (i) has the form :
\[
H(i)=I-\operatorname{tanq}{ }^{\star} v^{\star} v^{\prime} \text { and } G(i)=I-\operatorname{taup} * u^{\star} u^{\prime}
\]
\(w\) here tauq and taup are com plex scalars, and \(v\) and \(u\) are complex vectors; \(v(1: i-1)=0, v(i)=1\), and \(v(i+1 m)\) is stored on exitin \(A(i+1 m, i) ; u(1: i)=0, u(i+1)=1\), and \(u(i+2 m)\) is stored on exitin A \((i, i+2 m)\); tauq is stored in TAUQ (i) and taup in TAUP (i).
Ifm < n,
\[
Q=H(1) H(2) \ldots H(m-1) \text { and } P=G(1) G(2) \ldots G(m)
\]

Each H (i) and G (i) has the form :
\[
H(i)=I-\operatorname{tanq}{ }^{\star} v^{\star} v^{\prime} \text { and } G(i)=I-\operatorname{taup}{ }^{\star} u^{\star} u^{\prime}
\]
\(w\) here tauq and taup are com plex scalars, and \(v\) and \(u\) are complex vectors; \(v(1: i)=0, v(i+1)=1\), and \(v(i+2 \mathrm{~m})\) is stored on exitin \(A(i+2 m, i) ; u(1: i-1)=0, u(i)=1\), and \(u(i+1 \mathrm{~m})\) is stored on exitin \(A(i, i+1 \mathrm{~m})\); tauq is stored in TAUQ (i) and taup in TA UP (i).

The contents of A on exitare illustrated by the follow ing exam ples:
\(m=6\) and \(n=5(m>n): \quad m=5\) and \(n=6(m<n):\)
( d e u1 u1 u1 ) ( d u1 u1 u1 u1
u1)
(v1 d e u2 u2) ( e d u2 u2 u2
u2 )
( v1 v2 d e u3) (v1 e d u3 u3
u3)
( v1 v2 v3 d e ) (v1 v2 e d u4 u4)
( v1 v2 v3 v4 d ) (v1 v2 v3 e d u5 )
( v1 v2 v3 v4 v5 )
where d and e denote diagonal and off-diagonal elem ents of \(B\), videnotes an elem ent of the vectordefining \(H\) (i), and ui an elem ent of the vectordefining \(G\) (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgecon -estim ate the reciprocal of the condition num ber of a general com plex matrix \(A\), in either the 1 -nom or the infinity-norm, using the LU factorization com puted by CGETRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGECON NORM,N,A,LDA,ANORM,RCOND,W ORK,W ORK 2, INFO)}
CHARACTER * 1NORM
DOUBLE COM PLEX A (LDA,*),W ORK (*)
INTEGERN,LDA,}\mathbb{N}F
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION W ORK 2 (*)
SUBROUT\mathbb{NE ZGECON_64 NORM,N,A,LDA,ANORM,RCOND,W ORK,W ORK2,}
\mathbb{NFO)}
CHARACTER * 1 NORM
DOUBLE COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGER*8N,LDA,INFO}
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION W ORK2 (*)

```
F95 INTERFACE
    SUBROUTINE GECON \(\mathbb{N} O R M, \mathbb{N}], A,[L D A], A N O R M, R C O N D,[\mathbb{W} O R K],[\mathbb{W} O R K 2]\),
        [ \(\mathbb{N} F O\) ])
    CHARACTER (LEN=1) ::NORM
    COMPLEX (8), D IM ENSION (:) ::W ORK
    COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N} T E G E R:: N, L D A, \mathbb{N} F O\)
    REAL (8) ::ANORM,RCOND

SU BROUTINE GECON_64 \(\mathbb{N} O R M, \mathbb{N}], A,[L D A], A N O R M, R C O N D,[W O R K],[\mathbb{W} O R K 2]\), [ \(\mathbb{N}\) FO ])

CHARACTER ( \(几 E N=1):: N O R M\)
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: \(:\) :) ::A
\(\mathbb{N}\) TEGER (8) :: N , LD A , \(\mathbb{N}\) FO
REAL (8) ::ANORM,RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zgecon (charnorm, intn, doublecom plex *a, int lda, double anorm, double *rcond, int *info);
void zgecon_64 (charnorm, long n, doublecom plex *a, long lda, double anorm , double *roond, long *info);

\section*{PURPOSE}
zgecon estim ates the reciprocal of the condition num ber of a general com plex \(m\) atrix \(A\), in either the 1 -norm orthe infinity-norm, using the LU factorization com puted by CGETRF.

A \(n\) estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as
\(\operatorname{RCOND}=1 /(\operatorname{norm}(A) * \operatorname{norm}(\operatorname{inv}(A)))\).

\section*{ARGUMENTS}

NORM (input)
Specifies w hether the 1-norm condition num ber or the infinity-norm condition num ber is required:
= ' 'or \(\mathrm{D}^{\prime}\) : 1-norm;
\(=1 ': \quad\) Infinity-norm .

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input) The factors \(L\) and \(U\) from the factorization \(A=\) \(P * L * U\) as com puted by CGETRF .

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

\section*{ANORM (input)}

IfNORM = 1 'or 0 ', the 1 -nom of the original
\(m\) atrix \(A\). IfNORM = \(I\) ', the infinity-norm of the originalm atrix A.

RCOND (output)
The reciprocal of the condition num ber of the
\(m\) atrix \(A\), computed as \(R C O N D=1 /(\) norm (A) * norm (inv (A))).

W ORK (w orkspace)
dim ension ( \(2 \star \mathrm{~N}\) )

W ORK 2 (w orkspace)
dim ension ( \(2 * N\) )
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgeequ - com pute row and colum n scalings intended to equilibrate an \(M\)-by-N \(m\) atrix \(A\) and reduce its condition num ber

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZGEEQU M,N,A,LDA,R,C,ROW CN,COLCN,AMAX,}
\mathbb{NFO)}
DOUBLE COM PLEX A (LDA,*)
\mathbb{NTEGER M,N,LDA,}\mathbb{NFO}
DOUBLE PRECISION ROW CN,COLCN,AM AX
DOUBLE PRECISION R (*),C (*)
SU BROUT\mathbb{NE ZGEEQU_64 M ,N,A,LDA,R,C,ROW CN,COLCN,AM AX,}
INFO)
DOUBLE COM PLEX A (LDA,*)
INTEGER*8M,N,LDA,INFO
DOUBLE PRECISION ROW CN,COLCN,AMAX
DOUBLE PRECISION R (*),C (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GEEQU (M ], \(\mathbb{N}], A,[L D A], R, C, R O W C N, C O L C N\), AMAX, [ \(\mathbb{N F O}\) ])

COM PLEX (8),D IM ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, L D A, \mathbb{N} F O\)
REAL (8) ::ROW CN,COLCN,AMAX
REAL (8),D IM ENSION (:) ::R,C
SU BROUTINE GEEQU_64 (M) \(\mathbb{M}, \mathbb{N}], A,[L D A], R, C, R O W C N, C O L C N\), AMAX, \([\mathbb{N} F O]\) )

COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER (8) :: M , N , LDA, \(\mathbb{I N} F O\)
REAL (8) ::ROW CN, COLCN, AMAX
REAL (8), D \(\mathbb{M}\) ENSION (:) ::R , C

\section*{C INTERFACE}
\#include <sunperfh>
void zgeequ (intm, intn, doublecom plex *a, int lda, double
* \(r_{r}\) double * c , double *row cn, double *colen, double *am ax, int*info);
void zgeequ_64 (long m, long n, doublecom plex *a, long lda, double \({ }^{*} r_{r}\) double \({ }^{*}\) c, double *row cn, double *colen, double *am ax, long *info);

\section*{PURPOSE}
zgeequ com putes row and colum n scalings intended to equilibrate an M -by-N m atrix A and reduce its condition num ber. R retums the row scale factors and \(C\) the colum n scale factors, chosen to try to \(m\) ake the largestelem ent in each row and column of the \(m\) atrix \(B \quad w\) ith elements \(B(i, j)=R(i) \star A(i, j) * C(j)\) have absolute value 1.
\(R\) (i) and C (i) are restricted to be betw een SM LN UM = sm allest safe num ber and B IG N U M = largest safe num ber. U se of these scaling factors is notguaranteed to reduce the condition num berofA butw orks w ellin practice.

\section*{ARGUMENTS}

M (input) The num ber of row s of them atrix A. M >=0.

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input) The M -by- N m atrix w hose equilibration factors are to be com puted.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, M)\).

R (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O>M, R\) contains the row scale factors forA.

C (output)
If \(\mathbb{N} F O=0, C\) contains the colum \(n\) scale factors forA.

ROW CN (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O>M, R O W C N\) contains the ratio
of the sm allest \(R\) (i) to the largestR (i). If
ROW CN >= 0.1 and AM AX is neither too large nortoo sm all, it is notw orth scaling by R .

COLCN (output)
If \(\mathbb{N} F O=0, C O L C N\) contains the ratio of the sm allest C (i) to the largestC (i). IfC O LCN >=0.1, it is notw orth scaling by \(C\).

AMAX (output)
A bsolute value of largestm atrix elem ent. IfA M A X
is very close to overflow orvery close to underflow , the m atrix should be scaled.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N F O}=i\), and \(i\) is
\(<=\mathrm{M}\) : the i-th row of A is exactly zero
> M : the ( \(\mathrm{i}-\mathrm{M}\) ) -th colum n of A is exactly zero

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgees - com pute foran \(N\)-by -N com plex nonsym \(m\) etric \(m\) atrix A, the eigenvalues, the Schur form T, and, optionally, the \(m\) atrix of Schurvectors \(Z\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGEES (JOBZ,SORTEV,SELECT,N,A,LDA,NOUT,W ,Z,LDZ,}
W ORK,LDW ORK,W ORK2,W ORK 3, INFO)
CHARACTER * 1 JOBZ,SORTEV
DOUBLE COM PLEX A (LDA,*),W (*),Z (LD Z,*),W ORK (*)
INTEGERN,LDA,NOUT,LDZ,LDW ORK,INFO
LOG ICAL SELECT
LOG ICAL W ORK 3 (*)
DOUBLE PRECISION W ORK 2 (*)
SU BROUT\mathbb{NE ZGEES_64(JOBZ,SORTEV,SELECT,N,A,LDA,NOUT,W ,Z,LD Z,}
W ORK,LDW ORK,W ORK2,W ORK3, INFO)
CHARACTER * 1 JOBZ,SORTEV
DOUBLE COM PLEX A (LDA,*),W (*),Z (LD Z,*),W ORK (*)
\mathbb{NTEGER*8N,LDA,NOUT,LD Z,LDW ORK,INFO}
LOG ICAL*8 SELECT
LOG ICAL*8 W ORK 3(*)
DOUBLE PRECISION W ORK2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GEES (JOBZ,SORTEV, [SELECT], \(\mathbb{N}], A,[L D A], ~ N O U T], W,[Z],[L D Z]\), [W ORK], [LDW ORK ], [W ORK 2], [W ORK 3], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1)::JOBZ,SORTEV
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ,W ORK

COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::A, Z
\(\mathbb{N}\) TEGER ::N,LDA,NOUT,LDZ,LDW ORK, \(\mathbb{N} F O\)
LOGICAL :: SELECT
LOG ICAL,D IM ENSION (:) ::W ORK 3
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ORK2
SU BROUTINE GEES_64 (JOBZ,SORTEV, [SELECT], \(\mathbb{N}], A,[L D A], \mathbb{N O U T}], W,[Z]\), [LD Z], [W ORK], [LDW ORK], [W ORK 2], [W ORK 3], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) :: JOBZ,SORTEV
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A, Z
\(\mathbb{N}\) TEGER (8) ::N,LDA,NOUT,LD Z,LDW ORK, \(\mathbb{N} F O\)
LOG ICAL (8) :: SELECT
LO G ICAL (8),D IM EN SION (:) ::W ORK 3
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zgees(char jobz, char sortev, int(*select) (doublecom plex), intn, doublecom plex *a, int lda, int *nout, doublecom plex *w, doublecom plex *z, intldz, int*info);
void zgees_64 (char jobz, char sortev, long (*select) (doublecom plex), long n, doublecom plex *a, long lda, long *nout, doublecom plex *w, doublecom plex *z, long ldz, long *info);

\section*{PURPOSE}
zgees com putes foran N -by -N com plex nonsym \(m\) etric \(m\) atrix \(A\), the eigenvalues, the Schur form T, and, optionally, the \(m\) atrix of Schurvectors \(Z\). This gives the Schur factorization \(A=Z * T *(Z * * H)\).

Optionally, it also orders the eigenvalues on the diagonal of the Schur form so that selected eigenvalues are at the top left. The leading colum ns of \(Z\) then form an orthonorm al basis for the invariant subspace comesponding to the selected eigenvalues.

A com plex \(m\) atrix is in Schur form if it is uppertriangular.

\section*{ARGUMENTS}
\(=N\) ': Schurvectors are notcom puted;
\(=\mathrm{V}\) : Schurvectors are com puted.

\section*{SORTEV (input)}

Specifies w hether or not to order the eigenvalues
on the diagonalof the Schur form . = N ': Eigenvalues are notordered:
\(=S^{\prime}:\) Eigenvalues are ordered (see SELECT) .

SELECT (input)
SELECT m ustbe declared EXTERNAL in the calling subroutine. If SORTEV \(=S^{\prime}\) ', SELECT is used to selecteigenvalues to order to the top left of the Schurform. IfSORTEV \(=N^{\prime}\) ', SELECT is notreferenced. The eigenvalue \(W\) ( \()\) is selected if SELECT (W (J) is true.
N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the \(N\) boy -N m atrix A. On exit, A has been overw rilten by its Schur form \(T\).

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

NOUT (output)
If SORTEV \(=\mathrm{N}^{\prime}, \mathrm{NOUT}=0\). If SORTEV \(=\mathrm{S}^{\prime}, \mathrm{NOUT}\)
= num ber ofeigenvalues forw hich SELECT is tue.

W (output)
W contains the com puted eigenvalues, in the sam e order that they appear on the diagonal of the outputSchur form \(T\).

Z (output)
If \(\mathrm{OBBZ}=\mathrm{V}^{\prime}, \mathrm{Z}\) contains the untary \(m\) atrix \(Z\) of Schur vectors. If \(\mathrm{OBZ}=\mathrm{N}^{\prime}, \mathrm{Z}\) is notreferenced.

LD Z (input)
The leading dim ension of the array \(\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1\); if \(\mathrm{OOBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{N}\) 。

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LDWORK.

LDW ORK (input)
The dimension of the aray W ORK. LDW ORK >=
\(\max (1,2 \star \mathrm{~N})\). For good perform ance, LDW ORK must generally be larger.

IfLDW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
W ORK 2 (w orkspace)
dim ension \(\mathbb{N}\) )

W ORK 3 (w orkspace)
dim ension (N) N ot referenced ifSORTEV = \(\mathrm{N}^{\prime}\).
\(\mathbb{I N} F O\) (output)
= 0 : successfiulexit
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an illegalvahe.
\(>0:\) if \(\mathbb{N F O}=i\), and \(i\) is
<= N : the QR algorithm failed to com pute all the eigenvalues; elem ents \(1: \mathbb{I} \mathrm{O}_{-1}^{-1}\) and i+1 \(\mathbb{N}\) ofW contain those eigenvalues which have converged; if JOBZ \(=V\) ', \(Z\) contains the \(m\) atrix which reduces \(A\) to its partially converged Schur form . \(=\mathrm{N}+1\) : the eigenvalues could notbe reordered because some eigenvalues were too close to separate (the problem is very ill-conditioned); \(=\mathrm{N}+2\) : after reordering, roundoff changed values of som e com plex eigenvalues so that leading eigenvalues in the Schur form no longer satisfy SELECT = TRUE.. This could also be caused by underflow due to scaling.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgeesx - com pute foran \(N\)-by -N com plex nonsym \(m\) etric \(m\) atrix \(A\), the eigenvalues, the Schur form \(T\), and, optionally, the \(m\) atrix of Schurvectors Z

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGEESX (JOBZ,SORTEV,SELECT,SENSE,N,A,LDA,NOUT,W,Z,}
LDZ,RCONE,RCONV,W ORK,LDW ORK,W ORK 2,BW ORK 3, INFO)

```
CHARACTER * 1 JOBZ, SORTEV, SEN SE
DOUBLE COM PLEX A (LDA,*),W (*), Z (LD Z,*),W ORK (*)
\(\mathbb{N}\) TEGERN,LDA,NOUT,LDZ,LDW ORK, \(\mathbb{N} F\) F
LOG ICAL SELECT
LOG ICAL BW ORK 3 (*)
DOUBLE PRECISION RCONE,RCONV
DOUBLE PRECISION W ORK 2 (*)
SU BROUTINE ZGEESX_64 (OOBZ,SORTEV,SELECT,SENSE,N,A,LDA,NOUT,W,
    Z,LDZ,RCONE,RCONV,W ORK,LDW ORK,W ORK2,BW ORK 3, \(\mathbb{N} F O\) )
CHARACTER * 1 JOBZ,SORTEV, SENSE
DOUBLE COM PLEX A (LDA, *), W (*), Z (LD Z, \(\left.{ }^{\star}\right), \mathrm{W} O R K\left({ }^{*}\right)\)
\(\mathbb{N} T E G E R * 8 N, L D A, N O U T, L D Z, L D W\) ORK, \(\mathbb{N} F O\)
LO G ICAL*8 SELECT
LOG ICAL* 8 BW ORK 3 (*)
DOUBLE PRECISION RCONE,RCONV
DOUBLE PRECISION W ORK 2 (*)

\section*{F95 INTERFACE}

SU BROUTINE GEESX (JOBZ, SORTEV, [SELECT],SEN SE, \(\mathbb{N}], A,[L D A], N O U T, W\), [Z], [LD Z],RCONE,RCONV, [W ORK], [LDW ORK], [W ORK2], BW ORK 3], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) :: JOBZ, SORTEV, SEN SE
COMPLEX (8), D \(\mathbb{I M} E N S I O N(:):: W\),W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION ( \(:\) : : : : : A , Z
\(\mathbb{N} T E G E R:: N, L D A, N O U T, L D Z, L D W O R K, \mathbb{N F O}\)
LOGICAL : SELECT
LOGICAL,D \(\mathbb{M}\) ENSION (:) ::BW ORK 3
REAL (8) ::RCONE,RCONV
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ORK2

SU BROUTINE GEESX_64 (OOBZ,SORTEV, [SELECT],SENSE, \(\mathbb{N}], A,[L D A], N O U T\), W , [Z], [LDZ],RCONE,RCONV, [WORK], [LDW ORK], [WORK2], [BWORK3], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) :: JOBZ, SORTEV, SENSE
COM PLEX (8), D \(\mathbb{M} E N S I O N\) (:) ::W ,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A, Z
\(\mathbb{N}\) TEGER (8) :: N , LDA , NOUT, LD Z, LDW ORK, \(\mathbb{N} F O\)
LOGICAL (8) :: SELECT
LOGICAL (8), D IM ENSION (:) ::BW ORK 3
REAL (8) ::RCONE,RCONV
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zgeesx (char jobz, char sontev, int(*select) (doublecom plex), char sense, intn, doublecom plex *a, int lda, int *nout, doublecom plex *W , doublecom plex *z, int ldz, double *roone, double *rconv, int *info);
void zgeesx_64 (char jobz, char sortev, long (*select) (doublecom plex), charsense, long n, doublecom plex *a, long lda, long *nout, doublecom plex \({ }^{*}\) w, doublecom plex \({ }^{*} z\), long ldz, double *rcone, double *rconv, long *info);

\section*{PURPOSE}
zgeesx com putes for an \(N\) boy N com plex nonsym m etric \(m\) atrix \(A\), the eigenvalues, the Schur form \(T\), and, optionally, the \(m\) atrix of Schurvectors \(Z\). This gives the Schur factorization \(A=Z * T *(Z * * H)\).

Optionally, italso orders the eigenvalues on the diagonal of the Schur form so that selected eigenvalues are at the top left; com putes a reciprocal condition num ber for the average of the selected eigenvalues (RCONDE); and com putes a
reciprocal condition num ber for the right invariant subspace comesponding to the selected eigenvalues (RCONDV). The leading colum ns of \(Z\) form an orthonorm al basis for this invariant subspace.

For furtherexplanation of the reciprocalcondition num bers RCONDE and RCONDV, see Section 4.10 of the LAPACK U sers' G uide (w here these quantities are called s and sep respectively).

A com plex \(m\) atrix is in Schur form if it is upper triangular.

\section*{ARGUMENTS}

JOBZ (input)
\(=N\) ':Schurvectors are not com puted;
= V ':Schurvectors are com puted.
SORTEV (input)
Specifies w hether ornot to order the eigenvalues on the diagonal of the Schur form . = \(\mathrm{N}^{\prime}\) : Eigen-
values are not ordered;
= S': E igenvalues are ordered (see SELEC T).
SELECT (input)
SELECT mustbe declaredEXTERNAL in the calling
subroutine. If SORTEV \(=\) S', SELECT is used to
selecteigenvalues to order to the top left of the
Schurform. IfSORTEV = N', SELECT is notrefer-
enced. An eigenvalue \(W(\mathcal{H})\) is selected if
SELECT (N) ( \(\mathcal{j}\) ) is true.
SEN SE (input)
D eterm ines which reciprocal condition num bers are com puted. = N ': N one are com puted;
\(=\mathrm{E}:\) : C om puted for average of selected eigen-
values only;
= V ': C om puted for selected right invariant subspace only;
= B ': C om puted forboth. If SEN \(S E=\mathrm{E}^{\prime}, \mathrm{V}^{\prime}\) or B',SORTEV mustequal \(S^{\prime}\).

N (input) The order of the m atrix \(\mathrm{A} \cdot \mathrm{N}>=0\).

A (input/output)
On entry, the N toy- N m atrix A . On exit, A is overw rilten by its Schur form \(T\).

LD A (input)

The leading dim ension of the array A. LDA >= \(\max (1, \mathbb{N})\).

\section*{NOUT (output)}

If SORTEV = N',NOUT = 0. IfSORTEV = S', NOUT
= num ber ofeigenvalues forw hich SELEC \(T\) is true.
W (output)
W contains the com puted eigenvalues, in the same order that they appear on the diagonal of the outputSchur form \(T\).

Z (output)
If \(\mathrm{OOBZ}=\mathrm{V}^{\prime}, \mathrm{Z}\) contains the unitary m atrix Z of Schur vectors. If \(J O B Z=N^{\prime}, Z\) is not referenced.

LD \(Z\) (input)
The leading dm ension of the array Z . LD \(\mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD Z >=N.

RCONE (output)
IfSENSE = E 'or B',RCONE contains the reciprocal condition number for the average of the selected eigenvalues. N ot referenced if SEN SE \(=\) N 'or V '.

RCONV (output)
If SEN SE = V 'or B ', RCONV contains the reciprocal condition num ber for the selected right invariant subspace. N ot referenced if \(\operatorname{SENSE}=\mathrm{N}^{\prime}\) or E'.

W ORK (w orkspace)
dim ension (LD W ORK) On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim allD W ORK .

LDW ORK (input)
The dimension of the array W ORK. LDW ORK >= \(\max \left(1,2 \star^{*}\right)\). A lso, ifSENSE = E'or V'or B',
LDW ORK >= \(2 \star\) NOUT* \((\mathbb{N}-N O U T)\), where NOUT is the num ber of selected eigenvalues com puted by this routine. N ote that \(2 * \mathrm{NOUT} *(\mathbb{N}-\mathrm{NOUT})<=\mathrm{N} * \mathrm{~N} / 2\). For good perform ance, LD W O RK m ustgenerally be larger.

W ORK 2 (w orkspace)
dim ension \((\mathbb{N})\)
BW ORK 3 (w orkspace)
dim ension \((\mathbb{N}) \mathrm{N}\) ot referenced if \(S O\) RTEV \(=\mathrm{N}^{\prime}\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the \(i\)-th argum enthad an illegalvalue.
> 0 : if \(\mathbb{N} F O=\) i, and is
<= N : the Q R algorithm failed to com pute all the eigenvalues; elem ents \(1: \mathbb{I} \mathrm{O}^{-1}\) and i+1 \(\mathbb{N}\) ofW contain those eigenvalues which have converged; if \(J O B Z=V ', Z\) contains the transform ation which reduces A to its partially converged Schur form . \(=\mathrm{N}+1\) : the eigenvalues could not be reordered because some eigenvahues were too close to separate (the problem is very ill-conditioned); = \(\mathrm{N}+2\) : after reordering, roundoff changed values of som e com plex eigenvalues so that leading eigenvalues in the Schur form no longer satisfy SELEC T = TRU E. This could also be caused by underflow due to scaling.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgeev -com pute for an \(N\)-by -N com plex nonsym \(m\) etric \(m\) atrix A , the eigenvalues and, optionally, the left and/orright eigenvectors

\section*{SYNOPSIS}
```

SUBROUTINE ZGEEV (JOBVL,JOBVR,N,A,LDA,W ,VL,LDVL,VR,LDVR,
W ORK,LDW ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 J BVL,NOBVR
DOUBLE COMPLEX A (LDA,*),W (*), VL (LDVL,*), VR (LDVR,*),
W ORK (*)
INTEGER N,LDA,LDVL,LDVR,LDW ORK, INFO
DOUBLE PRECISION W ORK 2 (*)
SUBROUT\mathbb{NE ZGEEV_64 (JOBVL,JOBVR,N,A,LDA,W ,VL,LDVL,VR,LDVR,}
W ORK,LDW ORK,W ORK2, INFO)
CHARACTER * 1 JOBVL, JOBVR
DOUBLE COMPLEX A (LDA,*),W (*),VL (LDVL,*), VR (LDVR,*),
W ORK (*)
\mathbb{NTEGER*8N,LDA,LDVL,LDVR,LDW ORK, INFO}
DOUBLE PRECISION W ORK2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GEEV (OOBVL, JOBVR, \(\mathbb{N}], A,[L D A], W, V L,[L D V L], V R,[L D V R]\), [W ORK ], [LDW ORK], \(\mathbb{W}\) ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1)::JOBVL, JOBVR
COMPLEX (8),D \(\mathbb{I}\) ENSION (:) ::W ,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : : : : A,VL,VR
\(\mathbb{N} T E G E R:: N, L D A, L D V L, L D V R, L D W O R K, \mathbb{N} F O\)

REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ORK2

SU BROUTINE GEEV_64 (JOBVL, \(\operatorname{OBCR}, \mathbb{N}], A,[L D A], W, V L,[L D V L], V R\), \([[L D V R],[W O R K],[L D W O R K],[W O R K 2],[\mathbb{N F O}])\)

CHARACTER ( \(L E N=1\) ) :: JOBVL, OOBVR
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,\(:\) : : : A, VL, VR
\(\mathbb{N}\) TEGER (8) :: N , LDA , LDVL, LDVR, LDW ORK , \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::WORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zgeev (char jंbvl, char j̣bvr, intn, doublecom plex *a, int lda, doublecom plex \({ }^{*}\), doublecom plex \({ }^{*}\) vl, int ldvl, doublecom plex *vr, int ldvr, int *info);
void zgeev_64 (char jobvıl, char jobvr, long n, doublecom plex
*a, long lda, doublecom plex *w, doublecom plex *vl,
long ldvl, doublecom plex *vr, long ldvr, long
*info);

\section*{PURPOSE}
zgeev com putes foran \(N\)-by -N com plex nonsym m etric m atrix \(A\), the eigenvalues and, optionally, the left and/or right eigenvectors.

The righteigenvectorv (i) of satisfies
\(A * v(j)=\operatorname{lam} \operatorname{bda}(\mathcal{j}) * v(j)\)
w here lam boda ( 7 ) is its eigenvalue.
The lefteigenvectoru ( 7 ) ofA satisfies

where \(u(j){ }^{\star *} H\) denotes the conjugate transpose ofu ( \(\mathcal{j}\) ).

The com puted eigenvectors are norm alized to have Euclidean norm equal to 1 and largest com ponentreal.

\section*{ARGUMENTS}

JOBVL (input)
\(=\mathrm{N}\) ': lefteigenvectors of A are notcom puted;
\(=\mathrm{V}\) ': lefteigenvectors of are com puted.

JO BVR (input)
\(=\mathrm{N}\) ': righteigenvectors of \(A\) are not com puted;
\(=\mathrm{V}\) ': righteigenvectors of A are com puted.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
A (input/output)
On entry, the \(N-b y-N\) m atrix A. On exit, A has been overw ritten.

LD A (input)
The leading dim ension of the anay A. LDA >= \(\max (1, N)\).

W (output)
W contains the com puted eigenvalues.

\section*{VL (output)}

If \(\mathrm{JOBVL}=\mathrm{V}\) ', the left eigenvectors \(u(1)\) are stored one after another in the colum ns of V L, in the sam e order as theireigenvalues. If \(\mathrm{JOBVL}=\) \(N^{\prime}, V L\) is not referenced. \(u(\mathcal{F})=V L(:, 7)\), the \(j\) th collumn of VL.

LDVL (input)
The leading dim ension of the array VL. LD V L >=1; if \(\mathrm{JOBVL}=\mathrm{V}^{\prime}, \mathrm{LD} V \mathrm{~L}>=\mathrm{N}\).

VR (input)
If \(\mathrm{JOBVR}=\mathrm{V}\) ', the right eigenvectors \(\mathrm{V}(\mathrm{I})\) are stored one after another in the colum ns of VR, in the sam e order as theireigenvalues. If \(J O B V R=\) \(N^{\prime}, \operatorname{VR}\) is notreferenced. \(\mathrm{V}(\mathcal{1})=\operatorname{VR}(:, 7)\), the \(j\) th colum \(n\) ofVR.

LDVR (input)
The leading dim ension of the array VR. LD VR >=1; if \(J 0 B V R=V ', L D V R>=N\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the amray W ORK. LDW ORK >= \(\max (1,2 \star \mathrm{~N})\). For good perform ance, LDW ORK must generally be larger.

If LD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 2 (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N N F O}=-i\), the \(i\)-th argum enthad an illegalvalue.
\(>0:\) if \(\mathbb{N F O}=i\), the \(Q R\) algorithm failed to com pute all the eigenvalues, and no eigenvectors have been com puted; elem ents and i+1 N of W contain eigenvalues which have converged.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgeevx -com pute foran \(N\)-by -N com plex nonsym \(m\) etric \(m\) atrix A, the eigenvalues and, optionally, the leftand/or right eigenvectors

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGEEVX (BALANC,JOBVL,JOBVR,SENSE,N,A,LDA,W,VL,}
LDVL,VR,LDVR,\PiO,IHI,SCALE,ABNRM,RCONE,RCONV,W ORK,
LDW ORK,W ORK2,INFO)
CHARACTER * 1 BALANC,JOBVL,JOBVR,SENSE
DOUBLE COMPLEX A (LDA,*),W (*),VL (LDVL,*), VR (LDVR,*),
W ORK (*)

```

```

DOUBLE PRECISION ABNRM
DOUBLE PRECISION SCALE (*),RCONE (*),RCONV (*),W ORK 2 (*)
SU BROUT\mathbb{NE ZGEEVX_64(BALANC,NOBVL,JOBVR,SENSE,N,A,LDA,W ,V L,}
LDVL,VR,LDVR, \#O,HI,SCALE,ABNRM,RCONE,RCONV,WORK,
LDW ORK,W ORK2,INFO)
CHARACTER * 1 BALANC,JOBVL,NOBVR,SENSE
DOUBLE COMPLEX A (LDA,*),W (*),VL (LDVL,*), VR (LDVR,*),
W ORK (*)

```

```

DOUBLE PRECISION ABNRM
DOUBLE PRECISION SCALE (*),RCONE (*),RCONV (*),W ORK2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GEEVX BALANC, JOBVL, JOBVR,SENSE, \(\mathbb{N}], A,[L D A], W, V L\), [LDVL],VR, [LDVR], \(\mathbb{L} O, \mathbb{H} I, S C A L E, A B N R M, R C O N E, R C O N V,[W O R K]\), LDW ORK, [W ORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::BALANC, JOBVL, JO BVR , SEN SE
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: ::) : : A , VL, VR
\(\mathbb{N}\) TEGER :: N, LDA, LDVL, LDVR, \(\Pi O, \mathbb{H} I, L D W O R K, \mathbb{N F O}\)
REAL (8) ::ABNRM
REAL (8), D \(\mathbb{M}\) ENSION (:) :: SCALE,RCONE,RCONV,W ORK 2
 \(\mathrm{VL},[\mathrm{LDVL}], \mathrm{VR},[\mathrm{LDVR}], \Pi O, \mathbb{H} \mathrm{I}, \mathrm{SCALE}, \mathrm{ABNRM}, \mathrm{RCONE}, \mathrm{RCONV}\), [W ORK ], LDW ORK, [W ORK2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::BALANC, JOBVL, JOBVR, SEN SE
COM PLEX (8), D \(\mathbb{I M} E N S \mathbb{O N}(:):: W\),W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : : A, VL, VR
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{LD} A, L D V L, L D V R, \mathbb{L}, \mathbb{H} \mathrm{I}, \mathrm{LD} W\) ORK , \(\mathbb{N} F O\)
REAL (8) ::ABNRM
REAL (8), D \(\mathbb{M} E N S I O N(:):: S C A L E, R C O N E, R C O N V, W O R K 2\)

\section*{C INTERFACE}
\#include <sunperfh>
void zgeevx (char, char, char, char, int, doublecom plex*, int, doublecom plex*, doublecom plex*, int, doublecom plex*, int, int*, int*, double*, double*, double*, double*, int*);
void zgeevx_64 (char, char, char, char, long, doublecom plex*, long, doublecom plex*, doublecom plex*, long, doublecom plex*, long, long*, long*, double*, double*, double*, double*, long*);

\section*{PURPOSE}
zgeevx com putes for an N -by -N com plex nonsym m etric m atrix A , the eigenvalues and, optionally, the left and/or right eigenvectors.

Optionally also, itcom putes a balancing transform ation to im prove the conditioning of the eigenvalues and eigenvectors ( \(\mathbb{L} O, \mathbb{H}\) I, SCA LE , and A BN RM ), reciprocal condition num bers for the eigenvalues (RCONDE), and reciprocal condition num bers for the right
eigenvectors (RCONDV).

The righteigenvectorv (i) ofA satisfies
\[
A * v(\mathcal{J})=\operatorname{lam} \operatorname{bda}(\mathcal{j}) * v(\mathcal{J})
\]
w here lam boda ( 7 ) is its eigenvalue.
The lefteigenvectoru (7) ofA satisfies
where \(u(\mathcal{j}) * * H\) denotes the conjugate transpose of \(u(\mathcal{j})\).
The com puted eigenvectors are norm alized to have Euclidean norm equalto 1 and largest com ponent real.

B alancing a \(m\) atrix \(m\) eans perm uting the row sand colum ns to m ake itm ore nearly upper triangular, and applying a diagonalsim ilarity transform ation \(D\) * \(A\) * \(D * *(-1)\), where \(D\) is a diagonalm atrix, to \(m\) ake its row \(s\) and colum ns closer in norm and the condition num bers of its eigenvalues and eigenvectors sm aller. The com puted reciprocalcondition num bers comespond to the balanced matrix. Perm uting row s and columns will not change the condition num bers (in exact arithm etic) butdiagonal scaling w ill. For further explanation of balancing, see section 4.102 of the LA PA CK U sers' G uide.

\section*{ARGUMENTS}

BALANC (input)
Indicates how the inputm atrix should be diagonally scaled and/orperm uted to im prove the conditioning of its eigenvalues. \(=\mathrm{N}\) ': D o not diagonally scale orperm ute;
\(=\mathrm{P}\) ': Perform permutations to m ake the m atrix m ore nearly upper triangular. D o notdiagonally scale; = S :: D iagonally scale the matrix, ie. replace \(A\) by \(D * A * D * *(-1)\), where \(D\) is a diagonal \(m\) atrix chosen to \(m\) ake the row \(s\) and colum ns of \(A\) m ore equal in norm .D o notperm ute; \(=\mathrm{B}\) ':Both diagonally scale and perm ute A.

C om puted reciprocalcondition num bers willbe for the \(m\) atrix afterbalancing and/orperm uting. Per\(m\) uting does not change condition num bers (in exact arithm etic), butbalancing does.

JOBVL (input)
\(=\mathrm{N}\) ': lefteigenvectors of A are notcom puted;
= V': lefteigenvectors of A are computed. If
SEN SE = E 'or B', JO BV L m ust= V'.
JO BVR (input)
\(=\mathrm{N}\) ': righteigenvectors of A are not com puted;
\(=\mathrm{V}\) ': righteigenvectors of A are com puted. If
SEN SE = E 'or B', JOBVR m ust= V'.

D eterm ines which reciprocal condition num bers are com puted. = N ':N one are com puted;
\(=\mathrm{E}\) : \(: \mathrm{C}\) om puted foreigenvalues only;
\(=\mathrm{V}\) : C om puted for righteigenvectors only;
= B ': C om puted foreigenvalues and right eigenvectors.

If SEN SE = E 'or B ', both leftand right eigenvectors must also be computed ( \(\mathrm{JOBVL}=\mathrm{V}\) 'and JobVR \(=V\) ) .

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the N -by -N m atrix A. On exit, A has been overw ritten. If JO BVL \(=\mathrm{V}\) 'orJOBVR \(=\mathrm{V}\) ', \(A\) contains the Schur form of the balanced version of the m atrix A.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, \mathbb{N})\).

W (output)
W contains the com puted eigenvalues.

\section*{VL (output)}

If \(\operatorname{JOBVL}=\mathrm{V}\) ', the left eigenvectors \(\mathrm{u}(\mathrm{y})\) are stored one afteranother in the colum ns of V L, in the sam e order as their eigenvalues. If \(J 0 \mathrm{BVL}=\) \(N^{\prime}, \mathrm{VL}\) is not referenced. \(\mathrm{u}(\mathcal{j})=\mathrm{VL}(:, r)\), the \(j\) th © Clumn of VL.

LDVL (input)
The leading dim ension of the array V L. LD V L >=1; if \(\mathrm{JOBVL}=\mathrm{V}\) ', LDVL \(>=\mathrm{N}\) 。

VR (input)
If \(\mathrm{JOBVR}=\mathrm{V}\) ', the right eigenvectors \(\mathrm{v}(\mathrm{I})\) are stored one after another in the colum ns of VR, in the sam e order as theireigenvalues. If \(J O B V R=\) \(N^{\prime}, \operatorname{VR}\) is notreferenced. \(\left.\operatorname{V(~}\right)=\operatorname{VR}(:, 7)\), the \(j\) th colum n ofVR.

LDVR (input)
The leading dim ension of the array \(V R . L D V R>=1\); if \(J 0 B V R=V ', L D V R>=N\).

ㅍㅇ (output)

ㅍO and \(\mathbb{H}\) I are integervalues determ ined when \(A\) \(w\) as balanced. The balanced \(A(i, 7)=0\) if \(I>J\) and \(J=1, \ldots, I L O-1\) or \(I=\mathbb{H} I+1, \ldots, N\).

IH I (output)
HO and \(\mathbb{H}\) I are integervalues determ ined when \(A\) \(w\) as balanced. The balanced \(A(i, 7)=0\) if I> J and \(J=1, \ldots, I L O-1\) or \(I=\mathbb{H} I+1, \ldots, N\).

SCALE (output)
D etails of the perm utations and scaling factors applied w hen balancing \(A\). IfP ( 7 ) is the index of the row and column interchanged w ith row and colum \(n j\), and \(D(j)\) is the scaling factorapplied to row and column \(j\) then SCALE \((J)=P(J)\), for \(J=1, \ldots\), IIO \(-1=D(J)\), for \(J=\mathbb{L O}, \ldots, \mathbb{H} I=\) \(P(J) \quad\) for \(J=\mathbb{H} I+1, \ldots, N\). The order in which the interchanges are \(m\) ade is \(N\) to \(\mathbb{H} I+1\), then 1 to 핑․

\section*{ABNRM (output)}

The one-norm of the balanced \(m\) atrix the \(m\) axim um of the sum of absolute values of elem ents of any colum n).

RCONE (output)
RCONE ( \()\) ) is the reciprocalcondition num ber of the \(j\) th eigenvalue.

RCONV (output)
RCONV ( \(j\) ) is the reciprocalcondition num ber of the \(j\) th righteigenvector.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK ( 1 ) retums the optim al LDW ORK.

LDW ORK (output)
The dim ension of the array \(W\) ORK. IfSENSE \(=N^{\prime}\) or E', LDW ORK \(>=\mathrm{max}\left(1,2 \star^{*} \mathrm{~N}\right)\), and ifSENSE \(=\mathrm{V}^{\prime}\) or \(B\) ', LD W ORK \(>=N * N+2 * N\). Forgood perform ance, LDW ORK m ust generally be larger.

IfLDW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 2 (w orkspace)
dim ension (2*N)
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue.
\(>0:\) if \(\mathbb{N F O}=\) i, the Q R algorithm failed to com pute all the eigenvalues, and no eigenvectors or condition num bers have been com puted; elem ents 1: IHO-1 and i+1 N ofW contain eigenvalues which have converged.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgegs - routine is deprecated and has been replaced by routine CGGES

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZGEGS (OOBVSL,JO BV SR,N,A LDA,B,LDB,ALPHA,BETA,VSL,}
LDVSL,VSR,LDVSR,W ORK,LDW ORK,W ORK 2, INFO)
CHARACTER * 1 JOBVSL,NOBVSR
DOUBLE COM PLEX A (LDA,*), B (LDB,*), ALPHA (*), BETA (*),
VSL (LDVSL,*),VSR (LDVSR,*),W ORK (*)
INTEGERN,LDA,LDB,LDVSL,LDVSR,LDW ORK,INFO
DOUBLE PRECISION W ORK 2 (*)
SUBROUT\mathbb{NE ZGEGS_64(JOBVSL,JOBVSR,N,A,LDA,B,LDB,ALPHA,BETA,}
VSL,LDVSL,VSR,LDVSR,W ORK,LDW ORK,WORK2,\mathbb{NFO)}
CHARACTER * 1 JOBVSL,NOBVSR
DOUBLE COM PLEX A (LDA,*), B (LDB,*), ALPHA (*), BETA (*),
VSL (LDVSL,*),VSR (LDVSR,\star),W ORK (*)
INTEGER*8N,LDA,LDB,LDVSL,LDVSR,LDW ORK,INFO
DOUBLE PRECISION W ORK2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GEGS (OOBVSL, \(\mathfrak{J} 0 \mathrm{BV}\) SR, \(\mathbb{N}], A,[L D A], B,[L D B], A L P H A, B E T A\), VSL, [LDVSL],VSR, [LDVSR], [W ORK], [LDW ORK], [W ORK 2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) :: JO BV SL, JO BV SR
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::ALPHA,BETA,W ORK
COM PLEX (8),D IM ENSION (:,:) ::A, B,VSL,VSR
\(\mathbb{N} T E G E R:: N, L D A, L D B, L D V S L, L D V S R, L D W O R K, \mathbb{N F O}\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ORK 2

SU BROU T \(\mathbb{N}\) E G EG S_64 (JOBV SL , JO BV SR , \(\mathbb{N}], A,[L D A], B,[L D B], A L P H A\), BETA, VSL, [LDVSL],VSR, [LDVSR], [W ORK], [LDW ORK], [W ORK2], [ \(\mathbb{N}\) FO ])

CHARACTER ( \(\llcorner E N=1\) ) :: JOBVSL, JOBV SR
COMPLEX (8), D \(\mathbb{M}\) ENSION (:) ::ALPHA, BETA, W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: ::) ::A, B, VSL , VSR
\(\mathbb{N}\) TEGER (8) :: N , LD A , LD B , LDV SL, LDV SR, LDW ORK, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zgegs (char jobvsl, char jंbvsr, int n, doublecom plex
*a, int lda, doublecom plex *b, int ldb, doublecom -
plex *alpha, doublecom plex *beta, doublecom plex
*Vsl, int ldvsl, doublecom plex *vsr, int ldvsr, int*info);
void zgegs_64 (char jobvssl, char jobvssr, long n, doublecom plex *a, long lda, doublecom plex *b, long ldb, doublecom plex *alpha, doublecom plex *beta, doublecom plex *Vsl, long ldvsl, doublecom plex *Vsr, long ldvsr, long *info);

\section*{PURPOSE}
zgegs routine is deprecated and has been replaced by routine CGGES.

CGEGS com putes fora pair of N łoy-N com plex nonsymm etric \(m\) atrices A, B : the generalized eigenvalues (alpha, beta), the com plex Schur form (A,B), and optionally left and/or rightSchurvectors ( V SL and V SR).
(Ifonly the generalized eigenvalues are needed, use the driverCGEGV instead.)

A generalized eigenvalue for a pair of \(m\) atrices ( \(A, B\) ) is, roughly speaking, a scalar \(w\) ora ratio alpha/beta \(=\mathrm{w}\), such that A -w *B is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation forbeta \(=0\), and even forboth being zero. A good beginning reference is the book, "M atrix C om putations", by G.Golub \& C . van Loan (Johns H opkins U .Press)

The (generalized) Schur form of a pair of \(m\) atrices is the result ofm ultiplying both \(m\) atrices on the leftby one unitary \(m\) atrix and both on the rightby anotherunitary matrix,
these tw o unitary \(m\) atrices being chosen so as to bring the pair ofm atrioes into upper triangular form w ith the diagonal elem ents ofB being non-negative realnum bers this is also called com plex Schur form .)

The left and rightSchurvectors are the colum ns ofV SL and VSR, respectively, where VSL and VSR are the unitary \(m\) atrices
which reduce A and B to Schur form :

Schurform of \((A, B)=(N S L) \star * H A(N S R),(N S L) * * H B(N S R))\)

\section*{ARGUMENTS}

JO BV SL (input)
\(=N^{\prime}:\) do notcom pute the left:Schurvectors;
\(=\mathrm{V}\) ': com pute the leftSchur vectors.
\(J 0 B V S R\) (input)
\(=N\) : do notcom pute the rightSchurvectors;
\(=\mathrm{V}\) ': com pute the rightSchurvectors.
N (input) The order of the m atrices \(\mathrm{A}, \mathrm{B}, \mathrm{VSL}\), and VSR. N \(>=0\).

A (input/output)
O \(n\) entry, the first of the pairofm atrices w hose generalized eigenvalues and (optionally) Schur vectors are to be com puted. On exit, the generalized Schur form of A.

LD A (input)
The leading dim ension ofA. LD A \(>=\max (1, N)\).
B (input/output)
O \(n\) entry, the second of the pair of \(m\) atrices \(w\) hose generalized eigenvalues and (optionally) Schur vectors are to be com puted. On exit, the generalized Schur form ofB.

LD B (input)
The leading dim ension ofB. LD B \(>=m\) ax \((1, N)\).

\section*{ALPHA (output)}

On exit, ALPHA ( \(\mathcal{)}\) ) BETA ( \(\mathcal{J}\), \(\mathcal{=} 1, \ldots, N\), w illbe the generalized eigenvalues. A LPHA ( \(\mathcal{\rho}, \dot{于} 1, \ldots, N\) and BETA ( \(\mathcal{i}, \dot{于} 1, \ldots, N\) are the diagonals of the com plex Schur form ( \(A, B\) ) output by CGEGS. The

BETA ( \()\) w illlbe non-negative real.

N ote: the quotients A LPHA ( \(\mathcal{7}\) ) \(B\) ETA ( \()\) may easily over- orunderflow, and BETA ( \()\) m ay even be zero. Thus, the user should avoid naively com puting the ratio alpha/beta. H ow ever, A LPH A w illbe alw ays less than and usually com parable w ith norm (A) in m agnitude, and BETA alw ays less than and usually com parable \(w\) th norm (B).

\section*{BETA (output)}

See the description of A LPH A .
VSL (input)
If 0 OVVSL = V',VSL willcontain the left Schur vectors. (See "Puppose", above.) N ot referenced if \(J 0 \mathrm{BV}\) SL \(=\mathrm{N}\) '.

LD V SL (input)
The leading dim ension of the \(m\) atrix V SL. LD V SL >= 1 , and if JO BV SL \(=\mathrm{V}\) ', LD V SL \(>=\mathrm{N}\).

VSR (input)
If \(J 0\) BV SR \(=V\) ', VSR willcontain the right Schur vectors. (See "Purpose", above.) N ot referenced if \(J 0 \mathrm{BV}\) SR = \(\mathrm{N}^{\prime}\).

LD VSR (input)
The leading dim ension of the \(m\) atrix \(V\) SR .LD V SR >= 1 , and if \(J O B V S R=V\) ', LD VSR \(>=N\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the aray \(W\) ORK. LDW ORK >= \(\max (1,2 \star \mathrm{~N})\). For good perform ance, LDW ORK must generally be larger. To com pute the optim alvahue of LDW ORK, call ILAENV to getblocksizes (for CGEQRF, CUNM QR, and CUNGQR.) Then com pute: NB as the MAX of the blocksizes forCGEQRF, CUNM \(Q R\), and CUNGQR; the optim allDW ORK is \(N\) * \(N B+1\) ).

If LD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA .
dim ension ( \(3 * N\) )
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue.
\(=1, \ldots, N\) : The Q Z teration failed. (A , B ) are not in Schur form, butA LPHA ( 1 ) andBETA ( 1 ) should be correct for \(\mathcal{F} \mathbb{N} F O+1, \ldots, N .>N\) : errors that usually indicate LA PA C K problem s:
\(=\mathrm{N}+1\) : error retum from CGGBAL
\(=\mathrm{N}+2\) : error retum from \(C\) G EQRF
\(=\mathrm{N}+3\) : error retum from \(C U N M Q R\)
\(=\mathrm{N}+4\) : error retum from CUNGQR
\(=\mathrm{N}+5\) : error retum from CGGHRD
\(=\mathrm{N}+6\) : emor retum from CHGEQZ (otherthan failed
iteration) \(=\mathrm{N}+7\) : emor retum from CGGBAK (com put-
ing V SL )
\(=\mathrm{N}+8\) : error retum from CGGBAK (com puting VSR)
\(=\mathrm{N}+9\) : error retum from CLA.SCL (various places)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zgegv - routine is deprecated and has been replaced by routine CGGEV

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGEGV (JOBVL,JOBVR,N,A,LDA,B,LDB,ALPHA,BETA,VL,}
LDVL,VR,LDVR,W ORK,LDW ORK,W ORK 2, \mathbb{NFO)}

```
CHARACTER * 1 JOBVL, JOBVR
DOUBLE COM PLEX A (LDA, *), B (LDB,*), ALPHA (*), BETA (*),
VL (LDVL, \(\left.{ }^{\star}\right), V R(L D V R, \star), W\) ORK ( \(\left.{ }^{*}\right)\)
\(\mathbb{N}\) TEGER N,LDA,LDB,LDVL,LDVR,LDW ORK, \(\mathbb{N} F O\)
DOUBLE PRECISION W ORK 2 (*)
SU BROUTINE ZGEGV_64(OBVL, JOBVR,N,A,LDA,B,LDB,ALPHA,BETA,VL,
    LDVL,VR,LDVR,W ORK,LDW ORK,W ORK2, \(\mathbb{N} F O\) )
CHARACTER * 1 JOBVL, JOBVR
DOUBLE COMPLEX A (LDA,*), B (LDB,*), ALPHA (*), BETA (*),
VL (LDVL, *) , VR (LDVR, *), W ORK (*)
\(\mathbb{N}\) TEGER*8N,LDA,LDB,LDVL,LDVR,LDW ORK, \(\mathbb{N} F O\)
DOUBLE PRECISION W ORK 2 (*)

\section*{F95 INTERFACE}

SUBROUTINE GEGV (JOBVL, JOBVR, \(\mathbb{N}\) ], A, [LDA], B, [LDB],ALPHA,BETA, VL, [LDVL],VR, [LDVR], \(\mathbb{W}\) ORK], [LDW ORK], [W ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) :: JOBVL, JOBVR
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::ALPHA,BETA,W ORK
COMPLEX (8), D \(\mathbb{M}\) ENSION (:,:) :: A, B,VL,VR
\(\mathbb{N} T E G E R:: N, L D A, L D B, L D V L, L D V R, L D W O R K, \mathbb{N F O}\)

SU BROUTINE GEGV_64 (OBVL, OBVR, \(\mathbb{N}], A,[L D A], B,[L D B], A L P H A\), BETA, \(\mathrm{VL},[\mathrm{LDVL}], \mathrm{VR},[L D V R],[\mathbb{W} O R K],[L D W O R K],[W O R K 2],[\mathbb{N F O}])\)

CHARACTER ( \(L E N=1\) ) :: JOBVL, DOBVR
COMPLEX (8), D \(\mathbb{M}\) ENSION (:) ::ALPHA, BETA, W ORK
COM PLEX (8), D \(\mathbb{M} E N S I O N(:,:):: A, B, V L, V R\)
\(\mathbb{N} \operatorname{TEGER}(8):: N, L D A, L D B, L D V L, L D V R, L D W O R K, \mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zgegv (char jobvl, char jobvr, intn, doublecom plex *a, int lda, doublecom plex *b, int ldb, doublecom plex *alpha, doublecom plex *beta, doublecom plex *vl, int \(1 d v l\), doublecom plex *vr, int ldvr, int *info);
void zgegv_64 (char j́jbvl, char jobvr, long n, doublecom plex
*a, long lda, doublecom plex *b, long ldb, doublecom plex *alpha, doublecom plex *beta, doublecom plex *vl, long ldvl, doublecom plex *vr, long ldvr, long *info);

\section*{PURPOSE}
zgegv routine is deprecated and has been replaced by routine CGGEV .

CGEGV com putes for a pair of N boy- N com plex nonsym m etric \(m\) atrices A and B, the generalized eigenvalues (alpha, beta), and optionally, the left and/or right generalized eigenvectors (VL and VR).

A generalized eigenvahe for pair of \(m\) atriges \((A, B)\) is, roughly speaking, a scalar w or a ratio alpha/beta \(=\mathrm{w}\), such that \(A-w * B\) is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation forbeta=0, and even forboth being zero. A good beginning reference is the book, "M atrix C om putations", by G.G olub \& C .van Loan (Johns H opkins U .P ress)

A rightgeneralized eigenvector corresponding to a generalized eigenvalue \(w\) for pair ofm atriges ( \(A, B\) ) is a vector \(r\) such that \((A-w B) r=0\). A left generalized eigenvector is a vector lsuch that \(l^{* *} H *(A-w B)=0\), where l**H is the conjugate-transpose ofl.

N ote: this routine perform s "fullbalancing" on A and B. See "FurtherD etails", below .

\section*{ARGUMENTS}

JO BVL (input)
\(=\mathrm{N}\) ': do notcom pute the leftgeneralized eigenvectors;
\(=V^{\prime}\) : com pute the left generalized eigenvectors.

JO BVR (input)
\(=\mathrm{N}\) ': do not com pute the right generalized eigenvectors;
\(=\mathrm{V}\) : com pute the right generalized eigenvectors.
N (input) The order of the m atrioes \(\mathrm{A}, \mathrm{B}, \mathrm{VL}\), and VR. \(\mathrm{N}>=\) 0.

A (input/output)
O n entry, the first of the pair ofm atrices w hose generalized eigenvalues and (optionally) generalized eigenvectors are to be com puted. On exit, the contents will have been destroyed. Fora description of the contents of A on exit, see "FurtherD etails", below .)

LD A (input)
The leading dim ension ofA. LD \(A>=\max (1, N)\).

B (input/output)
O n entry, the second of the pair ofm atrices w hose generalized eigenvalues and (optionally) generalized eigenvectors are to be com puted. On exit, the contents will have been destroyed. Fora description of the contents of \(B\) on exit, see "FurtherD etails", below .)

LD B (input)
The leading dim ension ofB. LD B \(>=m\) ax \((1, N)\).

ALPHA (output)
On exit, A LPHA ( 7 ) \(N L(\mathcal{7}), \dot{F} 1, \ldots, N\), will be the generalized eigenvahues.

N ote: the quotients ALPHA (J) NL (J) m ay easily over or underflow, and VL ( 7 ) \(m\) ay even be zero. Thus, the user should avoid naively com puting the
ratio alpha/beta. H ow ever, A LPH A w illbe alw ays less than and usually com parable w th norm (A) in m agnitude, and VL alw ays less than and usually com parable \(w\) th norm (B).

BETA (output)
If \(\mathrm{JO} \mathrm{BVL}=\mathrm{V}\) ', the left generalized eigenvectors.
(See "Purpose", above.) Each eigenvectorw ill.be scaled so the largest com ponentw ill have abs (real part) + abs(mag. part) \(=1\), *exœept \({ }^{\star}\) that for eigenvalues \(w\) th alpha=beta=0, a zero vector \(w i l l\) be retumed as the corresponding eigenvector. N ot referenced if \(\mathrm{JOBVL}=\mathrm{N}^{\prime}\).
VL (output)
If \(\mathrm{JOBVL}=\mathrm{V}\) ', the left generalized eigenvectors.
(See "Pupose", above.) Each eigenvectorw ill.be scaled so the largest com ponentw ill have abs (real
part) \(+\mathrm{abs}(\) ( m ag. part) \(=1\), *exœept* that for eigenvalues w ith alpha=beta=0, a zero vector w ill be retumed as the comesponding eigenvector. N ot referenced if \(\mathrm{JO} \mathrm{BVL}=\mathrm{N}\) '.

LDVL (input)
The leading dim ension of the \(m\) atrix \(V \mathrm{~L} . \mathrm{LD} V \mathrm{~L}>=1\), and if \(\mathrm{JOBVL}=\mathrm{V}\) ', LDVL >=N.

\section*{VR (output)}

If \(\mathrm{JOBVR}=\mathrm{V}\) ', the right generalized eigenvectors. (See "Pupose", above.) Each eigenvector w illbe scaled so the largest com ponentw ill have abs(real part) +abs (im ag. part) \(=1\),*except* that foreigenvalues w th alph \(a=\) beta \(=0\), a zero vector \(w\) ill be retumed as the corresponding eigenvector. N ot referenced if \(\mathrm{JO} B V R=\mathrm{N}^{\prime}\).

LDVR (input)
The leading dim ension of the \(m\) atrix \(V R\).LD \(V R>=1\), and if \(J O B V R=V\) ', LDVR \(>=N\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F} F=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the amray W ORK. LDW ORK >= \(\mathrm{max}(1,2 \star \mathrm{~N})\). For good perform ance, LDW ORK must generally be larger. To com pute the optim alvalue of LDW ORK, call IIA ENV to getblocksizes (for CGEQRF, CUNM QR, and CUNGQR.) Then com pute: NB as the MAX of the blocksizes forCGEQRF, CUNM \(Q R\), and

CUNGQR; The optim alldW ORK is MAX \((2 * N, N * \mathbb{N}+1)\)
).
IfLDW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
W ORK 2 (w orkspace)
dim ension ( \(8 * \mathrm{~N}\) )
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{N}\) FO \(=-\) i, the i-th argum enthad an illegalvahue.
\(=1, \ldots, N\) : The Q Z iteration failed. N o eigenvectors have been calculated, butA LPH A ( \(\mathcal{j}\) ) and VL ( \(\mathcal{j}\) )
should be correct for \(\mathcal{j} \mathbb{N} F O+1, \ldots, N .>N\) :
enrors that usually indicate LA PACK problem s:
\(=\mathrm{N}+1\) : error retum from CGGBAL
\(=\mathrm{N}+2\) : error retum from CGEQRF
\(=N+3\) : error retum from CUNM \(Q R\)
\(=\mathrm{N}+4\) : error retum from CUNGQR
\(=\mathrm{N}+5\) : error retum from CGGHRD
\(=\mathrm{N}+6\) : error retum from CHGEQZ (other than failed iteration) \(=\mathrm{N}+7\) : emor retum from CTGEVC
\(=\mathrm{N}+8\) : enror retum from CGGBAK (com puting VL)
\(=\mathrm{N}+9\) : error retum from CGGBAK (com puting VR)
\(=\mathrm{N}+10\) : enror retum from CLASCL (various calls)

\section*{FURTHER DETAILS}

Balancing

This driver calls C G G B A L to both perm ute and scale row s and colum ns of \(A\) and \(B\). The perm utationsPL and PR are chosen so that \(\mathrm{PL} * \mathrm{~A} * P R\) and \(P L * B * R\) w illlbe upper triangular except for the diagonal blocksA ( \(i: j i: j\) ) and \(B(i: j i: j\), w ith \(i\) and jas close together as possible. The diagonal scaling \(m\) atrices \(D L\) and \(D R\) are chosen so that the pair DL*PL*A*PR*DR,DL*PL*B*PR*DR have elem ents close to one (except for the elem ents that start out zero .)

A fter the eigenvalues and eigenvectors of the balanced \(m\) atrices have been com puted, C G G B A K transform s the eigenvectors back to \(w\) hat they w ould have been (in perfect arithm etic) if they had notbeen balanced.

Contents of \(A\) and \(B\) on Exit

If any eigenvectors are com puted (either \(J 0 B V L=V\) ' or JO \(B V R=V\) ' or both), then on exit the arrays \(A\) and \(B\) will contain the com plex Schur form [*] of the "balanced" versions of \(A\) and \(B\). If no eigenvectors are com puted, then only the diagonalblocksw illbe conect.
[ \(\star\) ] In otherw ords, uppertriangular form .

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zgehrd -reduce a com plex generalm atrix A to upper H essen-
berg form H by a unitary sim ilarity transform ation

\section*{SYNOPSIS}

```

DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK IN (*)

```


```

DOUBLE COM PLEXA (LDA,*),TAU (*),W ORK\mathbb{N (*)}

```


\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E \operatorname{GEHRD}(\mathbb{N}], \mathbb{L O}, \mathbb{H} I, A,[L D A], T A U,[W O R K \mathbb{N}],[L W O R K \mathbb{N}]\), [ \(\mathbb{N}\) FO ])

COM PLEX (8),D \(\mathbb{I M} E N S \mathbb{O N}(:):: T A U, W\) ORK \(\mathbb{N}\)
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathbb{L} \mathrm{O}, \mathbb{H} \mathrm{I}, \mathrm{LDA}, \mathrm{LW}\) ORK \(\mathbb{N}, \mathbb{N} F O\)
SU BROUTINEGEHRD_64( \(\mathbb{N}], \mathbb{L} O, \mathbb{H} I, A,[\operatorname{LD} A], T A U,[\mathbb{W} O R K \mathbb{N}],[L W O R K \mathbb{N}]\), [ \(\mathbb{N}\) FO ])

COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK \(\mathbb{N}\)
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R(8):: N, \mathbb{L} O, \mathbb{H} I, L D A, L W O R K \mathbb{N}, \mathbb{N} F O\)
\#include < sunperfh>
void zgehrd (intn, int ilo, int ihis, doublecom plex *a, int lda, doublecom plex *tau, int*info);
void zgehrd_64 (long n, long ilo, long ihi, doublecom plex *a, long lda, doublecom plex *tau, long *info);

\section*{PURPOSE}
zgehrd reduces a com plex generalm atrix A to upper \(H\) essenberg form \(H\) by a unitary sim ilarity transform ation: \(Q^{\prime *} A\) * \(\mathrm{Q}=\mathrm{H}\).

\section*{ARGUMENTS}

N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).
ㅍO (input)
It is assum ed that A is already upper triangular in row s and colum ns \(1: \mathbb{I L O - 1}\) and \(\mathbb{H} \mathrm{I}+1 \mathrm{~N} . \mathbb{W}\) and IH I are norm ally setby a previous call to C G EBA L; otherw ise they should be set to 1 and \(N\) respectively. See FurtherD etails.

IH I (input)
See the description of IIO .
A (input/output)
O \(n\) entry, the N -by -N generalm atrix to be reduced.
O n exit, the upper triangle and the firstsubdiagonalofA are overw ritten \(w\) ith the upper H essenberg \(m\) atrix \(H\), and the elem ents below the first subdiagonal, w ith the array TAU, represent the unitary \(m\) atrix \(Q\) as a product of elem entary reflectors. See FurtherD etails.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, \mathbb{N})\).

TAU (output)
The scalar factors of the elem entary reflectors (see Further Details). Elem ents 1:IWO-1 and IH IN -1 of TAU are setto zero.

W ORK \(\mathbb{N}\) (w orkspace)

On exit, if \(\mathbb{N F O}=0, W O R K \mathbb{N}\) (1) retums the optim allW ORK \(\mathbb{N}\).

LW ORK \(\mathbb{N}\) (input)
The length of the array \(W O R K \mathbb{N}\). LW ORK \(\mathbb{N}>=\) \(\max (1, \mathbb{N})\). For optim um perform ance LW ORK \(\mathbb{N}>=\) N *N B , w here N B is the optim alblocksize.

If LW ORK \(\mathbb{N}=-1\), then a workspace query is assum ed; the routine only calculates the optim al size of the \(W\) ORK \(\mathbb{N}\) array, retums this value as the firstentry of the \(W\) ORK \(\mathbb{N}\) array, and no error \(m\) essage related to LW ORK \(\mathbb{N}\) is issued by XERBLA.
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the i-th argum enthad an illegalvahue.

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a productof (hi-ib) ele\(m\) entary reflectors

Each H (i) has the form
\[
\mathrm{H}(\mathrm{i})=\mathrm{I}-\tan * \mathrm{~V}^{*} \mathrm{~V}^{\prime}
\]
\(w\) here tau is a com plex scalar, and \(v\) is a com plex vector w ith \(\mathrm{v}(1: i)=0, \mathrm{v}(i+1)=1\) and \(v(i h i+1 \mathrm{~m})=0 ; \mathrm{v}(i+2\) : ihi \()\) is stored on exitin A (i+2:ihi,i), and tau in TAU (i).

The contents ofA are illustrated by the follow ing exam ple, w ith \(\mathrm{n}=7\), \(\mathrm{il}=2\) and ihi \(=6\) :
```

on entry, on ex斗,

```
(a a a a a a a) (a a h h h h a) ( a a a a a a) ( a h h h \(h a)(a \operatorname{a} a \operatorname{a} a)(\mathrm{h} h \mathrm{~h}\) h h h ) ( a a a a a a) ( v2 h h h h h) ( a a a a a a) ( v2 v3 h h h h ) ( a a a a a a) ( v2 v3 v4 h h h ) (a) ( a)
where a denotes an elem ent of the original \(m\) atrix \(A, h\) denotes a \(m\) odified elem ent of the upper \(H\) essenberg \(m\) atrix \(H\), and videnotes an elem ent of the vector defining \(H\) (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zgelgf-com pute an LQ factorization of a com plex M by -N \(m\) atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGELQFM,N,A,LDA,TAU,W ORK,LDW ORK, INFO)}

```
DOUBLE COM PLEXA (LDA , *),TAU (*), W ORK (*)
\(\mathbb{N}\) TEGER M, N,LDA, LDW ORK, \(\mathbb{N} F O\)
SU BROUTINE ZGELQF_64 M,N,A,LDA,TAU,WORK,LDWORK, \(\mathbb{N} F O\) )
D OUBLE COM PLEX A (LDA , *), TAU (*), W ORK (*)
\(\mathbb{N}\) TEGER* \(8 \mathrm{M}, \mathrm{N}\), LDA, LDW ORK, \(\mathbb{N} F O\)

\section*{F95 INTERFACE}

SU BROUTINE GELQF (M ], \(\mathbb{N}], A,[L D A], T A U, \mathbb{W} O R K],[L D W O R K],[\mathbb{N F O}])\)
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, L D A, L D W\) ORK, \(\mathbb{N} F O\)
SU BROUTINE GELQF_64 ( \(\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{N} O R K],[L D W O R K],[\mathbb{N} F O])\)

COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU ,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER (8) ::M , N,LDA, LDW ORK, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include < sunperfh>
void zgelqf(intm , intn, doublecom plex *a, int lda, doublecom plex *tau, int *info);
void zgelqf_64 (long m, long n, doublecom plex *a, long lda, doublecom plex *tau, long *info);

\section*{PURPOSE}
zgelqfoom putes an LQ factorization of a com plex M -by-N \(m\) atrix \(A: A=L * Q\).

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the M-by-N matrixA. On exit, the ele\(m\) ents on and below the diagonalof the array contain the \(m\)-by-m in \((m, n)\) low er trapezoidalm atrix \(L\) ( \(L\) is lower triangular ifm <= n); the elem ents above the diagonal, w ith the array TAU, represent the unitary \(m\) atrix \(Q\) as a productofelem entary reflectors (see FurtherD etails).

LD A (input)
The leading dim ension of the array A. LDA >= max (1, M).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N F} F=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the aray \(W\) ORK. LDW ORK >= \(m a x(1, M)\). Foroptim um perform ance LDW ORK \(>=M * N B\), where NB is the optim alblocksize.

IfLDW ORK = -1 , then aw orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LDW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\(Q=H(k)^{\prime} \ldots H(2)\) 'H \((1)\) ', where \(k=m\) in \((m, n)\).
Each \(H\) (i) has the form
H (i) \(=I-\tan * V^{*} V^{\prime}\)
\(w\) here tau is a com plex scalar, and \(v\) is a com plex vector w ith \(v(1: i-1)=0\) and \(v(i)=1\); con \(\dot{g}(v(i+1 n))\) is stored on exitin A ( \(i, i+1 \mathrm{n}\) ), and tau in TAU (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgels - solve overdeterm ined or underdeterm ined com plex linear system \(s\) involving an \(M\) boy N matrix \(A\), or its conjugate-transpose, using a QR orLQ factorization of

\section*{SYNOPSIS}
```

SUBROUTINE ZGELS (TRANSA,M ,N,NRHS,A,LDA,B,LDB,W ORK,LDW ORK,
\mathbb{NFO)}
CHARACTER * 1 TRANSA
DOUBLE COM PLEX A (LDA,*),B (LDB,*),W ORK (*)

```

```

SUBROUT\mathbb{NE ZGELS_64 (TRANSA,M,N,NRHS,A,LDA,B,LDB,W ORK,LDW ORK,}
INFO)
CHARACTER * 1 TRANSA
DOUBLE COM PLEX A (LDA,*),B (LDB,*),W ORK (*)
\mathbb{NTEGER*8M,N,NRHS,LDA,LDB,LDW ORK,INFO}

```

\section*{F95 INTERFACE}
```

SU BROUTINE GELS ([TRANSA], $\mathbb{M}], \mathbb{N}], \mathbb{N R H S}], A,[L D A], B,[L D B],[W$ ORK ], LDW ORK, [ $\mathbb{N} F \mathrm{~F}$ ])
CHARACTER (LEN=1) ::TRANSA
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D IM ENSION (:,:) ::A,B
$\mathbb{N}$ TEGER ::M,N,NRHS,LDA,LDB,LDW ORK, $\mathbb{N} F O$
SU BROU T $\mathbb{N} E$ GELS_64 ([TRANSA ], $\mathbb{M}], \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B]$, $\mathbb{W}$ ORK ],LDW ORK, [ $\mathbb{N} F O]$ )

```

CHARACTER (LEN=1) ::TRANSA
COM PLEX (8), D \(\mathbb{M} \operatorname{ENSION(:)::WORK}\)
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : A , B
\(\mathbb{N} \operatorname{TEGER}\) (8) ::M , \(\mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LDA}, \mathrm{LD} \mathrm{B}, \mathrm{LDW} O R K, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zgels (char, int, int, int, doublecom plex*, int, doublecom plex*, int, int*);
void zgels_64 (char, long, long, long, doublecom plex*, long, doublecom plex*, long, long*);

\section*{PURPOSE}
zgels solves overdeterm ined or underdeterm ined complex linear system \(s\) involving an \(M\) boy \(N \mathrm{~N}\) atrix \(A\), or its conjugate-transpose, using a QR orLQ factorization of A. It is assum ed thatA has full rank.

The follow ing options are provided:
1. If TRA \(\mathrm{N} S=\mathrm{N}\) 'and \(\mathrm{m}>=\mathrm{n}\) : find the least squares solution of
an overdeterm ined system, ie., solve the least squares problem
\[
\mathrm{m} \text { inim ize }\|\mathrm{B}-\mathrm{A} * \mathrm{X}\| .
\]
2. If TRA \(N S=N\) 'and \(m<n\) : find the \(m\) inim um norm solution of an underdeterm ined system \(A * X=B\).
3. If TRAN \(S=C\) 'and \(m>=n\) : find them inim um norm solution of
an undeterm ined system \(A * * H * X=B\).
4. IfTRAN \(S=C\) 'and \(m<n\) : find the least squares solution of
an overdeterm ined system, i.e., solve the least squares problem
\[
\mathrm{m} \text { inim ize }\|\mathrm{B}-\mathrm{A} * * \mathrm{H} * \mathrm{X}\| .
\]

Several righthand side vectors b and solution vectors x can be handled in a single call; they are stored as the colum ns of the M -by-NRHS righthand side m atrix B and the N boy-NRHS solution \(m\) atrix X .

\section*{ARGUMENTS}

TRANSA (input)
\(=\mathrm{N}\) : the linearsystem involves A;
\(=C\) ': the linear system involves \(A * * H\).

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

M (input) The num ber of row s of the matrix A. M >=0.

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atriges B and X .NRH S >=0.
A (input/output)
On entry, the M boy-N m atrix A. ifM \(>=N, A\) is overw ritten by details of its \(Q R\) factorization as retumed by CGEQRF; ifM < N, A is overw ritten by details of its LQ factorization as retumed by CGELQF.

LDA (input)
The leading dim ension of the anay A. LDA >= \(\max (1, M)\).

B (input/output)
O n entry, the m atrix B of righthand side vectors,
stored colum nw ise; B is M byy-NRHS ifTRANSA = N',
orN byy-NRHS ifTRANSA = C'. On exit, B is
overw rilten by the solution vectors, stored colum nw ise: ifTRANSA \(=N\) 'and \(m>=n\), row 1 to n ofB contain the least squares solution vectors; the residual.sum ofsquares for the solution in each colum \(n\) is given by the sum of squares of ele\(m\) ents \(N+1\) to \(M\) in thatcolum \(n\); ifTRA \(N S A=N\) 'and \(m<n\), row \(s 1\) to \(N\) ofB contain them inim um norm solution vectors; ifTRANSA \(=C\) 'and \(m>=n\), row \(s\) 1 to M ofB contain them inim um norm solution vectors; ifTRANSA \(=C\) 'and \(m<n\), row 1 to \(M\) of \(B\) contain the least squares solution vectors; the residualsum of squares for the solution in each colum \(n\) is given by the sum of squares of elem ents \(M+1\) to \(N\) in that colum \(n\).

LD B (input)
The leading dim ension of the amay \(\mathrm{B} . \mathrm{LD} \mathrm{B}>=\) M AX ( \(1, \mathrm{M}, N\) ) 。

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (output)
The dim ension of the amay W ORK. LDW ORK >= \(\max (\)
1, \(\mathrm{M} N+\max (\mathrm{M} N, \mathrm{NRHS})\) ). Foroptim alperfor
\(m\) ance, LDW ORK \(>=\max (1, M N+\max (M N, N R H S) * N B\)
). where \(M N=m\) in \(M N\) ) and \(N B\) is the optim um
block size.

IfLD W ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
theW ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LD W ORK is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an ille-
galvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zgelsd -com pute the m inim um -norm solution to a real linear
least squares problem

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGELSD M ,N,NRHS,A,LDA,B,LDB,S,RCOND,RANK,W ORK,}
LW ORK,RW ORK,IN ORK,\mathbb{NFO)}
DOUBLE COM PLEX A (LDA,*),B (LDB,*),W ORK (*)
INTEGERM,N,NRHS,LDA,LDB,RANK,LW ORK,\mathbb{NFO}
INTEGER IN ORK (*)
DOUBLE PRECISION RCOND
DOUBLE PRECISION S (*),RW ORK (*)

```
SU BROUTINE ZGELSD_64M,N,NRHS,A,LDA,B,LDB,S,RCOND,RANK,
    W ORK,LW ORK,RW ORK, IW ORK, \(\mathbb{N} F O\) )
D OUBLE COM PLEX A (LDA , \(\left.{ }^{\star}\right), \mathrm{B}(\mathrm{LD} B, \star), \mathrm{W} O R K(*)\)
\(\mathbb{N} T E G E R * 8 M, N, N R H S, L D A, L D B, R A N K, L W O R K, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER*8 \(\mathbb{I N}\) ORK ( \({ }^{*}\) )
DOUBLE PRECISION RCOND
DOUBLE PRECISION S (*),RW ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GELSD ( \(\mathbb{M}], \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], S, R C O N D\), RANK, [W ORK ], [LW ORK], RW ORK ], [IW ORK], [ \(\mathbb{N F O}\) ])

COM PLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, B
\(\mathbb{N}\) TEGER ::M,N,NRHS,LDA,LDB,RANK,LW ORK, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{I N}\) ORK

REAL (8) ::RCOND
REAL (8),D \(\mathbb{I M}\) ENSION (:) ::S,RW ORK
SU BROUTINE GELSD_64 (M) \(\mathbb{M}, \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], S, R C O N D\), RANK, [W ORK ], [LW ORK], RW ORK ], [IW ORK], [ \(\mathbb{N} F O\) ])

COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B
\(\mathbb{N}\) TEGER (8) ::M,N,NRHS,LDA,LDB,RANK,LW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL (8) :: RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::S,RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zgelsd (intm, intn, intnrhs, doublecom plex *a, int lda, doublecom plex *b, int ldb, double *s, double roond, int *rank, int*info);
void zgelsd_64 (long m , long n, long nrhs, doublecom plex *a, long lda, doublecom plex *b, long ldb, double *s, double rcond, long *rank, long *info);

\section*{PURPOSE}
zgelsd com putes the m inim um -norm solution to a real linear least squares problem :
\(m\) inim ize 2 -norm (|, \(\mathrm{b}-\mathrm{A}\) * \(\mathrm{x} \mid\) )
using the singularvalue decom position (SVD ) ofA.A is an M -by-N m atrix which m ay be rank-deficient.

Several righthand side vectors b and solution vectors x can be handled in a single call; they are stored as the colum ns of the M -by-NRHS righthand side m atrix B and the N by-NRH S solution \(m\) atrix \(X\).

The problem is solved in three steps:
(1) Reduce the coefficientm atrix A to bidiagonal form \(w\) ith H ouseholder tranform ations, reducing the original problem
into a "bidiagonal least squares problem " (BLS)
(2) Solve the BLS using a divide and conquer approach.
(3) A pply back all the H ouseholder tranform ations to solve the original least squares problem .

The effective rank of A is determ ined by treating as zero those singular values which are less than RCOND tim es the largest singularvalue.

The divide and conquer algorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on \(m\) achines w ith a guard digit in add/subtract, or on those binary \(m\) achines \(w\) thout guard digits w hich subtract like the \(C\) ray \(X-M P, C\) ray \(Y-M P, C\) ray \(C-90\), or C ray-2. It could conceivably fail on hexadecim al or decim al machines w thout guard digits, butw e know of none.

\section*{ARGUMENTS}

M (input) The num ber of row sof the \(m\) atrix \(A . M>=0\).
N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRH S (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the \(m\) atrices \(B\) and X.NRHS \(>=0\).

A (input/output)
On entry, the M -by -N m atrix A. On exit, A has been destroyed.

LD A (input)
The leading dim ension of the array A. LDA >= max (1,M).

B (input/output)
On entry, the M-by-NRHS righthand sidem atrix B . On exit, B is overw ritten by the N -by-NRH S solution \(m\) atrix \(X\). If \(m>=n\) and RANK \(=n\), the residual sum -of-squares for the solution in the \(i\)-th colum \(n\) is given by the sum of squares of elem ents \(\mathrm{n}+1 \mathrm{~m}\) in thatcolumn.

LD B (input)
The leading dim ension of the amay \(B\). LD B >= \(\max (1, M, N)\).

S (output)
The singular values ofA in decreasing order. The condition number of \(A\) in the 2 -norm \(=\) \(S(1) / S(m\) in \((m, n))\).

RCOND (input)
RCOND is used to determ ine the effective rank of
A. Singularvalues \(S(i)<=\) RCOND *S ( 1 ) are treated as zero. IfRCOND \(<0, m\) achine precision is used instead.

RANK (output)
The effective rank of A, ie., the num ber of singular values w hich are greater than RCOND *S (1).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the anray \(W\) ORK. LW ORK \(>=1\). The exact \(m\) inim um am ount of \(w\) orkspace needed depends on \(\mathrm{M}, \mathrm{N}\) andNRHS. IfM >= N , LW ORK >= 2*N + \(N * N R H S\). If \(M<N\), LW ORK \(>=2 * M+M * N R H S\). For good penform ance, LW ORK should generally be larger.

IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of theW ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

RW ORK (w orkspace)
IfM \(>=\mathrm{N}\), LRW ORK \(>=8{ }^{*} \mathrm{~N}+2{ }^{\star} \mathrm{N} *\) SM LSIZ \(+8{ }^{*} \mathrm{~N} * \mathrm{NLVL}+\) \(\mathrm{N} * \mathrm{~N}\) RHS. If \(\mathrm{M}<\mathrm{N}, \mathrm{LRW}\) ORK \(>=8 * \mathrm{M}+2 * \mathrm{M} *\) SM LSZ + \(8 * M * N L V L+M * N R H S . S M L S Z\) is retumed by HAENV and is equal to the \(m\) axim um size of the subproblem sat the bottom of the com putation tree (usually about 25), and NLVL= \(\mathbb{N} T\left(L O G \_2(M \mathbb{N}(M, N\right.\)
)/(SM LSTZ+1)) ) + 1
IV ORK (w orkspace)
LIV ORK >= \(3 * M \mathbb{N} M N * N L V L+11 * M \mathbb{N} M N\), where \(M \mathbb{N} M N=M \mathbb{N}(M, N)\).
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthad an illegalvalue.
>0: the algorithm forcom puting the SVD failed
to converge; if \(\mathbb{N} F O=\) i, ioff-diagonalelem ents of an interm ediate bidiagonal form did not converge to zero.

\section*{FURTHER DETAILS}

B ased on contributions by
M ing Gu and Ren-Cang Li, Computer Science D ivision,
U niversity of Califomia atBerkeley, U SA

O sniM arques, LBNLN ER SC , U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgelss -com pute the minim um norm solution to a complex linear least squares problem

\section*{SYNOPSIS}

SU BROUTINE ZGELSS \(M, N, N R H S, A, L D A, B, L D B, S \mathbb{N} G, R C O N D, \mathbb{R A N K}\), W ORK,LDW ORK, WORK2, \(\mathbb{N} F O\) )

DOUBLE COM PLEX A (LDA, \(\left.{ }^{*}\right)\), \(\operatorname{B}(\mathbb{L D} B, \star), W\) ORK (*)
\(\mathbb{N}\) TEGERM,N,NRHS,LDA,LDB, \(\mathbb{R} A N K, L D W O R K, \mathbb{N} F O\)
DOUBLE PRECISION RCOND
DOUBLE PRECISIONSTNG (*),WORK2 (*)

SU BROUTINE ZGELSS_64M,N,NRHS,A,LDA,B,LDB,SING,RCOND, \(\mathbb{R} A N K\), W ORK,LDW ORK, WORK2, \(\mathbb{N} F O\) )

DOUBLE COM PLEX A (LDA, \(\left.{ }^{*}\right)\), \(B(\mathbb{L D} B, \star), W\) ORK (*)
\(\mathbb{N} T E G E R * 8 M, N, N R H S, L D A, L D B, \mathbb{R A N K}, L D W O R K, \mathbb{N} F O\)
DOUBLE PRECISION RCOND
DOUBLE PRECISION S \(\mathbb{N} G(*), W\) ORK 2 (*)

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E \operatorname{GELSS}(\mathbb{M}], \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], S \mathbb{N} G, R C O N D\), \(\mathbb{R} A N K,[\mathbb{W}\) ORK ], [LDW ORK ], [W ORK2], [ \(\mathbb{N} F \mathrm{O}]\) )

COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : : : A , B
\(\mathbb{N} T E G E R:: M, N, N R H S, L D A, L D B, \mathbb{R A N K}, L D W O R K, \mathbb{N} F O\)
REAL (8) ::RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::SING,W ORK2
SU BROUTINE GELSS_64 (M) \(\mathbb{M}], \mathbb{N} R H S], A,[\operatorname{LDA}], B,[L D B], S \mathbb{N} G\),

RCOND, \(\mathbb{R} A N K,[\mathbb{W} O R K],[L D W O R K],[W O R K 2],[\mathbb{N} F O])\)
COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B
\(\mathbb{N}\) TEGER (8) ::M , N,NRHS,LDA,LDB, \(\mathbb{R} A N K, L D W O R K, \mathbb{N} F O\)
REAL (8) :: RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::SNG,W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zgelss (intm, intn, intnrhs, doublecom plex *a, int lda, doublecom plex *b, int ldb, double *sing, double roond, int *irank, int *info);
void zgelss_64 (long m, long n, long nrhs, doublecom plex *a, long lda, doublecom plex *b, long ldb, double *sing, double rcond, long *irank, long *info);

\section*{PURPOSE}
zgelss com putes the \(m\) inim um norm solution to a complex linear least squares problem :
\(M\) inin ize 2 -norm (|b-A *x \()\).
using the singularvalue decom position (SVD) ofA.A is an M -by-N m atrix which m ay be rank-deficient.

Several righthand side vectors b and solution vectors x can be handled in a single call; they are stored as the colum ns of the M -by-NRH S righthand sidem atrix \(B\) and the \(N\) by-NRH S solution \(m\) atrix \(X\).

The effective rank ofA is determ ined by treating as zero those singular values which are less than RCOND tim es the largest singularvalue.

\section*{ARGUMENTS}

M (input) The num ber of row sof the \(m\) atrix \(A . M>=0\).
\(N\) (input) The num ber of collm ns of the m atrix \(A . N>=0\).

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of the \(m\) atrioes \(B\) and \(X\).NRH \(S>=0\).

A (input/output)
On entry, the M -by -N m atrix A. On exit, the first \(m\) in \((m, n)\) row \(s\) of A are overw ritten \(w\) ith its right singularvectors, stored row w ise.

LD A (input)
The leading dim ension of the array A. LDA >= \(m a x(1, M)\).

B (input/output)
On entry, the M -by NRHS righthand side m atrix B .
On exit, B is overw ritten by the \(N\) by \(-\mathrm{NRH} S\) solution \(m\) atrix \(X\). Ifm \(>=n\) and \(\mathbb{R A N K}=n\), the residual sum -of-squares for the solution in the i-th colum \(n\) is given by the sum of squares of ele\(m\) ents \(n+1 m\) in that \(c o l u m n\).

LD B (input)
The leading dim ension of the anay \(B\). LD B >= \(\max (1, M, N)\).

\section*{SNN (output)}

The singularvalues ofA in decreasing order. The condition number of \(A\) in the 2 -norm \(=\) \(S \mathbb{N} G(1) / S \mathbb{N} G(m\) in \((m, n))\).

RCOND (input)
RCOND is used to determ ine the effective rank of
A. Singular values SING (i) <= RCOND *SING (1) are treated as zero. IfRCOND \(<0, \mathrm{~m}\) achine precision is used instead.

RANK (output)
The effective rank of A, i.e., the num ber of singular values which are greater than RCOND *SING (1).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dim ension of the amray \(W\) ORK. LDW ORK \(>=1\), and also: LDW ORK \(>=2 \star_{m}\) in \(\left.(M, N)+m a x M, N, N R H S\right)\) For good perform ance, LDW ORK should generally be larger.

If LDW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK anay, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 2 (w orkspace)
dim ension \(\left(5{ }^{*} \mathrm{~m}\right.\) in \(\left.\mathrm{M}, \mathrm{N}\right)\) )
\(\mathbb{N F O}\) (output)
\(=0\) : successfulexit
< 0: if \(\mathbb{N N}\) FO = -i, the i-th argum ent had an illegalvalue.
> 0: the algorithm forcom puting the SVD failed to converge; if \(\mathbb{N} F O=\) i, ioff-diagonalelem ents of an interm ediate bidiagonal form did not converge to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgelsx -routine is deprecated and has been replaced by routine CGELSY

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGELSX M,N,NRHS,A,LDA,B,LDB,JPIVOT,RCOND,\mathbb{RANK,}}\mathbf{M},\mp@code{L}
W ORK,W ORK2,INFO)
DOUBLE COM PLEX A (LDA,*),B (LDB ,*),W ORK (*)
NNTEGER M,N,NRHS,LDA,LDB,\mathbb{RANK,}\mathbb{N}FO
INTEGER JPIVOT (*)
DOUBLE PRECISION RCOND
DOUBLE PRECISION W ORK 2 (*)
SUBROUT\mathbb{NE ZGELSX_64M,N,NRHS,A,LDA,B,LDB,JPIVOT,RCOND,}
RANK,W ORK,WORK2,\mathbb{NFO)}
DOUBLE COM PLEX A (LDA,*),B (LDB,*),W ORK (*)
\mathbb{NTEGER*8M,N,NRHS,LDA,LDB,\mathbb{RANK,INFO}}\mathbf{N},\mp@code{N}
INTEGER*8 \mathbb{PIVOT (*)}
DOUBLE PRECISION RCOND
DOUBLE PRECISION W ORK2 (*)

```

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E \operatorname{GELSX}(\mathbb{M}], \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], J P I V O T, R C O N D\), \(\mathbb{R A N K},[\mathbb{W}\) ORK], [W ORK2], [ \(\mathbb{N} F O])\)

COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A, B
\(\mathbb{N} T E G E R:: M, N, N R H S, L D A, L D B, \mathbb{R} A N K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION (:) :: \(\mathbb{J} \mathbb{I V O T}\)
REAL (8) :: RCOND

REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ORK2

SU BROUTINE GELSX_64 ( \(\mathbb{M}], \mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], \mathbb{P} \mathbb{V} O T\), RCOND, \(\mathbb{R} A N K,[W O R K],[W O R K 2],[\mathbb{N} F O])\)

COM PLEX (8), D \(\mathbb{M} \operatorname{ENSION}(:):: W\) ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: \(:\) :) : : A , B
\(\mathbb{N}\) TEGER (8) :: M , N , NRHS, LDA, LDB, \(\mathbb{R} A N K, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{M}\) ENSION (:) :: JPIVOT
REAL (8) :: RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zgelsx (intm, intn, intnrhs, doublecom plex *a, int lda, doublecom plex *b, int ldb, int * jipivot, double roond, int *irank, int *info);
void zgelsx_64 (long m, long n, long nrhs, doublecom plex *a, long lda, doublecom plex *b, long ldb, long * jipivot, double rcond, long *irank, long *info);

\section*{PURPOSE}
zgelsx routine is deprecated and has been replaced by routine CGELSY .

CGELSX computes the \(m\) inim um-norm solution to a complex linear least squares problem :
\(m\) inim ize \(\|A * X-B\|\)
using a com plete orthogonal factorization of A. A is an \(M-\) by-N m atrix w hich \(m\) ay be rank-deficient.

Several righthand side vectors b and solution vectors x can be handled in a single call; they are stored as the colum ns of the M -by-NRH S righthand side m atrix B and the N boy-NRHS solution m atrix X .

The routine firstcom putes a QR factorization with colum \(n\) pivoting:
\(A * P=Q *[R 11 R 12]\)
[ 0 R22]
w ith R 11 defined as the largest leading subm atrix whose estim ated condition num ber is less than \(1 \notin\) COND. The order ofR 11, RANK, is the effective rank ofA.

Then, R 22 is considered to be negligible, and R 12 is annihilated by unitary transform ations from the right, arriving at the com plete orthogonal factorization:
\(A * P=Q *[T 110] * Z\)
[ 0 0]
Them inim um norm solution is then
\(\mathrm{X}=\mathrm{P} * \mathrm{Z}\) ' [inv (T11)*Q 1 *B ]
[ 0 ]
where Q 1 consists of the firstRANK colum ns of Q .

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, ie., the num ber of collm ns ofm atrices B and X.NRHS \(>=0\).

A (input/output)
On entry, the \(M\)-by -N matrix A. On exit, A has been overw rilten by details of its complete orthogonal factorization.

LD A (input)
The leading dim ension of the array A. LDA >= max (1,M).

B (input/output)
On entry, the M-by-NRHS righthand sidem atrix B . On exit, the N -by-N RH S solution \(m\) atrix X . Ifm >= n and \(\mathbb{R} A N K=n\), the residual sum-of-squares for the solution in the \(i\)-th colum \(n\) is given by the sum of squares of elem ents \(N+1 \mathrm{M}\) in thatcolum \(n\).

LD B (input)
The leading dim ension of the array \(\mathrm{B} . \operatorname{LD} \mathrm{B}>=\) \(\max (1, M, N)\).

JPIVOT (input/output)
On entry, if \(\mathbb{P} \mathbb{I V O T}\) (i) ne.0, the i-th column of \(A\) is an initial colum \(n\), otherw ise it is a free colum \(n\). Before the \(Q R\) factorization of \(A\), all initial colum ns are perm uted to the leading positions; only the rem aining free colum ns are m oved as a resultof colum \(n\) pivoting during the factorization. On exit, if \(\mathbb{P} \operatorname{IVOT}(i)=k\), then the \(i\)-th colum \(n\) of A *P was the \(k\)-th collm \(n\) of A.

RCOND (input)

RCOND is used to determ ine the effective rank of A, which is defined as the order of the largest leading triangular subm atrix R 11 in the \(Q\) R factorization w ith pivoting ofA , whose estim ated condition num ber \(<1\) RCOND.

RANK (output)
The effective rank of A, i.e., the order of the subm atrix R11. This is the sam e as the orderof the subm atrix T11 in the com plete orthogonal factorization of A.
W ORK (w orkspace)
\((m\) in \(M, N)+\max \left(N, 2 \star_{m}\right.\) in \(\left.\left.(M, N)+N R H S\right)\right)\),

W ORK 2 (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the i-th argum enthad an ille-
galvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zgelsy -com pute the m inim um-norm solution to a complex linear least squares problem

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGELSY M,N,NRHS,A,LDA,B,LDB,JPVT,RCOND,RANK,}
WORK,LW ORK,RW ORK,INFO)

```
DOUBLE COM PLEX A (LDA ,*), B (LDB,*), W ORK (*)
\(\mathbb{N}\) TEGERM,N,NRHS,LDA,LDB,RANK,LW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R \operatorname{JPV}\) ( \({ }^{*}\) )
DOUBLE PRECISION RCOND
DOUBLE PRECISION RW ORK (*)
SU BROUTINE ZGELSY_64M,N,NRHS,A,LDA,B,LDB, UPVT,RCOND,RANK,
    W ORK,LW ORK,RW ORK, \(\mathbb{N} F O\) )
D OUBLE COM PLEX A (LDA , \(\left.{ }^{\star}\right), \mathrm{B}(\mathrm{LD} B, \star), \mathrm{W} O R K(*)\)
\(\mathbb{N}\) TEGER*8M,N,NRHS,LDA,LDB,RANK,LW ORK, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER *8 \(\left.\mathbb{J P V}{ }^{( }{ }^{\star}\right)\)
DOUBLE PRECISION RCOND
DOUBLE PRECISION RW ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GELSY (M ], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], J P V T, R C O N D\), RANK, \(\mathbb{W}\) ORK ], [LW ORK ], [RW ORK ], [ \(\mathbb{N F O}]\) )

COM PLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : ) ::A, B
\(\mathbb{N} T E G E R:: M, N, N R H S, L D A, L D B, R A N K, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION(:)::JPVT}\)

REAL (8) ::RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::RW ORK
SU BROUTINE GELSY_64 (M ], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B], \mathbb{P V} T\), RCOND,RANK, \(\mathbb{W}\) ORK], [LW ORK], RW ORK], [ \(\mathbb{N F O}])\)

COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : : A, B
\(\mathbb{N}\) TEGER (8) :: M , N,NRHS,LDA,LDB,RANK,LW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{J V T}\)
REAL (8) ::RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::RW ORK

\section*{C INTERFACE}
\#include < sunperfh>
void zgelsy (intm, intn, intnrhs, doublecom plex *a, int lda, doublecom plex *b, int ldb, int * pivt, double rcond, int *rank, int *info);
void zgelsy_64 (long m, long n, long nihs, doublecom plex *a, long lda, doublecom plex *b, long ldb, long * jpvt, double rcond, long *rank, long *info);

\section*{PURPOSE}
zgelsy com putes the m inim um-norm solution to a com plex linear least squares problem :
\(m\) inim ize \(\|A * X-B\|\)
using a com plete orthogonal factorization of . A is an \(M\) -by-N \(m\) atrix \(w\) hich \(m\) ay be rank-deficient.

Several righthand side vectors b and solution vectors x can be handled in a single call; they are stored as the colum ns of the M by \(-N\) RH S righthand side \(m\) atrix \(B\) and the \(N\) by-NRHS solution \(m\) atrix \(X\).

The routine firstcom putes \(a Q R\) factorization \(w\) th \(c o l u m n\) pivoting:

A * \(\mathrm{P}=\mathrm{Q}\) * [R11R12]
[ 0 R22]
w ith R 11 defined as the largest leading subm atrix whose estim ated condition num ber is less than 1 RCOND. The order ofR11,RANK, is the effective rank ofA.

Then, R22 is considered to be negligible, and R 12 is annihilated by unitary transform ations from the right, amiving at the com plete orthogonal factorization:
\[
A * P=Q *[T 110] * Z
\]

Them inim um norm solution is then
\(\mathrm{X}=\mathrm{P}\) * Z ' [ inv (T11)*Q 1 *B ]
[ 0 ]
where Q 1 consists of the firstRANK colum ns of \(Q\).
This routine is basically identical to the original xG ELSX except three differences:
- The perm utation ofm atrix \(B\) (the right hand side) is faster and
m ore sim ple.
o The call to the subroutine \(x G E Q P F\) has been substituted by the
the call to the subroutine \(x G E Q P 3\). This subroutine is a B las-3
version of the \(Q R\) factorization \(w\) th colum \(n\) pivoting. \(O M\) atrix \(B\) (the righthand side) is updated \(w\) ith \(B\) las-3.

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of collm ns of the \(m\) atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRH S (input)
The num ber of righthand sides, ie., the num ber of colum ns ofm atrices \(B\) and \(X . N R H S>=0\).

A (input/output)
On entry, the M by -N matrix A. On exit, A has been overw ritten by details of its com plete orthogonal factorization.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, \mathrm{M})\).

B (input/output)
On entry, the M -by - N RH S righthand side m atrix B . On exit, the \(N\)-by N RH S solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray \(B . L D B>=\) \(\max (1, M, N)\).

JPVT (input/output)
On entry, if JPV T (i) ne. 0, the i-th collm n of A is perm uted to the frontofA \(P\), otherw ise colum n i is a free colum \(n\). On exit, if JPV \(T(i)=k\), then
the i-th column of A *P w as the \(k\)-th colum \(n\) of A.

\section*{RCOND (input)}

RCOND is used to determ ine the effective rank of A, which is defined as the order of the largest leading triangular subm atrix R 11 in the \(Q R\) factorization \(w\) th pivoting ofA , whose estim ated condition num ber \(<1\) RCOND.

\section*{RANK (output)}

The effective rank ofA, ie., the order of the subm atrix R11. This is the sam e as the orderof the subm atrix T11 in the complete orthogonal factorization of A.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array W ORK. The unblocked strategy requires that: LW ORK >=MN+MAX (2*M N, \(\mathrm{N}+1, \mathrm{M} \mathrm{N}+\mathrm{NRH} \mathrm{S}\) ) where \(\mathrm{M} \mathrm{N}=\mathrm{m}\) in \(\mathrm{M}, \mathrm{N})\). The block algorithm requires that: LW ORK \(>=\mathrm{MN}+\mathrm{MAX}(2 * \mathrm{M} N\),
 upper bound on the blocksize retumed by IUA EN V for the routines CGEQP3, CTZRZF, CTZRQF, CUNM QR, and CUNMRZ.

If LW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
theW ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

RW ORK (w orkspace)
dim ension \((2 * N)\)
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{FURTHER DETAILS}
```

B ased on contributions by
A .Petitet, C om puterScience D ept, U niv . ofTenn., K nox-
ville, U SA
E.Q uintana-O rti, D epto. de Inform atica,U niversidad Jaim e
I, Spain

```
G. Q uintana-O rti, D epto. de Inform atica, U niversidad Jaim e I, Spain

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgem m -perform one of the \(m\) atrix-m atrix operations \(C:=\) alpha*op (A ) *op (B) + beta*C

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGEMM (TRANSA,TRANSB,M,N,K,ALPHA,A,LDA,B,LDB,}
BETA,C,LDC)
CHARACTER * 1 TRANSA,TRANSB
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (LDA,*),B (LDB,*),C (LD C ,*)
INTEGERM,N,K,LDA,LDB,LDC
SU BROUTINE ZGEM M _64 (TRANSA,TRANSB,M ,N ,K,A LPHA,A ,LDA,B,LD B,
BETA,C,LDC)
CHARACTER * 1 TRANSA,TRANSB
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (LDA,*),B (LDB,*),C (LD C,*)
INTEGER*8M,N,K,LDA,LDB,LDC

```

\section*{F95 INTERFACE}

SU BROUTINE GEM M ([TRANSA], [TRANSB], \(\mathbb{M}], \mathbb{N}], \mathbb{K}], A L P H A, A,[L D A]\), B, [LD B ], BETA , C , [LD C ])

CHARACTER (LEN=1) ::TRANSA,TRANSB
COM PLEX (8) ::ALPHA,BETA
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, B, C
\(\mathbb{N} T E G E R:: M, N, K, L D A, L D B, L D C\)
SU BROUTINE GEMM _64 ([TRANSA], [TRANSB], \(\mathbb{M}], \mathbb{N}],[K], A L P H A, A,[L D A]\), B, [LD B],BETA, C, [LD C])

CHARACTER ( \(L E N=1\) ) : : TRAN SA, TRAN SB
COM PLEX (8) ::ALPHA,BETA
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : ) :: A , B , C
\(\mathbb{N}\) TEGER (8) :: M , N , K , LD A , LD B , LD C

\section*{C INTERFACE}
\#include <sunperfh>
void zgem m (chartransa, chartransb, intm , int \(n\), int \(k\), doublecom plex *alpha, doublecom plex *a, intlda, doublecom plex *b, int ldlo, doublecom plex *beta, doublecom plex \({ }^{\star}\) c, int ldc);
void zgem m _64 (chartransa, chartransb, long m, long n, long k, doublecom plex *alpha, doublecom plex *a, long lda, doublecom plex *b, long ldb, doublecom plex *beta, doublecom plex * c, long ldc);

\section*{PURPOSE}
zgem \(m\) perform s one of the \(m\) atrix-m atrix operations

C : alpha*op (A )*op (B ) + beta*C
where op (X ) is one of
op \((X)=X\) or \(o p(X)=X^{\prime}\) or \(o p(X)=\) con \(\dot{j}(X)\), alpha and beta are scalars, and \(A, B\) and \(C\) arem atrices, \(w\) ith op (A) an \(m\) by \(k m\) atrix, op (B) a \(k\) by \(n m\) atrix and \(C\) an \(m\) by \(n m\) atrix.

\section*{ARGUMENTS}

TRANSA (input)
On entry, TRAN SA specifies the form of op (A ) to be used in the \(m\) atrix \(m\) ultiplication as follow \(s\) :

TRANSA \(=N\) 'or \(h\) ', op \((A)=A\).

TRANSA = T'ort', op (A) =A'.

TRANSA \(=C^{\prime}\) or \(E^{\prime}, o p(A)=\operatorname{conjg}\left(A^{\prime}\right)\).

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

TRANSB (input)
On entry, TRAN SB specifies the form of op (B) to be used in the \(m\) atrix \(m\) ultiplication as follow \(s:\)

TRAN SB \(=\) N'or \(h\) ', op (B) \(=\mathrm{B}\).

TRANSB = T'or \(t^{\prime}\), op (B) \()=B^{\prime}\).

TRANSB = C'or \(C^{\prime}, ~ o p(B)=\) con \(\dot{g}\left(B^{\prime}\right)\).

U nchanged on exit.

TRAN SB is defaulted to \(N\) 'forF95 \(\mathbb{I N T E R F A C E .}\)

M (input)
O n entry, \(M\) specifies the num ber of rows of the \(m\) atrix op (A ) and of the \(m\) atrix C. \(M>=\) 0 . U nchanged on exit.

N (input)
O n entry, \(N\) specifies the num ber of colum ns of the \(m\) atrix op ( B ) and the num ber of colum ns of the \(m\) atrix \(C . N>=0\). U nchanged on exit.

K (input)
On entry, \(K\) specifies the num ber of colum ns of the \(m\) atrix op (A) and the num ber of row s of the \(m\) atrix op (B).K \(>=0\). U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
COM PLEX *16 aray ofD \(\mathbb{M}\) ENSION (LDA, ka ), where ka is \(\mathrm{K} w\) hen TRANSA \(=\mathrm{N}\) 'or h ', and is M otherwise. Before entry w ith TRANSA = N 'or h', the leading \(M\) by \(K\) partof the array A m ust contain the \(m\) atrix \(A\), otherw ise the leading \(K\) by \(M\) part of the array A m ustcontain the m atrix A. U nchanged on exit.

LDA (input)
O n entry, LD A specifies the first dim ension of A as declared in the calling (sub) program .W hen TRANSA \(=N\) 'or \(h\) 'then LD \(A>=\max (1, M)\), otherw ise LD \(A>=\max (1, K)\). U nchanged on exit.

B (input)
COM PLEX *16 anay ofD \(\mathbb{I M} E N S I O N(L D B, k b)\), where
kb is n when TRANSB \(=\mathrm{N}\) 'or h ', and is k otherw ise. Before entry w th TRANSB \(=\mathrm{N}\) ' or h ', the leading k by n partof the array B \(m\) ustcontain the \(m\) atrix \(B\), otherw ise the leading \(n\) by \(k\) partof the aray B mustcontain the \(m\) atrix \(B\). U nchanged on exit.

LD B (input)
O n entry, LD B specifies the firstdimension of B as declared in the calling (sub) program . W hen TRANSB \(=N^{\prime}\) 'or \(h\) 'then LD \(B>=m a x(1, k)\), otherw ise LDB \(>=\max (1, n)\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then C need not.be set on input. U nchanged on exit.

C (input/output)
COM PLEX *16 array ofD \(\mathbb{M}\) ENSION (LD C, n). Before
entry, the leading \(m\) by \(n\) part of the amay \(C\)
\(m\) ustcontain the \(m\) atrix \(C\), exceptw hen beta is zero, in which case \(C\) need notbe seton entry. On exit, the array \(C\) is overw ritten by the \(m\) by \(\mathrm{n} m\) atrix (alpha*op (A ) *op (B ) + beta*C).

LD C (input)
On entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program. LD C \(>=m a x(1, m)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgem v -perform one of the m atrix-vectoroperations \(\mathrm{y}:=\) alpha*A *x + beta* \(y\), ory \(:=\) alpha*A * \(x+\) beta* \(y\), or \(y:=\) alpha*con \(\dot{g}\left(A^{\prime}\right){ }^{\prime} x+\) beta*y

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGEMV (TRANSA,M,N,ALPHA,A,LDA,X,NNCX,BETA,Y,\mathbb{NCY)}}\mathbf{M}\mathrm{ , (NA,}
CHARACTER * 1 TRANSA
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (LDA,*),X (*),Y (*)
\mathbb{NTEGERM,N,LDA,INCX,INCY}
SUBROUT\mathbb{NE ZGEM V_64(TRANSA,M ,N,ALPHA,A,LDA,X,INCX,BETA,Y,}
INCY)
CHARACTER * 1 TRANSA
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (LDA,*),X (*),Y (*)
\mathbb{NTEGER*8M,N,LDA,}\mathbb{NCX,}\mathbb{N}CY

```

\section*{F95 INTERFACE}

SU BROUTINE GEMV ([TRANSA ], \(\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A], X,[\mathbb{N C X}], B E T A\), Y, [ \(\mathbb{N} C Y])\)

CHARACTER (LEN=1) ::TRANSA
COMPLEX (8) ::ALPHA,BETA
COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::X,Y
COM PLEX (8),D \(\mathbb{M}\) ENSIDN (: : : : : A
\(\mathbb{N} T E G E R:: M, N, L D A, \mathbb{N} C X, \mathbb{N} C Y\)
SU BROUTINE GEM V_64 ([TRANSA], \(\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A], X,[\mathbb{N C X}]\),

BETA, Y, [ \(\mathbb{N C Y}])\)

CHARACTER (LEN=1) ::TRANSA
COM PLEX (8) ::ALPHA,BETA
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::X,Y
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{LD} \mathrm{A}, \mathbb{N C X}, \mathbb{N C Y}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zgem v (chartransa, intm, intn, doublecom plex *alpha, doublecom plex *a, intlda, doublecom plex \({ }^{*}\) x, int incx, doublecom plex *beta, doublecom plex *y, int incy);
void zgem v_64 (chartransa, long m, long n, doublecom plex *alpha, doublecom plex *a, long lda, doublecom plex
\({ }^{*}\) x, long incx, doublecom plex *beta, doublecom plex
*y, long incy);

\section*{PURPOSE}
zgem v perform sone of the \(m\) atrix-vector operations \(y:=\) alpha*A *x + beta* \(y\), ory \(:=\) alpha*A \({ }^{*} x+\) beta* \(y\), or \(y:=\) alpha*con \(\dot{9}\left(A^{\prime}\right)^{\star} x+b e t a * y ~ w h e r e ~ a l p h a ~ a n d ~ b e t a ~ a r e ~\) scalars, \(x\) and \(y\) are vectors and \(A\) is an \(m\) by \(n m\) atrix.

\section*{ARGUMENTS}

TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} y:=\) alpha*A *x + beta* \(y\).

TRANSA \(=T^{\prime}\) or \(t^{\prime} y:=\) alpha*A * \(\mathrm{x}+\) beta* \(^{\mathrm{y}}\).

TRANSA = C'or \(\mathrm{C}^{\prime} \mathrm{y}:=\) alpha*con \(\dot{g}\left(\mathrm{~A}^{\prime}\right){ }^{*} \mathrm{x}+\) beta*y.

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

M (input)
O n entry, M specifies the num ber of row s of the \(m\) atrix A. M >=0. U nchanged on exit.

N (input)
O n entry, N specifies the num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\). Unchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
Before entry, the leading \(m\) by \(n\) part of the anay A must contain the matrix of coefficients. U nchanged on exit.
LDA (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A >= \(\max (1, m)\). U nchanged on exit.

X (input)
\((1+(\mathrm{n}-1) \star \operatorname{abs}(\mathbb{N} C X))\) when TRANSA \(=\mathrm{N}\) 'or \(h^{\prime}\) and at least \((1+(m-1) * a b s(\mathbb{N} C X))\) otherw ise. Before entry, the increm ented array \(X\) \(m\) ustcontain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X\) m ustnotbe zero. U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then \(Y\) need notbe set on input. U nchanged on exit.

Y (input/output)
\((1+(m-1) \star \operatorname{abs}(\mathbb{N} C Y))\) when TRANSA \(=N\) 'or \(h^{\prime}\) and at least \((1+(n-1) \star a b s(\mathbb{N} C Y))\)
otherw ise. Before entry w ith BETA non-zero, the increm ented array \(Y\) m ust contain the vectory. On exit, \(Y\) is overw rilten by the updated vectory.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y . \mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zgeqlf-com pute a Q L factorization of a com plex M-by N \(m\) atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGEQLFM,N,A,LDA,TAU,W ORK,LDW ORK, INFO)}

```
DOUBLE COM PLEXA (LDA , *),TAU (*), W ORK (*)
\(\mathbb{N}\) TEGER M, N,LDA, LDW ORK, \(\mathbb{N} F O\)
SU BROUTINE ZGEQLF_64 M,N,A,LDA,TAU,W ORK,LDW ORK, \(\mathbb{N} F O\) )
D OUBLE COM PLEX A (LDA , *), TAU (*), W ORK (*)
\(\mathbb{N} T E G E R * 8 \mathrm{M}, \mathrm{N}\), LDA, LDW ORK, \(\mathbb{N} F \mathrm{O}\)

\section*{F95 INTERFACE}

SU BROUTINE GEQLF (M ], \(\mathbb{N}], A,[L D A], T A U, \mathbb{W} O R K],[L D W O R K],[\mathbb{N F O}])\)
COM PLEX (8),D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, L D A, L D W\) ORK, \(\mathbb{N} F O\)
SU BROUTINE GEQLF_64 ( \(\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N} F O])\)

COM PLEX (8),D \(\mathbb{I}\) ENSION (:) ::TAU ,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER (8) ::M , N,LDA, LDW ORK, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include < sunperfh>
void zgeqlf(intm , intn, doublecom plex *a, int lda, doublecom plex *tau, int *info);
void zgeqlif_64 (long m, long n, doublecom plex *a, long lda, doublecom plex *tau, long *info);

\section*{PURPOSE}
zgeqlf com putes a Q L factorization of a com plex M -by-N \(m\) atrix \(A: A=Q * L\).

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix A. M \(>=0\).
N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the M -by -N m atrix A. On exit, if \(m>=\) \(n\), the lower triangle of the subanay A ( \(m\) \(\mathrm{n}+1 \mathrm{~m}, 1 \mathrm{n}\) ) contains the N by N low er triangular \(m\) atrix \(L\); ifm <= \(n\), the elem ents on and below the ( \(n-m\) )-th superdiagonal contain the M -by -N lower trapezoidalm atrix \(L\); the rem aining elem ents, w th the anray \(T A U\), represent the unitary \(m\) atrix \(Q\) as a product of elem entary reflectors (see Further D etails).

LD A (input)
The leading dim ension of the array A. LDA >= \(m a x(1, M)\).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N F} F=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the array W ORK. LDW ORK >= \(m\) ax \((1, N)\). Foroptim um perform ance LD \(W\) ORK \(>=N * N B\), where NB is the optim alblocksize.

IfLDW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N F O}\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{I N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(k) \ldots H(2) H(1), \text { where } k=m \text { in }(m, n) .
\]

Each \(H\) (i) has the form

H (i) \(=I-\tan * V^{*} V^{\prime}\)
\(w\) here tau is a com plex scalar, and \(v\) is a com plex vector w ith \(v(m-k+i+1 \mathrm{~m})=0\) and \(v(m-k+i)=1 ; v(1 m-k+i-1)\) is stored on exitin A ( \(1 \mathrm{~m}-\mathrm{k}+\mathrm{i}-1, \mathrm{n}-\mathrm{k}+\mathrm{i}\) ), and tau in TAU (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zgeqp3 - com pute a Q \(R\) factorization \(w\) ith colum \(n\) pivoting of a matrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGEQP3M,N,A,LDA,JPVT,TAU,W ORK,LW ORK,RW ORK,\mathbb{NFO)}}\mathbf{M}\mathrm{ (NA,}

```
DOUBLE COM PLEXA (LDA , *),TAU (*), W ORK (*)
\(\mathbb{N}\) TEGER M,N,LDA,LWORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R \operatorname{JPV}\) ( \({ }^{( }\))
D OUBLE PRECISION RW ORK (*)
SU BROUTINE ZGEQP3_64M,N,A,LDA, JPVT,TAU,W ORK,LW ORK,RW ORK,
        \(\mathbb{N} F O\) )
DOUBLE COM PLEXA (LDA , *), TAU (*), WORK (*)
\(\mathbb{N}\) TEGER* \(8 \mathrm{M}, \mathrm{N}\), LD A , LW ORK, \(\mathbb{N} F \mathrm{O}\)
\(\mathbb{N}\) TEGER*8 \(\mathbb{J P V}\) ( \({ }^{*}\) )
DOUBLE PRECISION RW ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE GEQP3 (M ], \(\mathbb{N}], A,[L D A], J P V T, T A U,[W O R K],[L W ~ O R K]\), [RW ORK], [ \(\mathbb{N} F O]\) )

COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU ,W ORK
COM PLEX (8),D IM ENSION (:,:) ::A
\(\mathbb{N}\) TEGER ::M,N,LDA,LW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION (:) :: JPVT
REAL (8),D \(\mathbb{M}\) ENSION (:) ::RW ORK

SU BROUTINE GEQP3_64 (M ], \(\mathbb{N}], A,[L D A], J P V T, T A U,[W O R K],[L W\) ORK ],
\(\mathbb{R W}\) ORK], [ \(\mathbb{N} F O]\) )
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R(8):: M, N, L D A, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{J V T}\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zgeqp3 (intm , intn, doublecom plex *a, int lda, int *jpvt, doublecom plex *tau, int*info);
void zgeqp3_64 (long m, long n, doublecom plex *a, long lda, long * jpvt, doublecom plex *tau, long *info);

\section*{PURPOSE}
zgeqp3 com putes a \(Q R\) factorization \(w\) ith colum \(n\) pivoting of a \(m\) atrix \(A: A * P=Q * R\) using Level3 BLA \(S\).

\section*{ARGUMENTS}

M (input) The num ber of row sof the \(m\) atrix \(A . M>=0\).
N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input/output)
On entry, the \(M\)-by-N m atrix A. On exit, the upper triangle of the array contains the \(m\) in \((M, N)\)-by \(-N\) upper trapezoidalm atrix \(R\); the elem ents below the diagonal, together \(w\) th the array TA \(U\), represent the unitary \(m\) atrix \(Q\) as a productofm in \(M, N\) ) ele\(m\) entary reflectors.

LDA (input)
The leading dim ension of the array A. LDA >= \(m a x(1, M)\).

JPVT (input/output)
On entry, if \(\mathbb{P V} T(J)\) ne. 0 , the \(J\) th colum \(n\) of \(A\) is perm uted to the frontof \(A\) * (a leading colum \(n\) );
if JPVT \((J)=0\), the \(J\) th column of \(A\) is a free column. On exit, if \(\mathbb{J P V}(\mathcal{J})=\mathrm{K}\), then the J th colum \(n\) of A *P was the the \(K\)-th colum \(n\) of A.

TAU (output)
The scalar factors of the elem entary reflectors.
W ORK (w orkspace)
Onexit, if \(\mathbb{N} F O=0, \mathrm{~W}\) ORK (1) retums the optim al
LW ORK.

LW ORK (input)
The dim ension of the aray W ORK. LW ORK >= N+1. For optim alperform ance LW ORK >=( \(\mathrm{N}+1\) ) N B , where N B is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

RW ORK (w orkspace)
dim ension ( \(2 \star \mathrm{~N}\) )
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit.
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an illegalvalue.

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(1) H(2) \ldots H(k), \text { where } k=m \text { in }(m, n) .
\]

Each \(H\) (i) has the form

H (i) \(=I-\tan * V^{*} V^{\prime}\)
w here tau is a real/com plex scalar, and \(v\) is a real/com plex vectorw ith \(v(1: i-1)=0\) and \(v(i)=1 ; v(i+1 \mathrm{~m})\) is stored on exit in A (i+1 \(m, i)\), and tau in TAU (i).

B ased on contributions by
G . Q uintana-O rti, D epto. de Inform atica, U niversidad Jaim e I, Spain
X . Sun, C om puter Science D ept., D uke U niversity, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zgeqpf-routine is deprecated and has been replaced by routine C GEQP3

\section*{SYNOPSIS}

SUBROUTINE ZGEQPF M,N,A,LDA, \(\mathbb{P} \mathbb{I V O T}, T A U, W\) ORK,W ORK 2, \(\mathbb{N} F O\) )
D OUBLE COM PLEX A (LDA , *), TAU (*), W ORK (*)
\(\mathbb{N}\) TEGERM,N,LDA, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{J} \mathbb{V} O T(*)\)
DOUBLE PRECISION W ORK 2 (*)
SUBROUTINE ZGEQPF_64 M,N,A,LDA, JPIVOT,TAU,WORK,WORK2, \(\mathbb{N} F O\) )
D OUBLE COM PLEX A (LDA ,*), TAU (*), W ORK (*)
\(\mathbb{N}\) TEGER*8M,N,LDA, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{J P I V O T}\) ( \({ }^{*}\) )
DOUBLE PRECISION W ORK 2 (*)

\section*{F95 INTERFACE}

SU BROUTINE GEQPF ( \(\mathbb{M}], \mathbb{N}], A,[L D A], J P I V O T, T A U,[\mathbb{W} O R K],[W O R K 2]\), [ \(\mathbb{N}\) FO ])

COM PLEX (8),D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER ::M,N,LDA, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: JPIVOT
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ORK2
SU BROUTINE GEQPF_64 (M) \(\mathbb{M}, \mathbb{N}], A,[L D A], \mathbb{U} \mathbb{I} O T, T A U,[\mathbb{W} O R K],[\mathbb{W}\) ORK 2], [ \(\mathbb{N}\) FO ])

\section*{C INTERFACE}
\#include <sunperfh>
void zgeqpf(intm, intn, doublecom plex *a, int lda, int
* jpivot, doublecom plex *tau, int *info);
void zgeqpf_64 (long m, long n, doublecom plex *a, long lda, long * jíivot, doublecom plex *tau, long *info);

\section*{PURPOSE}
zgeqpf routine is deprecated and has been replaced by routine CGEQP3.

CGEQPF com putes a \(Q\) R factorization \(w\) ith colum \(n\) pivoting of a com plex \(M\) by \(-N\) m atrix \(A: A * P=Q * R\).

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix A. \(\mathrm{M}>=0\).
N (input) The num ber of colum ns of the \(m\) atrix \(\mathrm{A} \cdot \mathrm{N}>=0\)

A (input/output)
On entry, the M -by-N m atrix A. On exit, the upper triangle of the array contains the \(m\) in \((M, N)\)-by \(-N\) upper triangularm atrix \(R\); the elem ents below the diagonal, together w ith the array TAU, represent the unitary \(m\) atrix \(Q\) as a productofm in \((m, n)\) ele\(m\) entary reflectors.

LDA (input)
The leading dim ension of the array A. LD A >= \(m a x(1, M)\).

JPIVOT (input/output)
On entry, if \(\mathbb{P P I V O T}\) (i) ne. 0 , the \(i\)-th colum n of \(A\) is perm uted to the front of A *P (a leading colum n); if \(\mathbb{P P I V O T}\) (i) = 0, the i-th colum \(n\) of A is a free column. On exit, if \(\operatorname{PPIVOT}(i)=k\), then
the i-th colum \(n\) of \(A * P\) was the \(k\)-th colum \(n\) of \(A\).

TAU (output)
The scalar factors of the elem entary reflectors.

W ORK (w orkspace)
dim ension \(\mathbb{N}\) )

W ORK 2 (w orkspace)
dim ension ( \(2 \star \mathrm{~N}\) )
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N F O}=-\mathrm{i}\), the i -th argum ent had an illegal value

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\(Q=H(1) H(2) \ldots H(n)\)
Each \(H\) (i) has the form
\(\mathrm{H}=\mathrm{I}-\tan { }^{*} \mathrm{~V}^{*} \mathrm{~V}^{\prime}\)
\(w\) here tau is a com plex scalar, and \(v\) is a com plex vector with \(v(1: i-1)=0\) and \(v(i)=1 ; v(i+1 \mathrm{~m})\) is stored on exit in \(A(i+1 m, i)\).

The \(m\) atrix \(P\) is represented in jpvtas follow s: If put \((\mathcal{j})=i\)
then the th colum n ofP is the ith canonicalunitvector.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zgeqrf-com pute \(a Q R\) factorization of a com plex \(M-b y-N\)
\(m\) atrix \(A\)

\section*{SYNOPSIS}

DOUBLE COM PLEXA (LDA , *),TAU (*), W ORK (*)
\(\mathbb{N}\) TEGER M, N,LDA,LDW ORK, \(\mathbb{N} F O\)
SU BROUTINE ZGEQRF_64 \(M, N, A, L D A, T A U, W O R K, L D W O R K, \mathbb{N} F O)\)
D OUBLE COM PLEX A (LDA , *), TAU (*), W ORK (*)
\(\mathbb{N}\) TEGER* \(8 \mathrm{M}, \mathrm{N}\), LDA, LDW ORK, \(\mathbb{N} F O\)

\section*{F95 INTERFACE}
\(\operatorname{SUBROUT\mathbb {NE}GEQRF}(\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N F O}])\)
COM PLEX (8),D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, L D A, L D W\) ORK, \(\mathbb{N} F O\)
SUBROUTINE GEQRF_64 (M) \(\mathbb{N} \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N} F O])\)

COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU ,W ORK
COM PLEX (8), D \(\mathbb{I M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER (8) ::M , N,LDA,LDW ORK, \(\mathbb{N}\) FO

\section*{C INTERFACE}
\#include < sunperfh>
void zgeqrf(intm, intn, doublecom plex *a, int lda, doublecom plex *tau, int *info);
void zgeqnf_64 (long m, long n, doublecom plex *a, long lda, doublecom plex *tau, long *info);

\section*{PURPOSE}
zgeqnf com putes a QR factorization of a complex M -by-N \(m \operatorname{atrix} A: A=Q * R\).

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of colum ns of the matrix A. \(\mathrm{N}>=0\).

A (input/output)
On entry, the M -by -N m atrix A. On exit, the ele\(m\) ents on and above the diagonal of the array contain the \(m\) in \(M, N\) )-by \(-N\) uppertrapezoidalm atrix \(R\) \((R\) is upper triangular if \(m>=n\) ); the elem ents below the diagonal, w ith the amay TAU, represent the unitary \(m\) atrix \(Q\) as a productofm in \((m, n)\) ele\(m\) entary reflectors (see FurtherD etails).

LDA (input)
The leading dim ension of the array A. LD A >= max (1, M).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the array W ORK. LDW ORK >= \(\max (1, N)\). Foroptim um perform ance LDW ORK >=N N NB, where NB is the optim alblocksize.

If LD W ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim al size of
theW ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LDW ORK is issued by XERBLA .
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\(Q=H(1) H(2) \ldots H(k)\), where \(k=m\) in \((m, n)\).
Each \(H\) (i) has the form
H (i) \(=I-\tan * V^{*} V^{\prime}\)
\(w\) here tau is a com plex scalar, and \(v\) is a com plex vector w ith \(\mathrm{v}(1: i-1)=0\) and \(v(i)=1 ; \mathrm{v}(i+1 \mathrm{~m})\) is stored on exit in A (i+1 \(\mathrm{m}, \mathrm{i}\) ), and tau in TAU (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgerc-perform the rank 1 operation A := alpha*x*conjg ( \(\left.y^{\prime}\right)+A\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGERC M,N,ALPHA,X,INCX,Y, INCY,A,LDA)}
D OUBLE COM PLEX ALPHA
DOUBLE COM PLEXX (*),Y(*),A (LDA,*)

```


```

DOUBLE COM PLEX ALPHA
DOUBLE COM PLEXX (*),Y (*),A (LDA,*)
\mathbb{NTEGER*8M,N,INCX,INCY,LDA}

```

\section*{F95 INTERFACE}

SUBROUTINE GERC ( \(\mathbb{M}], \mathbb{N}], A L P H A, X,[\mathbb{N C X}], Y,[\mathbb{N C Y}], A,[\operatorname{LDA}])\)
COM PLEX (8) ::ALPHA
COMPLEX (8),D IM ENSION (:) ::X,Y
COM PLEX (8),D \(\mathbb{I}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, \mathbb{N C X}, \mathbb{N C Y}, L D A\)
SUBROUTINE GERC_64 ( \(\mathbb{M}], \mathbb{N}], A L P H A, X,[\mathbb{N C X}], Y,[\mathbb{N C Y}], A,[L D A])\)
COM PLEX (8) ::ALPHA
COMPLEX (8), D IM ENSION (:) ::X,Y
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R(8):: M, N, \mathbb{N C X}, \mathbb{N} C Y, L D A\)

\section*{C INTERFACE}
\#include <sunperfh>
void zgerc (intm, intn, doublecom plex *alpha, doublecom plex
*x, int incx, doublecom plex *y, int incy, doublecom plex *a, int lda);
void zgerc_64 (long m, long n, doublecom plex *alpha, doublecom plex *x, long incx, doublecom plex *y, long incy, doublecom plex *a, long lda);

\section*{PURPOSE}
zgerc perform sthe rank 1 operation \(A:=a l p h a * x^{\star} c o n g\left(y^{\prime}\right)\)
\(+A\) where alpha is a scalar, \(x\) is an \(m\) elem entvector, \(y\) is an \(n\) elem entvector and \(A\) is an \(m\) by \(n m\) atrix.

\section*{ARGUMENTS}

M (input)
O n entry, \(M\) specifies the num ber of row s of the \(m\) atrix A. M >= 0 . U nchanged on exit.

N (input)
O n entry, \(N\) specifies the num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\). U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
\((1+(m-1) * a b s(\mathbb{N C X}))\). Before entry, the increm ented array \(X\) m ust contain the \(m\) elem ent vectorx. U nchanged on exit.
\(\mathbb{N} C X\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X \mathrm{~m}\) ustnotbe zero. U nchanged on exit.
\(Y\) (input)
\((1+(n-1) \star a b s(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) m ust contain the \(n\) elem ent vectory. U nchanged on exit.
\(\mathbb{N C C Y}\) (input)

O n entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of Y. \(\mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

A (input/output)
Before entry, the leading \(m\) by \(n\) part of the array
A must contain the matrix ofcoefficients. On exit, A is overw rilten by the updated \(m\) atrix.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A \(>=\) \(\max (1, m)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgerfs - im prove the com puted solution to a system of linear equations and provides errorbounds and backw ard emoresti\(m\) ates for the solution

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGERFS (TRANSA,N,NRHS,A,LDA,AF,LDAF, PPIVOT,B,LDB,}
X,LDX,FERR,BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 TRANSA
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
\mathbb{N TEGER N,NRHS,LDA,LDAF,LDB,LDX, INFO}
\mathbb{NTEGER IPIVOT (*)}
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)
SU BROUTINE ZGERFS_64 (TRANSA,N,NRHS,A,LDA,AF,LDAF,\mathbb{P IV OT,B,}
LD B,X,LDX,FERR,BERR,W ORK,W ORK 2, INFO)

```
CHARACTER * 1 TRANSA
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
\(\mathbb{I N}\) TEGER*8 N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER*8 \(\mathbb{P} \mathbb{I V O T}\) (*)
D OUBLE PREC ISION FERR (*), BERR (*), W ORK 2 (*)

\section*{F95 INTERFACE}

SU BROUTINE GERFS ([TRANSA], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F], \mathbb{P} \mathbb{I} O T\), B, [LDB], X, [LDX],FERR,BERR, [W ORK], [W ORK2], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::TRANSA
COMPLEX (8),D IM ENSION (:) ::W ORK

COM PLEX (8), D \(\mathbb{I M} E N S \mathbb{O N}(:,:\) : : \(: A, A F, B, X\)
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{I} O T\)
REAL (8), D \(\mathbb{I}\) ENSION (:) :: FERR,BERR,W ORK2

SU BROUTINE GERFS_64 ([TRANSA ], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F]\), \(\mathbb{P} \mathbb{I} O T, B,[L D B], X,[L D X], F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::TRAN SA
COM PLEX (8), D \(\mathbb{M} E N S I O N(:):: W O R K\)
COM PLEX (8), D \(\mathbb{I}\) ENSION (: : : : : A, AF, B, X
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{I} \mathrm{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8), D \(\mathbb{M}\) ENSION (:) :: FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void zgerfs (char transa, intn, intnrhs, doublecom plex *a, int lda, doublecom plex *af, int ldaf, int *ipivot, doublecom plex *b, int ldlo, doublecom plex *x, int ldx, double * ferr, double *berr, int *info);
void zgerfs_64 (chartransa, long n, long nrhs, doublecom plex
*a, long lda, doublecom plex *af, long ldaf, long
*ípivot, doublecom plex *b, long ldb, doublecom plex
*x, long ldx, double *ferr, double *berr, long
*info);

\section*{PURPOSE}
zgerfs im proves the com puted solution to a system of linear equations and provides errorbounds and backw ard erroresti\(m\) ates for the solution.

\section*{ARGUMENTS}

TRANSA (input)
Specifies the form of the system ofequations:
\(=N^{\prime}: A * X=B \quad\) (Notranspose)
\(=T\) ': \(A * * T * X=B \quad\) (Transpose)
\(=C\) ': A**H * X = B (C onjugate transpose)

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)

The num ber of righthand sides, i.e., the num ber of collm ns of the \(m\) atrioes \(B\) and \(X\). NRH \(S>=0\).

A (input) The originaln by N m atrix A .
LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

AF (input)
The factors \(L\) and \(U\) from the factorization \(\mathrm{A}=\) \(\mathrm{P} * \mathrm{~L} * \mathrm{U}\) as com puted by CGETRF.
LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, \mathbb{N})\).

PIVOT (input)
The pinot indices from CGETRF ; for \(1<=i<=N\), row \(i\) of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P}\) IV OT (i).
\(B\) (input) The righthand side \(m\) atrix \(B\).
LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, \mathbb{N})\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by CGETRS. On exit, the im proved solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the array X. LD X >= \(\max (1, \mathbb{N})\).

FERR (output)
The estim ated forw ard enrorbound for each solution vector \(X(\mathcal{)}\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution comesponding to \(X(\mathcal{O}), \operatorname{FERR}(\mathcal{O})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{D})\)-XTRU \(E\) ) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

BERR (output)
The com ponentw ise relative backw ard emor of each solution vector X (j) (ie., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace) dim ension ( \(2 * N\) )

W ORK 2 (w orkspace) dim ension \(\mathbb{N}\) )
INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zgergf-com pute an \(R Q\) factorization of a com plex \(M-b y-N\) m atrix A

\section*{SYNOPSIS}

DOUBLE COM PLEXA (LDA , *),TAU (*), W ORK (*)
\(\mathbb{N}\) TEGER M, N,LDA,LDW ORK, \(\mathbb{N} F O\)
SU BROUTINE ZGERQF_64 M,N,A,LDA,TAU,W ORK,LDW ORK, \(\mathbb{N} F O\) )
D OUBLE COM PLEX A (LDA , *), TAU (*), W ORK (*)
\(\mathbb{N}\) TEGER* \(8 \mathrm{M}, \mathrm{N}\), LDA, LDW ORK, \(\mathbb{N} F O\)

\section*{F95 INTERFACE}
\(\operatorname{SUBROUT\mathbb {NE}GERQF}(\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N F O}])\)
COM PLEX (8),D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, L D A, L D W\) ORK, \(\mathbb{N} F O\)
SU BROUTINE GERQF_64 (M) \(\mathbb{N} \mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L D W O R K],[\mathbb{N} F O])\)

COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU ,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER (8) ::M , N,LDA,LDW ORK, \(\mathbb{N}\) FO

\section*{C INTERFACE}
\#include < sunperfh>
void zgerqf(intm, intn, doublecom plex *a, int lda, doub-
lecom plex *tau, int *info);
void zgerqf_64 (long m, long n, doublecom plex *a, long lda, doublecom plex *tau, long *info);

\section*{PURPOSE}
zgergf com putes an RQ factorization of a com plex \(M\)-by \(-\mathbb{N}\) \(m\) atrix \(A: A=R * Q\).

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
Onentry, the \(M\) by N matrix A. On exit, if \(\mathrm{m}<=\) \(n\), the upper triangle of the subarray \(A(1 \mathrm{~m}, \mathrm{n}\) \(\mathrm{m}+1 \mathrm{n}\) ) contains the M boy -M upper triangularm atrix \(R\); if \(m>=n\), the elem ents on and above the \(m\) n )-th subdiagonalcontain the M -by -N upper trapezoidal \(m\) atrix \(R\); the rem aining elem ents, \(w\) ith the amay TA \(U\), represent the unitary \(m\) atrix \(Q\) as a product of \(m\) in \((m, n\) ) elem entary reflectors (see FurtherD etails).

LDA (input)
The leading dim ension of the amay A. LDA >= max (1,M).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N F} F=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dim ension of the array \(W\) ORK. LDW ORK >= \(m \operatorname{ax}(1, M)\). Foroptim um perform ance LDW ORK \(>=M * N B\), where NB is the optim alblocksize.

IfLDW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(1) \text { 'H (2)' } \ldots H(k) \text { ', where } k=m \text { in }(m, n) \text {. }
\]

Each \(H\) (i) has the form

H (i) \(=I-\tan * V^{*} V^{\prime}\)
\(w\) here tau is a com plex scalar, and \(v\) is a com plex vector with \(v(n-k+i+1 n)=0\) and \(v(n-k+i)=1\); con \(\operatorname{jg}(v(1 n-k+i-1))\) is stored on exitin A \((m-k+i, 1 m-k+i-1)\), and tau in TAU (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgenu -penform the rank 1 operation A := alpha*x*y'+A

\section*{SYNOPSIS}

```

D OUBLE COM PLEX ALPHA
DOUBLE COM PLEXX (*),Y (*),A (LDA,*)
\mathbb{NTEGERM,N,INCX,INCY,LDA}

```

```

DOUBLE COM PLEX ALPHA
DOUBLE COM PLEXX (*),Y (*),A (LDA,*)

```

F95 INTERFACE
    SUBROUTINE GER (M ], \(\mathbb{N}], A \operatorname{LPHA}, \mathrm{X},[\mathbb{N} C X], Y,[\mathbb{N C Y}], A,[L D A])\)
    COM PLEX (8) ::A LPHA
    COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::X,Y
    COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N}\) TEGER ::M,N, \(\mathbb{N C X}, \mathbb{N C Y}, L D A\)
    SUBROUTINE GER_64 (M ], \(\mathbb{N}], A L P H A, X,[\mathbb{N C X}], Y,[\mathbb{N} C Y], A,[L D A])\)
    COM PLEX (8) ::A LPHA
    COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::X,Y
    COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathbb{I N C X}, \mathbb{N} C Y, L D A\)
\#include <sunperfh>
void zgenu (intm, intn, doublecom plex *alpha, doublecom plex
\({ }^{*} x\), int incx, doublecom plex *y, int incy, doublecom plex *a, int lda);
void zgenu_64 (long m, long n, doublecom plex *alpha, doublecom plex *x, long incx, doublecom plex *y, long incy, doublecom plex *a, long lda);

\section*{PURPOSE}
zgenu perform sthe rank 1 operation \(A:=a l p h{ }^{\star} x^{*} y^{\prime}+A\) \(w\) here aloha is a scalar, \(x\) is an \(m\) elem entvector, \(y\) is an \(n\) elem entvectorand \(A\) is an \(m\) by \(n m\) atrix.

\section*{ARGUMENTS}

M (input)
O n entry, M specifies the num ber of row s of the \(m\) atrix A. M >=0. U nchanged on exit.

N (input)
O n entry, \(N\) specifies the num ber of colum ns of the \(m\) atrix \(A . N>=0\). Unchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
\((1+(m-1) \star \operatorname{abs}(\mathbb{N C X}))\). Before entry, the increm ented amay \(X\) m ust contain the \(m\) elem ent vectorx. U nchanged on exit.
\(\mathbb{N} C X\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X \mathrm{~m}\) ustnotbe zero. U nchanged on exit.

Y (input)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y \mathrm{~m}\) ust contain the n elem ent vectory. U nchanged on exit.
\(\mathbb{N} C Y\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the
elem ents of \(Y . \mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

A (input/output)
Before entry, the leading \(m\) by \(n\) part of the array
A must contain the matrix ofcoefficients. O n exit, A is overw rilten by the updated \(m\) atrix.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A \(>=\) \(\max (1, m)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zgesdd - com pute the singular value decom position (SV D ) of a complex M -by-N matrix A, optionally com puting the left and/orrightsingularvectors, by using divide-and-conquer \(m\) ethod

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZGESDD (OOBZ,M ,N,A,LDA,S,U,LDU,VT,LDVT,W ORK,}
LW ORK,RW ORK,IN ORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ
DOUBLE COM PLEX A (LDA,*),U (LDU,*),VT (LDVT,*),W ORK (*)
INTEGERM,N,LDA,LDU,LDVT,LW ORK,\mathbb{NFO}
INTEGER IN ORK (*)
DOUBLE PRECISION S (*),RW ORK (*)
SU BROUTINE ZGESDD_64(JOBZ,M,N,A,LDA,S,U,LDU,VT,LDVT,W ORK,
LW ORK,RW ORK,\mathbb{N ORK,\mathbb{NFO)}}\mathbf{N}=(
CHARACTER * 1 JOBZ
DOUBLE COM PLEX A (LDA,*),U (LDU,*),VT (LDVT,*),W ORK (*)
\mathbb{NTEGER*8M,N,LDA,LDU,LDVT,LW ORK, INFO}
INTEGER*8 IN ORK (*)
DOUBLE PRECISION S (*),RW ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GESDD (JOBZ, \(\mathbb{M}], \mathbb{N}], A,[L D A], S, U,[L D U], V T,[L D V T]\), [W ORK ], [LW ORK], RW ORK ], [IW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1): : JOBZ
COMPLEX (8),D IM ENSION (:) ::W ORK

COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : :: A, U, VT
\(\mathbb{N} T E G E R:: M, N, L D A, L D U, L D V T, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL (8),D \(\mathbb{M}\) ENSION (:) ::S,RW ORK
SU BROUTINE GESDD_64 (OBBZ, \(\mathbb{M}], \mathbb{N}], A,[L D A], S, U,[L D U], V T,[L D V T]\), [W ORK ], [LW ORK ], RW ORK ], [IW ORK ], [ \(\mathbb{N} F \mathrm{~F}\) ])

CHARACTER (LEN=1): : JOBZ
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D IM ENSION (:,:) ::A, U,VT
\(\mathbb{N}\) TEGER ( 8 ) :: M , N,LDA, LDU ,LDVT,LW ORK, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
REAL (8),D \(\mathbb{M}\) ENSION (:) ::S,RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zgesdd (char jंbz, intm, intn, doublecom plex *a, int lda, double *s, doublecom plex *u, int ldu, doublecom plex *vt, int ldvt, int *info);
void zgesdd_64 (char jobz, long m, long n, doublecom plex *a, long lda, double *s, doublecom plex *u, long ldu, doublecom plex *vt, long ldvt, long *info);

\section*{PURPOSE}
zgesdd com putes the singular value decom position (SVD ) of a complex M -by-N matrix A, optionally com puting the left and/orrightsingularvectors, by using divide-and-conquer \(m\) ethod. The SVD isw ritten \(=U * S I G M A *\) conjugate-transpose \(N\) )
where SIGM A is an \(M\)-by \(-N \mathrm{~m}\) atrix which is zero except for its \(m\) in \((m, n\) ) diagonal elem ents, \(U\) is an \(M\) by \(M\) unitary \(m\) atrix, and \(V\) is an \(N\)-by \(-N\) unitary \(m\) atrix. The diagonalelem ents of SIGMA are the singularvalues ofA; they are realand nonnegative, and are retumed in descending order. The first m in ( \(\mathrm{m}, \mathrm{n}\) ) colum \(n s\) of \(U\) and \(V\) are the left and right singular vectors of A.

N ote that the routine retums \(\mathrm{VT}=\mathrm{V}\) **H, notV .

The divide and conquer algorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on \(m\) achines w ith a guard digit in add/subtract, or on those binary \(m\) achines \(w\) ithout guard digits which subtract like the \(C\) ray \(X-M P\), C ray Y \(-M P\), C ray C-90, orC ray-2. It could conceivably fail on hexadecim al or decim al machines \(w\) thout guard

\section*{ARGUMENTS}

JOBZ (input)
Specifies options for com puting allorpart of the
m atrix U :
= \(A\) ': allM colum ns of U and all N row sof \(\mathrm{V} * * H\) are retumed in the arays \(U\) and \(V T ;=S\) : the firstm in \((M, N)\) colum ns of \(U\) and the firstm in \((M, N)\) row s of V ** H are retumed in the arrays U and VT ; \(=\mathrm{O}^{\prime}\) : If \(\mathrm{M}>=\mathrm{N}\), the first N colum ns of U are overw rilten on the array \(A\) and allrow sofV **H are retumed in the array VT; otherw ise, all colum \(n s\) of \(U\) are retumed in the anray \(U\) and the firstM row sofV \({ }^{* * H}\) are overw rilten in the amay \(\mathrm{VT} ;=\mathrm{N}\) ': no colum ns of U or row sof V ** H are com puted.

M (input) The num ber of row s of the inputm atrix \(A . M>=0\).

N (input) The num ber of colum ns of the inputm atrix \(\mathrm{A} . \mathrm{N}>=\) 0.

A (input/output)
On entry, the M -by-N m atrix A. On exit, if OBB = \(O^{\prime}\) ' \(A\) is overw rilten \(w\) ith the first \(N\) colum \(n s\) of
U (the leftsingularvectors, stored colum nw ise) if \(\mathrm{M}>=\mathrm{N}\); A is overw ritten w ith the firstM row S of \(V * * H\) the right singularvectors, stored row wise) otherw ise. if JOBZ ne. O', the contents of A are destroyed.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, M)\).

S (output)
The singularvalues of \(A\), sorted so that \(S\) (i) \(>=\) \(S(i+1)\) 。

U (output)
\(\mathrm{UCOL}=\mathrm{M}\) if \(\mathrm{OBZ}=\mathrm{A}^{\prime}\) or \(\mathrm{OBZ}=\mathrm{D}^{\prime}\) 'and \(\mathrm{M}<\mathrm{N}\);
\(\mathrm{UCOL}=\mathrm{m}\) in \((\mathrm{M}, \mathrm{N})\) if \(J O B Z=S^{\prime}\). If \(\mathrm{OOBZ}=A^{\prime}\) or
JOBZ \(=0\) 'and \(M<N\), U contains the \(M\) boy \(M\) unitary \(m\) atrix \(U\); if \(J O B Z=S\) ', \(U\) contains the first
\(m\) in \((M, N)\) colum ns of \(U\) (the leftsingular vectors, stored colum nw ise); if \(\mathrm{OBBZ}=0\) 'and \(\mathrm{M}>=\mathrm{N}\), or
\(J O B Z=N ', U\) is not referenced.

LD U (input)
The leading dim ension of the array \(U\). LD U >= 1 ; if \(\mathrm{OBZ}=\mathrm{S}^{\prime}\) or A 'or \(\mathrm{OB} \mathrm{BZ}=\mathrm{O}^{\prime}\) 'and \(\mathrm{M}<\mathrm{N}\), LDU \(>=M\).

VT (output)
If \(\mathrm{JOBZ}=\mathrm{A}\) 'or \(\mathrm{JOBZ}=\mathrm{O}\) 'and \(\mathrm{M}>=\mathrm{N}, \mathrm{VT}\) contains the N -by N unitary m atrix \(\mathrm{V} * * \mathrm{H}\); if \(\mathrm{JOBZ}=\) \(S ', V T\) contains the firstm in \(M, N\) ) row s of \(V * * H\) (the right singularvectors, stored row w ise); if \(\mathrm{JOBZ}=\mathrm{O}\) 'andM \(<\mathrm{N}\), or \(\mathrm{OBBZ}=\mathrm{N}\) ', VT is not referenced.

LDVT (input)
The leading dim ension of the array V T. LD V T >=1;
if \(J 0 B Z=A\) 'or \(O B Z=0\) 'and \(M>=N, L D V T>=N\);
if \(J O B Z=S \prime, L D V T>=m\) in \(M, N)\).
W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK.LW ORK >= 1. if \(\operatorname{JOBZ}=N\) ', LW ORK \(>=2 \star \mathrm{~m}\) in \(M, N)+m a x(M, N)\). if JOBZ
\(=\quad 0!\quad L W O R K \quad>=\)

\(=S^{\prime}\) or \(A^{\prime}\), LWORK >=
\(m\) in \((M, N){ }^{*} m\) in \(\left.M, N\right)+2 \star_{m}\) in \(\left.M, N\right)+m\) ax \((M, N)\). For good perform ance, LW ORK should generally be larger. If LW ORK < 0 but other input argum ents are legal, W ORK (1) retums optim alLW ORK.

RW ORK (w orkspace)
If \(\mathrm{JOBZ}=\mathrm{N}\) ', LRW ORK \(>=7 * \mathrm{~m}\) in \(M, N)\). O therw ise, LRW ORK \(>=5 \star^{m}\) in \((M, N) *_{m}\) in \((M, N)+5 \star_{m}\) in \(\left.M, N\right)\)

IV ORK (w orkspace)
dim ension ( \(8 \star \mathrm{M} \mathbb{I} \mathbb{M}, N)\) )
\(\mathbb{N} F O\) (output)
= 0 : successfulexit.
< 0: if \(\mathbb{N N}\) FO = -i, the i-th argum ent had an illegalvahue.
\(>0\) : The updating process of SBD SD C did not converge.

\section*{B ased on contributions by}

M ing \(G u\) and \(H\) uan Ren, \(C\) om puterScience D ívision, U niversity of
C alifomia atB erkeley, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgesv -com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}

```

DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGERN,NRHS,LDA,LDB,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}

```

```

DOUBLE COM PLEX A (LDA,*),B (LDB,*)
\mathbb{NTEGER*8N,NRHS,LDA,LDB, INFO}
INTEGER*8\mathbb{PIVOT (*)}

```
F95 INTERFACE
    SUBROUTINE GESV ( \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I V O T}, \mathrm{B},[\operatorname{LDB}],[\mathbb{N} F O])\)
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A,B
    \(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, \mathbb{N F O}\)
    \(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
    SUBROUTINEGESV_64 ( \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I V O T}, \mathrm{B},[\operatorname{LDB}],[\mathbb{N} F O])\)
    COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B
    \(\mathbb{N}\) TEGER (8) ::N,NRHS,LDA,LDB, \(\mathbb{N} F O\)
    \(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
C INTERFACE
    \#include <sunperfh>
void zgesv (intn, intnins, doublecom plex *a, int lda, int *ípívot, doublecom plex *b, int ldl, int *info);
void zgesv_64 (long n, long nihs, doublecom plex *a, long lda, long *ípivot, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
zgesv com putes the solution to a com plex system of linear equations
\(A * X=B, w h e r e A\) is an \(N\) boy \(-N m\) atrix and \(X\) and \(B\) are N -by-N R H S m atrices.
The LU decom position w ith partialpivoting and row interchanges is used to factorA as
\(A=P * L * U\),
\(w\) here \(P\) is a perm utation \(m\) atrix, \(L\) is unit low er triangular, and \(U\) is upper triangular. The factored form of \(A\) is then used to solve the system ofequations \(A * X=B\).

\section*{ARGUMENTS}

N (input) The num ber of linear equations, ie., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS \(>=0\).

A (input/output)
O n entry, the \(N\) boy -N coefficient \(m\) atrix \(A\). On exit, the factors \(L\) and \(U\) from the factorization \(A\) \(=\mathrm{P} * \mathrm{~L} * \mathrm{U}\); the unitdiagonalelem ents of \(L\) are not stored.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

IPIVOT (output)
The pivot indices that define the perm utation \(m\) atrix \(P\); row i of the \(m\) atrix \(w\) as interchanged w ith row \(\mathbb{P}\) IVOT (i).

B (input/output)
On entry, the N -by-N RH S m atrix of righthand side
\(m\) atrix \(B\). On exit, if \(\mathbb{N} F O=0\), the \(N\) boy-NRHS solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray \(B\). LD B \(>=\) \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO = -i, the i-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{U}(i, i)\) is exactly zero. The factorization has been com pleted, but the factor \(U\) is exactly singular, so the solution could not be com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgesvd - com pute the singular value decom position (SVD ) of a com plex M -by -N m atrix A , optionally com puting the left and/or right singularvectors

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGESVD(JOBU,NOBVT,M,N,A,LDA,SNNG,U,LDU,VT,LDVT,} W ORK,LDW ORK,WORK2, $\mathbb{N} F O$ )

```

CHARACTER * 1 JOBU, JOBVT
 \(\mathbb{N}\) TEGERM,N,LDA,LDU,LDVT,LDW ORK, \(\mathbb{N} F O\) DOUBLE PRECISION STNG (*),W ORK 2 (*)

SU BROUTINE ZGESVD_64 (JOBU, JOBVT,M,N,A,LDA,SING,U,LDU,VT, LDVT, W ORK,LDW ORK,W ORK \(2, \mathbb{I N F O )}\)

CHARACTER * 1 JOBU, JOBVT

\(\mathbb{N}\) TEGER*8M,N,LDA,LDU,LDVT,LDW ORK, \(\mathbb{N} F O\)
DOUBLE PRECISION SING (*),W ORK 2 (*)

\section*{F95 INTERFACE}

SU BROUTINE GESVD (JOBU, JO BVT, \(\mathbb{M}], \mathbb{N}], A,[L D A], S \mathbb{N G}, \mathrm{U},[L D U], V T\), [LDVT], [W ORK], [LDW ORK], [W ORK2], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) :: JOBU,NOBVT
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSIDN (: : : : : A , U , VT
\(\mathbb{N} T E G E R:: M, N, L D A, L D U, L D V T, L D W O R K, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::SING,W ORK2
 VT, [LDVT], [W ORK ], [LDW ORK ], [W ORK 2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) :: JOBU, JOBVT
COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::A, U ,VT
\(\mathbb{N} T E G E R(8):: M, N, L D A, L D U, L D V T, L D W O R K, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::SING,W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zgesvd (char j̀bu, char jobvt, intm , intn, doublecom plex *a, intlda, double *sing, doublecom plex *u, int ldu, doublecom plex *vt, int ldvt, int *info);
void zgesvd_64 (char jobu, char jobvt, long m, long n, doublecom plex *a, long lda, double *sing, doublecom plex *u, long ldu, doublecom plex *vt, long ldvt, long *info);

\section*{PURPOSE}
zgesvd com putes the singular value decom position (SVD ) of a complex M -by-N matrix A, optionally com puting the left and/or rightsingularvectors. The SVD isw ritten \(=U\) * SIGMA * conjugate-transpose \((N)\)
where SIG M A is an M -by -N m atrix which is zero except for its \(m\) in ( \(m, n\) ) diagonal elem ents, \(U\) is an \(M\) by \(M\) unitary \(m\) atrix, and \(V\) is an \(N\) by -N unitary \(m\) atrix. The diagonalelem ents of SIGM A are the singularvalues ofA ; they are realand nonnegative, and are retumed in descending order. The first m in ( \(\mathrm{m}, \mathrm{n}\) ) colum ns of U and V are the leftand rightsingular vectors ofA.

N ote that the routine retums V ** H , not V .

\section*{ARGUMENTS}
\(J 0\) BU (input)
Specifies options for com puting allorpart of the m atrix U :
= A ': allM colum ns ofU are retumed in array U :
= S ': the firstm in \((\mathrm{m}, \mathrm{n})\) colum nsofU the left singular vectors) are retumed in the array \(U\); \(=\) \(O^{\prime}\) ': the firstm in \((m, n)\) colum ns of \(U\) (the left
singular vectors) are overw ritten on the array A;
\(=\mathrm{N}\) : no colum ns of U (no leftsingularvectors) are com puted.

JO BVT (input)
Specifies options for com puting allorpart of the m atrix V ** H :
\(=A\) ': allN row sofV \({ }^{* *} \mathrm{H}\) are retumed in the amay VT;
\(=S\) : the firstm in \((m, n)\) row s ofV \(* * H\) (the right singular vectors) are retumed in the array \(\mathrm{VT} ;=\) \(O^{\prime}\) : the firstm in \((m, n)\) row s ofV **H the right singular vectors) are overw rilten on the array A; \(=\mathrm{N}^{\prime}\) : no row sofV \(* * \mathrm{H}\) (no right singular vectors) are com puted.

JO BVT and JOBU cannotboth be D'.
\(M\) (input) The num ber of row s of the inputm atrix \(A . M>=0\).

N (input) The num ber of colum ns of the inputm atrix \(\mathrm{A} . \mathrm{N}>=\) 0 .

A (input/output)
On entry, the M -by -N m atrix A . On exit, if \(\mathrm{OBU}=\) 0 , A is overw ritten \(w\) ith the firstm in \((m, n)\)
colum ns of \(U\) (the left singular vectors, stored colum nw ise); if \(\mathrm{JOBVT}=\mathrm{O}^{\prime}\), A is overw rilten w th the firstm in \((m, n)\) row s of \(V * * H\) (the right singularvectors, stored row w ise); if OB BU ne. O 'and JOBVT ne. 0 ', the contents of A are destroyed.

LDA (input)
The leading dim ension of the aray \(A . L D A>=\) \(\max (1, M)\).

SIN G (output)
The singularvalues of \(A\), sorted so that \(\operatorname{SING}\) (i) \(>=S \mathbb{N G}(i+1)\).

\(S^{\prime}\). If \(O B U=A \prime, U\) contains the \(M\)-by \(M\) unitary m atrix U ; if \(\mathrm{OB} \mathrm{B}=\mathrm{S}^{\prime}\), U contains the first \(m\) in ( \(m, n\) ) colum ns of \(U\) (the leftsingular vectors, stored colum nw ise); if JOBU = N 'or O', U is not referenced.

LD U (input)
The leading dim ension of the array \(\mathrm{U} . \mathrm{LDU}>=1\); if \(J O B U=S\) 'or \(A\) ',LDU >=M.

\section*{VT (input)}

If \(\mathrm{JOBVT}=\mathrm{A}\) ', VT contains the N -by-N unitary \(m\) atrix \(\mathrm{V} * * \mathrm{H}\); if \(\mathrm{JO} \mathrm{BV} \mathrm{T}=\mathrm{S}\) ', VT contains the first m in \((\mathrm{m}, \mathrm{n})\) row sof \(\mathrm{V} * * \mathrm{H}\) (the right singularvectors, stored row wise); if JOBVT = N 'or \(\mathrm{O}^{\prime}\) ', VT is not referenced.

LDVT (input)
The leading dim ension of the array V T. LD V T >=1; if \(\mathrm{JOBVT}=\mathrm{A}\) ', LDVT >= ; if \(\mathrm{if} \mathrm{OBVT}=\mathrm{S}\) ', LDVT >= \(m\) in \(M, N\) ).
W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dim ension of the array W ORK. LDW ORK >= 1 . LDW ORK >= \(2 * M \mathbb{N}(M, N)+M A X(M, N)\) Forgood perform ance, LD W ORK should generally be larger.

IfLDW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of theW ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 2 (w orkspace)
D \(\mathbb{M}\) ENSION ( \(5 * \mathrm{M} \mathbb{N} \mathbb{M}, N)\) ). On exit, if \(\mathbb{N} F O>0\), W ORK \(2(\mathbb{M} \mathbb{N} \mathbb{M}, \mathbb{N})-1)\) contains the unconverged superdiagonal elem ents of an upper bidiagonal m atrix B whose diagonal is in \(\mathrm{S} \mathbb{N} G\) (notnecessarily sorted). \(B\) satisfies \(A=U * B * V T, s o\) it has the sam e singular values as A, and singular vectors related by \(U\) and \(V T\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit.
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum enthad an illegalvahue.
> 0 : ifCBD \(S Q R\) did notconverge, \(\mathbb{N F O}\) specifies how \(m\) any superdiagonals of an interm ediate bidiagonalform B did not converge to zero. See the description ofW ORK 2 above for details.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgesvx -use the LU factorization to com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}

```

    EQUED,R,C,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,
    W ORK2,\mathbb{NFO)}
    CHARACTER * 1 FACT,TRANSA,EQUED
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
\mathbb{N TEGER N,NRHS,LDA,LDAF,LDB,LDX, INFO}
\mathbb{NTEGER IPIVOT (*)}
DOUBLE PRECISION RCOND
DOUBLE PRECISIONR(*),C (*),FERR (*),BERR (*),WORK2 (*)
SUBROUT\mathbb{NE ZGESVX_64(FACT,TRANSA,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,}}\mathbf{N},\mp@code{N},
EQUED,R,C,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,
W ORK2,\mathbb{NFO)}
CHARACTER * 1FACT,TRANSA,EQUED
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
INTEGER*8N,NRHS,LDA,LDAF,LDB,LDX, \mathbb{NFO}
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION RCOND
DOUBLE PRECISION R (*),C (*),FERR (*),BERR (*),W ORK 2 (*)

```

F95 INTERFACE
SU BROUTINE GESVX (FACT, [TRANSA], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F]\), \(\mathbb{P} \mathbb{V} O T, E Q U E D, R, C, B,[L D B], X,[L D X], R C O N D, F E R R\),

BERR, [W ORK], [WORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::FACT,TRANSA, EQUED
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : A , AF, B, X
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL (8) :: RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::R,C,FERR,BERR,W ORK 2
SU BROUTINE GESVX_64(EACT, [TRANSA], \(\mathbb{N}], \mathbb{N R H S}], A,[L D A], A F,[L D A F]\), \(\mathbb{P} I V O T, E Q U E D, R, C, B,[L D B], X,[L D X], R C O N D, F E R R\), BERR, [W ORK], [W ORK2], [NFO])

CHARACTER (LEN=1)::FACT,TRANSA,EQUED
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : : : A , AF, B, X
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8) :: RCOND
REAL (8),D IM ENSION (:) ::R,C ,FERR,BERR,W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zgesvx (char fact, chartransa, intn, int nihs, doublecom plex *a, int lda, doublecom plex *af, int ldaf, int *ipivot, char equed, double *r, double \({ }^{*} \mathrm{c}\), doublecom plex *b, int ldb, doublecom plex *x, int ldx, double *roond, double *ferr, double *berr, int*info);
void zgesvx_64 (char fact, chartransa, long n, long nihs, doublecom plex *a, long lda, doublecom plex *af, long ldaf, long *ipívot, char equed, double *r, double *c, doublecom plex *b, long ldb, doublecom plex *x, long ldx, double *roond, double *ferr, double *berr, long *info);

\section*{PURPOSE}
zgesvx uses the LU factorization to com pute the solution to a com plex system of linear equations
\(A\) * \(X=B\), where \(A\) is an \(N\) boy \(N \mathrm{~m}\) atrix and \(X\) and \(B\) are N -by-N R H S m atrices.

Errorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT = E', real scaling factors are com puted to equilibrate
the system :
TRANS \(=\mathrm{N}^{\prime}: \operatorname{diag}(\mathbb{R}) * \mathrm{~A} * \operatorname{diag}(\mathrm{C}) \quad * \operatorname{inv}(\operatorname{diag}(\mathrm{C})) * \mathrm{X}=\) \(\operatorname{diag}(\mathbb{R}) * B\)

TRANS \(=T^{\prime}:(\operatorname{diag}(R) * A * \operatorname{diag}(C)) * * T * \operatorname{inv}(\operatorname{diag}(R)) * X=\) diag (C) *B

TRANS \(=C^{\prime}:(\operatorname{diag}(R) * A * \operatorname{diag}(C)) * * H * \operatorname{inv}(\operatorname{diag}(R)) * X=\) diag (C) *B
W hether or not the system w illbe equilibrated depends on the
scaling of the m atrix A, but if equilibration is used, A is
overw ritten by diag \((\mathbb{R}) * A\) *diag ( \(C\) ) and \(B\) by \(\operatorname{diag}(R) * B\) (if TRANS \(=N\) )
ordiag (C)*B (if TRANS = T'orC).
2. IfFACT \(=N\) 'or \(E\) ', the LU decom position is used to factor the
\(m\) atrix A (afterequilibration ifFACT = E) as
\(A=P * L * U\),
\(w\) here \(P\) is a perm utation \(m\) atrix, \(L\) is a unit low er triangular
\(m\) atrix, and \(U\) is upper triangular.
3. If som e \(U(i, i)=0\), so that \(U\) is exactly singular, then the routine
retums w ith \(\mathbb{N} F O=\) i. O therw ise, the factored form of A is used
to estim ate the condition num ber of the \(m\) atrix \(A\). If the reciprocal of the condition num ber is less than \(m\) achine precision,
\(\mathbb{N} F O=N+1\) is retumed as a waming, but the routine stillgoes on
to solve for X and com pute error bounds as described below .
4.The system ofequations is solved for X using the factored form
of A.
5. Herative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw and error estim ates
for 五.
6. Ifequilibration \(w\) as used, the \(m\) atrix \(X\) is prem ultiplied by
diag (C) (iftRANS = N) ordiag \((\mathbb{R})\) (ifTRANS = \(T^{\prime}\) or C) so
that it solves the original system before equilibration.

\section*{ARGUMENTS}

FACT (input)
Specifies whether ornot the factored form of the \(m\) atrix A is supplied on entry, and if not, w hether the m atrix A should be equilibrated before it is factored. \(=\mathrm{F}\) ': On entry, AF and \(\mathbb{P I V O T}\) contain the factored form of A. IfEQUED is not \(N\) ', the \(m\) atrix A has been equilibrated \(w\) th scaling factors given by \(R\) and \(C . A, A F\), and \(\mathbb{P} I V O T\) are not m odified. \(=\mathrm{N}\) ': The m atrix A w illbe copied to AF and factored.
= E ': The matrix A will be equilibrated if necessary, then copied to AF and factored.
TRANSA (input)
Specifies the form of the system of equations:
\(=N: A * X=B \quad\) N \(\circ\) transpose)
\(=T\) ': A ** T * \(\mathrm{X}=\mathrm{B}\) ( T ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (Conjugate transpose)

N (input) The num ber of linear equations, i.e., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the m atrices B and X. NRHS \(>=0\).

A (input/output)
On entry, the N -by -N m atrix A . IfFA CT = \(\mathrm{F}^{\prime}\) and EQUED is not \(N\) ', then \(A\) musthave been equilibrated by the scaling factors in R and/orC. A is not modified if FACT \(=\) F'or \(N\) ', or if FACT \(=\) E'and EQUED = N 'on exit.

Onexit, ifEQUED ne. \(N\) ', A is scaled as follows: EQUED \(=R\) ': \(A=\operatorname{diag}(R) * A\)
\(\mathrm{EQUED}=\mathrm{C}: \mathrm{A}:=\mathrm{A} * \operatorname{diag}(\mathrm{C})\)
EQUED = \(B^{\prime}: A:=\operatorname{diag}(\mathbb{R}) * A * \operatorname{diag}(C)\).
LDA (input)
The leading dim ension of the anay A. LDA >= \(\max (1, N)\).

AF (input/output)
If \(F A C T=F\) ', then \(A F\) is an inputargum ent and on
entry contains the factors \(L\) and \(U\) from the factorization \(A=P * L * U\) as com puted by CGETRF. If EQUED ne. \(N\) ', then \(A F\) is the factored form of the equilibrated \(m\) atrix \(A\).

If \(\mathrm{FACT}=\mathrm{N}\) ', then AF is an output argum ent and on exit retums the factors \(L\) and \(U\) from the factorization \(A=P * L * U\) of the originalm atrix \(A\).

IfFACT \(=\mathrm{E}\) ', then \(A F\) is an output argum ent and on exit retums the factors \(L\) and \(U\) from the factorization \(\mathrm{A}=\mathrm{P} * \mathrm{~L} * \mathrm{U}\) of the equilibrated m atrix A (see the description of \(A\) for the form of the equilibrated \(m\) atrix).

\section*{LDAF (input)}

The leading dim ension of the array AF. LD AF >= \(\max (1, N)\).

\section*{IPIVOT (inputoroutput)}

IfFACT=F', then \(\mathbb{P} \mathbb{V} O T\) is an input argum ent and on entry contains the pivot indioes from the factorization \(\mathrm{A}=\mathrm{P} * \mathrm{~L} * \mathrm{U}\) as com puted by CGETRF; row \(i\) of the \(m\) atrix \(w\) as interchanged \(w\) th row \(\mathbb{P} \mathbb{V} O T\) (i).

IfFACT \(=N\) ', then \(\mathbb{P} \mathbb{I V O T}\) is an output argum ent and on exit contains the pivot indiges from the factorization \(\mathrm{A}=\mathrm{P} * \mathrm{~L} * \mathrm{U}\) of the originalm atrix A .

IfFACT=E', then \(\mathbb{P I V O T}\) is an output argum ent and on exit contains the pivot indices from the factorization \(A=P * L * U\) of the equilibrated \(m\) atrix A.

EQUED (input/output)
Specifies the form of equilibration thatw as done.
\(=N^{\prime}:\) N o equilibration (alw ays true ifFACT =
\(\mathrm{N})\).
\(=R\) ': Row equilibration, ie., A has been
prem ultiplied by diag \((R)\) ) = C': Colum n equilibration, ie., A has been postm ultiplied by diag (C ). = B ': B oth row and colum \(n\) equilibration, ie., A has been replaced by diag \((\mathbb{R})\) * A * diag (C). EQUED is an inputargum entifFACT= F '; otherw ise, it is an output argum ent.
\(R\) (input/output)
The row scale factors forA. IfEQUED = R' or
\(B\) ', A is multiplied on the leftby diag \((\mathbb{R})\); if
\(E Q U E D=N\) 'or \(C\) ', \(R\) is notaccessed. \(R\) is an
input argum ent if \(\mathrm{FACT}=\mathrm{F}\) '; otherw ise, R is an output argum ent. IfFACT = F'andEQUED = R'or \(B\) ',each elem entofR \(m\) ustbe positive.

C (input/output)
The colum \(n\) scale factors for \(A\). IfEQUED = C 'or
B ', A is multiplied on the right.by diag ( C ); if EQUED \(=N\) 'or \(R\) ', \(C\) is notaccessed. \(C\) is an input argum ent ifFACT = F '; otherw ise, C is an outputargum ent. IfFACT = F 'and \(\mathrm{EQUED}=\mathrm{C}\) 'or B ', each elem entofC m ustbe positive.
B (input/output)
On entry, the N -by-NRH S righthand side m atrix B .
On exit, if EQUED \(=N\) ', \(B\) is notm odified; if TRANSA \(=N^{\prime}\) and EQUED \(=R\) 'or \(B\) ', \(B\) is overw rilten by diag \((R) * B\); ifTRANSA \(=T^{\prime}\) 'or \(C^{\prime}\) and EQUED \(=C^{\prime}\) or \(B^{\prime}, B\) is overw ritten by diag (C) *B .

LD B (input)
The leading dim ension of the array \(B\). LD B >= \(\max (1, N)\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=N+1\), the N -by -N RH \(S\) solution
\(m\) atrix \(X\) to the original system of equations.
\(N\) ote that \(A\) and \(B\) arem odified on exit if EQUED
ne. \(N\) ', and the solution to the equilibrated
system is inv (diag (C ))*X ifTRANSA = N 'and EQUED
\(=C\) 'or \(B^{\prime}\), orinv (diag \(\left.(R)\right) * X\) ifTRANSA \(=T\) 'or \(C^{\prime}\) and EQUED \(=R\) 'or \(B\) '.

LD X (input)
The leading dim ension of the anay X . LD X >= \(\max (1, N)\).

\section*{RCOND (output)}

The estim ate of the reciprocal condition num ber of the matrix A afterequilibration (if done). If
RCOND is less than them achine precision (in particular, ifRCOND \(=0\) ), the \(m\) atrix is singular to w orking precision. This condition is indicated by a retum code of \(\mathbb{N} \mathrm{FO}>0\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X()\) ) the \(j\) th column of the solution \(m\) atrix X). If XTRUE is the true solution comesponding to \(X(\mathcal{O}), \operatorname{FERR}(\mathcal{7})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele-
\(m\) ent in \((X(\mathcal{O})-X\) TRUE \()\) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{)}\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector \(X(\mathcal{)}\) (i.e., the sm allest relative change in any elem entof \(A\) orB thatm akes \(X(\mathcal{j})\) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension ( \(2 *\) N ) On exit, W ORK 2 (1) contains the reciprocal pivot grow th factornorm (A)/norm (U). The "max absolute elem ent" norm is used. If W ORK2(1) is much less than 1, then the stability of the LU factorization of the (equilibrated)
\(m\) atrix A could be poor. This also \(m\) eans that the solution X, condition estim atorRC OND, and forw ard error bound \(F E R R\) could be unreliable. If factorization fails \(w\) ith \(0<\mathbb{N} F O<=N\), then \(W\) ORK 2 (1) contains the reciprocalpivot grow th factor for the leading \(\mathbb{N}\) FO colum ns of A.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the i-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N F O}=i\), and \(i\) is
<= N : U (i,i) is exactly zero. The factorization
has been com pleted, but the factorU is exactly singular, so the solution and error bounds could not be com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1: \mathrm{U}\) is nonsingular, butRCOND is less than machine precision, \(m\) eaning that the \(m\) atrix is singular to w orking precision. N evertheless, the solution and error bounds are com puted because there are a num ber of situations w here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgetf2 -com pute an LU factorization of a general \(m\)-by-n \(m\) atrix A using partialpivoting \(w\) th row interchanges

\section*{SYNOPSIS}
```

SUBROUTINE ZGETF2M,N,A,LDA,\mathbb{PIV,INFO)}
D OUBLE COM PLEX A (LDA,*)
INTEGERM,N,LDA,INFO
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{*})
SU BROUT\mathbb{NE ZGETF2_64M,N,A,LDA, \mathbb{P IV,NNFO)}}\mathbf{M}\mathrm{ (N)}
DOUBLE COM PLEX A (LDA,*)
INTEGER*8M,N,LDA,INFO
\mathbb{NTEGER*8 \mathbb{P IV (*)}}\mathbf{(*)}
F95 INTERFACE

```

```

    COM PLEX (8),D IM ENSION (:,:) ::A
    \mathbb{NTEGER ::M,N,LDA,INFO}
    INTEGER,D IM ENSION (:) :: \mathbb{P IV}
    ```

```

    COM PLEX (8),D IM ENSION (:,:) ::A
    \mathbb{NTEGER (8) ::M,N,LDA,INFO}
    INTEGER (8),D IM ENSION (:) ::\mathbb{PIV}
    ```
C INTERFACE
    \#include < sunperfh>
void zgetf2 (intm, intn, doublecom plex *a, int lda, int *ịpiv, int *info);
void zgetf2_64 (long m, long n, doublecom plex *a, long lda, long *ịiv, long *info);

\section*{PURPOSE}
zgetff com putes an LU factorization of a general \(m\) łoy-n \(m\) atrix A using partialpivoting \(w\) ith row interchanges.

The factorization has the form
\[
A=P * L * U
\]
\(w\) here \(P\) is a perm utation \(m\) atrix, \(L\) is low ertriangular \(w\) ith unit diagonal elem ents (low ertrapezoidalifm > n), and U is upper triangular (uppertrapezoidalifm < n).

This is the right-looking Level2 B LA S version of the algorithm .

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, them by \(n m\) atrix to be factored. On
exit, the factors \(L\) and \(U\) from the factorization \(A\)
\(=\mathrm{P} * \mathrm{~L} * \mathrm{U}\); the unitdiagonalelem ents of L are not stored.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, M)\).
\(\mathbb{P} \mathbb{I V}\) (output)
The pivotindioes; for \(1<=i<=m\) in \((M)\) ), row i of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P} \mathbb{I V}\) (i).

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-\mathrm{k}\), the k -th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=k, U(k, k)\) is exactly zero. The factorization has been com pleted, but the factor \(U\) is
exactly singular, and division by zero will occur
if it is used to solve a system of equations.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgetrf-com pute an LU factorization of a general M by -N \(m\) atrix A using partialpivoting \(w\) th row interchanges

\section*{SYNOPSIS}

```

D OUBLE COM PLEX A (LDA,*)
INTEGERM,N,LDA,INFO
INTEGER \mathbb{PIVOT (*)}

```

```

DOUBLE COM PLEX A (LDA,*)
INTEGER*8M,N,LDA,INFO
INTEGER*8 \mathbb{PIVOT (*)}
F95 INTERFACE

```

```

    COM PLEX (8),D IM ENSION (:,:) ::A
    \mathbb{NTEGER ::M,N,LDA,INFO}
    \mathbb{NTEGER,D IM ENSION (:) :: \mathbb{PIVOT}}\mathbf{T}\mathrm{ O}
    ```

```

    COM PLEX (8),D IM ENSION (:,:) ::A
    \mathbb{NTEGER (8) ::M,N,LDA,INFO}
    \mathbb{NTEGER (8),D IM ENSION (:)::\mathbb{PIVOT}}\mathbf{~}=\mp@code{M}
    ```
C INTERFACE
    \#include < sunperfh>
void zgetrf(intm, intn, doublecom plex *a, int lda, int *ípívot, int*info);
void zgetrf_64 (long m, long n, doublecom plex *a, long lda, long *ipivot, long *info);

\section*{PURPOSE}
zgetrfoom putes an LU factorization of a general M boy-N \(m\) atrix A using partialpivoting \(w\) ith row interchanges.

The factorization has the form
\[
A=P * L * U
\]
\(w\) here \(P\) is a perm utation \(m\) atrix, \(L\) is low ertriangular \(w\) ith unit diagonal elem ents (low ertrapezoidalifm > n), and U is upper triangular (uppertrapezoidalifm < n).

This is the right-looking Level3 B LA S version of the algorithm .

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the M -by \(-\mathrm{N} m\) atrix to be factored. On exit, the factors \(L\) and \(U\) from the factorization \(A\)
\(=\mathrm{P} * \mathrm{~L} * \mathrm{U}\); the unitdiagonalelem ents of L are not stored.

LD A (input)
The leading dim ension of the aray A. LD A >= \(\max (1, M)\).

PIVOT (output)
The pivotindioes; for \(1<=i<=m\) in \((M)\) ), row i
of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P}\) IV OT (i).

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the \(i\) th argum enthad an ille-
galvalue
\(>0:\) if \(\mathbb{N} F O=i, U(i, i)\) is exactly zero. The factorization has been com pleted, but the factor \(U\)
is exactly singular, and division by zero will occur if it is used to solve a system ofequations.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgetri-com pute the inverse of a m atrix using the LU factorization com puted by CGETRF

\section*{SYNOPSIS}

SUBROUTINE ZGETRIN,A,LDA, \(\mathbb{P} \mathbb{I} O T, W\) ORK,LDW ORK, \(\mathbb{N} F O\) )
D OUBLE COM PLEX A (LDA, *),W ORK (*) \(\mathbb{N} T E G E R N, L D A, L D W O R K, \mathbb{N F O}\) \(\mathbb{N T E G E R} \mathbb{P} \mathbb{I V O T}{ }^{( }\))

SUBROUTINE ZGETRI_64 \(\mathbb{N}, A, L D A, \mathbb{P} \mathbb{V} O T, W\) ORK,LDW ORK, \(\mathbb{N} F O\) )
DOUBLE COM PLEX A (LDA, \(\left.{ }^{*}\right), \mathrm{W}\) ORK ( \(\left.{ }^{( }\right)\)
\(\mathbb{N} T E G E R * 8 N, L D A, L D W\) ORK, \(\mathbb{N} F O\)
\(\mathbb{N T E G E R} * 8 \mathbb{P} \mathbb{I V O T}\left({ }^{\star}\right)\)

\section*{F95 INTERFACE}

SUBROUTINE GETRI( \(\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V O T}, \mathbb{W}\) ORK \(]\), [LDW ORK], [ \(\mathbb{N F O}])\)
COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX (8), D IM ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: N, L D A, L D W\) ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V} O T\)
SU BROUTINE GETRI_64 (N ],A, [LDA], \(\mathbb{P} \mathbb{V} O T,[\mathbb{N} O R K],[L D W\) ORK \(],[\mathbb{N F O}])\)
COMPLEX (8), D IM ENSION (:) ::W ORK
COM PLEX (8), D IM ENSION (: : : : ::A
\(\mathbb{N}\) TEGER (8) ::N,LDA,LDW ORK, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)

\section*{C INTERFACE}
\#include <sunperfh>
void zgetri(intn, doublecom plex *a, int lda, int *ipivot, int*info);
void zgetri_64 (long n, doublecom plex *a, long lda, long *ipivot, long *info);

\section*{PURPOSE}
zgetricom putes the inverse of a m atrix using the LU factorization com puted by CGETRF .

Thism ethod inverts \(U\) and then com putes inv (A) by solving the system \(\operatorname{inv}(A) * L=\operatorname{inv}(U)\) forinv (A).

\section*{ARGUMENTS}

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the factors \(L\) and \(U\) from the factorization \(A=P * L * U\) as com puted by CGETRF. On exit, if \(\mathbb{N} F O=0\), the inverse of the originalm atrix \(A\).

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, N)\).
\(\mathbb{P} \mathbb{I V O T}\) (input)
The pivotindiges from CGETRF ; for \(1<=i<=N\), row \(i\) of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P I V O T}\) (i).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0\), then \(W\) ORK (1) retums the optim alLD W ORK .

LDW ORK (input)
The dim ension of the array \(W\) ORK. LDW ORK \(>=\) \(m\) ax \((1, N)\). Foroptim alperform ance LDW ORK \(>=N * N B\), where NB is the optim al blocksize retumed by แAENV.

If LD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK amay, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i, U(i, i)\) is exactly zero; the \(m\) atrix is singular and its inverse could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgetrs-solve a system of linearequations \(A * X=B, A * * T\)
* \(\mathrm{X}=\mathrm{B}\), or \(\mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B}\) w th a generaln toy -N m atrix A using the LU factorization com puted by C G ETRF

\section*{SYNOPSIS}

```

CHARACTER * 1 TRANSA
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGERN,NRHS,LDA,LDB,INFO
INTEGER \mathbb{PIVOT (*)}
SUBROUT\mathbb{NE ZGETRS_64(TRANSA,N,NRHS,A,LDA,\mathbb{PIVOT,B,LDB,INFO)}}\mathbf{~}\mathrm{ (IN,}
CHARACTER * 1 TRANSA
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGER*8N,NRHS,LDA,LDB,INFO
INTEGER*8 \mathbb{PIVOT (*)}

```

\section*{F95 INTERFACE}

SU BROUTINE GETRS ([TRANSA], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I V O T}, \mathrm{B},[\operatorname{LD} B]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::TRANSA
COMPLEX (8),D \(\mathbb{I}\) ENSION (:,:) ::A, B
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)

SU BROUTINE GETRS_64 ([TRANSA], \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I V O T}, B,[L D B]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::TRANSA
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : A , B
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDA,LDB, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zgetrs (char transa, intn, intnrhs, doublecom plex *a, int lda, int *ipivot, doublecom plex *b, int ldlo, int*info);
void zgetrs_64 (chartransa, long n, long nrhs, doublecom plex
*a, long lda, long *ípivot, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
zgetrs solves a system of linear equations
\(A * X=B, A * * T * X=B\), or \(A * * H * X=B\) w th \(A\) general \(N\) boy \(-N\) m atrix A using the LU factorization com puted by CGETRF.

\section*{ARGUMENTS}

TRANSA (input)
Specifies the form of the system ofequations:
\(=\mathrm{N}: A * X=B \quad\) (Notranspose)
\(=T ': A * * T * X=B \quad\) ( ranspose)
\(=C: A * * H * X=B \quad\) (C onjugate transpose)

TRANSA is defaulted to \(N\) 'forF95 \(\mathbb{I N T E R F A C E .}\)

N (input) The order of them atrix A. N >=0.

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B . NRH S >=0 .

A (input) The factors \(L\) and \(U\) from the factorization \(A=\) \(\mathrm{P} * \mathrm{~L} * \mathrm{U}\) as com puted by CGETRF .

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, N)\).

The pivotindices from CGETRF; for \(1<=\mathrm{i}<=\mathrm{N}\), row \(i\) of the \(m\) atrix \(w\) as interchanged \(w\) ith row \(\mathbb{P}\) IV OT (i).

B (input/output)
On entry, the righthand side m atrix B. On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).
\(\mathbb{I N} F O\) (output)
= 0 : successfiulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zggbak - form the rightor lefteigenvectors of a com plex
generalized eigenvalue problem \(A\) * \(x=\operatorname{lam}\) bda* \(B * x\), by backw ard
transform ation on the com puted eigenvectors of the balanced
pair ofm atrioes outputby C G GBAL

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGGBAK(JOB,SDDE,N, HO, HHI,LSCALE,RSCALE,M,V,LDV,}
\mathbb{NFO)}
CHARACTER * 1 JOB,SIDE
DOUBLE COM PLEX V (LDV,*)
\mathbb{NTEGER N, HO, HHI,M,LDV, NNFO}
DOUBLE PRECISION LSCALE (*),RSCALE (*)

```

```

    LDV,\mathbb{NFO)}
    CHARACTER * 1 JOB,S\mathbb{DE}
DOUBLE COM PLEX V (LDV,*)
\mathbb{NTEGER*8N, ILO, IHI,M,LDV,INFO}
DOUBLE PRECISION LSCALE (*),RSCALE (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE GGBAK (OB B,SDE, $\mathbb{N}], \mathbb{I} O, \mathbb{H} I, L S C A L E, R S C A L E, \mathbb{M}], V$, [LDV], [ $\mathbb{N F O}]$ )
CHARACTER (LEN=1):: JOB,$S D E$
COM PLEX (8),D $\mathbb{M}$ ENSION (: : : : : V
$\mathbb{N} T E G E R:: N, \mathbb{I} O, \mathbb{H} \mathrm{I}, \mathrm{M}, \mathrm{LDV}, \mathbb{N} F O$
REAL (8), D IM ENSION (:) ::LSCALE,RSCALE

```

SU BROUTINE GGBAK_64 (JOB,SDE, \(\mathbb{N}], \mathbb{L} O, \mathbb{H} I, L S C A L E, R S C A L E, ~ M ~], V\), [LDV], [iNFO])

CHARACTER (LEN=1):: JOB ,SDE
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::V
\(\mathbb{N} T E G E R(8):: N, \mathbb{L O}, \mathbb{H} \mathrm{I}, \mathrm{M}, \mathrm{LDV}, \mathbb{N} F \mathrm{O}\)
REAL (8), D \(\mathbb{I M}\) ENSION (:) :: LSCALE,RSCALE

\section*{C INTERFACE}
\#include <sunperfh>
void zggbak (char job, charside, intn, int ilo, int ini, double * lscale, double * rscale, intm, doublecom plex *v, int ldv, int*info);
void zggbak_64 (char j̀.b, char side, long n, long ilo, long ihi, double *lscale, double *rscale, long m, doublecom plex *v, long ldv, long *info);

\section*{PURPOSE}
zggbak form \(s\) the rightor left eigenvectors of a com plex generalized eigenvalue problem \(A\) * \(x=\) lam bda* \({ }^{*}\) \(x\), by backw ard transform ation on the com puted eigenvectors of the balanced pair ofm atrioes output.by C G G BAL .

\section*{ARGUMENTS}
\(J O B\) (input)
Specifies the type of backw ard transform ation
required:
\(=\mathrm{N}^{\prime}\) : do nothing, retum im m ediately;
\(=\mathrm{P}^{\prime}\) : do backw ard transform ation forperm utation
only;
= S': do backw ard transform ation for scaling
only;
= B ': do backw ard transform ations forboth per\(m\) utation and scaling. JO B m ustbe the sam e as the argum ent JO B supplied to C G G BAL .

SIDE (input)
= R : V V contains righteigenvectors;
\(=\mathrm{L} \cdot \mathrm{V}\) contains lefteigenvectors.
N (input) The num ber of row sof the m atrix \(\mathrm{V} . \mathrm{N}>=0\).

The integers \(\mathbb{H O}\) and \(\mathbb{H}\) I determ ined by CG GBAL. 1 \(<=\mathbb{H O}<=\mathbb{H} I<=N\), if \(N>0 ; \mathbb{H}=1\) and \(\mathbb{H} \mathrm{I}=0\), if \(\mathrm{N}=0\).

IH I (input)
The integers \(\mathbb{I L O}\) and \(\mathbb{H}\) I determ ined by CG GBAL. 1 \(<=\mathbb{L O}<=\mathbb{H} \mathrm{I}<=\mathrm{N}\), if \(\mathrm{N}>0\); \(\mathbb{H} \mathrm{O}=1\) and \(\mathbb{H} \mathrm{I}=0\), if \(\mathrm{N}=0\).

LSCALE (input)
D etails of the perm utations and/or scaling factors applied to the left side of \(A\) and \(B\), as retumed by CGGBAL .
RSCALE (input)
D etails of the perm utations and/or scaling factors applied to the right side of \(A\) and \(B\), as retumed by CGGBAL .

M (input) The num ber of colum ns of the matrix \(\mathrm{V} . \mathrm{M}>=0\).
V (input/output)
O \(n\) entry, the \(m\) atrix of right or lefteigenvectors to be transform ed, as retumed by CTGEVC. On exit, \(V\) is overw ritten by the transform ed eigenvectors.

LDV (input)
The leading \(d m\) ension of the \(m\) atrix \(V\). LDV >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0: successfulexit.
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvahue.

\section*{FURTHER DETAILS}

See R . .W ard, B alancing the generalized eigenvalue problem, SIAM J.Sci.Stat.C omp. 2 (1981),141-152.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zggbal-balance a pair of general com plex \(m\) atrices \((A, B)\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGGBAL (JOB,N,A,LDA,B,LDB,ILO,\mathbb{H I,LSCALE,RSCALE,}}\mathbf{N},\textrm{L},\textrm{L}
W ORK,INFO)
CHARACTER * 1 Job
DOUBLE COM PLEX A (LDA,*),B (LDB,*)

```

```

DOUBLE PRECISION LSCALE (*),RSCALE (*),W ORK (*)

```

```

    RSCALE,WORK,INFO)
    CHARACTER * 1 Job
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
\mathbb{NTEGER*8N,LDA,LDB,}\mathbb{NO},\mathbb{H}I,\mathbb{NFO}
DOUBLE PRECISION LSCALE (*),RSCALE (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GGBAL (JOB, \(\mathbb{N}\) ], A, [LDA ], B, [LD B], \(\mathbb{I} O, \mathbb{H} I, L S C A L E\), RSCALE, [W ORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1):: JOB
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A,B
\(\mathbb{N}\) TEGER :: N, LD A, LD B, \(\mathbb{H} O, \mathbb{H} \mathrm{I}, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::LSCALE,RSCALE,W ORK
SU BROUTINE GGBAL_64 (JOB, \(\mathbb{N}], A,[L D A], B,[L D B], \mathbb{L} O, \mathbb{H} I, L S C A L E\), RSCALE, [W ORK], [ \(\mathbb{N} F O]\) )

CHARACTER ( \(L E N=1\) ) :: JOB
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : A , B
\(\mathbb{N}\) TEGER (8) :: \(N\), LD A ,LD B , \(\mathbb{L} O, \mathbb{H} I, \mathbb{N}\) FO
REAL (8), D \(\mathbb{M} E N S I O N(:):: L S C A L E, R S C A L E, W\) ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zggbal(char jं.b, int n, doublecom plex *a, int lda, doublecom plex *b, int ldlo, int*ilo, int*ihi, double *lscale, double *rscale, int *info);
void zggbal 64 (char jंb, long n, doublecom plex *a, long lda, doublecom plex *b, long ldlo, long *ilo, long *ihi, double *lscale, double *rscale, long *info);

\section*{PURPOSE}
zggbalbalances a pair of general com plex matrices ( \(A, B\) ). This involves, first, perm uting \(A\) and \(B\) by sim ilarity transform ations to isolate eigenvalues in the first 1 to IIO \$-\$1 and last \(\mathbb{H}\) I+1 to \(N\) elem ents on the diagonal; and second, applying a diagonal sim ilarity transform ation to row \(s\) and colum ns \(\mathbb{H} O\) to \(\mathbb{H}\) Ito \(m\) ake the row \(s\) and colum ns as close in norm as possible. B oth steps are optional.

B alancing \(m\) ay reduce the 1-norm of the \(m\) atrices, and im prove the accuracy of the com puted eigenvalues and/oreigenvectors in the generalized eigenvalue problem \(A{ }^{*} \mathrm{x}=\operatorname{lam} . \mathrm{bda}{ }^{*} \mathrm{~B}{ }^{*} \mathrm{x}\).

\section*{ARGUMENTS}

JO B (input)
Specifies the operations to be perform ed on A and
B :
\(=\mathrm{N}^{\prime}\) : none: simply set \(\mathbb{H} O=1, \mathbb{H} I=\mathrm{N}\), LSCALE \((\mathbb{I})=1.0\) and RSCALE \((\mathbb{I})=1.0\) fori=1,...N ;
\(=P^{\prime}\) : perm ute only;
\(=S\) ': scale only;
\(=B:\) both perm ute and scale.

N (input) The order of them atriges A and \(\mathrm{B} . \mathrm{N}>=0\).

A (input/output)
On entry, the input \(m\) atrix A. On exit, A is overw rilten by the balanced \(m\) atrix. If \(\mathrm{OB}=\mathrm{N}^{\prime}\),

A is not referenced.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).
\(B\) (input) On entry, the input \(m\) atrix \(B\). On exit, \(B\) is overw rilten by the balanced \(m\) atrix. If \(J 0 B=N\) ', \(B\) is not referenced.

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

ㅍO (output)
IOO and \(\mathbb{H}\) Iare set to integers such thaton exit
\(A(i, j)=0\) and \(B(i, 1)=0\) if \(i>\) jand \(j=\) \(1, \ldots\), ILO O -1 or \(i=\mathbb{H}\) I \(+1, \ldots, N\). If \(J O B=N\) ' or \(S^{\prime}, \mathrm{HO}=1\) and \(\mathbb{H} \mathrm{I}=\mathrm{N}\).

IH I (output)
IIO and \(\mathbb{H}\) I are set to integers such that on exit
\(A(i, j)=0\) and \(B(i, 7)=0\) if \(i>j a n d i=\)


LSCALE (input)
D etails of the perm utations and scaling factors applied to the left side of \(A\) and \(B\). IfP \((\mathcal{J})\) is the index of the row interchanged \(w\) ith row \(j\) and D ( \(j\) ) is the scaling factor applied to row \(j\) then LSCALE ( \()=\mathrm{P}(\mathcal{i}\) ) for \(J=1, \ldots\), ILO-1 \(=\mathrm{D}(\mathcal{j})\) for \(J=\mathbb{L O}, \ldots, \mathbb{H} I=P(\mathcal{O}) \quad\) for \(J=\mathbb{H} I+1, \ldots, N\). The order in which the interchanges are m ade is N to \(\mathbb{H} \mathrm{I}+1\), then 1 to \(\mathbb{H} O-1\).

RSCALE (input)
D etails of the perm utations and scaling factors applied to the right side of \(A\) and \(B\). If \(P(i)\) is the index of the colum \(n\) interchanged \(w\) ith colum \(n\) \(j\) and \(D(j)\) is the scaling factorapplied to column \(j\) then RSCALE \((\mathcal{j})=P(\mathcal{j})\) for \(J=\)
 for \(J=\mathbb{H} I+1, \ldots, N\). The order in which the interchanges are m ade is N to \(\mathrm{IH} \mathrm{I}+1\), then 1 to [ H - 1 .

W ORK (w orkspace)
dim ension (6*N )
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum enthad an illegalvalue.

\section*{FURTHER DETAILS}

See R C.W A RD , B alancing the generalized eigenvalue problem, SIAM J.Sci.Stat.Comp. 2 (1981),141-152.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgges -com pute fora pairof N boy -N com plex nonsymm etric \(m\) atrices ( \(A, B\) ), the generalized eigenvalues, the generalized com plex Schur form ( \(\mathrm{S}, \mathrm{T}\) ), and optionally leftand/or right Schurvectors (VSL and V SR)

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZGGES (OOBVSL,JO BV SR,SORT,DELZTG,N,A,LDA,B,LDB,} SD $\mathbb{I}$, ALPHA, BETA, VSL,LDVSL, VSR,LDVSR,W ORK,LW ORK,RW ORK, BW ORK, $\mathbb{N} F O$ )

```
```

CHARACTER * 1 JOBVSL, JOBVSR,SORT
DOUBLE COM PLEX A (LDA,*), B (LDB,*), ALPHA (*), BETA (*),
VSL (LDVSL,*),VSR (LDVSR,*),W ORK (*)
INTEGERN,LDA,LDB,SD IM,LDVSL,LDVSR,LW ORK,\mathbb{NFO}
LOGICALDELZTG
LOG ICAL BW ORK (*)
DOUBLE PRECISION RW ORK (*)

```
SU BROUTINE ZGGES_64 (OOBVSL, OOBVSR,SORT,DELZTG,N,A,LDA,B,LDB,
    SD \(\mathbb{I M}, A L P H A, B E T A, V S L, L D V S L, V S R, L D V S R, W\) ORK,LW ORK,RW ORK,
    BW ORK, \(\mathbb{N} F O\) )
CHARACTER * 1 JOBVSL, JOBVSR, SORT
DOUBLE COM PLEX A (LDA, *), B (LDB,*), ALPHA (*), BETA (*),
VSL (LDVSL, \(\left.{ }^{\star}\right), V \mathrm{SR}(\mathrm{LDV} \mathrm{SR}, \star), \mathrm{W}\) ORK ( \(\left.{ }^{( }\right)\)
\(\mathbb{N} T E G E R * 8 N, L D A, L D B, S D \mathbb{M}, L D V S L, L D V S R, L W O R K, \mathbb{N} F O\)
LOG ICAL*8DELZTG
LO G ICAL*8BW ORK (*)
DOUBLE PRECISION RW ORK (*)

SUBROUTNE GGES (JOBVSL, JOBVSR, SORT, DELZTG], \(\mathbb{N}], A,[L D A], B,[L D B]\), SD \(\mathbb{I}\), ALPHA, BETA, VSL, [LDVSL],VSR, [LDVSR], [W ORK], [LW ORK], [RW ORK], [BW ORK], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) :: JO BV SL, JO BV SR, SO RT
COM PLEX (8),D IM ENSION (:) ::ALPHA,BETA,W ORK
COM PLEX (8), D IM ENSION (: : : : : A, B,VSL,VSR
\(\mathbb{N} T E G E R:: N, L D A, L D B, S D \mathbb{I M}, L D V S L, L D V S R, L W O R K, \mathbb{N} F O\)
LOGICAL ::DELZTG
LO G ICAL,D IM ENSION (:) ::BW ORK
REAL (8),D \(\mathbb{I M}\) ENSION (:) ::RW ORK
SU BROUTINE G GES_64 (JO BV SL, JO BV SR , SORT, DELZTG ], \(\mathbb{N}\) ], A, [LDA ], B, \([L D B], S D \mathbb{I}, A L P H A, B E T A, V S L,[L D V S L], V S R,[L D V S R],[W O R K]\), [LW ORK], RW ORK ], [BW ORK], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) :: JOBVSL, JOBVSR , SO RT
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::ALPHA,BETA,W ORK
COM PLEX (8), D IM ENSION (:,:) ::A, B,VSL,VSR
\(\mathbb{N} T E G E R(8):: N, L D A, L D B, S D \mathbb{M}, L D V S L, L D V S R, L W O R K, \mathbb{N} F O\)
LOG ICAL (8) ::DELZTG
LOG ICAL (8), D IM ENSIO N (:) ::BW ORK
REAL (8),D \(\mathbb{M}\) ENSION (:) ::RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zgges (char jobvsl, char jobvsr, char sort, int(*delztg) (doublecom plex,doublecom plex), intn, doublecom plex *a, int lda, doublecom plex *b, int ldb, int *sdim , doublecom plex *alpha, doublecom plex *beta, doublecom plex *vsl, int ldvsl, doublecom plex *vsr, int ldvsr, int *info);
void zgges_64 (char jobvsl, char jobvssr, char sort, long (*delztg) (doublecom plex,doublecom plex), long n, doublecom plex *a, long lda, doublecom plex *b, long ldb, long *sdim, doublecom plex *alpha, doublecom plex *beta, doublecom plex *vsl, long ldvsl, doublecom plex *vsr, long ldvss, long *info);

\section*{PURPOSE}
zgges com putes for a pair of N -by-N com plex nonsym \(m\) etric \(m\) atrices ( \(A, B\) ), the generalized eigenvalues, the generalized com plex Schur form (S, T), and optionally left and/or right Schur vectors (NSL and VSR). This gives the generalized Schur factorization
\[
\left.\left.(A, B)=(N S L) * S^{\star}(N S R) * * H,(N S L) * T * N S R\right) * * H\right)
\]
where ( NSR\()^{\star *_{H}}\) is the conjugate-transpose of VR .

Optionally, it also orders the eigenvalues so that a selected cluster of eigenvalues appears in the leading diagonalblocks of the upper triangularm atrix \(S\) and the upper triangularm atrix T. The leading colum ns of V SL and V SR then form an unitary basis for the comesponding left and right eigenspaces (deflating subspaces).
(If only the generalized eigenvalues are needed, use the driverC G G EV instead, which is faster.)

A generalized eigenvalue for a pairofm atrices \((A, B)\) is a scalar w or a ratio alpha/beta \(=\mathrm{w}\), such that \(A-\mathrm{w} * \mathrm{~B}\) is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation for beta=0, and even forboth being zero.

A pairofm atrices \((\mathrm{S}, \mathrm{T})\) is in generalized com plex Schur form if S and T are uppertriangularand, in addition, the diagonalelem ents of \(T\) are non-negative realnum bers.

\section*{ARGUMENTS}

JO BV SL (input)
= N ': do notcom pute the leftSchurvectors;
\(=\mathrm{V}\) : com pute the leftSchurvectors.

JO BV SR (input)
\(=\mathrm{N}\) ': do notcom pute the rightSchurvectors;
\(=\mathrm{V}\) ': com pute the rightSchurvectors.
SORT (input)
Specifies w hether ornot to order the eigenvalues on the diagonal of the generalized Schur form . =
N ': Eigenvalues are not ordered;
\(=S\) ': Eigenvalues are ordered (see D ELZTG).

DELZTG (input)
DELZTG mustbe declared EXTERNAL in the calling subroutine. If SORT \(=N\) ',DELZTG is notreferenced. IfSORT = S',DELZTG is used to select eigenvalues to sort to the top leftof the Schur form. A n eigenvalue A LPHA ( ) ( BETA ( \()\) ) is selected ifDELZTG (ALPHA ( \()\), BETA ( 7 ) is true.
\(N\) ote that a selected com plex eigenvalue \(m\) ay no
longer satisfy DELZTG (ALPHA ( \(\mathbf{j}\), BETA ( \(\boldsymbol{\jmath}\) ) = .TRUE.
afterordering, since ordering \(m\) ay change the value of com plex eigenvalues (especially if the eigenvalue is ill-conditioned), in this case \(\mathbb{N}\) FO is set to \(\mathrm{N}+2\) (See \(\mathbb{N}\) FO below ).

N (input) The order of the m atrices A, B, V SL, and V SR. N \(>=0\).

A (input/output)
O \(n\) entry, the firstof the pair of \(m\) atrices. On
exit, A has been overw ritten by its generalized Schur form \(S\).
LD A (input)
The leading dim ension ofA . LD A \(>=\max (1, \mathbb{N})\).
B (input/output)
O n entry, the second of the pairofm atrices. On exit, B has been overw ritten by its generalized Schur form T.

LD B (input)
The leading dim ension ofB. LD B \(>=m\) ax \((1, N)\).

SD \(\mathbb{I M}\) (output)
If \(S O R T=N^{\prime}, S D \mathbb{I M}=0\). IfSORT \(=S^{\prime}, S D \mathbb{M}=\) num ber of eigenvalues (aftersorting) forw hich DELZTG is true.

A LPHA (output)
On exit, A LPHA ( \(\mathfrak{j}\) ) BETA ( \(\mathcal{j}\) ) \(\mathfrak{j} 1, \ldots, N\), w illbe the generalized eigenvalues. A LPHA ( \(\mathcal{\jmath}), 1, \ldots, N\) and BETA ( \(\mathcal{i}, \dot{于} 1, \ldots, N\) are the diagonals of the com plex Schur form ( \(A, B\) ) output by CGGES. The BETA ( 7 ) w illbe non-negative real.

N ote: the quotients A LPHA ( \()\) ) BETA ( ) m ay easily over- or underflow, and BETA ( \()\) ) may even be zero. Thus, the user should avoid naively com puting the ratio alpha/beta. H ow ever, A LPH A w illbe alw ays less than and usually com parable w ith norm (A) in m agnitude, and BETA alw ays less than and usually com parable w ith norm (B).

BETA (output)
See description of A LPH A.
VSL (input)
If 0 OVVSL = V',VSL willcontain the left Schur
vectors. N ot referenced if O BV SL \(=\mathrm{N}^{\prime}\).

LD V SL (input)
The leading dim ension of the \(m\) atrix V SL.LD V SL >= 1 , and if \(\mathrm{OBVSL}=V^{\prime}, \mathrm{LDVSL}>=\mathrm{N}\).

VSR (input)
If \(\mathrm{OBV} \mathrm{BR}=\mathrm{V}\) ', VSR w illcontain the right Schur vectors. N ot referenced if \(\mathrm{OBV} \mathrm{BR}=\mathrm{N}^{\prime}\).

LDVSR (input)
The leading dim ension of the \(m\) atrix \(V\) SR .LD V SR >= 1 , and if \(\mathrm{OBVSR}=\mathrm{V}\) ', LDVSR \(>=\mathrm{N}\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK ( 1 ) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the amay \(W\) ORK. LW ORK \(>=\) \(\max (1,2 \star N)\). For good penform ance, LW ORK m ustgenerally be larger.

If LW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

RW ORK (w orkspace)
dim ension \((8 \star \mathrm{~N})\)

BW ORK (w orkspace)
dim ension \((\mathbb{N}) N\) ot referenced if \(S O R T=N^{\prime}\).
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue.
\(=1, \ldots, N\) : The \(\mathrm{Q} Z\) teration failed. ( \(\mathrm{A}, \mathrm{B}\) ) are not in Schur form, butA LPHA ( 1 ) andBETA ( 17 ) should be comect for \(\ddagger \mathbb{N} \mathrm{FO}+1, \ldots, \mathrm{~N} .>\mathrm{N}:=\mathrm{N}+1\) : other than Q Z iteration failed in CHGEQ Z
\(=N+2\) : after reordering, roundoff changed values of som e complex eigenvalues so thatleading eigenvalues in the Generalized Schur form no longer satisfy DELZTG=.TRUE. This could also be caused due to scaling. \(=\mathrm{N}+3\) : reordering falied in CTGSEN .

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zggesx - com pute for a pair of N -by -N com plex nonsym m etric \(m\) atrices \((A, B)\), the generalized eigenvalues, the com plex Schurform ( \(\mathrm{S}, \mathrm{T}\) ),

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZGGESX (JOBVSL,JOBVSR,SORT,DELCTG,SENSE,N,A,LDA,B,}
LDB,SD IM ,ALPHA,BETA,VSL,LDVSL,VSR,LDVSR,RCONDE,RCONDV,
W ORK,LW ORK,RW ORK,IN ORK,L\mathbb{IW ORK,BW ORK,INFO)}

```
CHARACTER * 1 JOBVSL, JOBVSR, SORT, SENSE
DOUBLE COM PLEX A (LDA,*), B (LDB,*), ALPHA (*), BETA (*),
VSL (LDVSL, \(\left.{ }^{\star}\right), V \mathrm{SR}(\mathrm{LDVSR}, \star), \mathrm{W} O R K(*)\)
\(\mathbb{N}\) TEGER N,LDA,LDB,SD \(\mathbb{I M}, L D V S L, L D V S R, L W\) ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{I N}\) ORK (*)
LOG ICALDELCTG
LOG ICAL BW ORK (*)
DOUBLE PRECISION RCONDE (*),RCONDV (*),RW ORK (*)
SU BROUTINE ZGGESX_64 (JOBVSL, JOBVSR,SORT,DELCTG,SENSE,N,A,LDA,
        B,LDB,SD \(\mathbb{I}\), ALPHA,BETA,VSL,LDVSL,VSR,LDVSR,RCONDE,
        RCONDV,W ORK,LW ORK,RW ORK, IW ORK,LIW ORK,BW ORK, \(\mathbb{N} F O\) )
CHARACTER * 1 JOBVSL, JOBVSR, SORT, SENSE
DOUBLE COMPLEX A (LDA, \(\left.{ }^{\star}\right)\), B (LDB,*), ALPHA ( \(\left.{ }^{\star}\right)\), BETA ( \({ }^{*}\) ),
VSL (LDVSL, \(\left.{ }^{\star}\right), \mathrm{VSR}(\mathrm{LDVSR}, \star), \mathrm{W} O R K\left({ }^{( }\right)\)
\(\mathbb{N}\) TEGER*8 N,LDA, LD B, SD \(\mathbb{I M}, L D V S L, ~ L D V S R, ~ L W ~ O R K, ~ L I N ~ O R K, ~\)
\(\mathbb{N F O}\)
\(\mathbb{N}\) TEGER * \(8 \mathbb{I N}\) ORK ( \({ }^{( }\))
LOG ICAL*8 DELCTG
LOG ICAL*8BW ORK (*)
DOUBLE PRECISION RCONDE (*),RCONDV (*),RW ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE G GESX (JOBV SL, JOBVSR, SORT, [DELCTG], SENSE, \(\mathbb{N}], A,[L D A]\), \(B,[L D B], S D \mathbb{M}, A L P H A, B E T A, V S L,[L D V S L], V S R,[L D V S R], R C O N D E\), RCONDV, [W ORK], [LW ORK], [RW ORK], [ \(\mathbb{W}\) ORK], [ \(\mathbb{I W}\) ORK], [BW ORK], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) :: JOBVSL, JOBVSR, SORT, SEN SE
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::ALPHA,BETA, W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : : A , B , V SL, V SR
\(\mathbb{N} T E G E R:: N, L D A, L D B, S D \mathbb{M}, L D V S L, L D V S R, L W O R K, L \mathbb{I N} O R K\), \(\mathbb{N} \mathrm{FO}\)
\(\mathbb{N}\) TEGER,D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{I W}\) ORK
LOGICAL ::DELCTG
LOGICAL, D \(\mathbb{M}\) ENSION (:) ::BW ORK
REAL (8), D \(\mathbb{M}\) ENSION (:) ::RCONDE,RCONDV,RW ORK
SU BROU T IN E G G ESX_64 (OOBVSL, JO BV SR , SORT, [D ELCTG ], SEN SE, \(\mathbb{N}], A,[L D A]\), \(B,[L D B], S D \mathbb{M}, A L P H A, B E T A, V S L,[L D V S L], V S R,[L D V S R], R C O N D E\), RCONDV, [W ORK], [LW ORK], [RW ORK], [ \(\mathbb{W}\) ORK], [LIWORK], [BWORK], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) :: JOBVSL, JOBV SR, SORT, SEN SE
COMPLEX (8), D \(\mathbb{M}\) ENSION (:) ::ALPHA,BETA, W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: \(:\) :) : : A , B , V SL , V SR
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{LDA}, \mathrm{LDB}, \mathrm{SD} \mathbb{M}\), LDVSL, LDVSR, LW ORK,
LIW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N} O R K\)
LOGICAL (8) :: DELCTG
LOGICAL (8), D \(\mathbb{M}\) ENSION (:) ::BW ORK
REAL (8), D \(\mathbb{M}\) ENSION (:) ::RCONDE,RCONDV,RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zggesx (char jobvsl, char jobvsr, char sort, int(*delctg) (doublecom plex,doublecom plex), char sense, intn, doublecom plex *a, int lda, doublecom plex *b, int ldb, int *sdim, doublecom plex *alpha, doublecom plex *beta, doublecom plex *Vsl, int ldvsl, doublecom plex *vsr, int ldvss, double * rconde, double *rcondv, int *info);
void zggesx_64 (char jobvsl, char jobvsr, char sort, long (*delctg) (doublecom plex,doublecom plex), char sense, long n, doublecom plex *a, long lda, doublecom plex *b, long ldb, long *sdim, doublecom plex *alpha, doublecom plex *beta, doublecom plex *Vsl, long ldvsl, doublecom plex *vsr, long ldvsr, double * rconde, double *rcondv, long *info);

\section*{PURPOSE}
zggesx com putes for a pair of N -by- N com plex nonsym m etric \(m\) atrices \((A, B)\), the generalized eigenvalues, the com plex Schur form ( \(\mathrm{S}, \mathrm{T}\) ), and, optionally, the left and/or right \(m\) atrices of Schur vectors (VSL and V SR). This gives the generalized Schur factorization A, B) \(=(\mathrm{NSL}) \mathrm{S}(\mathrm{NSR}) \star \star \mathrm{H}\), ( NSL ) T ( SR ) \({ }^{* *}{ }^{*}\) )
where ( NR\()^{* *}{ }^{*}\) is the conjugate-transpose of SR .
Optionally, it also orders the eigenvalues so that a selected chuster of eigenvalues appears in the leading diagonalblocks of the uppertriangularm atrix \(S\) and the upper triangular \(m\) atrix \(T\); com putes a reciprocalcondition num ber for the average of the selected eigenvalues (RCONDE); and com putes a reciprocal condition num ber for the rightand left deflating subspaces corresponding to the selected eigenvalues (RCONDV). The leading colum ns of VSL andVSR then form an orthonorm albasis for the corresponding left and righteigenspaces (deflating subspaces).

A generalized eigenvalue for a pairofm atrices \((A, B)\) is a scalar \(w\) or a ratio alpha/beta \(=w\), such that A \(-w\) *B is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation for beta=0 or forboth being zero.

A pairofm atrices ( \(\mathrm{S}, \mathrm{T}\) ) is in generalized complex Schur form if \(T\) is upper triangularw ith non-negative diagonaland \(S\) is upper triangular.

\section*{ARGUMENTS}

JO BV SL (input)
= N ': do notcom pute the leftSchurvectors;
= V ': com pute the leftSchurvectors.

JO BV SR (input)
\(=\mathrm{N}\) ': do notcom pute the rightSchurvectors;
\(=\mathrm{V}\) ': com pute the rightSchurvectors.

SORT (input)
Specifies w hether ornot to order the eigenvalues
on the diagonal of the generalized Schur form . =
N ': Eigenvalues are not ordered;
= \(S^{\prime}\) : Eigenvalues are ordered (see D ELC TG).

DELCTG (input)
DELCTG m ustbe declared EXTERNAL in the calling subroutine. If \(\mathrm{SORT}=\mathrm{N}\) ', DELCTG is notreferenced. If SORT = S',DELCTG is used to select eigenvalues to sort to the top leftof the Schur form. N ote that a selected com plex eigenvaluem ay no longer satisfy DELCTG (ALPHA ( ) , BETA ( 7 ) ) = .TRUE.afterordering, since ordering may change the value of com plex eigenvalues (especially if the eigenvalue is ill-conditioned), in this case \(\mathbb{N}\) FO is set to \(\mathrm{N}+3\) see \(\mathbb{N}\) FO below ).

SEN SE (input)
D eterm ines which reciprocal condition num bers are com puted. \(=\mathrm{N}^{\prime}\) : N one are com puted;
\(=E^{\prime}:\) C om puted for average of selected eigenvalues only;
\(=V^{\prime}:\) C om puted forselected deflating subspaces only;
\(=B^{\prime}:\) Computed forboth. IfSENSE = E', \(V^{\prime}\) ', or B', SORT m ustequal S'.

N (input) The order of the m atrioes A , B , V SL, and V SR. N \(>=0\).

A (input/output)
O \(n\) entry, the firstof the pair of \(m\) atrices. On
exit, A has been overw rilten by its generalized Schurform \(S\).

LDA (input)
The leading dim ension ofA. LD A \(>=\max (1, N)\).

B (input/output)
O \(n\) entry, the second of the pair ofm atrices. On exit, B has been overw ritten by its generalized Schurform \(T\).

LD B (input)
The leading dim ension ofB. LD B \(>=m\) ax \((1, N)\).

SD \(\mathbb{I M}\) (output)
If \(S O R T=N^{\prime}, S D \mathbb{M}=0\). IfSORT \(=S^{\prime}, S D \mathbb{M}=\) num ber of eigenvahues (aftersorting) forwhich DELCTG istrue.

\section*{ALPHA (output)}

On exit, ALPHA ( \()\) BETA ( 1 , \(=1, \ldots, N\), w illbe the generalized eigenvalues. ALPHA (J) and

BETA ( \(\mathcal{j}, \dot{=} 1, \ldots, \mathrm{~N}\) are the diagonals of the com plex Schur form ( \(\mathrm{S}, \mathrm{T}\) ). BETA ( \(\mathcal{j}\) ) will be nonnegative real.

N ote: the quotients A LPHA ( \(\mathcal{j}\) ) BETA ( \()\) may easily over- orunderflow, and BETA ( ) m ay even be zero. Thus, the user should avoid naively com puting the ratio alpha/beta. H ow ever, A LPH A w illbe alw ays less than and usually com parable w ith norm (A) in m agnitude, and BETA alw ays less than and usually com parable w ith nom (B).
BETA (output)
See description of A LPH A .
VSL (input)
If JO BV SL = V',V SL w illcontain the left Schur vectors. N ot referenced if \(\mathrm{JO} \mathrm{BVSL}=\mathrm{N}\).

LD VSL (input)
The leading dim ension of the \(m\) atrix VSL. LDVSL \(>=1\), and if J BVSL \(=\mathrm{V}^{\prime}, \mathrm{LDVSL}>=\mathrm{N}\).

VSR (input)
If \(J 0 B V S R=V\) ', VSR willcontain the right Schur vectors. N ot referenced if \(\mathrm{JO} \mathrm{BV} \mathrm{SR}=\mathrm{N}\) '.

LDV SR (input)
The leading dim ension of the \(m\) atrix \(V\) SR .LD V SR >= 1 , and if \(\operatorname{OOBVSR}=V\) ', LD V SR >= N .

\section*{RCONDE (output)}

IfSENSE = E'or B', RCONDE (1) and RCONDE (2) contain the reciprocalcondition num bers for the average of the selected eigenvalues. N ot referenced ifSEN SE = N 'or V'.

\section*{RCONDV (output)}

If SENSE = V 'or B', RCONDV (1) and RCONDV (2) contain the reciprocal condition num ber for the selected deflating subspaces. N ot referenced if SENSE = N'or E'.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al
LW ORK.
LW ORK (input)
The dim ension of the anay W ORK. LW ORK >= 2*N. If SENSE = E', V',or B',LW ORK >=MAX (2*N, \(2 * S D \mathbb{M} * \mathbb{N}-S D \mathbb{I})\) ).
dim ension (8*N ) Realw orkspace.
\(\mathbb{I W}\) ORK (w orkspace/output)
N otreferenced if \(\mathrm{SEN} S E=\mathrm{N}^{\prime}\). On exit, if \(\mathbb{N} F O=\) \(0, \mathbb{I W}\) ORK (1) retums the optim alL \(\mathbb{I W}\) ORK.

LIV ORK (input)
The dim ension of the aray \(W\) ORK.LIW ORK \(>=N+2\).

BW ORK (w orkspace)
dim ension \((\mathbb{N}) N\) ot referenced ifSORT \(=N^{\prime}\).

IN FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue.
\(=1, \ldots, N\) : The \(\mathrm{Q} Z\) iteration failed. ( \(\mathrm{A}, \mathrm{B}\) ) are not in Schur form , butA LPH A ( 1 ) and BETA ( 1 ) should be correct for \(\mathcal{F} \mathbb{N}\) FO \(+1, \ldots, N .>N:=N+1\) : other than Q Z iteration failed in C H G EQ Z
\(=\mathrm{N}+2\) : after reordering, roundoff changed values of som e complex eigenvalues so that leading eigenvalues in the Generalized Schur form no longer satisfy DELCTG=.TRUE . This could also be caused due to scaling. \(=\mathrm{N}+3\) : reordering failed in CTGSEN .

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zggev - com pute for a pair of N -by -N com plex nonsym \(m\) etric \(m\) atrices ( \(A, B\) ), the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGGEV (JOBVL,JOBVR,N,A,LDA,B,LDB,ALPHA,BETA,VL,}
LDVL,VR,LDVR,W ORK,LW ORK,RW ORK,INFO)
CHARACTER * 1 JOBVL, JO BVR
DOUBLE COM PLEX A (LDA,*), B (LDB,*), ALPHA (*), BETA (*),
VL (LDVL,*),VR (LDVR,*),W ORK (*)
INTEGERN,LDA,LDB,LDVL,LDVR,LW ORK,INFO
DOUBLE PRECISION RW ORK (*)
SU BROUTINE ZGGEV_64(OOBVL,JOBVR,N,A,LDA,B,LDB,A LPHA,BETA,VL,
LDVL,VR,LDVR,W ORK,LW ORK,RW ORK,\mathbb{NFO)}
CHARACTER * 1 JOBVL,JOBVR
DOUBLE COM PLEX A (LDA,*), B (LDB,*), ALPHA (*), BETA (*),
VL (LDVL,*),VR (LDVR,\star),W ORK (*)
INTEGER*8N,LDA,LDB,LDVL,LDVR,LW ORK,INFO
DOUBLE PRECISION RW ORK (*)

```

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N E}\) GGEV (JOBVL, \(\operatorname{JOBVR}, \mathbb{N}], A,[L D A], B,[L D B], A L P H A, B E T A\), VL, [LDVL],VR, [LDVR], [W ORK ], [LW ORK], RW ORK], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1) :: JOBVL, JOBVR
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::ALPHA,BETA,W ORK
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:): : A, B,VL,VR
\(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathrm{LDA}, \mathrm{LD} B, L D V L, L D V R, L W O R K, \mathbb{N} F O\)

SU BROUTINE G GEV_64 (JOBVL, JOBVR, \(\mathbb{N}], A,[L D A], B,[L D B], A L P H A\), BETA, VL, [LDVL],VR, [LDVR], \(\mathbb{W}\) ORK], [LW ORK], \([R W\) ORK ], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1) :: JOBVL, JOBVR
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::ALPHA,BETA,W ORK
COM PLEX (8), D \(\mathbb{I M}\) ENSION (:,:) :: A, B, VL, VR
\(\mathbb{N}\) TEGER (8) ::N,LDA,LDB,LDVL,LDVR,LW ORK, \(\mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zggev (char jobvl, char jobvr, intn, doublecom plex *a, int lda, doublecom plex *b, int ldb, doublecom plex *alpha, doublecom plex *beta, doublecom plex *vl, int Idvl, doublecom plex *vr, int ldvr, int *info);
void zggev_64 (char jंbvl, char jobvr, long n, doublecom plex *a, long lda, doublecom plex *b, long ldb, doublecom plex *alpha, doublecom plex *beta, doublecom plex *vl, long ldvl, doublecom plex *vr, long ldvr, long *info);

\section*{PURPOSE}
zggev com putes fora pair of \(N\)-by -N com plex nonsym \(m\) etric \(m\) atrices ( \(A, B\) ), the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors.

A generalized eigenvalue for a pairofm atrices ( \(A, B\) ) is a scalar lam bda or a ratio alpha/beta = lam bda, such thatA lam bda*B is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation for beta \(=0\), and even forboth being zero.

The rightgeneralized eigenvectorv ( \()\) ) corresponding to the generalized eigenvalue lam boda ( \(\mathcal{I}\) ) of \((A, B)\) satisfies
\[
A * v(\mathcal{j})=\operatorname{lam} \operatorname{bda}(\mathcal{I}) * B * V(\mathcal{J} .
\]

The leftgeneralized eigenvector \(u(j)\) comesponding to the generalized eigenvalues lam bda ( \()\) ) of \((A, B\) ) satisfies
\[
\mathrm{u}(\mathcal{j}) * * \mathrm{H} * \mathrm{~A}=\operatorname{lam} \operatorname{bda}(\mathcal{j}) * \mathrm{u}(\mathcal{j}) * * \mathrm{H} * \mathrm{~B}
\]
where \(u(\mathcal{j}) * * \mathrm{H}\) is the conjugate-transpose ofu ( \()\).

\section*{ARGUMENTS}

JO BVL (input)
\(=\mathrm{N}\) : : do notcom pute the leftgeneralized eigenvectors;
\(=\mathrm{V}\) : com pute the leftgeneralized eigenvectors.
\(J 0\) BVR (input)
\(=\mathrm{N}\) ': do not com pute the right generalized eigenvectors;
\(=\mathrm{V}\) : com pute the right generalized eigenvectors.

N (input) The order of the \(m\) atriges \(\mathrm{A}, \mathrm{B}, \mathrm{VL}\), and \(V \mathrm{R} . \mathrm{N}>=\) 0 .

A (input/output)
On entry, the \(m\) atrix \(A\) in the pair \((A, B)\). On exit, A has been overw rilten.

LD A (input)
The leading dim ension ofA. LD A \(>=\max (1, N)\).

B (input/output)
On entry, them atrix \(B\) in the pair \((A, B)\). On exit, B has been overw ritten.

LD B (input)
The leading dim ension ofB. LD B >=max \((1, N)\).

\section*{ALPHA (output)}

On exit, ALPHA ( 1 ) BETA ( 1 ) , \(1, \ldots, N\), w illbe the generalized eigenvahues.

N ote: the quotients A LPHA ( 1 ) BETA ( 1 ) m ay easily over- orunderflow, and BETA ( 7 ) m ay even be zero. Thus, the usershould avoid naively com puting the ratio alpha/beta. H ow ever, A LPH A w illbe alw ays less than and usually com parable w ith norm (A) in m agnitude, and BETA alw ays less than and usually com parable w th norm (B).

BETA (output)
See description of A LPH A.

VL (output)
If \(\mathrm{OBV} \mathrm{B}=\mathrm{V}\) ', the leftgeneralized eigenvectors \(u(j)\) are stored one after another in the colum ns ofVL, in the sam e order as their eigenvalues.

Each eigenvector will be scaled so the largest com ponentw illhave abs(real part) + abs(m ag. part) \(=1\). N ot referenced if \(J 0 B V L=N '\).

LDVL (input)
The leading dim ension of the \(m\) atrix \(V \mathrm{~L}\). LD V L >=1, and if \(\mathrm{JOBVL}=\mathrm{V}\) ', LDVL \(>=\mathrm{N}\).
VR (output)
If JO BVR = V', the right generalized eigenvectors
\(\mathrm{v}(\mathrm{j})\) are stored one after another in the colum ns ofVR, in the sam e order as their eigenvalues. Each eigenvector will be scaled so the largest com ponentw ill have abs(real part) + abs(m ag. part) \(=1\). N ot referenced if \(J 0 B V R=N^{\prime}\).

LDVR (input)
The leading dim ension of the \(m\) atrix \(V R\).LD VR \(>=1\), and if \(J O B V R=V\) ', LDVR \(>=N\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al
LW ORK.

\section*{LW ORK (input)}

The dim ension of the array W ORK. LW ORK >= max ( \(1,2 * \mathrm{~N}\) ). For good perform ance, LW O RK m ustgenerally be larger.

If LW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{RW ORK (w orkspace)}
dim ension ( \(8 * \mathrm{~N}\) )
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvahue.
\(=1, \ldots, N\) : The Q Z iteration failed. No eigenvec-
tors have been calculated, but A LPHA ( \()\) ) and
BETA ( \(\mathcal{j}\) ) should be comect for \(\mathcal{于} \mathbb{N} F O+1, \ldots, N . \quad>\)
\(\mathrm{N}:=\mathrm{N}+1\) : other then Q Z teration failed in
SH GEQ Z,
\(=\mathrm{N}+2\) : enror retum from STGEVC.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zggevx - com pute fora pairof \(N\)-by- N com plex nonsym m etric \(m\) atrices ( \(A, B\) ) the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZGGEVX (BALANC,JOBVL,JOBVR,SENSE,N,A,LDA,B,LDB,}
ALPHA,BETA,VL,LDVL,VR,LDVR,\mathbb{IO,}\mathbb{H}I,LSCALE,RSCALE,ABNRM,
BBNRM,RCONDE,RCONDV,W ORK,LW ORK,RW ORK,IN ORK,BW ORK,\mathbb{NFO)}

```
CHARACTER * 1 BALANC, JOBVL, JOBVR, SENSE
DOUBLE COM PLEX A (LDA,*), B (LDB,*), ALPHA (*), BETA (*),
VL (LDVL, \(\left.{ }^{\star}\right), \mathrm{VR}(\mathrm{LDVR}, \star), \mathrm{W} O R K(*)\)
\(\mathbb{N}\) TEGERN,LDA,LDB,LDVL,LDVR, \(\mathbb{I} O, \mathbb{H} I, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{I N}\) ORK (*)
LOG ICALBW ORK (*)
DOUBLE PRECISION ABNRM,BBNRM
D OUBLE PRECISION LSCALE (*), RSCALE (*),RCONDE (*),RCONDV (*),
RW ORK (*)
SU BROUTINE ZGGEVX_64 BALANC, JO BVL, JOBVR, SEN SE, N, A, LDA, B, LD B,
    A LPHA, BETA,VL,LDVL,VR,LDVR, \(\mathbb{O}, \mathbb{H} I, L S C A L E, R S C A L E, A B N R M\),
    BBNRM,RCONDE,RCONDV,WORK,LWORK,RWORK, IW ORK,BWORK, \(\mathbb{N} F O\) )

CHARACTER * 1 BALANC, JOBVL, JOBVR,SENSE
DOUBLE COMPLEX A (LDA,*), B (LDB,*), ALPHA (*), BETA (*),
VL (LDVL, , ), VR (LDVR, \(), \mathrm{W} O R K(\star)\)
\(\mathbb{N}\) TEGER*8N,LDA,LDB,LDVL,LDVR, \(\mathbb{I} O, \mathbb{H} I, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{I N}\) ORK (*)
LO G ICAL*8BW ORK (*)
DOUBLE PRECISION ABNRM,BBNRM

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N E}\) G GEVX (BALANC, JOBVL, JOBVR, SENSE, \(\mathbb{N}], A,[L D A], B,[L D B]\), A LPHA, BETA, VL, [LDVL],VR, [LDVR], \(H O, \mathbb{H} I, ~ L S C A L E, R S C A L E\), ABNRM, BBNRM,RCONDE,RCONDV, [W ORK], [LW ORK], RW ORK], [IW ORK], [BW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::BALANC, JOBVL, JOBVR, SEN SE
COM PLEX (8), D \(\mathbb{M} E N S I O N(:):: A L P H A, B E T A, W\) ORK
COM PLEX (8), D \(\mathbb{I M} \operatorname{ENSION}(:,:\) : : \(: \mathrm{A}, \mathrm{B}, \mathrm{VL}, \mathrm{VR}\)
\(\mathbb{N} T E G E R:: N, L D A, L D B, L D V L, L D V R, \mathbb{L} O, \mathbb{H} I, L W O R K, \mathbb{N F O}\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{I W}\) ORK
LOGICAL,D \(\mathbb{I M} E N S I O N(:):\) BW ORK
REAL (8) ::ABNRM,BBNRM
REAL (8), D \(\mathbb{M}\) ENSION (:) :: LSCALE, RSCALE, RCONDE, RCONDV,
RW ORK
SU BROUTINE G GEVX_64 (BALANC, OBVL, JOBVR, SENSE, \(\mathbb{N}], A,[L D A], B\), [LD B ], A LPHA, BETA , VL, [LDVL],VR, [LDVR], ILO, IH I, LSCALE, RSCALE, ABNRM, BBNRM,RCONDE,RCONDV, [WORK], [LW ORK], \(\mathbb{R W} O R K]\), [ \(\mathbb{W}\) ORK], [BW ORK], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1) ::BALANC, JOBVL, JOBVR, SEN SE
COM PLEX (8), D \(\mathbb{M} E N S I O N(:):: A L P H A, B E T A, W\) ORK
COM PLEX (8), D \(\mathbb{I M} E N S I O N(:,:):: A, B, V L, V R\)
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{LD} A, L D B, L D V L, L D V R, \mathbb{L}, \mathbb{H} \mathrm{I}, \mathrm{LW} O R K, \mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}\) (8),D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{I W}\) ORK
LOGICAL (8), D \(\mathbb{M}\) ENSION (:) ::BW ORK
REAL (8) ::ABNRM ,BBNRM
REAL (8), D \(\mathbb{M}\) ENSION (:) :: LSCALE, RSCALE, RCONDE, RCONDV, RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zggevx (charbalanc, char jobvl, char jobvr, char sense, int \(n\), doublecom plex *a, int lda, doublecom plex *b, int ldb, doublecom plex *alpha, doublecom plex *beta, doublecom plex *vl, int ldvl, doublecom plex *vr, int ldvr, int *ilo, int *ihi, double *lscale, double *rscale, double *abnrm, double *bbnrm, double *roonde, double *roondv, int *info);
void zggevx_64 (charbalanc, char jंbvl, char jobvr, char sense, long n, doublecom plex *a, long lda, doublecom plex *b, long ldb, doublecom plex *alpha, doublecom plex *beta, doublecom plex *vl, long ldvl, doublecom plex *vr, long ldvr, long *ilo, long *ihi, double *lscale, double *rscale, double
*abnrm, double *bbnım, double *rconde, double
*rcondv, long *info);

\section*{PURPOSE}
zggevx com putes for a pair of N -by-N com plex nonsym m etric \(m\) atrices \((A, B)\) the generalized eigenvalues, and optionally, the leftand/or right generalized eigenvectors.

O ptionally, italso com putes a balancing transform ation to im prove the conditioning of the eigenvalues and eigenvectors ( \(\mathbb{L} O, \mathbb{H} I, L S C A L E, R S C A L E, A B N R M\), and BBNRM ), reciprocal
 calcondition num bers for the righteigenvectors (RCONDV).

A generalized eigenvalue for a pairofm atrices ( \(A, B\) ) is a scalar lam bda or a ratio alpha/beta = lam bda, such thatA lam bda*B is singular. It is usually represented as the pair (alpha,beta), as there is a reasonable interpretation for beta=0, and even forboth being zero.

The righteigenvectorv (i) comesponding to the eigenvalue lam bda \((\rightarrow)\) of \((A, B)\) satisfies
\[
A * v(\mathcal{j})=\operatorname{lam} \operatorname{bda}(\bar{j}) * B * v(\mathcal{j}) .
\]

The lefteigenvectoru ( \(\mathcal{I}\) ) corresponding to the eigenvalue lam bda ( \(\mathcal{F}\) ) of \((A, B)\) satisfies
\[
\mathrm{u}(\boldsymbol{j}) * * \mathrm{H} * \mathrm{~A}=\operatorname{lam} \operatorname{bda}(\mathcal{j}) * \mathrm{u}(\boldsymbol{j}) * * \mathrm{H} * \mathrm{~B} .
\]
where \(u(j) * * H\) is the conjugate transpose of \(u()\).

\section*{ARGUMENTS}

BALANC (input)
Specifies the balance option to be perform ed:
= N ': do notdiagonally scale orperm ute;
= \(\mathrm{P}^{\prime}\) : perm ute only;
= S ': scale only;
= B ': both perm ute and scale. C om puted reciprocal condition num bers \(w i l l\) be forthe \(m\) atrices afterperm uting and/orbalancing. Perm uting does not change condition num bers (in exactarithm etic), but.balancing does.

JO BVL (input)
= \(\mathrm{N}^{\prime}\) : do not com pute the left generalized eigenvectors;
\(=\mathrm{V}^{\prime}\) : com pute the left generalized eigenvectors.

JOBVR (input)
\(=\mathrm{N}^{\prime}\) : do not com pute the right generalized eigenvectors;
\(=\mathrm{V}\) : com pute the right generalized eigenvectors.

SENSE (input)
D eterm ines which reciprocal condition num bers are com puted. = N ': none are com puted;
= E ': com puted foreigenvalues only;
\(=\mathrm{V}\) : com puted foreigenvectors only;
= B ': com puted foreigenvalues and eigenvectors.
N (input) The order of the m atrices \(\mathrm{A}, \mathrm{B}, \mathrm{VL}\), and \(\mathrm{VR} . \mathrm{N}>=\) 0.

A (input/output)
Onentry, them atrix \(A\) in the pair \((A, B)\). On
exit, A has been overw ritten. If \(J 0 B V L=V\) 'or \(J O B V R=V\) 'orboth, then A contains the first part of the com plex Schur form of the "balanced" versions of the inputA and \(B\).

LDA (input)
The leading dim ension ofA. LD A \(>=\max (1, N)\).
B (input/output)
Onentry, them atrix \(B\) in the pair \((A, B)\). On exit, \(B\) has been overw rilten. If \(J O B V L=V\) 'or JO BVR=V 'orboth, then B contains the second part of the com plex Schur form of the "balanced" versions of the input \(A\) and \(B\).

LD B (input)
The leading dim ension ofB. LD B \(>=m\) ax \((1, N)\).
ALPHA (output)
On exit, ALPHA ( \(\rightarrow\) ) BETA \((\rightarrow\), \(\dot{=} 1, \ldots, N\), willbe the generalized eigenvalues.

N ote: the quotientA LPHA (J) BETA (j) ) may easily over- orunderflow, and BETA ( \()\) m ay even be zero. Thus, the user should avoid naively com puting the ratio A LPHA BETA. H ow ever, A LPH A w illbe alw ays less than and usually com parable w ith norm (A) in \(m\) agnitude, and BETA alw ays less than and usually com parable w ith norm (B).

BETA (output)
See description of A LPH A .

VL (output)
If \(\mathrm{OOBVL}=\mathrm{V}\) ', the leftgeneralized eigenvectors
\(u(7)\) are stored one after another in the colum ns of VL, in the same order as their eigenvalues. Each eigenvector will be scaled so the largest com ponentw illhave abs(real part) + abs(im ag. part) \(=1\). N otreferenced if \(\mathrm{OOBVL}=\mathrm{N}^{\prime}\).

LDVL (input)
The leading dim ension of the m atrix \(\mathrm{VL} . \mathrm{LD} V \mathrm{~L}>=1\), and if \(\mathrm{JOBVL}=V^{\prime}, \mathrm{LDVL}>=\mathrm{N}\).

VR (output)
If O BVR \(=\mathrm{V}\) ', the right generalized eigenvectors
V (7) are stored one after another in the colum ns ofVR, in the sam e order as their eigenvalues. Each eigenvector will be scaled so the largest com ponentw illhave abs(real part) + abs(im ag. part) \(=1\). N otreferenced if \(\mathrm{OOBVR}=\mathrm{N}^{\prime}\).

LDVR (input)
The leading dim ension of the \(m\) atrix \(V R . L D V R>=1\), and if \(J O B V R=V ', L D V R>=N\).

IIO (output)
IUO is an integervalue such thaton exitA \((i, j)=\) 0 and \(B(i, j)=0\) if \(i>\) jand \(j=1, \ldots, \Pi O-1\) or \(i\)
\(=\mathbb{H} I+1, \ldots, N\). IfBALANC=N'or \(S^{\prime}, \Pi O=1\) and \(\mathbb{H} I=N\).

IH I (output)
\(\mathbb{H} I\) is an integervalue such that on exitA \((i, j)=\) 0 and \(B(i, j)=0\) if \(i>\) jand \(j=1, \ldots\), HO-1 ori
\(=\mathbb{H} I+1, \ldots, N\). IfBALANC \(=N^{\prime}\) or \(S^{\prime}, ~ I L O=1\) and \(\mathbb{H} I=N\).

LSCALE (output)
D etails of the perm utations and scaling factors applied to the left side of A and B. IfPL ( \()\) is the index of the row interchanged \(w\) ith row \(j\) and D L ( ) is the scaling factor applied to row \(j\) then \(\operatorname{LSCALE}(7)=P L(j)\) for \(j=1, \ldots\) ILO \(-1=D L(1)\) for \(j=\mathbb{H}, \ldots, \mathbb{H} I=P L(j)\) for \(j=\mathbb{H} I+1, \ldots, N\). The order in which the interchanges are m ade is N to \(\mathbb{H} \mathrm{I}+1\), then 1 to \(\mathrm{HO}-1\).

RSCALE (output)
D etails of the perm utations and scaling factors applied to the right side of \(A\) and \(B\). IfPR ( \()\) is the index of the colum \(n\) interchanged \(w\) ith colum \(n\)
\(j\) and \(\operatorname{DR}(\mathcal{j})\) is the scaling factor applied to
collmm \(j\) then RSCALE \((j)=\operatorname{PR}(\mathcal{j})\) for \(j=\)
 for \(j=\mathbb{H} \mathrm{I}+1, \ldots, \mathrm{~N}\) The order in \(w\) hich the interchanges are \(m\) ade is \(N\) to \(\mathbb{H} \mathrm{I}+1\), then 1 to \(\mathbb{I L O}-1\). ABNRM (output) The one-norm of the balanced m atrix A .

\section*{BBNRM (output)}

The one-norm of the balanced \(m\) atrix B .

\section*{RCONDE (output)}

IfSENSE = E'or B', the reciprocal condition num bers of the selected eigenvalues, stored in consecutive elem ents of the array. If SEN \(\mathrm{SE}=\) V',RCONDE is not referenced.

RCONDV (output)
If JO B \(=\mathrm{V}\) 'or B ', the estim ated reciprocal condition numbers of the selected eigenvectors, stored in consecutive elem ents of the array. If the eigenvalues cannot be reordered to com pute RCONDV \((\mathcal{j})\),RCONDV ( \(\mathcal{j}\) ) is setto 0 ; this can only occurw hen the true value w ould be very sm allanyway. IfSENSE = E',RCONDV is not referenced. N ot referenced if \(\mathrm{JOB}=\mathrm{E}\) '.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, \mathrm{~W} O R K(1)\) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the anay \(W\) ORK. LW ORK >= m ax \(\left(1,2 \star^{*} \mathrm{~N}\right)\). If SEN \(S E=\mathrm{N}\) 'or E ', LW ORK \(>=2 \star \mathrm{~N}\). If SENSE \(=\mathrm{V}\) 'or \(\mathrm{B}^{\prime}\), LW ORK \(>=2 * \mathrm{~N} * \mathrm{~N}+2 \star \mathrm{~N}\).

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of theW ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA .

RW ORK (w orkspace)
dim ension (6*N ) Realw orkspace.

IW ORK (w orkspace)
dim ension \((\mathbb{N}+2)\) If SEN \(S E=E ', \mathbb{I N} O R K\) is notreferenced.
BW ORK (w orkspace)
dim ension ( \(N\) ) If SENSE \(=N\) ', BW ORK is not refer-
enced.
\(\mathbb{N}\) FO (output)
= 0 : successfulexit
<0: if \(\mathbb{N N}\) FO = -i, the \(i\)-th argum ent had an illegalvahue.
\(=1, \ldots, N\) : The Q Z iteration failed. N o eigenvectors have been calculated, but A LPHA ( \(j\) ) and BETA ( \(\mathcal{j}\) ) should be comect for \(\mathcal{j} \mathbb{N} F O+1, \ldots, N .>\) \(\mathrm{N}:=\mathrm{N}+1\) : other than Q Z teration failed in CHGEQZ.
\(=\mathrm{N}+2\) : error retum from CTGEVC.

\section*{FURTHER DETAILS}

Balancing am atrix pair ( \(A, B\) ) includes, first, perm uting row \(s\) and colum ns to isolate eigenvalues, second, applying diagonal sim ilarity transform ation to the row s and colum ns to m ake the row s and colum ns as close in norm as possible. The com puted reciprocal condition num bers comespond to the balanced \(m\) atrix. Perm uting row sand colum ns w ill not change the condition num bers (in exact arithm etic) but diagonal scaling w ill. For further explanation of balancing, see section 4.11.12 of LAPACK U sers'G uide.

A n approxim ate errorbound on the chordal distance betw een the i-th computed generalized eigenvalue \(w\) and the comesponding exacteigenvalue lam boda is hord (w, lam bda) <= EPS * norm (ABNRM,BBNRM) /RCONDE (I)

A \(n\) approxim ate errorbound for the angle betw een the \(i\)-th com puted eigenvectorV L (i) orVR (i) is given by PS * norm (ABNRM,BBNRM)/D \(\mathbb{F}\) (i).

For further explanation of the reciprocal condition num bers RCONDE and RCONDV, see section 4.11 ofLAPACK U sers G uide.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}

\section*{zggglm -solve a general G auss M arkov linear model (G LM)} problem

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGGGLM N,M,P,A,LDA,B,LDB,D,X,Y,W ORK,LDW ORK,}
INFO)

```
DOUBLE COM PLEXA (LDA,\(\star), B(L D B, *), D(*), X(*), Y(*), W O R K(*)\)
\(\mathbb{N} T E G E R N, M, P, L D A, L D B, L D W O R K, \mathbb{N} F O\)
SU BROUTINE ZGGGLM_64 \(\mathbb{N}, M, P, A, L D A, B, L D B, D, X, Y, W\) ORK,LDW ORK,
    \(\mathbb{N} F O\) )

\(\mathbb{N} T E G E R * 8 N, M, P, L D A, L D B, L D W O R K, \mathbb{N} F O\)

\section*{F95 INTERFACE}

SU BROUTINE GGGLM ( \(\mathbb{N}], \mathbb{M}], \mathbb{P}], A,[\operatorname{LDA}], B,[L D B], D, X, Y,[\mathbb{W}\) ORK \(]\), [LDW ORK], [ \(\mathbb{N} F O]\) )

COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::D , X,Y,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : : A, B
\(\mathbb{N} T E G E R:: N, M, P, L D A, L D B, L D W\) ORK, \(\mathbb{N} F O\)
SU BROUTINE GGGLM_64 (N) \(\mathbb{N}, \mathbb{M}],[\mathbb{P}], A,[L D A], B,[L D B], D, X, Y,[\mathbb{O}\) ORK ], [LDW ORK], [ \(\mathbb{N} F O]\) )

COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::D , X,Y,W ORK
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::A,B
\(\mathbb{N} T E G E R(8):: N, M, P, L D A, L D B, L D W O R K, \mathbb{N F O}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zggglm (intn, intm, intp, doublecom plex *a, int lda, doublecom plex *b, int ldl, doublecom plex *d, doublecom plex *x, doublecom plex *y, int *info);
void zggglm _64 (long n, long m, long p, doublecom plex *a, long lda, doublecom plex *b, long ldlo, doublecom plex *d, doublecom plex *x, doublecom plex *y, long *info);

\section*{PURPOSE}
zggglm solves a general Gauss M arkov linear model (G LM ) problem :
\(m\) inim ize \(\|y\| 2\) subject to \(d=A *^{\prime} x+B * y\)
x
\(w\) here \(A\) is an \(N\) boy \(-M m\) atrix, \(B\) is an \(N\) boy \(P m\) atrix, and \(d\) is a given N -vector. It is assum ed thatM \(<=\mathrm{N}<=\mathrm{M}+\mathrm{P}\), and
\[
\operatorname{rank}(A)=M \quad \text { and } \quad \operatorname{rank}(A B)=N
\]

U nder these assum ptions, the constrained equation is alw ays consistent, and there is a unique solution \(x\) and \(a m\) inim al 2-norm solution \(y\), w hich is obtained using a generalized Q R factorization of \(A\) and \(B\).

In particular, ifm atrix \(B\) is square nonsingular, then the problem GLM is equivalent to the follow ing w eighted linear least squares problem
\(m\) inim ize \(\left\|\operatorname{inv}(B)^{\star}\left(d-A *_{x}\right)\right\| 2\)
x
w here inv \((B)\) denotes the inverse of \(B\).

\section*{ARGUMENTS}

N (input) The num ber of row s of the m atriges A and \(\mathrm{B} . \mathrm{N}>=\) 0 .
\(M\) (input) The num ber of colum ns of the m atrix A. \(0<=M<=\) N .
\(P\) (input) The num ber of colum ns of the \(m\) atrix \(B . P>=N-M\).
A (input/output)
On entry, the N -by -M m atrix A. On exit, A is destroyed.

LD A (input)
The leading din ension of the array A. LDA >= \(\max (1, N)\).

B (input/output)
On entry, the N -by Pm atrix B. On exit, B is destroyed.
LD B (input)
The leading dim ension of the array \(\mathrm{B} . \operatorname{LD} \mathrm{B}>=\) \(\max (1, N)\).

D (input/output)
O \(n\) entry, \(D\) is the lefthand side of the G LM equation. On exit, D is destroyed.

X (output)
On exit, X and Y are the solutions of the G LM problem.

Y (output)
On exit, \(X\) and \(Y\) are the solutions of the GLM problem.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the amay W ORK. LDW ORK >= \(m a x(1, N+M+P)\). Foroptim um perform ance, LD W ORK >= \(M+m\) in \((\mathbb{N}, P)+m\) ax \(\mathbb{N}, P)^{\star} N B\), where \(N B\) is an upperbound for the optim al blocksizes forCGEQRF, CGERQF, CUNMQR and CUNMRQ.

If LD W ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of theW ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0: successfulexit.
<0: if \(\mathbb{N N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvahue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zgghrd - reduce a pair of com plex \(m\) atrices ( \(A, B\) ) to generalized upper H essenberg form using unitary transform ations, \(w\) here \(A\) is a generalm atrix and \(B\) is upper triangular

\section*{SYNOPSIS}

```

    Z,LDZ,INFO)
    ```
CHARACTER * 1 COMPQ,COMPZ
D OUBLE COM PLEX A (LDA,*), B (LDB,*), Q (LD Q ,*), Z (LD Z,*)
\(\mathbb{N}\) TEGER \(N, \mathbb{L O}, \mathbb{H} I, L D A, L D B, L D Q, L D Z, \mathbb{N} F O\)
SU BROUTINE ZGGHRD_64 (COMPQ, COMPZ,N, \(\mathbb{L O}, \mathbb{H} I, A, L D A, B, L D B, Q\),
    LD Q , Z, LD Z, \(\mathbb{N}\) FO)
CHARACTER * 1 COMPQ,COMPZ
D OUBLE COM PLEX A (LDA, *), B (LDB, \(), \mathrm{Q}(\mathrm{LD} \mathrm{Q}, \star), \mathrm{Z}(\mathrm{LD} \mathrm{Z}, \star)\)
\(\mathbb{N}\) TEGER* \(8 \mathrm{~N}, \mathbb{L} \mathrm{O}, \mathbb{H} \mathrm{I}, \mathrm{LD} A, L D B, L D Q, L D Z, \mathbb{N} F O\)

\section*{F95 INTERFACE}

SU BROUTINE GGHRD (COMPQ,COMPZ, \(\mathbb{N}], \mathbb{H}, \mathbb{H} I, A,[L D A], B,[L D B], Q\), [LDQ], Z, [LD Z], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::COM PQ,COM PZ
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) :: A, B, \(\mathrm{Q}, \mathrm{Z}\)
\(\mathbb{N} T E G E R:: N, \mathbb{H}, \mathbb{H} I, L D A, L D B, L D Q, L D Z, \mathbb{N} F O\)
SU BROUTINE GGHRD_64 (COMPQ, COMPZ, \(\mathbb{N}], \mathbb{H}, \mathbb{H} I, A,[L D A], B,[L D B]\), \(Q,[\operatorname{LD} Q], Z,[L D Z],[\mathbb{N F O}])\)

CHARACTER (LEN=1): :COMPQ,COMPZ
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::A, B, \(\mathrm{Q}, \mathrm{Z}\)
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathbb{L O}, \mathbb{H} \mathrm{I}, \mathrm{LD} A, L D B, L D Q, L D Z, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zgghrd (char com pq, charcom pz, intn, int ilo, int ini, doublecom plex *a, int lda, doublecom plex *b, int ldb, doublecom plex *q, int ldq, doublecom plex *z, int ldz, int*info);
void zgghrd_64 (charcom pq, charcom pz, long n, long ilo, long ihi, doublecom plex *a, long lda, doublecom plex *b, long ldb, doublecom plex *q, long ldq, doublecom plex *z, long ldz, long *info);

\section*{PURPOSE}
zgghrd reduces a pair of com plex \(m\) atrices \((A, B)\) to generalized upper H essenberg form using unitary transform ations, \(w\) here \(A\) is a generalm atrix and \(B\) is upper triangular: \(Q\) '* \(A * Z=H\) and \(Q ' * B * Z=T\), where \(H\) is upper \(H\) essenberg, \(T\) is upper triangular, and Q and Z are unitary, and ' m eans conjugate transpose.

The unitary \(m\) atrioes \(Q\) and \(Z\) are determ ined as products of G ivens rotations. They m ay eitherbe form ed explicitly, or they \(m\) ay be postm ultiplied into inputm atrioes Q 1 and \(\mathrm{Z1}\), so that 1 * \(A * 1^{\prime}=(Q 1 * Q) * H *(Z 1 * Z) '\)

\section*{ARGUMENTS}

COMPQ (input)
\(=\mathrm{N}\) ': do not com pute Q ;
\(=\mathrm{I}: \mathrm{Q}\) is initialized to the unit m atrix, and the unitary matrix \(Q\) is retumed; = V : Q must contain a unitary \(m\) atrix \(Q 1\) on entry, and the product \(\mathrm{Q} 1 * \mathrm{Q}\) is retumed.

COMPZ (input)
\(=\mathrm{N}\) ': do not com pute Q ;
\(=I^{\prime}: Q\) is initialized to the unit \(m\) atrix, and the unitary m atrix Q is retumed; \(=\mathrm{V}: \mathrm{Q} \mathrm{m}\) ust contain a unitary \(m\) atrix \(Q 1\) on entry, and the product \(\mathrm{Q} 1 * \mathrm{Q}\) is retumed.

N (input) The order of the m atrioes A and \(\mathrm{B} . \mathrm{N}>=0\).
ㅍO (input)
It is assum ed thatA is already upper triangular in row sand colum ns \(1: \mathbb{I H}-1\) and \(\mathbb{H} \mathrm{I}+1 \mathbb{N} . \mathbb{H O}\) and HH I are norm ally set.by a previous callto C G G BA L; otherw ise they should be set to 1 and N respec-
tively. \(1<=\mathbb{L} \mathrm{O}<=\mathbb{H} \mathrm{I}<=\mathrm{N}\), if \(\mathrm{N}>0\); \(\mathbb{H} \mathrm{O}=1\) and \(\mathbb{H} \mathrm{I}=0\), if \(\mathrm{N}=0\).

IH I (input)
See description of IIO .

A (input/output)
O n entry, the N -by N generalm atrix to be reduced. On exit, the upper triangle and the first subdiagonalofA are overw rilten with the upper H essenberg \(m\) atrix \(H\), and the rest is set to zero.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, \mathbb{N})\).

B (input/output)
\(\mathrm{O} n\) entry, the \(\mathrm{N}-\) by -N upper triangular m atrix B .
On exit, the upper triangularm atrix \(T=Q\) ' B Z .
The elem entsbelow the diagonalare setto zero.
LD B (input)
The leading dim ension of the array B . LD B >= \(\max (1, \mathbb{N})\).

Q (input/output)
If COMPQ \(=\mathrm{N}^{\prime}: \mathrm{Q}\) is notreferenced.
If \(C O M P Q=\) ' \('\) : on entry, \(Q\) need notbe set, and on exit it contains the unitary \(m\) atrix \(Q\), where \(Q\) 'is the product of the \(G\) ivens transform ations which are applied to \(A\) and \(B\) on the left. If \(C O M P Q=V\) ': on entry, \(\mathrm{Q} m\) ustcontain a unitary m atrix Q 1 , and on exit this is overw rilten by \(\mathrm{Q} 1 * \mathrm{Q}\).

LD Q (input)
The leading dim ension of the array \(Q . L D Q>=N\) if COM PQ = V 'or I'' LD Q >= 1 otherw ise.

Z (input/output)
If COMPZ=N': Z is notreferenced.
If COM PZ = 'I': on entry, Z need not.be set, and on
exit it contains the unitary \(m\) atrix \(Z\), which is
the product of the \(G\) ivens transform ations which are applied to \(A\) and \(B\) on the right. If COMPZ=V': on entry, Z must contain a unitary \(m\) atrix \(Z 1\), and on exit this is overw ritten by Z1*Z.

LD \(Z\) (input)
The leading dim ension of the array \(\mathrm{Z} . \operatorname{LD} \mathrm{Z}>=\mathrm{N}\) if \(\mathrm{COM} \mathrm{PZ}=\mathrm{V}\) 'or I'; LD Z >= 1 otherw ise.
\(\mathbb{I N} F \mathrm{O}\) (output)
= 0: successfulexit.
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvahue.

\section*{FURTHER DETAILS}

This routine reduces \(A\) to \(H\) essenberg and \(B\) to triangular form by an unblocked reduction, as described in _M atrix_C om putations_, by G olub and van Loan (Johns H opkins Press).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgglse - solve the linear equality-constrained least squares (LSE) problem

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGGLSE M,N,P,A,LDA,B,LDB,C,D,X,W ORK,LDW ORK,}
INFO)

```
D OUBLE COM PLEX A (LDA,\(\star), B(L D B, *), C(*), D(*), X(*), W O R K(*)\)
\(\mathbb{N}\) TEGER M,N,P,LDA,LDB,LDW ORK, \(\mathbb{N} F O\)
SU BROUTINE ZGGLSE_64M,N,P,A,LDA,B,LDB,C,D,X,WORK,LDWORK,
    \(\mathbb{N} F O\) )

\(\mathbb{N} T E G E R * 8 M, N, P, L D A, L D B, L D W O R K, \mathbb{N} F O\)

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E \operatorname{GGLSE}(\mathbb{M}], \mathbb{N}],[\mathbb{P}], A,[L D A], B,[L D B], C, D, X,[\mathbb{N}\) ORK], [LDW ORK], [ \(\mathbb{N} F O]\) )

COMPLEX (8),D \(\mathbb{M}\) ENSION (:) :: C, D, X,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : : A, B
\(\mathbb{N}\) TEGER :: M , N, P,LDA,LDB,LDW ORK, \(\mathbb{N} F O\)
SUBROUTINE GGLSE_64 ( \(\mathbb{M}], \mathbb{N}], \mathbb{P}], A,[\operatorname{LDA}], B,[\operatorname{LDB}], C, D, X,[\mathbb{O}\) ORK \(]\), [LDW ORK], [ \(\mathbb{N} F \mathrm{~F}]\) )

COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::C,D ,X,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A,B
\(\mathbb{N} T E G E R(8):: M, N, P, L D A, L D B, L D W O R K, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zgglse (intm, intn, intp, doublecom plex *a, int lda, doublecom plex *b, int ldl, doublecom plex *c, doublecom plex *d, doublecom plex *x, int *info);
void zgglse_64 (long m, long \(n\), long p, doublecom plex *a, long lda, doublecom plex *b, long ldb, doublecom plex *c, doublecom plex *d, doublecom plex *x, long *info);

\section*{PURPOSE}
zgglse solves the linearequally-constrained least squares (LSE ) problem :
\(m\) inim ize \(\left\|C-A *_{x}\right\| 2\) subject to \(B *_{x}=d\)
where \(A\) is an \(M\) boy \(N\) m atrix, \(B\) is a P-by \(N\) m atrix, \(c\) is a given \(M\)-vector, and \(d\) is a given \(P\)-vector. 化 is assum ed that
\(P<=N<=M+P\), and
\(\operatorname{rank}(B)=P\) and \(\operatorname{rank}((A))=N\).
( (B ) )

These conditions ensure that the LSE problem has a unique solution, w hich is obtained using a G RQ factorization of the \(m\) atrices \(B\) and \(A\).

\section*{ARGUMENTS}

M (input) The num ber of row s of the matrix A. \(\mathrm{M}>=0\).

N (input) The num ber of colum ns of the \(m\) atrioes \(A\) and \(B . N\) \(>=0\).

P (input) The num ber of row s of the m atrix B. \(0<=\mathrm{P}<=\mathrm{N}<=\) \(M+P\).

A (input/output)
On entry, the M boy- N m atrix A. On exit, A is des troyed.

LD A (input)
The leading dim ension of the aray A. LD A >=
\(\max (1, \mathrm{M})\).
B (input/output)
On entry, the \(P\)-by \(-N\) m atrix B. On exit, B is destroyed.

LD B (input)
The leading dim ension of the aray \(\mathrm{B} . \operatorname{LD} \mathrm{B}>=\) \(\max (1, P)\).

C (input/output)
On entry, C contains the right hand side vector for the leastsquares part of the LSE problem. On exit, the residual sum of squares for the solution is given by the sum of squares of elem ents \(\mathrm{N}+\mathrm{P}+1\) to \(M\) ofvector \(C\).

D (input/output)
O n entry, D contains the right hand side vector for the constrained equation. On exit, \(D\) is destroyed.

X (output)
On exit, \(X\) is the solution of the LSE problem .

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The dimension of the array W ORK. LDW ORK >= \(m a x(1, M+N+P)\). For optim um perform ance LDW ORK >= \(P+m\) in \((M, N)+m\) ax \(M, N) * N B\),w here \(N B\) is an upperbound for the optim al blocksizes forCGEQRF,CGERQF, CUNM QR and CUNMRQ.

IfLDW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit.
<0: if \(\mathbb{N N}\) FO = -i, the i-th argum ent had an illegalvahue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zggqrf - com pute a generalized \(Q \mathrm{R}\) factorization of an N -by -M \(m\) atrix \(A\) and an \(N\) by \(P m\) atrix \(B\).

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGGQRFN,M,P,A,LDA,TAUA,B,LDB,TAUB,W ORK,LW ORK,}
\mathbb{NFO )}
DOUBLE COM PLEX A (LDA,*),TAUA (*),B (LDB ,*),TAUB (*),W ORK (*)
INTEGERN,M,P,LDA,LDB,LW ORK,\mathbb{NFO}
SUBROUT\mathbb{NE ZGGQRF_64 N,M,P,A,LDA,TAUA,B,LDB,TAUB,W ORK,}
LW ORK,\mathbb{NFO)}
DOUBLE COM PLEX A (LDA,*),TAUA (*),B (LDB,*),TAUB (*),W ORK (*)
INTEGER*8N,M,P,LDA,LDB,LW ORK, INFO

```

\section*{F95 INTERFACE}

SUBROUTINE GGQRF ( \(\mathbb{N}], \mathbb{M}], \mathbb{P}], A,[L D A], T A U A, B,[L D B], T A U B,[\mathbb{N} O R K]\), [LW ORK], [ \(\mathbb{N F O}\) ])
```

COM PLEX (8),D IM ENSION (:) ::TAUA,TAUB,W ORK

```
COM PLEX (8),D \(\mathbb{M}\) ENSIDN (:,:): A, B
\(\mathbb{N} T E G E R:: N, M, P, L D A, L D B, L W O R K, \mathbb{N} F O\)

SU BROUTINEGGQRF_64 ( \(\mathbb{N}], \mathbb{M}],[\mathbb{P}], A,[L D A], T A U A, B,[L D B], T A U B\), [W ORK], [LW ORK], [NFO])

COM PLEX (8),D IM ENSION (:) ::TAUA,TAUB,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : : A , B
\(\mathbb{N}\) TEGER (8) ::N,M,P,LDA,LDB,LW ORK, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zggqrf(intn, intm, intp, doublecom plex *a, int lda, doublecom plex *taua, doublecom plex *b, int ladb, doublecom plex *taub, int *info);
void zggqnf_64 (long \(n\), long m, long p, doublecom plex *a, long lda, doublecom plex *taua, doublecom plex *b, long lab, doublecom plex *taub, long *info);

\section*{PURPOSE}
zggqrf com putes a generalized \(Q R\) factorization of an \(N-b y-M\) \(m\) atrix \(A\) and an \(N\) by \(P m\) atrix \(B\) :
\[
A=Q * R, \quad B=Q * T * Z,
\]
where \(Q\) is an \(N\) boy \(N\) unitary \(m\) atrix, \(Z\) is a \(P\)-by \(P\) unitary \(m\) atrix, and \(R\) and \(T\) assum e one of the form \(s\) :
if \(N>=M, R=(R 11) M\), orif \(N<M, R=(R 11 R 12\)
) N ,
( 0 ) N M
N \(\mathrm{M}-\mathrm{N}\)
M
where R11 is uppertriangular, and
if \(\mathrm{N}<=\mathrm{P}, \mathrm{T}=(0 \mathrm{~T} 12) \mathrm{N}\), orifN \(>\mathrm{P}, \mathrm{T}=(\mathrm{T} 11\) )
\(\mathrm{N}-\mathrm{P}\),
\[
P \oplus N \quad N
\]
(T21) P
P
where T12 or T 21 is uppertriangular.
In particular, if \(B\) is square and nonsingular, the \(G Q R\) factorization of \(A\) and \(B\) im plicitly gives the \(Q R\) factorization of inv (B)*A :
\[
\operatorname{inv}(B) \star A=Z *(\operatorname{inv}(T) \star R)
\]
\(w\) here inv ( \(B\) ) denotes the inverse of the \(m\) atrix \(B\), and \(Z^{\prime}\) denotes the conjugate transpose ofm atrix \(Z\).

\section*{ARGUMENTS}

N (input) The num ber of row s of the m atrices A and B . \(\mathrm{N}>=\)
0.

M (input) The num ber of colum ns of the m atrix A. M >=0.
\(P\) (input) The num ber of colum ns of the \(m\) atrix \(B . P>=0\).

A (input/output)
On entry, the \(N\) boy \(M \mathrm{~m}\) atrix A. On exit, the ele\(m\) ents on and above the diagonal of the amay contain the \(m\) in \(\mathbb{N}, M\) )-by \(-M\) uppertrapezoidalm atrix \(R\) \((R\) is upper triangular if \(N>=M)\); the elem ents below the diagonal, w th the array TA U A , represent the unitary \(m\) atrix \(Q\) as a productofm in \((\mathbb{N}, M)\) ele\(m\) entary reflectors (see FurtherD etails).

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

TAUA (output)
The scalar factors of the elem entary reflectors \(w\) hich represent the unitary \(m\) atrix \(Q\) (see Further D etails).

B (input/output)
On entry, the N -by P m atrix B. On exit, if \(\mathrm{N}<=\) P , the upper triangle of the subaray \(B(\mathcal{N}, \mathrm{P}-\) \(\mathrm{N}+1: \mathrm{P}\) ) contains the N -by N uppertriangularm atrix
\(T\); ifN > P, the elem ents on and above the \((\mathbb{N} P)-\) th subdiagonal contain the N Hoy -P upper trapezoidal m atrix T ; the rem aining elem ents, w th the array TA UB, represent the unitary matrix Z as a product of elem entary reflectors (see Further D etails).

LD B (input)
The leading dim ension of the aray \(\mathrm{B} . \mathrm{LD} \mathrm{B}>=\) \(\max (1, N)\).

TAUB (output)
The scalar factors of the elem entary reflectors which represent the unitary \(m\) atrix \(Z\) (see Further D etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the amay W ORK. LW ORK >=
\(\max (1, N, M, P)\). For optim um perform ance LW ORK \(>=\) \(\max (\mathbb{N}, \mathbb{M}, P){ }^{m} \max (\mathbb{N} 1, N B 2, N B 3)\), where \(N B 1\) is the optim al blocksize forthe QR factorization of an N toy -M m atrix, N B 2 is the optim al blocksize for the RQ factorization of an \(N\) boy \(P m\) atrix, and NB3 is the optim alblocksize for a call of CUNM QR .

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue.

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(1) H(2) \ldots H(k), w \text { here } k=m \text { in }(n, m) .
\]

Each H (i) has the form
\[
H(i)=I-\operatorname{tana}{ }^{*} v^{*} v^{\prime}
\]
\(w\) here taua is a com plex scalar, and \(v\) is a com plex vector \(w\) ith \(v(1: i-1)=0\) and \(v(i)=1 ; v(i+1 m)\) is stored on exit in A \((i+1 m, i)\), and taua in TAUA (i).
To form Q explicitly, use LA PACK subroutine CUNGQR. To use Q to update another matrix, use LAPACK subroutine CUNM QR.

Them atrix \(Z\) is represented as a product of elem entary reflectors
\(Z=H(1) H(2) \ldots H(k), w\) here \(k=m\) in \((n, p)\).
Each H (i) has the form
\[
H(i)=I-\operatorname{taub} * v^{*} v^{\prime}
\]
\(w\) here taub is a com plex scalar, and \(v\) is a com plex vector \(w\) ith \(v(p-k+i+1 \mathrm{p})=0\) and \(v(p-k+i)=1 ; v(1 \mathrm{p}-k+i-1)\) is stored on exitin B ( \(n-k+i, 1\) p \(k+i-1\) ), and taub in TA U B (i). To form Z explicitly, use LAPACK subroutine CUNGRQ. To use Z to update another matrix, use LAPACK subroutine CUNMRQ.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zggrqf-com pute a generalized \(R Q\) factorization of an \(M\) by \(-N\) \(m\) atrix \(A\) and \(a P-b y-N\) m atrix B

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGGRQFM,P,N,A,LDA,TAUA,B,LDB,TAUB,W ORK,LW ORK,}
\mathbb{NFO)}
DOUBLE COM PLEX A (LDA,*),TAUA (*),B (LDB ,*),TAUB (*),W ORK (*)
INTEGERM,P,N,LDA,LDB,LW ORK, INFO
SUBROUT\mathbb{NE ZGGRQF_64M,P,N,A,LDA,TAUA,B,LDB,TAUB,W ORK,}
LW ORK,\mathbb{NFO)}
DOUBLE COM PLEX A (LDA,*),TAUA (*),B (LDB,*),TAUB (*),W ORK (*)
INTEGER*8M,P,N,LDA,LDB,LW ORK, IN FO

```

\section*{F95 INTERFACE}

SU BROUTINE GGRQF ( \(\mathbb{M}],[\mathbb{P}], \mathbb{N}], A,[L D A], T A U A, B,[L D B], T A U B,[\mathbb{O} O K]\), [LW ORK], [ \(\mathbb{N F O}\) ])
```

COM PLEX (8),D IM ENSION (:) ::TAUA,TAUB,W ORK

```
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A,B
\(\mathbb{N} T E G E R:: M, P, N, L D A, L D B, L W O R K, \mathbb{N} F O\)

SU BROUTINE GGRQF_64 (M) \(\mathbb{M}, \mathbb{P}], \mathbb{N}], A,[L D A], T A U A, B,[L D B], T A U B\), [W ORK ], [LW ORK], [ \(\mathbb{N F O}\) ])

COM PLEX (8),D IM ENSION (:) ::TAUA,TAUB,W ORK
COMPLEX (8), D \(\mathbb{M}\) ENSION (: : : ) :: A, B
\(\mathbb{N}\) TEGER (8) ::M, P, N,LDA,LDB,LW ORK, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include < sunperfh>
void zggrqf(intm, intp, intn, doublecom plex *a, int lda, doublecom plex *taua, doublecom plex *b, int ladb, doublecom plex *taub, int*info);
void zggrqf_64 (long m, long p, long n, doublecom plex *a, long lda, doublecom plex *taua, doublecom plex *b, long ldb, doublecom plex *taub, long *info);

\section*{PURPOSE}
zggrqf com putes a generalized RQ factorization of an M by -N \(m\) atrix \(A\) and \(a P-b y-N m\) atrix \(B\) :
\[
A=R * Q, \quad B=Z * T * Q,
\]
where \(Q\) is an \(N\) boy \(N\) unitary \(m\) atrix, \(Z\) is a \(P\)-by \(P\) unitary \(m\) atrix, and \(R\) and \(T\) assum e one of the form \(s\) :
```

ifM <= N, R = (0 R12)M, orifM > N, R = ( R11 )

```
\(\mathrm{M}-\mathrm{N}\),
\[
\mathrm{N}-\mathrm{M} \mathrm{M} \quad \text { (R21)N }
\]

N
where R12 orR21 is upper triangular, and
if \(\mathrm{P}>=\mathrm{N}, \mathrm{T}=(\mathrm{T} 11) \mathrm{N}\), orifP \(<\mathrm{N}, \mathrm{T}=(\mathrm{T} 11 \mathrm{~T} 12\)
) P,
( 0 ) P-N
P NP
N
where T11 is upper triangular.
In particular, if \(B\) is square and nonsingular, the \(G R Q\) factorization ofA and \(B\) im plicitly gives the \(R Q\) factorization ofA *inv (B):
\[
A * \operatorname{inv}(B)=(R * \operatorname{inv}(I)) * Z^{\prime}
\]
where inv ( \(B\) ) denotes the inverse of the \(m\) atrix \(B\), and \(Z^{\prime}\) denotes the conjugate transpose of the \(m\) atrix \(Z\).

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
\(P\) (input) The num ber of row sof the \(m\) atrix \(B . P>=0\).
N (input) The num ber of colum ns of the \(m\) atrioes \(A\) and \(B . N\) \(>=0\).

A (input/output)
On entry, the M -by -N m atrix A. On exit, if M <=
N , the upper triangle of the subaray A ( \(1 \mathrm{M}, \mathrm{N}\) -
\(\mathrm{M}+1 \mathrm{~N}\) ) contains the M -by -M uppertriangularm atrix
\(R\); if \(M>N\), the elem ents on and above the \(M-N\) )th subdiagonal contain the \(\mathrm{M}-\) by -N upper trapezoidal \(m\) atrix \(R\); the rem aining elem ents, \(w\) ith the amay TAUA, represent the unitary \(m\) atrix \(Q\) as a product of elem entary reflectors (see Further D etails).
LDA (input)
The leading dim ension of the array A. LDA >= max (1, M).

TAUA (output)
The scalar factors of the elem entary reflectors which represent the unitary \(m\) atrix \(Q\) (see Further D etails).

B (input/output)
On entry, the \(P-b y-N m\) atrix B. On exit, the ele\(m\) ents on and above the diagonal of the array contain the \(m\) in \((\mathbb{P}, N)-b y-N\) uppertrapezoidalm atrix \(T\) ( \(T\) is upper triangular if \(P>=N\) ); the elem ents below the diagonal, w ith the amray TA U B, represent the unitary \(m\) atrix \(Z\) as a productofelem entary reflectors (see FurtherD etails).

LD B (input)
The leading dim ension of the array \(\mathrm{B} . \operatorname{LD} \mathrm{B}>=\) \(\max (1, P)\).

TAUB (output)
The scalar factors of the elem entary reflectors which represent the unitary \(m\) atrix \(Z\) (see Further D etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the array W ORK. LW ORK >= \(m a x(1, N, M, \mathbb{P})\). For optim um perform ance LW \(O R K>=\)
\(\max (\mathbb{N}, \mathbb{M}, \mathbb{P})_{m a x}(N B 1, N B 2, N B 3)\), where NB1 is the optim al blocksize forthe \(R Q\) factorization of an M by \(-\mathrm{N} m\) atrix, NB2 is the optim al blocksize for the \(Q R\) factorization of \(P\)-by \(-N m\) atrix, and NB3 is the optim alblocksize for a callof \(C U N M R Q\).

If LW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
< 0: if \(\mathbb{N F O}=-i\), the \(i\)-th argum enthad an illegal value.

\section*{FURTHER DETAILS}

Them atrix \(Q\) is represented as a product of elem entary reflectors
\(Q=H(1) H(2) \ldots H(k)\), where \(k=m\) in \((m, n)\).
Each H (i) has the form
\[
\mathrm{H}(\mathrm{i})=\mathrm{I}-\operatorname{tana} * \mathrm{~V}^{*} \mathrm{~V}^{\prime}
\]
where taua is a com plex scalar, and \(v\) is a complex vector \(w\) ith \(v(n-k+i+1 n)=0\) and \(v(n-k+i)=1 ; v(1 n-k+i-1)\) is stored on exitin A ( \(m-k+i, 1 n-k+i-1\) ), and taua in TAUA (i). To form \(Q\) explicitly, use LAPACK subroutine CUNGRQ. To use \(Q\) to update another \(m\) atrix, use LAPACK subroutine CUNMRQ.

Them atrix \(Z\) is represented as a product of elem entary reflectors
\[
Z=H(1) H(2) \ldots H(k), w \text { here } k=m \text { in }(p, n) .
\]

Each H (i) has the form
\[
\mathrm{H}(\mathrm{i})=\mathrm{I}-\operatorname{tanb} * \mathrm{~V}^{*} \mathrm{v}^{\prime}
\]
where taub is a com plex scalar, and \(v\) is a com plex vector w ith \(\mathrm{v}(1: i-1)=0\) and \(v(i)=1 ; \mathrm{v}(i+1: p)\) is stored on exit in B (i+1 \(\mathrm{P}, \mathrm{i}\) ), and taub in TA UB (i).
To form \(Z\) explicitly, use LAPACK subroutine CUNGQR. To use \(Z\) to update another matrix, use LAPACK subroutine CUNMQR.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zggsvd -com pute the generalized singular value decom position (G SVD) of an M -by-N com plex m atrix A and P-by-N com plex \(m\) atrix B

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZGGSVD (JOBU, JOBV,JOBQ,M,N,P,K,L,A,LDA,B,LDB,}
ALPHA,BETA,U,LDU,V,LDV,Q,LDQ,W ORK,W ORK2,IN ORK 3, INFO)
CHARACTER * 1 JOBU,NOBV,NO BQ
DOUBLE COM PLEX A (LDA,*), B (LDB,*), U (LDU,*), V (LDV ,*),
Q (LDQ,*),W ORK (*)
\mathbb{NTEGERM,N,P,K,L,LDA,LDB,LDU,LDV,LDQ,INFO}
\mathbb{NTEGER IN ORK3(*)}
DOUBLE PRECISION ALPHA (*),BETA (*),W ORK2 (*)
SU BROUTINE ZGGSVD_64 (JOBU,JO BV ,JOBQ ,M ,N,P,K,L,A ,LD A,B,LD B,
ALPHA,BETA,U,LDU,V,LDV,Q,LDQ,W ORK,W ORK2,INORK 3, \mathbb{NFO)}
CHARACTER * 1 JOBU,NOBV,NOBQ
DOUBLE COM PLEX A (LDA,*), B (LDB,*), U (LDU ,*), V (LDV,*),
Q (LDQ,*),W ORK (*)
\mathbb{N TEGER*8M ,N,P,K,L,LDA,LDB,LDU,LDV,LDQ, IN FO}
\mathbb{NTEGER*8 IN ORK 3 (*)}
DOUBLE PRECISION ALPHA (*),BETA (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}
 [LDB],ALPHA,BETA, U, [LDU],V, [LDV],Q, [LDQ], [W ORK], [W ORK2], \(\mathbb{I N}\) ORK \(3,[\mathbb{N F O}])\)


COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COMPLEX (8), D \(\mathbb{M}\) ENSION (:,:) :: A, B, U, V, Q
\(\mathbb{N}\) TEGER ::M,N,P,K,L,LDA,LDB,LDU,LDV,LDQ, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{I N}\) ORK 3
REAL (8),D \(\mathbb{M}\) ENSION (:) ::ALPHA,BETA,W ORK 2
SU BROUTINE G GSVD_64 (JOBU, JO BV , JOBQ, \(\mathbb{M}], \mathbb{N}],[P], K, L, A,[L D A]\), \(B,[L D B], A L P H A, B E T A, U,[L D U], V,[L D V], Q,[L D Q],[W O R K]\), [W ORK2], \(\mathbb{I N}\) ORK3, [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) :: JO BU , JO BV , \(\mathcal{J O B Q}\)
COMPLEX (8),D \(\mathbb{I M} E N S I O N(:):: W\) ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) :: A, B, U, V, Q
\(\mathbb{N}\) TEGER (8) ::M ,N,P,K,L,LDA,LDB,LDU,LDV,LDQ, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) :: \(\mathbb{I W}\) ORK 3
REAL (8),D \(\mathbb{M}\) ENSION (:) ::ALPHA,BETA,W ORK 2

\section*{C INTERFACE}
\#include < sunperfh>
void zggsvd (char j̀juu, char j̀bv, char jobq, intm, int n, int p , int *k, int *l, doublecom plex *a, int lda, doublecom plex *b, int ldb, double *alpha, double *beta, doublecom plex *u, int ldu, doublecom plex *v, int ldv, doublecom plex *q, int ldq, int *inv ork3, int*info);
void zggsvd_64 (char jobu, char jobv, char joboq, long m, long n , long p , long *k, long *l, doublecom plex *a, long lda, doublecom plex *b, long ldb, double *alpha, double *beta, doublecom plex *u, long ldu, doublecom plex *v, long ldv, doublecom plex *q, long ldq, long *iv ork3, long *info);

\section*{PURPOSE}
zggsvd com putes the generalized singularvalue decom position
(G SVD ) of an \(M-b y-N\) complex \(m\) atrix \(A\) and \(P-b y-N\) com plex \(m\) atrix B :
\[
U{ }^{*} A * Q=D 1^{*}(0 R), \quad V{ }^{*} B * Q=D 2^{\star}(0 R)
\]
\(w\) here \(U, V\) and \(Q\) are unitary \(m\) atrices, and \(Z\) ' \(m\) eans the conjugate transpose of \(Z\). Let \(K+L=\) the effective num erical rank of them atrix ( \(A\) ' \(B)^{\prime}\) ', then \(R\) is a \((K+L)\)-by- \((K+L)\) nonsingular upper triangularm atrix, D 1 and \(D 2\) are \(M\)-by-( \((\mathbb{K}+L)\) and \(\mathrm{P}-\mathrm{by}-(\mathrm{K}+\mathrm{L})\) "diagonal" \(m\) atrioes and of the follow ing structures, respectively:

If \(M \mathrm{~K}-\mathrm{L}>=0\),
```

            K L
    DI= K(I O)
            L (OC )
    M K 工 (O O )
            K L
    D2 = L (0 S )
    P-\leftharpoonup (0 0)
        NK\dashvK L
    (0R ) = K (0 R11 R12 )
L(0 0 R22)
where

```
```

C = diag(ALPHA (K+1), .., ALPHA (K+L)),
S = diag(BETA (K+1), ...,BETA (K+L)),
C**2+S**2 = I.
R is stored in A (1:K +L,N K 工+1 N ) on exit.

```

IfM \(\mathrm{K}-\mathrm{L}<0\),

K \(\mathrm{M}-\mathrm{K} \mathrm{K}+\mathrm{L}-\mathrm{M}\)
\(D 1=K\left(\begin{array}{ll}I & 0\end{array}\right)\)
\(M K(0 C 0)\)

K M -K K \(+\mathrm{L}+\mathrm{M}\)
\(D 2=M K(0 S 0)\)
\(\mathrm{K}+\mathrm{L} \mathrm{M}\) ( 0 O I )
\(\mathrm{P}-\left(\begin{array}{lll}0 & 0 & 0\end{array}\right)\)
\[
\begin{aligned}
& \mathrm{N} \text { K ـ K M K K + L } \mathrm{M} \\
& (0 R)=K(0 \quad R 11 R 12 R 13) \\
& M-K \text { ( } 0 \quad 0 \quad \text { R22 R23) } \\
& \mathrm{K}+\mathrm{L} \mathrm{M} \text { (0 0 0 R } 33 \text { ) }
\end{aligned}
\]
where
\[
\begin{aligned}
& C=\operatorname{diag}(A L P H A(K+1), \ldots, A L P H A M)), \\
& S=\operatorname{diag}(\operatorname{BETA}(K+1), \ldots, \text { BETA } M)), \\
& C \star \star 2+S \star \star 2=I .
\end{aligned}
\]
(R11 R12R13) is stored in \(A(\mathbb{M}, N-K-4+1 \mathbb{N})\), and R 33 is stored
( 0 R 22 R 23 )
in \(B(M-K+1: L, N+M-K-\longleftarrow+1 N)\) on exit.

The routine com putes \(C, S, R\), and optionally the unitary transform ation \(m\) atrices \(U, V\) and \(Q\).

In particular, if \(B\) is an \(N\) by N nonsingular \(m\) atrix, then the G SVD of A and B im plicitly gives the SVD of *inv (B):
\[
\mathrm{A} * \operatorname{inv}(\mathrm{~B})=\mathrm{U} \text { * (D 1*inv (D 2))*V '. }
\]

If ( \(A\) ' \(B\) )'has orthnorm alcolum ns, then the GSVD of A and \(B\) is also equal to the CS decom position of \(A\) and \(B\). Further\(m\) ore, the GSVD can be used to derive the solution of the eigenvalue problem :
A *A x = lam bda* B *B x.

In som e literature, the GSVD ofA and \(B\) is presented in the form
\[
\mathrm{U} * \mathrm{~A} * \mathrm{X}=(0 \mathrm{D} 1), \quad \mathrm{V} * \mathrm{~B} * \mathrm{X}=(0 \mathrm{D} 2)
\]
where U and V are orthogonaland X is nonsingular, and D 1 and D 2 are "diagonal". The form erGSVD form can be converted to the latter form by taking the nonsingularm atrix \(X\) as
\[
\begin{aligned}
X= & Q *\left(\begin{array}{ll}
I & 0
\end{array}\right) \\
& (0 \operatorname{inv}(R))
\end{aligned}
\]

\section*{ARGUMENTS}
\(J 0 \mathrm{BU}\) (input)
\(=\mathrm{U}\) : \(: \mathrm{U}\) nitary m atrix U is com puted;
\(=N^{\prime}: \mathrm{U}\) is not com puted.
\(J O B V\) (input)
\(=\mathrm{V}\) : U nitary m atrix V is com puted;
= N ': V is not com puted.
\(J O B Q\) (input)
\(=Q\) : U nitary \(m\) atrix \(Q\) is com puted;
\(=\mathrm{N}^{\prime}: \mathrm{Q}\) is notcom puted.

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).
N (input) The num ber of collm ns of the \(m\) atrices \(A\) and \(B . N\) \(>=0\).
\(P\) (input) The num ber of row sof the \(m\) atrix \(B . P>=0\).

K (output)
On exit, \(K\) and L specify the dim ension of the subblocks described in Punpose. \(\mathrm{K}+\mathrm{L}=\) effective num ericalrank of (A 'B )'.

L (output)
On exit, \(K\) and \(L\) specify the dim ension of the subblocks described in Pupose. K \(+\mathrm{L}=\) effective num erical rank of (A 'B ').

A (input/output)
On entry, the M by-N m atrix A. On exit, A contains the triangularm atrix \(R\), orpartof \(R\). See
Purpose fordetails.

LD A (input)
The leading dim ension of the aray A. LDA >= \(m a x(1, M)\).

B (input/output)
On entry, the P -by-N m atrix B. On exit, B contains part of the triangularm atrix \(R\) if \(M K-4<\) 0 . Se Pupose fordetails.

LD B (input)
The leading dim ension of the array \(B . L D B>=\) \(\max (1, \mathrm{P})\).

ALPHA (output)
On exit, A LPHA and BETA contain the generalized singular value pairs of A andB;ALPHA \((1: K)=1\),
ALPHA \((1: K)=1\),
\(\operatorname{BETA}(1: K)=0\), and ifM \(K-L>=0\), ALPHA \((\mathbb{K}+1 \mathbb{K}+L)\)
= C,
BETA \((K+1 K+L)=S\), orifM \(K-L<0\), ALPHA \((K+1 M)=\)
C, ALPHA \(M+1: K+L)=0\)
\(\operatorname{BETA}(K+1 M)=S, B E T A(M+1: K+L)=1\) and
A LPHA \((\mathbb{K}+L+1 \mathbb{N})=0\)
\(\operatorname{BETA}(\mathbb{K}+\mathrm{L}+1 \mathbb{N})=0\)
BETA (output)
See description of A LPH A.

U (output)
If \(J O B U=U ', U\) contains the \(M-b y \neq M\) unitary
\(m\) atrix \(U\). If \(J O B U=N ', U\) is not referenced.
LD U (input)
The leading dim ension of the array \(U\). LD U >= \(m a x(1, M)\) if \(\mathrm{JOBU}=\mathrm{U}\) '; LD U >= 1 otherw ise.

V (output)
If \(\mathrm{JOBV}=\mathrm{V}\) ', V contains the P -by P unitary
m atrix V . If \(\mathrm{JO} \mathrm{BV}=\mathrm{N}\) ', V is not referenced.

LDV (input)
The leading dim ension of the array V . LDV >= \(\max (1, \mathrm{P})\) if \(\mathrm{JOBV}=\mathrm{V} ; \mathrm{LDV}>=1\) otherw ise.

Q (output)
If \(\mathrm{OOBQ}=\mathrm{Q}, \mathrm{Q}\) contains the N -by N unitary \(m\) atrix \(Q\). If \(J O B Q=N\) ', \(Q\) is not referenced.

LD Q (input)
The leading dimension of the array \(Q . L D Q>=\) \(\max (1, N)\) if \(J B Q=Q ; L D Q>=1\) otherw ise.

W ORK (w orkspace)
dim ension \(M A X(3 * N, M, P)+N)\)

W ORK 2 (w orkspac)
dim ension ( \(2 * \mathrm{~N}\) )

IV ORK 3 (output)
dim ension (N) On exit, \(\mathbb{I N}\) ORK 3 stores the sorting inform ation. M ore precisely, the follow ing loop w illsortA LPHA for \(I=K+1, m\) in \(M, K+L\) ) sw ap A LPHA (I) and ALPHA (IW ORK 3 (I)) endfor such that ALPHA (1) >=ALPHA (2) >= ... \(>=A L P H A(N)\).
\(\mathbb{N} F O\) (output)
= 0: successfulexit.
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvahue.
\(>0\) : if \(\mathbb{N N F O}=1\), the Jacobi-type procedure failed to converge. For further details, see subroutine C TG SJA .

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
```

zggsvp - com pute unitary m atrices U,V and Q such that N -

```
\(K-\mathrm{K} L \mathrm{U} * \mathrm{~A} * \mathrm{Q}=\mathrm{K}\) (0A12A13) ifM K- \(\mathrm{K}>=0\)

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZG G SVP (JOBU,NOBV,NOBQ,M ,P,N,A,LDA,B,LDB,TOLA,}
TOLB,K,L,U,LDU,V,LDV,Q,LDQ,IN ORK,RW ORK,TAU,W ORK,
\mathbb{NFO)}
CHARACTER * 1 JOBU,NOBV,JOBQ
DOUBLE COM PLEX A (LDA,*), B (LDB,*), U (LDU ,*), V (LDV,*),
Q (LDQ,*),TAU (*),W ORK (*)
\mathbb{NTEGERM,P,N,LDA,LDB,K,L,LDU,LDV,LDQ, INFO}
INTEGER IN ORK (*)
DOUBLE PRECISION TOLA,TOLB
DOUBLE PRECISION RW ORK (*)

```
SU BROUTINE ZGGSVP_64 (JOBU , JOBV, JOBQ,M,P,N,A,LDA,B,LDB,TOLA,
    TOLB,K,L,U,LDU,V,LDV,Q,LDQ,IWORK,RWORK,TAU,WORK,
    \(\mathbb{N} F O\) )
CHARACTER * 1 JOBU, JOBV , JOBQ
DOUBLE COMPLEX A (LDA,*), B (LDB,*), U (LDU ,*), V (LDV,*),
Q (LDQ,*),TAU (*), WORK (*)
\(\mathbb{I N}\) TEGER*8 M , P, N,LDA,LDB, \(\mathrm{K}, \mathrm{L}, \mathrm{LD} \mathrm{U}, \mathrm{LD} V, L D Q, \mathbb{N}\) FO
\(\mathbb{N}\) TEGER*8 \(\mathbb{I N}\) ORK (*)
DOUBLE PRECISION TOLA, TOLB
DOUBLE PRECISION RW ORK (*)

\section*{F95 INTERFACE}


TOLA,TOLB,K,L,U,[LDU],V,[LDV],Q,[LDQ],[IN ORK], [RW ORK], [TAU], [W ORK], [NFO])

CHARACTER (LEN=1): : JOBU, JOBV, JOBQ
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) :: A, B, U, V, Q
\(\mathbb{N} T E G E R:: M, P, N, L D A, L D B, K, L, L D U, L D V, L D Q, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL (8) ::TOLA,TOLB
REAL (8), D IM ENSION (:) ::RW ORK

SU BROUTINE G GSVP_64 (JOBU, \(\mathcal{J O B V}, ~ J O B Q, \mathbb{M}],[\mathbb{P}], \mathbb{N}], A,[L D A], B\), [LD B ], TO LA ,TOLB , K, L, U , [LDU ], V, [LDV], Q, [LDQ], [IW ORK], [RW ORK], [TAU ], [W ORK ], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1):: \(00 \mathrm{BU}, 50 \mathrm{BV}, 50 \mathrm{BQ}\)
COMPLEX (8),D \(\mathbb{D}\) ENSION (:) ::TAU,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) :: A, B, U, V, Q
\(\mathbb{N}\) TEGER (8) ::M, \(\mathrm{P}, \mathrm{N}, \mathrm{LDA}, \mathrm{LD} \mathrm{B}, \mathrm{K}, \mathrm{L}, \mathrm{LD} \mathrm{U}, \mathrm{LDV}, \mathrm{LD} Q, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK
REAL (8) ::TOLA,TOLB
REAL (8),D \(\mathbb{M}\) ENSION (:) ::RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zggsvp (char j̀bu, char j̀bv, char jobq, intm, int p, int \(n\), doublecom plex *a, int lda, doublecom plex *b, int ldb, double tola, double tolb, int *k, int *l, doublecom plex *u, int ldu, doublecom plex *v, int ldv, doublecom plex *q, int ldq, int *info);
void zggsvp_64 (char jंbu, char jंbv, char jobq, long m, long p, long \(n\), doublecom plex *a, long lda, doublecom plex *b, long lab, double tola, double tolb, long *k, long *l, doublecom plex *u, long ldu, doublecom plex *v, long ldv, doublecom plex *q, long ldq, long *info);

\section*{PURPOSE}
zggsvp com putes unitary \(m\) atrices \(\mathrm{U}, \mathrm{V}\) and Q such that
L (0 A A23) M K \(\mathrm{L}\left(\begin{array}{lll}0 & 0 & 0\end{array}\right)\)

N K L K
\(=\mathrm{K}(0\) A12 A13) ifM \(\mathrm{K} \mathrm{L}<0\);
MK (0 0 A 23)
```

            N-KЧ K L
    V *B*Q = L (0 0 B13)
P-工 (0 0 0 )

```
where the K -by-K m atrix A 12 and L-by-L matrix B13 are nonsingularuppertriangular; A 23 is L-by-L uppertriangular if M K \(-\mathrm{l}=0\), otherw ise A 23 is \(\mathrm{M}-\mathrm{K}\) )-by-L upper trapezoidal. \(K+L=\) the effective num erical rank of the \(M+P\) )-by \(-\mathbb{N}\) m atrix
(A ', B )'. Z 'denotes the conjugate transpose of Z .

This decom position is the preprocessing step for com puting the Generalized Singular V alue D ecom position (GSVD ), see subroutine CG G SV D .

\section*{ARGUMENTS}

JOBU (input)
= U ': U nitary matrix U is com puted;
\(=\mathrm{N}^{\prime}: \mathrm{U}\) is not com puted.

JO BV (input)
\(=\mathrm{V}\) : U nitary m atrix V is com puted;
\(=\mathrm{N}: \mathrm{V}\) is not com puted.
\(J O B Q\) (input)
\(=Q\) : U nitary m atrix Q is com puted;
\(=\mathrm{N}^{\prime}: \mathrm{Q}\) is notcom puted.
M (input) The num ber of row s of the matrix \(\mathrm{A} . \mathrm{M}>=0\).

P (input) The num ber of row sof the matrix B. P \(>=0\).

N (input) The num ber of colum ns of the \(m\) atrioes \(A\) and \(B\). \(N\) \(>=0\).

A (input/output)
On entry, the M by-N matrix A. On exit, A contains the triangular (or trapezoidal) matrix described in the Purpose section.

LD A (input)
The leading dim ension of the array A. LD A >= max (1,M).

B (input/output)
On entry, the \(P-b y-N m\) atrix B. On exit, B contains the triangularm atrix described in the Purpose section.

LD B (input)
The leading dim ension of the array \(\mathrm{B} . \mathrm{LDB}>=\) \(\max (1, P)\).

\section*{TOLA (input)}

TOLA and TOLB are the thresholds to determ ine the effective num erical rank ofm atrix \(B\) and a subblock of A. Generally, they are set to TOLA = \(\mathrm{MAX}(\mathrm{M}, \mathrm{N}) \star\) norm ( A\() \star \mathrm{M}\) ACHEPS, TOLB = \(M A X(\mathbb{P} N){ }^{*}\) norm ( \(\left.B\right)^{*}\) M ACHEPS. The size of TOLA and TO LB m ay affect the size of backw ard errors of the decom position.

TOLB (input)
See description of TO LA .

K (output)
On exit, \(K\) and L specify the dim ension of the subblocks described in Purpose section. \(\mathrm{K}+\mathrm{L}=\) effective num erical rank of (A ', \(\mathrm{B}^{\prime}\) )'.

L (output)
See the description ofK .

U (input) If \(\mathrm{OB} \mathrm{U}=\mathrm{U}\) ', U contains the unitary matrix U . If \(\mathrm{OBBU}=\mathrm{N}\) ', U is notreferenced.

LD U (input)
The leading dim ension of the array \(\mathrm{U} . \operatorname{LDU}>=\) \(\max (1, M)\) if \(\mathrm{OB} U=U ' ; L D U>=1\) otherw ise.

V (input) If \(\mathrm{OBV}=\mathrm{V}\) ', V contains the unitary matrix V . If \(\mathrm{OBV}=\mathrm{N}^{\prime}\), \(V\) is notreferenced.

LDV (input)
The leading dim ension of the array \(\mathrm{V} . \mathrm{LDV}>=\) \(\mathrm{max}(1, \mathrm{P})\) if \(\mathrm{OBV}=\mathrm{V} ; \mathrm{LDV}>=1\) otherw ise.
\(Q\) (input) If \(O B Q=Q\) ' \(Q\) contains the unitary \(m\) atrix \(Q\). If \(\mathrm{OBQ}=\mathrm{N}^{\prime}, \mathrm{Q}\) is notreferenced.

LD Q (input)
The leading dim ension of the array \(\mathrm{Q} \cdot \mathrm{LDQ}>=\) \(\max (1, N)\) if \(\mathrm{OBQ}=\mathrm{Q} ; \mathrm{LD} Q>=1\) otherw ise.

IW ORK (w orkspace)
dim ension \(\mathbb{N}\) )

RW ORK (w orkspace)
dim ension \((2 * N)\)

TAU (w orkspace)
dim ension \((\mathbb{N})\)
W ORK (w orkspace)
dim ension \(M A X(3 * N, M, P))\)
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue.

\section*{FURTHER DETAILS}

The subroutine uses LA PACK subroutine CGEQPF for the QR factorization \(w\) ith column pivoting to detect the effective num erical rank of the a \(m\) atrix. It \(m\) ay be replaced by a better rank determ ination strategy.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
zgssco -G eneral sparse solver condition num ber estim ate.

\section*{SYNOPSIS}

SUBROUTINE ZGSSCO (COND,HANDLE, \(\mathbb{E R}\) )
\(\mathbb{N} T E G E R \quad \mathbb{E R}\)
DOUBLE PRECISION COND
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

ZGSSCO -C ondition num berestim ate.

\section*{PARAMETERS}

COND -DOUBLE PRECISION
On exit, an estim ate of the condition num berof the factored \(m\) atrix. M ustbe called after the num erical factorization subroutine, ZGSSFA ().

HANDLE (150) -D OUBLE PREC IS IO N anay
On entry, HANDLE ( \(*\) ) is an array containing inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine.
M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Enrornum ber. If no errorencountered, unchanged on
exit. If emorencountered, it is set to a non-zero
integer. Errornum bers setby this subroutine:
-700 : Invalid calling sequence - need to call ZG SSFA first.
-710 : C ondition num ber estim ate not available (notim plem ented for this H A N D LE sm atix type).

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
zgssda -D eallocate w orking storage for the general sparse solver.

\section*{SYNOPSIS}

SUBROUTINE ZGSSDA (HANDLE, \(\mathbb{E R}\) )
\(\mathbb{N}\) TEGER \(\quad \mathbb{E R}\)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

ZGSSDA -D eallocate dynam ically allocated w orking storage.

\section*{PARAMETERS}

HANDLE (150) -D OUBLE PREC IS IO N array
On entry, HANDLE (*) is an array containing inform ation needed by the solver, and \(m\) ustbe passed unchanged to each sparse solver subroutine. M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Errornum ber. If no errorencountered, unchanged on exit. If errorencountered, it is set to a non-zero integer. Errornum bers set.by this subroutine:
none

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
zgssfa -G eneral sparse solvernum eric factorization.

\section*{SYNOPSIS}

SUBROUTINE ZGSSFA (NEQNS,COLSTR,ROW \(\mathbb{N D}, V A L U E S, H A N D L E, \mathbb{E R}\) )
\(\mathbb{N} T E G E R \quad\) NEQNS, COLSTR (*),ROW \(\mathbb{N D}(*), \mathbb{E R}\)
DOUBLE COM PLEX VALUES (*)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

ZGSSFA -N um eric factorization of a sparse m atrix.

\section*{PARAMETERS}

NEQNS - \(\mathbb{N}\) TEGER
On entry, NEQNS specifies the num ber of equations in coefficientm atrix. U nchanged on exit.
\(\operatorname{COLSTR}\left(^{*}\right)-\mathbb{N}\) TEG ER array
On entry, \(\operatorname{COLSTR}\left(^{*}\right)\) is an array of size \((\mathbb{N} E \mathrm{~N}+1\) ), containing the pointers of them atrix structure.
U nchanged on exit.
ROWIND ( \(\left.{ }^{( }\right)-\mathbb{N}\) TEGER array
On entry, ROWIND ( \({ }^{*}\) ) is an array of size
CO LSTR \(\mathbb{N} E Q N S+1)-1\), containing the indices of the \(m\) atrix structure. U nchanged on exit.

VALUES (*) -D OUBLE COM PLEX aray
On entry, VALUES (*) is an array of size
CO LSTR (NEQNS+1)-1, containing the num eric values of
the sparse \(m\) atrix to be factored. U nchanged on exit.

HANDLE (150) -D OUBLE PRECISIO N array On entry, HANDLE (*) is an array containing inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine. M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Errornum ber. If no emrorencountered, unchanged on exit. If errorencountered, it is set to a non-zero integer. Enrornum bers set.by this subroutine:
-300 : Invalid calling sequence - need to call Z G SSO R first.
-301 : Failure to dynam ically allocate \(m\) em ory.
-666 : Intemalenror.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
zgssis -G eneral sparse solver one call interface.

\section*{SYNOPSIS}

> SUBROUTINE ZGSSFS (M TXTYP,PIVOT,NEQNS,COLSTR,ROW \(\mathbb{N} D\), VALUES,NRHS,RHS ,LDRHS,ORDMTHD, OUTUNT,MSGLVL,HANDLE, \(\mathbb{E R}\) )

CHARACTER*2 M TXTYP
CHARACTER*1 PIVOT
\(\mathbb{N}\) TEGER NEQNS,COLSTR (*),ROW \(\mathbb{N} D\left({ }^{*}\right)\),NRHS,LDRHS, OUTUNT,MSGLVL, \(\mathbb{E R}\)
CHARACTER*3 ORDMTHD
D OUBLE COM PLEX VALUES (*), RHS (*)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

ZGSSFS -G eneral sparse solver one call interface.

\section*{PARAMETERS}

MTXTYP -CHARACTER*2
On entry, M TX TY P specifies the coefficientm atrix
type. Specifically, the valid options are:
Sp 'or SP '-sym m etric structure, H erm itian positive definite
values
ss'or SS '-sym m etric structure, sym \(m\) etric values
su 'or SU '-sym m etric structure, unsym \(m\) etric values
uu 'or UU '-unsym \(m\) etric structure, unsym \(m\) etric values
U nchanged on exit.

\section*{PIVOT -CHARACTER*1}

O n entry, pivot specifies w hether ornotpivoting is used in the course of the num eric factorization. The valid options are:
h'or \(\mathrm{N}^{\prime}\)-no pivoting is used
(Pivoting is not supported forthis release).
U nchanged on exit.

\section*{NEQNS - \(\mathbb{N}\) TEGER}

On entry, NEQNS specifies the num ber ofequations in the coefficientm atrix. N EQ N S m ustbe at leastone. U nchanged on exit.

\section*{\(\operatorname{COLSTR}\) ( \(\left.^{*}\right)-\mathbb{N}\) TEG ER anay}

On entry, COLSTR (*) is an array of size \(\operatorname{NEQNS+1),~}\) containing the pointers of them atrix structure.
U nchanged on exit.
ROWIND (*) - \(\mathbb{N}\) TEG ER array
On entry, ROWIND ( \({ }^{\star}\) ) is an array of size COLSTR \(\mathbb{N E Q N S + 1 ) - 1 , ~ c o n t a i n i n g ~ t h e ~ i n d i c e s ~ o f ~ t h e ~}\) \(m\) atrix structure. U nchanged on exit.

\section*{VALUES (*) -D OUBLE COM PLEX aray}

On entry, VALUES (*) is an array of size
CO LSTR \(\mathbb{N} E Q N S+1)-1\), containing the non-zero num eric values of the sparse \(m\) atrix to be factored. U nchanged on exit.

NRHS - \(\mathbb{N}\) TEGER
On entry, N RH S specifies the num ber of righthand sides to solve for. U nchanged on exit.

RHS (*) -DOUBLE COM PLEX amay
On entry, RH S (LD RH S NRHS) contains the NRH S right hand sides. On exit, it contains the solutions.

LDRHS - \(\mathbb{N}\) TEGER
On entry, LD RH S specifies the leading dim ension of the RH S array. U nchanged on exit.

ORDMTHD -CHARACTER*3
On entry, ORDM THD specifies the fill-reducing ordering to be used by the sparse solver.
Specifically, the valid options are:
hat'or NA T '-natural ordering (no ordering) mmd'or M M D'-multiplem inim um degree
gnd 'or GND '-general nested dissection
uso 'or USO '-user specified ordering (see ZG SSU O )
U nchanged on exit.

OUTUNT - \(\mathbb{N}\) TEGER
O utputunit. U nchanged on exit.
M SGLVL - \(\mathbb{N}\) TEGER
M essage level.

0 -no output from solver.
( \(N\) o m essages supported for this release.)
U nchanged on exit.
HANDLE (150) -DOUBLE PRECISIO N aray
On entry, HANDLE (*) is an array of containing
inform ation needed by the solver, and \(m\) ustbe passed
unchanged to each sparse solver subroutine.
M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Enrornum ber. If no errorencountered, unchanged on
exit. If emorencountered, it is set to a non-zero
integer. Errornum bers set.by this subroutine:
-101: Failure to dynam ically allocate mem ory.
-102 : Invalid m atrix type.
-103 : Invalid pivotoption.
-104: N um berofnonzeros is less than N EQ N S .
-105: NEQNS < 1
-201 : Failure to dynam ically allocate \(m\) em ory.
-301 : Failure to dynam ically allocate \(m\) em ory.
-401 : Failure to dynam ically allocate \(m\) em ory.
-402 :NRHS < 1
-403 : NEQN S > LD RHS
-666 : Intemalerror.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
zgssin -Initialize the general sparse solver.

\section*{SYNOPSIS}
```

SUBROUTINE ZGSSIN (M TXTYP,PIVOT,NEQNS,COLSTR,ROW IND,OUTUNT,
M SGLVL,HANDLE,\mathbb{ER )}
CHARACTER*2 M TXTYP
CHARACTER*1 PIVOT
\mathbb{NTEGER NEQNS,COLSTR (*),ROW IND (*),OUTUNT,MSGLVL,\mathbb{ER}}\mathbf{N}\mathrm{ (*)}
DOUBLE PRECISION HANDLE (150)

```

\section*{PURPOSE}

ZGSSIN -Initialize the sparse solver and input the m atrix
structure.

\section*{PARAMETERS}

\section*{MTXTYP -CHARACTER*2}

On entry, M TX TY P specifies the coefficientm atrix type. Specifically, the valid options are:

Sp 'or SP '-sym m etric structure, H erm itian positive definite
ss'or SS '-sym m etric structure, sym m etric values
su 'or SU '-sym m etric structure, unsym \(m\) etric values
uu 'or UU '-unsym \(m\) etric structure, unsym \(m\) etric values
U nchanged on exit.

On entry, PIV OT specifies whetherornotpivoting is
used in the course of the num eric factorization.
The valid options are:
h'or \(\mathrm{N}^{\prime}\)-no pivoting is used
(Pivoting is not supported forthis release).

U nchanged on exit.
NEQNS - \(\mathbb{N} T E G E R\)
On entry, NEQNS specifies the num berofequations in the coefficientm atrix. N EQ N S m ustbe at leastone. U nchanged on exit.
\(\operatorname{COLSTR}\) ( \(\left.^{*}\right)-\mathbb{N}\) TEG ER amay
On entry, \(\operatorname{COLSTR}\) ( \({ }^{*}\) ) is an array of size \(\mathbb{N E Q N S + 1 ) \text { , }}\) containing the pointers of them atrix structure.
U nchanged on exit.
ROWIND ( \({ }^{\star}\) ) - \(\mathbb{N}\) TEG ER array
On entry, ROWIND ( \({ }^{\star}\) ) is an array of size
CO LSTR \(\mathbb{N} E Q N S+1\) )-1, containing the indices of the \(m\) atrix structure. U nchanged on exit.

HANDLE (150) -D OUBLE PRECIS IO N array
On entry, \(H A N D L E\) (*) is an array containing inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine. M odified on exit.

OUTUNT - \(\mathbb{N}\) TEGER
O utputunit. U nchanged on exit.

M SGLVL - \(\mathbb{N}\) TEGER
M essage level.
0 -no output from solver.
(N o m essages supported for this release.)
U nchanged on exit.
\(\mathbb{E R} \quad\) - \(\mathbb{N}\) TEGER
Enrornum ber. If no errorencountered, unchanged on
exit. If emrorencountered, it is set to a non-zero
integer. Enrornum bers setby this subroutine:
-101: Failure to dynam ically allocate mem ory.
-102 : Invalid m atrix type.
-103 : Invalid pivotoption.
-104 : N um berofnonzeros less than N EQ N S .
-105:NEQNS<1

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
zgssor-G eneral sparse solver ordering and sym bolic factorization.

\section*{SYNOPSIS}

```

CHARACTER*3 ORDMTHD
\mathbb{NTEGER ER}
DOUBLE PRECISION HANDLE (150)

```

\section*{PURPOSE}

ZGSSOR -O rders and sym bolically factors a sparse m atrix.

\section*{PARAMETERS}

ORDMTHD -CHARACTER*3
On entry, ORDM THD specifies the fill-reducing ordering to be used by the sparse solver. Specifically, the valid options are:
hat'or NAT '-naturalordering (no ordering)
mmd'or M M D '-m ultiplem inim um degree
gnd 'or GND '-generalnested dissection
uso 'or U SO '-user specified ordering (see ZG SSU O )
U nchanged on exit.
HANDLE (150) -DOUBLE PRECISIO N aray
On entry, HANDLE (*) is an array containing
inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine.

M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Errornum ber. If no errorencountered, unchanged on
exit. If emrorencountered, it is set to a non-zero
integer. Enrornum bers set.by this subroutine:
-200 : Invalid calling sequence - need to call ZG SSIN first.
-201 : Failure to dynam ically allocate \(m\) em ory.
-666 : Intemalerror.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
zgssps -Print general.sparse solver statics.

\section*{SYNOPSIS}

SUBROUTINE ZGSSPS (HANDLE, \(\mathbb{E R}\) )
\(\mathbb{N} T E G E R \quad \mathbb{E R}\)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

ZGSSPS -Printsolver statistics.

\section*{PARAMETERS}

HANDLE (150) -D OUBLE PREC IS IO N array
On entry, HANDLE ( \(*\) ) is an amay containing
inform ation needed by the solver, and \(m\) ust.be passed
unchanged to each sparse solver subroutine.
M odified on exit.

ER
- \(\mathbb{N}\) TEGER

E rrornum ber. If no errorencountered, unchanged on
exit. If errorencountered, it is set to a non-zero
integer. Errornum bers setby this subroutine:
-800: Invalid calling sequence -need to call ZG SSSL first.
-899 : Printed solver statistics not supported this release.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
zgssup -Retum perm utation used by the general sparse solver.

\section*{SYNOPSIS}

SUBROUTINE ZGSSRP (PERM, HANDLE, \(\mathbb{E R}\) )
\(\mathbb{N} T E G E R \quad\) PERM ( \(\left.{ }^{( }\right)\), \(\mathbb{E R}\)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

ZGSSRP -Retums the perm utation used by the solver for the fill-reducing ordering.

\section*{PARAMETERS}

PERM \(\mathbb{N} E Q N S\) ) - \(\mathbb{N} T E G E R\) aray
U ndefined on entry. PERM \(\mathbb{N E Q N S}\) ) is the perm utation array used by the sparse solver for the fillreducing ordering. M odified on exit.

HANDLE (150) -D OUBLE PRECISION array
On entry, HANDLE ( \(*\) ) is an array containing
inform ation needed by the solver, and \(m\) ust.be passed unchanged to each sparse solver subroutine.
M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Errornum ber. If no errorencountered, unchanged on exit. If errorencountered, it is set to a non-zero
integer. Errornum bers set.by this subroutine:
-600 : Invalid calling sequence - need to call Z G SSO R first.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
zgsssl-Solve routine for the general sparse solver.

\section*{SYNOPSIS}

SUBROUTINE ZGSSSL (NRHS,RHS,LDRHS,HANDLE, \(\mathbb{E R}\) )
\(\mathbb{N}\) TEGER NRHS,LDRHS, ER
D OUBLE COM PLEX RHS (LDRHS,NRHS)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

ZGSSSL -Triangular solve of a factored sparse m atrix.

\section*{PARAMETERS}

NRHS - \(\mathbb{N}\) TEGER
On entry, N RH S specifies the num ber of righthand sides to solve for. U nchanged on exit.

RHS (LDRHS,*) -DOUBLE COM PLEX array
On entry, RH S (LDRHS,NRHS) contains the NRHS right hand sides. On exit, itcontains the solutions.

LDRHS - \(\mathbb{N}\) TEGER
On entry, LD RH S specifies the leading dim ension of the RH S array. U nchanged on exit.

HANDLE (150) -D OUBLE PREC ISIO N aray
O n entry, HANDLE ( \({ }^{\star}\) ) is an array containing
inform ation needed by the solver, and \(m\) ustbe passed
unchanged to each sparse solver subroutine.
M odified on exit.

ER
- \(\mathbb{N}\) TEGER

Errornum ber. If no errorencountered, unchanged on
exit. If errorencountered, it is set to a non-zero
integer. Errornum bers setby this subroutine:
-400 : Invalid calling sequence - need to callZ G SSFA first.
-401 : Failure to dynam ically allocate \(m\) em ory.
-402 : NRHS < 1
-403 : NEQN S > LD RHS

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- PARAMETERS

\section*{NAME}
zgssuo -U ser supplied perm utation for ordering used in the general sparse solver.

\section*{SYNOPSIS}

SUBROUTINE ZGSSUO (PERM,HANDLE, \(\mathbb{E R}\) )
\(\mathbb{N}\) TEGER PERM (*), \(\mathbb{E R}\)
DOUBLE PRECISION HANDLE (150)

\section*{PURPOSE}

ZGSSUO -U ser supplied perm utation for ordering. M ust.be called after ZGSS IN 0 (sparse solver initialization) and before ZGSSOR () (sparse solver ordering).

\section*{PARAMETERS}

PERM \(\mathbb{N} E Q N S\) ) - \(\mathbb{N}\) TEGER array
On entry, PERM (NEQNS) is a perm utation array supplied by the user for the fill-reducing ordering.
U nchanged on exit.

HANDLE (150) -D OUBLE PREC IS IO N array
On entry, HANDLE (*) is an array containing
inform ation needed by the solver, and \(m\) ustbe passed
unchanged to each sparse solver subroutine.
M odified on exit.
\(\mathbb{E R} \quad-\mathbb{N} T E G E R\)
Errornum ber. If no errorencountered, unchanged on
exit. If emorencountered, it is set to a non-zero
integer. Errornum bers setby this subroutine:
-500 : Invalid calling sequence - need to call Z G SSIN first.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgtcon -estim ate the reciprocal of the condition num ber of a com plex tridiagonalm atrix A using the LU factorization as com puted by CG TTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGTCON NORM,N,LOW ,DIAG,UP1,UP2, \mathbb{PIVOT,ANORM,RCOND,}}\mathbf{N},
W ORK,INFO)
CHARACTER * 1 NORM
DOUBLE COM PLEX LOW (*),D IAG (*),UP1 (*),UP2 (*),W ORK (*)
INTEGERN,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION ANORM ,RCOND
SUBROUT\mathbb{NE ZGTCON_64 NORM,N,LOW,DIAG,UP1,UP2,\mathbb{PIVOT,ANORM,}}\mathbf{N},\mp@code{N},
RCOND,W ORK,\mathbb{NFO)}
CHARACTER * 1 NORM
DOUBLE COM PLEX LOW (*),D IAG (*),UP1 (*),UP2 (*),W ORK (*)
INTEGER*8N,\mathbb{NFO}
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION ANORM ,RCOND

```

\section*{F95 INTERFACE}
```

SU BROUTINE GTCON $\mathbb{N} O R M, \mathbb{N}], L O W, D I A G, U P 1, U P 2, \mathbb{P} \mathbb{I V O T}, A N O R M$, RCOND, [WORK], [ $\mathbb{N} F O]$ )
CHARACTER (LEN=1) ::NORM
COMPLEX (8),D $\mathbb{I M}$ ENSION (:) ::LOW ,D IAG,UP1,UP2,W ORK
$\mathbb{N} T E G E R:: N, \mathbb{N F O}$
$\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T$

```

REAL (8) ::ANORM,RCOND
SU BROUTINE GTCON_64 NORM, \(\mathbb{N}], L O W, D \mathbb{I A G}, U P 1, U P 2, \mathbb{P} \mathbb{I} O T, A N O R M\), RCOND, \([\mathbb{W} O R K],[\mathbb{N} F O])\)

CHARACTER (LEN=1)::NORM
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::LOW ,D \(\mathbb{A G}, \mathrm{UP} 1, \mathrm{UP} 2, \mathrm{~W} O R K\)
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathbb{I N} F \mathrm{O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V} O T\)
REAL (8) ::ANORM,RCOND

\section*{C INTERFACE}
\#include <sunperfh>
void zgtoon (charnorm, intn, doublecom plex *low, doublecom plex *diag, doublecom plex *up1, doublecom plex *up2, int*ipivot, double anorm, double *roond, int*info);
void zgtcon_64 (charnorm , long n, doublecom plex *low , doublecom plex *diag, doublecom plex *up1, doublecom plex *up2, long *ipivivot, double anorm, double *roond, long *info);

\section*{PURPOSE}
zgtcon estim ates the reciprocal of the condition num ber of a com plex tridiagonal matrix A using the LU factorization as com puted by CG TTRF.

A \(n\) estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND \(=1 /\) (ANORM * norm (inv (A))).

\section*{ARGUMENTS}

NORM (input)
Specifies w hether the 1-norm condition number or
the infinity-norm condition num ber is required:
= 1 'or \(0^{\prime}\) : 1 -nom ;
= I': Infinity-norm.
N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

LOW (input)
The \((n-1) m\) ultipliers that define the \(m\) atrix \(L\)
from the LU factorization of \(A\) as com puted by
CGTTRF.

D IA G (input)
The \(n\) diagonalelem ents of the upper triangular \(m\) atrix \(U\) from the \(L U\) factorization ofA .

UP1 (input)
The ( \(n-1\) ) elem ents of the first superdiagonal of U .

UP2 (input)
The ( \(n-2\) ) elem ents of the second superdiagonal of U.

IPIVOT (input)
The pivotindiges; for \(1<=i<=n\), row \(i\) of the matrix was interchanged w ith row \(\mathbb{P I V O T}(i)\). \(\mathbb{P} \mathbb{V} O T\) (i) will alw ays be either \(i\) or \(i+1\); \(\mathbb{P} \mathbb{I V O T}\) (i) = iindicates a row interchange w as not required.

ANORM (input)
IfNORM = I'or \(\mathrm{D}^{\prime}\) ', the 1-norm of the original \(m\) atrix \(A\). If NORM = ' 1 ', the infinity-norm of the originalm atrix A.

RCOND (output)
The reciprocal of the condition num ber of the \(m\) atrix \(A\), com puted as \(R C O N D=1 /(A N O R M * A \mathbb{N V N M})\),
\(w\) here \(A \mathbb{N} V N M\) is an estim ate of the 1 -norm of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension \((2 * N)\)
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~ = ~ - i , ~ t h e ~ i - t h ~ a r g u m ~ e n t h a d ~ a n ~ i l l e - ~}\) galvalue

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgthr-G athers specified elem ents from \(y\) into \(x\).

\section*{SYNOPSIS}
```

SUBROUTINE ZGTHR NZ,Y,X,\mathbb{NDX)}

```
DOUBLE COM PLEX Y (*), X (*)
\(\mathbb{N}\) TEGER NZ
\(\mathbb{N} T E G E R \mathbb{N} D X(*)\)
SUBROUTINE ZGTHR_64 \(\mathbb{N} Z, Y, X, \mathbb{N} D X)\)
DOUBLE COM PLEX Y (*), X (*)
\(\mathbb{N} T E G E R * 8 N Z\)
\(\mathbb{I N} T E G E R * 8 \mathbb{N} D X(*)\)
F95 \(\mathbb{I N}\) TERFACE
SUBROUTINE GTHR ( \(\mathbb{N} Z], Y, X, \mathbb{N} D X)\)
COM PLEX (8),D \(\mathbb{I M}\) ENSTON (:) :: Y,X
\(\mathbb{N}\) TEGER ::NZ
\(\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{N D X}\)
SUBROUTINE GTHR_64 (NZ], \(\mathrm{Y}, \mathrm{X}, \mathbb{N} D \mathrm{X}\) )
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::Y,X
\(\mathbb{N}\) TEGER (8) ::NZ
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{I} \operatorname{ENSION}(:):: \mathbb{N D X}\)

\section*{PURPOSE}

ZG THR -G athers the specified elem ents from a vectory in fullstorage form into a vectorx in com pressed form. Only
the elem ents of \(y\) w hose indices are listed in indx are referenced.
do \(i=1, n\) \(x(i)=y(\) indx \((i))\)
enddo

\section*{ARGUMENTS}

N Z (input) - \(\mathbb{N}\) TEGER
\(N\) um ber of elem ents in the com pressed form .
U nchanged on exit.
\(Y\) (input)
V ectorin fullstorage form . U nchanged on exit.
X (output)
V ector in com pressed form. C ontains elem ents ofy
whose indices are listed in indx on exit.
\(\mathbb{I N D X}\) (input) - \(\mathbb{N}\) TEGER
\(V\) ector containing the indices of the com pressed
form. It is assum ed that the elem ents in \(\mathbb{N} D \mathrm{X}\) are
distinct and greater than zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgthrz -G ather and zero.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGTHRZ NZ,Y,X,INDX)}
DOUBLE COM PLEX Y (*),X (*)
INTEGER NZ
INTEGER \mathbb{NDX (*)}
SUBROUT\mathbb{NE ZGTHRZ_64 NZ,Y,X,\mathbb{NDX)}}\mathbf{N}=(
DOUBLE COM PLEX Y (*),X (*)
INTEGER*8NZ
INTEGER*8 \mathbb{NDX (*)}
F95 INTERFACE
SUBROUT\mathbb{NE GTHRZ (NZ],Y,X,NNDX)}
COM PLEX (8),D IM ENSION (:) ::Y,X
\mathbb{NTEGER ::NZ}
\mathbb{NTEGER,D IM ENSION (:) :: \mathbb{NDX}}\mathbf{N}=\mp@code{N}
SUBROUT\mathbb{NEGTHRZ_64(NZ],Y,X,\mathbb{NDX)}}\mathbf{N}=(\mathbb{N}
COM PLEX (8),D IM ENSION (:) ::Y,X
INTEGER (8)::NZ
\mathbb{NTEGER (8),D IM ENSION (:) :: \mathbb{NDX}}\mathbf{N}=\mp@code{N}

```

\section*{PURPOSE}

ZG THRZ -G athers the specified elem ents from a vectory in fullstorage form into a vectorx in com pressed form. The
gathered elem ents ofy are set to zero. O nly the elem ents ofy w hose indices are listed in indx are referenced.
```

do i=1,n
x (i) = y (indx (i))
y(indx (i)) = 0
enddo

```

\section*{ARGUMENTS}

N Z (input) - \(\mathbb{N}\) TEGER
N um ber of elem ents in the com pressed form .
U nchanged on exit.

Y (input/output)
V ector in fullstorage form. G athered elem ents are setto zero.
X (output)
V ector in com pressed form. C ontains elem ents ofy w hose indices are listed in indx on exit.
\(\mathbb{N} D X\) (input) - \(\mathbb{N} T E G E R\)
V ector containing the indiges of the com pressed form. It is assum ed that the elem ents in \(\mathbb{N D} X\) are distinctand greater than zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgtrifs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is tridiagonal, and provides errorbounds and backw ard error estim ates for the solution

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGTRFS (TRANSA,N,NRHS,LOW ,D IA G,UP,LOW F,D IA GF,UPF1,}
UPF2,\mathbb{PIVOT,B,LDB,X,LDX,FERR,BERR,W ORK,W ORK 2, INFO)}
CHARACTER * 1 TRANSA
DOUBLE COM PLEX LOW (*),DIAG (*), UP (*), LOW F (*), DIAGF (*),
UPF1 (*),UPF2 (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,INFO
\mathbb{NTEGER \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)
SUBROUT\mathbb{NE ZGTRFS_64(IRANSA,N,NRHS,LOW ,D IAG,UP,LOW F,D IAGF,}
UPF1,UPF2,\mathbb{PIVOT,B,LDB,X,LDX,FERR,BERR,W ORK,W ORK2,}
INFO)
CHARACTER * 1 TRANSA
DOUBLE COM PLEX LOW (*),DIAG (*), UP (*), LOW F (*), D IA GF (*),
UPF1 (*),UPF2 (*),B (LD B ,*),X (LDX,*),W ORK (*)
\mathbb{NTEGER*8N,NRHS,LDB,LDX,INFO}
INTEGER *8 \mathbb{PIVOT (*)}
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE GTRFS ([TRANSA ], \(\mathbb{N}], \mathbb{N} R H S], L O W, D I A G, U P, L O W ~ F, D ~ I A G F\),
 [W ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::TRANSA

UPF1,UPF2,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : B , X
\(\mathbb{N}\) TEGER : \(: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} \mathrm{B}, \mathrm{LD} \mathrm{X}, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL (8), D \(\mathbb{I}\) ENSION (:) :: FERR,BERR,W ORK2

SU BROUTINE GTRFS_64 ([TRANSA ], \(\mathbb{N}], \mathbb{N} R H S\) ],LOW,D IAG,UP,LOW F, D \(\mathbb{I A G F}, \mathrm{UPF} 1, \mathrm{UPF} 2, \mathbb{P} \mathbb{V} O T, B,[L D B], X,[L D X], F E R R, B E R R,[\mathbb{O} O R K]\), [ W ORK2], [ \(\mathbb{N} \mathrm{FO}\) ])

CHARACTER (LEN=1) ::TRANSA
COM PLEX (8), D \(\mathbb{M} \operatorname{ENSION}\) (:) :: LOW , D \(\mathbb{I A G}, ~ U P, L O W F, D \mathbb{A G F}\), UPF1, UPF2, W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: \(:\) : : : B , X
\(\mathbb{N}\) TEGER (8) :: N , NRHS, LD B , LD X , \(\mathbb{N}\) FO
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8), D \(\mathbb{I}\) ENSION (:) ::FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void zgtrfs (char transa, int \(n\), int nins, doublecom plex *low, doublecom plex *diag, doublecom plex *up, doublecom plex *low f, doublecom plex *diagf, doublecom plex *upfl, doublecom plex *upf2, int *ipivot, doublecom plex *b, int ldb, doublecom plex *x, int ldx, double * ferr, double *berr, int *info);
void zgtrfs_64 (chartransa, long n, long nrhs, doublecom plex *low, doublecom plex *diag, doublecom plex *up, doublecom plex * low f, doublecom plex *diagf, doublecomplex *upfl, doublecomplex *upf2, long *ípivot, doublecom plex *b, long lalb, doublecom plex \({ }^{*} \mathrm{x}\), long ldx, double *ferr, double *berr, long *info);

\section*{PURPOSE}
zgtrfs im proves the com puted solution to a system of linear equations w hen the coefficientm atrix is tridiagonal, and provides errorbounds and backw ard errorestim ates for the solution.

\section*{ARGUMENTS}

TRANSA (input)
Specifies the form of the system of equations:
\(=\mathrm{N}\) ': A * \(\mathrm{X}=\mathrm{B} \quad\) ( o transpose)
\(=T\) ': A ** \(T * X=B \quad(T\) ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (Conjugate transpose)
TRANSA is defaulted to N 'forF95 \(\mathbb{I N}\) TERFACE.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS >=0.

LOW (input)
The \((n-1)\) subdiagonalelem ents of .

D IA G (input)
The diagonalelem ents ofA .

UP (input)
The \((n-1)\) superdiagonalelem ents of .
LOW \(F\) (input)
The ( \(n-1\) ) multipliers that define the \(m\) atrix \(L\) from the LU factorization of A as com puted by CGTTRF.

D IA GF (input)
Then diagonalelem ents of the upper triangular \(m\) atrix \(U\) from the LU factorization ofA.

UPF1 (input)
The \((n-1)\) elem ents of the first superdiagonal of U.

UPF2 (input)
The \((n-2)\) elem ents of the second superdiagonal of U.
\(\mathbb{P I V O T}\) (input)
The pivotindioes; for \(1<=i<=n\), row \(i\) of the \(m\) atrix \(w a s\) interchanged \(w\) th row \(\mathbb{P I V O T}(i)\). IPIVOT (i) will always be either \(i\) or i+1; PIVOT (i) = iindicates a row interchange was not required.
\(B\) (input) The righthand side \(m\) atrix \(B\).

\section*{LD B (input)}

The leading dim ension of the array \(B\). LD B >= \(\max (1, N)\).

\section*{X (input/output)}

O \(n\) entry, the solution \(m\) atrix \(X\), as com puted by CG TTRS. On exit, the im proved solution \(m\) atrix \(X\). LD X (input)

The leading dim ension of the array X. LD X >= \(\max (1, N)\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X()\) (the \(j\) th column of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(\mathrm{X}(\mathcal{)}, \mathrm{FERR}(\mathcal{)}\) ) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{\nu})-X\) TRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{)}\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each
solution vector \(X(\mathcal{i})\) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )

W ORK 2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N N}\) FO \(=-\) i, the i-th argum enthad an ille-
galvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgtsv -solve the equation \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGTSV N,NRHS,LOW,D IAG,UP,B,LDB,INFO)}
D OUBLE COM PLEX LOW (*),D IAG (*),UP (*),B (LDB,*)
\mathbb{NTEGERN,NRHS,LDB,INFO}
SUBROUT\mathbb{NE ZGTSV_64N,NRHS,LOW ,D IAG,UP,B,LDB, NNFO)}
DOUBLE COM PLEX LOW (*),D IAG (*),UP (*),B (LDB,*)
INTEGER*8N,NRHS,LDB,INFO

```

\section*{F95 INTERFACE}

SUBROUTINEGTSV ( \(\mathbb{N}], \mathbb{N} R H S], L O W, D \mathbb{I A G}, \mathrm{UP}, \mathrm{B},[\operatorname{LDB}],[\mathbb{N} F O])\)
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::LOW ,D \(\mathbb{I A G}, \mathrm{UP}\)
COM PLEX (8), D IM ENSION (:,:) :: B
\(\mathbb{N}\) TEGER :: N,NRHS,LDB, \(\mathbb{N}\) FO
SUBROUTINE GTSV_64 ( \(\mathbb{N}], \mathbb{N} R H S], L O W, D \mathbb{A} G, U P, B,[L D B],[\mathbb{N} F O])\)

COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::LOW ,D \(\mathbb{I A G}, \mathrm{UP}\)
COM PLEX (8), D IM ENSION (: : : : : B
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} \mathrm{B}, \mathbb{N} F \mathrm{O}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zgtsv (intn, intnrhs, doublecom plex *low, doublecom plex *diag, doublecom plex *up, doublecom plex *b,
intldb, int*info);
void zgtsv_64 (long n, long nrhs, doublecom plex *low, doublecom plex *diag, doublecom plex *up, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
zgtsv solves the equation
where \(A\) is an \(N\)-by \(\mathrm{N}_{\mathrm{N}}\) tridiagonalm atrix, by G aussian elim ination w ith partialpivoting.

Note that the equation A *X = B may be solved by interchanging the order of the argum ents \(\mathrm{D} U\) and \(\mathrm{D} L\).

\section*{ARGUMENTS}

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of the m atrix B. NRHS \(>=0\).

LOW (input/output)
On entry, LOW m ust contain the ( \(n-1\) ) subdiagonal elem ents ofA. On exit, LOW is overw ritten by the ( \(n-2\) ) elem ents of the second superdiagonal of the upper triangular \(m\) atrix \(U\) from the \(L U\) factorization ofA, in LOW (1), ..., LOW (n-2).

D IA G (input/output)
On entry, D IA G m ust contain the diagonal elem ents of A. On exit, D IA G is overw rilten by the \(n\) diagonalelem ents ofU .

UP (input/output)
O n entry, UP m ust contain the ( \(n-1\) ) superdiagonal elem ents of A. On exit, UP is overw ritten by the \((n-1)\) elem ents of the first superdiagonalof \(U\).

B (input/output)
On entry, the \(N-b y-N R H S\) righthand side \(m\) atrix \(B\).
On exit, if \(\mathbb{N F O}=0\), the \(N\) by-NRH S solution \(m\) atrix \(X\).

The leading dim ension of the aray \(B\). LD B \(>=\) \(\max (1, N)\).
\(\mathbb{N F O}\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{I N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N F O}=\mathrm{i}, \mathrm{U}(i, i)\) is exactly zero, and the solution has notbeen com puted. The factorization has notbeen com pleted unless \(i=N\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgtsvx -use the LU factorization to com pute the solution to a complex system of linearequations \(\mathrm{A} * \mathrm{X}=\mathrm{B}, \mathrm{A} * * \mathrm{~T} * \mathrm{X}=\) \(B\), orA \({ }^{* *} H * X=B\),

\section*{SYNOPSIS}
```

SUBROUTINE ZGTSVX (FACT,TRANSA,N,NRHS,LOW,DIAG,UP,LOW F,DIAGF,
UPF1,UPF2,\mathbb{PIVOT,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,}
W ORK2,\mathbb{NFO)}
CHARACTER * 1 FACT,TRANSA
DOUBLE COM PLEX LOW (*),DIAG (*), UP (*), LOW F (*), D IAGF (*),
UPF1 (*),UPF2 (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,INFO
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION RCOND
D OUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)
SUBROUTINE ZGTSVX_64 (FACT,TRANSA,N,NRHS,LOW,DIAG,UP,LOW F,
D IAGF,UPF1,UPF2,\mathbb{PIVOT,B,LDB,X,LDX,RCOND,FERR,BERR,}
W ORK,W ORK 2, INFO)
CHARACTER * 1 FACT,TRANSA
DOUBLE COM PLEX LOW (*),D IA G (*), UP (*), LOW F (*), D IAGF (*),
UPF1 (*),UPF2 (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGER*8N,NRHS,LDB,LDX,INFO
\mathbb{NTEGER*8 P\mathbb{IVOT (*)}}\mathbf{(})
DOUBLE PRECISION RCOND
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINEGTSVX (EACT, [TRANSA], \(\mathbb{N}], \mathbb{N} R H S], L O W, D \mathbb{A} G, U P, L O W F\),

D \(\mathbb{A} G \mathrm{G}, \mathrm{UPF} 1, \mathrm{UPF} 2, \mathbb{P} \mathbb{I V O T}, \mathrm{~B},[\mathrm{LD} \mathrm{B}], \mathrm{X},[\operatorname{LD}], R C O N D, F E R R, B E R R\), [W ORK], [W ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER ( \(L E N=1\) ) : :FACT,TRANSA
COMPLEX (8), D \(\mathbb{I}\) ENSION (:) :: LOW, D \(\mathbb{A} G\), UP, LOW F, D \(\mathbb{A} G F\),
UPF1,UPF2,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) :: B, X
\(\mathbb{N}\) TEGER ::N,NRHS,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL (8) ::RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

SU BROUTINE GTSVX_64 (FACT, [TRANSA], \(\mathbb{N}], \mathbb{N} R H S], L O W, D I A G, U P, L O W F\), D \(\mathbb{A} G F, U P F 1, U P F 2, \mathbb{P} \mathbb{I V O T}, B,[L D B], X,[L D X], R C O N D, F E R R, B E R R\), [W ORK], [W ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::FACT,TRANSA
COMPLEX (8), D IM ENSION (:) :: LOW, D \(\mathbb{A} G, U P\), LOW F, DIAGF,
UPF1,UPF2,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : B , X
\(\mathbb{N} \operatorname{TEGER}(8):: N, N R H S, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8) :: RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void zgtsvx (char fact, chartransa, intn, int nihs, doublecom plex *low , doublecom plex *diag, doublecom plex *up, doublecom plex *low f, doublecom plex *diagf, doublecom plex *upfl, doublecom plex *upf2, int *ịíivot, doublecom plex *b, int ldb, doublecom plex *x, int ldx, double *roond, double *ferr, double *berr, int*info);
void zgtsvx_64 (char fact, char transa, long n, long nus, doublecom plex *low, doublecom plex *diag, doublecom plex *up, doublecom plex *low f, doublecom plex *diagf, doublecom plex *upfl, doublecom plex *upf2, long *ipìivot, doublecom plex *b, long lab, doublecom plex *x, long ldx, double *rcond, double *ferr, double *berr, long *info);

\section*{PURPOSE}
zgtsvx uses the LU factorization to com pute the solution to a complex system of linearequations \(\mathrm{A} * \mathrm{X}=\mathrm{B}, \mathrm{A} * * \mathrm{~T} * \mathrm{X}=\) \(B\), or \(A * * H\) * \(X=B\), where \(A\) is a tridiagonalm atrix of order

N and X and B are N boy N R H S m atrioes.

E morbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT \(=\mathrm{N}\) ', the LU decom position is used to factor the \(m\) atrix A
as \(A=L * U\), where \(L\) is a product of perm utation and unitlow er
bidiagonal \(m\) atrices and \(U\) is upper triangular \(w\) ith nonzeros in
only the m ain diagonal and firsttw o superdiagonals.
2. If som e \(U(i, i)=0\), so that \(U\) is exactly singular, then the routine
retums w ith \(\mathbb{N} F O=\) i. O therw ise, the factored form of \(A\) is used
to estim ate the condition num ber of the \(m\) atrix \(A\). If the reciprocal of the condition num ber is less than machine precision,
\(\mathbb{N} F O=N+1\) is retumed as a waming, but the routine stillgoes on
to solve forX and com pute error bounds as described below.
3.The system ofequations is solved forX using the factored form of A.
4. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for 五.

\section*{ARGUMENTS}

FACT (input)
Specifies w hether ornot the factored form of A
has been supplied on entry \(.=F ': L O W\) F , D IA GF, UPF1, UPF2, and IPIVOT contain the factored form of \(A\); LOW , D \(\mathbb{A} G\), UP, LOW F, D IA GF, UPF1, UPF2 and \(\mathbb{P} \mathbb{I V}\) O T w illnotbe m odified. \(=\mathrm{N}\) ': The m atrix w ill be copied to LOW F, D IA GF , and UPF1 and factored.

TRANSA (input)
Specifies the form of the system of equations:
\(=N\) ': A * \(\mathrm{X}=\mathrm{B} \quad\) (Notranspose)
\(=T: A * * T * X=B \quad\) ( ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose)
TRANSA is defaulted to N 'forF95 \(\mathbb{I N}\) TERFACE.

N (input) The order of the matrix A. \(\mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber
of colum ns of the m atrix B. NRHS >=0.

LOW (input)
The \((n-1)\) subdiagonalelem ents ofA.
D IA G (input)
The \(n\) diagonalelem ents of \(A\).
UP (input/output)
The ( \(n-1\) ) superdiagonalelem ents of \(A\).
LOW F (input/output)
IfFACT = F ', then LOW F is an inputargum ent and on entry contains the ( \(n-1\) ) multipliers that define the \(m\) atrix \(L\) from the \(L U\) factorization of \(A\) as com puted by CGTTRF.

IfFACT \(=\mathrm{N}\) ', then LOW F is an outputargum ent and on exit contains the ( \(n-1\) ) m ultipliers that define them atrix \(L\) from the LU factorization of A.

D IA GF (input/output)
If \(F A C T=F\) ', then \(D I A G F\) is an input argum ent and on entry contains the n diagonalelem ents of the uppertriangularm atrix \(U\) from the LU factorization ofA.

IfFACT = \(N\) ', then \(D I A G F\) is an output argum ent and on exit contains the \(n\) diagonalelem ents of the upper triangularm atrix \(U\) from the \(L U\) factorization ofA .

UPF1 (input/output)
IfFACT = F ', then UPF 1 is an input argum ent and on entry contains the ( \(n-1\) ) elem ents of the first superdiagonalof \(U\).

IfFA C T = N ', then UPF1 is an outputargum ent and on exit contains the ( \(n-1\) ) elem ents of the first
superdiagonalofU.

UPF2 (input/output)
IfFACT = \(\mathrm{F}^{\prime}\), then UPF 2 is an input argum ent and on entry contains the ( \(n-2\) ) elem ents of the second superdiagonalofU .

IfFACT = \(N\) ', then UPF2 is an output argum ent and on exit contains the ( \(n-2\) ) elem ents of the second superdiagonalofU.

IPIVOT (input/output)
IfFACT= \(\mathrm{F}^{\prime}\), then \(\mathbb{P} \mathbb{I V O T}\) is an input argum ent and on entry contains the pivot indices from the LU factorization of A as com puted by CGTTRF.

IfFACT \(=\mathrm{N}\) ', then \(\mathbb{P} \mathbb{I V O T}\) is an output argum ent and on exit contains the pivot indioes from the LU factorization of ; row iof the \(m\) atrix \(w\) as inter changed w ith row \(\mathbb{P I V O T}\) (i). \(\mathbb{P} \mathbb{I V O T}\) (i) w illalw ays be either ior i+ \(1 ; \mathbb{P} \mathbb{V} O T(i)=i\) indicates a row interchange w as not required.
\(B\) (input) The \(N\) by -N RH \(S\) righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the aray \(B\). LD B \(>=\) \(\max (1, N)\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=N+1\), the \(N\)-by-NRH \(S\) solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the aray X . LD X >= \(\max (1, N)\).

RCOND (output)
The estim ate of the reciprocal condtion num ber of the \(m\) atrix \(A\). IfRCOND is less than the \(m\) achine precision (in particular, if RCOND \(=0\) ), the \(m\) atrix is singular to working precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO > 0 .

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X()\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{H}), \operatorname{FERR}(\mathcal{)})\) is an estim ated
upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(\mathrm{X}(\mathcal{j})\)-XTRUE) divided by the magnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vectorX (i) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )

W ORK 2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N}\) FO (output)
= 0 : successfulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvahue
\(>0\) : if \(\mathbb{N F O}=i\), and \(i\) is
\(<=\mathrm{N}: \mathrm{U}(i, i)\) is exactly zero. The factorization has notbeen com pleted unless \(i=N\), but the factorU is exactly singular, so the solution and error bounds could notbe com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1\) : U is nonsingular, but RCOND is less than \(m\) achine precision, \(m\) eaning that the \(m\) atrix is singular to working precision. N evertheless, the solution and emorbounds are com puted because there are a num ber of situations where the com puted solution can be m ore accurate than the value ofRCON D w ould suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgturf-com pute an LU factorization of a com plex tridiagonalm atrix A using elim ination \(w\) ith partialpivoting and row interchanges

\section*{SYNOPSIS}

```

DOUBLE COM PLEX LOW (*),D IAG (*),UP1 (*),UP2 (*)
INTEGERN,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}

```

```

DOUBLE COM PLEX LOW (*),D IAG (*),UP1 (*),UP2 (*)
INTEGER*8N,\mathbb{NFO}
\mathbb{NTEGER *8 \mathbb{PIVOT (*)}}\mathbf{(})

```

\section*{F95 INTERFACE}
```

SU BROUTINE GTTRF ( $\mathbb{N}$ ],LOW ,D $\mathbb{I A}, \mathrm{UP} 1, \mathrm{UP} 2, \mathbb{P} \mathbb{I V O T},[\mathbb{N} F O]$ )
COM PLEX (8),D $\mathbb{I M} E N S \mathbb{I O N}(:):: L O W, D \mathbb{I A G}, \mathrm{UP} 1, \mathrm{UP} 2$
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M}$ ENSION (:) :: $\mathbb{P} \mathbb{I V} O T$
SU BROUTINE GTTRF_64 (N ],LOW,D $\mathbb{I A G}, \mathrm{UP} 1, \mathrm{UP} 2, \mathbb{P} \mathbb{I} \operatorname{OT},[\mathbb{N} F O$ ])
COM PLEX (8), D $\mathbb{M}$ ENSION (:) ::LOW ,D IA G,UP1, UP2
$\mathbb{N}$ TEGER (8) ::N, $\mathbb{N} F O$
$\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T$

```
\#include < sunperfh>
void zgturf(intn, doublecom plex *low, doublecom plex *diag, doublecomplex *up1, doublecomplex *up2, int *ípívot, int*info);
void zgttrf_ 64 long \(n\), doublecom plex *low , doublecom plex *diag, doublecom plex *up1, doublecom plex *up2, long *ipivot, long *info);

\section*{PURPOSE}
zgttrf com putes an LU factorization of a com plex tridiagonal \(m\) atrix A using elim ination w ith partialpivoting and row interchanges.
The factorization has the form
\[
A=L * U
\]
where \(L\) is a product of perm utation and unit low er bidiagonalm atrices and \(U\) is upper triangularw ith nonzeros in only the \(m\) ain diagonal and first tw o superdiagonals.

\section*{ARGUMENTS}

N (input) The order of the m atrix A.
LOW (input/output)
On entry, LOW m ustcontain the ( \(n-1\) ) sub-diagonal elem ents ofA.

On exit, LOW is overw ritten by the ( \(n-1\) ) multipliers that define the \(m\) atrix \(L\) from the \(L U\) factorization of A.

D IA G (input/output)
On entry, D IA G m ust contain the diagonal elem ents of A.

On exit, D IA G is overw rilten by the n diagonal elem ents of the upper triangularm atrix \(U\) from the LU factorization ofA.

UP1 (input/output)
On entry, UP1 must contain the ( \(n-1\) ) superdiagonalelem ents ofA.

On exit, UP1 is overw rilten by the ( \(n-1\) ) elem ents of the first super-diagonalof \(U\).

UP2 (output)
On exit, UP2 is overw rilten by the ( \(n-2\) ) elem ents of the second super-diagonalofU .

IPIVOT (output)
The pivotindioes; for \(1<=i<=n\), row \(i\) of the \(m\) atrix \(w a s\) interchanged \(w\) th row \(\mathbb{P I V O T}\) (i). PIVOT (i) will alw ays be either \(i\) or i+1; PIVOT (i) = iindicates a row interchange \(w\) as not required.
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
< 0: if \(\mathbb{N} F O=-\mathrm{k}\), the k -th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=k, U(k, k)\) is exactly zero. The factorization has been com pleted, but the factor \(U\) is exactly singular, and division by zero will occur if it is used to solve a system ofequations.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zgttrs - solve one of the system sof equations \(A * X=B\), \(\mathrm{A} * * \mathrm{~T} * \mathrm{X}=\mathrm{B}\),orA \({ }^{*}{ }^{*} \mathrm{H} * \mathrm{X}=\mathrm{B}\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZGTTRS (TRANSA,N,NRHS,LOW ,D IAG,UP1,UP2, IPIVOT,B,}
LDB,INFO)
CHARACTER * 1 TRANSA
DOUBLE COM PLEX LOW (*),D IAG (*),UP1 (*),UP2 (*),B (LDB ,*)
\mathbb{NTEGERN,NRHS,LDB,NNFO}
INTEGER \mathbb{PIVOT (*)}
SU BROUTINE ZGTTRS_64 (TRANSA,N,NRHS,LOW ,D IA G,UP1,UP2, \mathbb{PIVOT,B,}
LDB,\mathbb{NFO)}
CHARACTER * 1 TRANSA
DOUBLE COM PLEX LOW (*),D IAG (*),UP1 (*),UP2 (*),B (LDB,*)
\mathbb{NTEGER*8N,NRHS,LDB,INFO}
INTEGER*8 \mathbb{PIVOT (*)}

```

\section*{F95 INTERFACE}

SU BROUTINE GTTRS ([TRANSA ], \(\mathbb{N}], \mathbb{N} R H S], L O W, D I A G, U P 1, U P 2, \mathbb{P} \mathbb{I V O T}\), B, [LDB], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::TRANSA
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::LOW ,D IA G ,UP1, UP2
COM PLEX (8),D IM ENSION (:,:) :: B
\(\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V} O T\)
SU BROUTINE GTTRS_64 ([TRANSA], \(\mathbb{N}], \mathbb{N} R H S], L O W, D I A G, U P 1, U P 2\),
\(\mathbb{P} \mathbb{V} O T, B,[\operatorname{LDB}],[\mathbb{N F O}])\)

CHARACTER (LEN=1) ::TRANSA
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::LOW ,D \(\mathbb{A} G, U P 1, U P 2\)
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : \(:\) B
\(\mathbb{N}\) TEGER (8) :: N,NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zgturs (chartransa, int \(n\), int nihs, doublecom plex *low, doublecom plex *diag, doublecom plex *up1, doublecom plex *up2, int *ịívot, doublecom plex *b, intlab, int*info);
void zgttrs_64 (chartransa, long n, long nihs, doublecom plex *low, doublecom plex *diag, doublecom plex *up1, doublecom plex *up2, long *ịìivot, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
zgturs solves one of the system sof equations
\(A * X=B, A * * T * X=B\), or \(A * * H * X=B\), with a tridiagonal \(m\) atrix A using the LU factorization com puted by CGTTRF.

\section*{ARGUMENTS}

TRANSA (input)
Specifies the form of the system of equations. =
N ': A * X = B N o transpose)
\(=T: A * * T * X=B \quad\) (Transpose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose)
TRANSA is defaulted to N 'forF95 \(\mathbb{I N}\) TERFACE.

N (input) The order of the m atrix A.
NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS \(>=0\).

LOW (input)
The \((n-1) m\) ultipliers that define the \(m\) atrix \(L\) from the LU factorization ofA.

D IA G (input)
The n diagonalelem ents of the upper triangular
\(m\) atrix \(U\) from the LU factorization ofA.
UP1 (input)
The \((n-1)\) elem ents of the first super-diagonal of U.

UP2 (input)
The ( \(n-2\) ) elem ents of the second super-diagonal of U.
\(\mathbb{P I V O T}\) (input)
The pivotindices; for \(1<=\mathrm{i}<=\mathrm{n}\), row i of the \(m\) atrix was interchanged w ith row \(\mathbb{P I V O T}\) (i). IPIVOT (i) will always be either \(i\) or i+1; PIV T (i) = iindicates a row interchange \(w\) as not required.

B (input/output)
O \(n\) entry, the \(m\) atrix of righthand side vectors \(B\). On exit, \(B\) is overw ritten by the solution vectors X.

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N F O}=-\mathrm{k}\), the k -th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhbev - com pute all the eigenvalues and, optionally, eigenvectors of a com plex Herm itian band m atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHBEV (JOBZ,UPLO,N,KD,A,LDA,W ,Z,LD Z,W ORK,}
W ORK2,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEX A (LDA,*),Z (LD Z,*),W ORK (*)
INTEGERN,KD,LDA,LD Z,INFO
DOUBLE PRECISION W (*),W ORK2 (*)
SUBROUTINE ZHBEV_64(JOBZ,UPLO,N,KD,A,LDA,W ,Z,LD Z,W ORK,
W ORK2,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEX A (LDA,*),Z (LD Z,*),W ORK (*)
\mathbb{NTEGER*8N,KD,LDA,LD Z,INFO}
DOUBLE PRECISION W (*),W ORK2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE HBEV ( \(\mathrm{OB} \mathrm{B}, \mathrm{UPLO}, \mathbb{N}], \mathrm{KD}, \mathrm{A},[\operatorname{LDA}], \mathrm{W}, \mathrm{Z},[\mathrm{LD} Z],[W \mathrm{ORK}]\), [W ORK 2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1)::JOBZ,UPLO
COM PLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, Z
\(\mathbb{N} T E G E R:: N, K D, L D A, L D Z, \mathbb{N} F O\)
REAL (8),D \(\mathbb{I}\) ENSION (:) ::W ,W ORK 2
SU BROUTINE HBEV_64 (JOBZ, UPLO , \(\mathbb{N}], K D, A,[L D A], W, Z,[L D Z]\),
[W ORK], [W ORK2], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1)::JOBZ, UPLO
COMPLEX (8), D IM ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : ) : : A , Z
\(\mathbb{N} T E G E R(8):: N, K D, L D A, L D Z, \mathbb{N} F O\)
REAL (8), D IM ENSION (:) ::W ,W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zhbev (char jobz, charuplo, intn, int kd, doublecom plex *a, int lda, double *w, doublecom plex *z, int ldz, int*info);
void zhbev_64 (char jojbz, char uplo, long n, long kd, doublecom plex *a, long lda, double *w, doublecom plex *z, long ldz, long *info);

\section*{PURPOSE}
zhbev com putes all the eigenvalues and, optionally, eigenvectors of a com plex H erm itian band m atrix A .

\section*{ARGUMENTS}

JOBZ (input)
\(=N^{\prime}:\) C om pute eigenvalues only;
\(=\mathrm{V}\) ': C om pute eigenvalues and eigenvectors.
UPLO (input)
= U ': U pper triangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
KD (input)
The num ber of superdiagonals of the \(m\) atrix A if UPLO = U',orthe num berof subdiagonals ifU PLO \(=\mathrm{L} \cdot \mathrm{KD}>=0\).

A (input/output)
O n entry, the upper or low ertriangle of the Her \(m\) itian band \(m\) atrix \(A\), stored in the firstK \(D+1\) row s of the amay. The \(j\) th colum \(n\) of A is stored in the \(j\) th colum n of the a may \(A\) as follow \(s\) : if \(\mathrm{UPLO}=\mathrm{U}, \mathrm{A}(\mathrm{kd}+1+i-j)=A(i, j)\) for \(\mathrm{max}(1, j\) \(\mathrm{kd})<=i<=j\) if UPLO \(=\mathrm{L}, \mathrm{A}(1+i-j, j)=A(i, j)\)
for \(\dot{j}=i<=m\) in \((n, j+k d)\).
On exit, A is overw rilten by values generated during the reduction to tridiagonal form. If P PLO = U ', the first superdiagonal and the diagonal of the tridiagonal \(m\) atrix \(T\) are retumed in row sK D and \(K D+1\) ofA, and if \(U P L O=L\) ', the diagonaland first subdiagonal of T are retumed in the first tw o row sofA.

LD A (input)
The leading dim ension of the array A. LD A >= KD + 1.

W (output) If \(\mathbb{N} F O=0\), the eigenvalues in ascending order.

Z (input) If \(\mathrm{JOBZ}=\mathrm{V}^{\prime}\), then if \(\mathbb{N F O}=0, \mathrm{Z}\) contains the orthonom aleigenvectors of the \(m\) atrix \(A, w\) ith the \(i\)-th colum \(n\) of \(Z\) holding the eigenvector associated w th \(W\) (i). If \(\mathrm{JOBZ}=\mathrm{N}\) ', then Z is not referenced.

LD Z (input)
The leading dim ension of the amray Z . LD \(\mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD Z \(>=\mathrm{max}(1, N)\).

W ORK (w orkspace)
dim ension (N)
W ORK 2 (w orkspace)
dim ension (max (1,3*N-2))
\(\mathbb{N} F O\) (output)
= 0 : successfulexit.
<0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvahue.
> 0 : if \(\mathbb{N} F O=\) i, the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhbevd - com pute all the eigenvalues and, optionally, eigenvectors of a com plex \(H\) erm itian band \(m\) atrix \(A\)

\section*{SYNOPSIS}

SU BROUTINE ZHBEVD (JOBZ, UPLO,N,KD,AB,LDAB,W, Z,LDZ,WORK, LW ORK,RW ORK,LRW ORK, IW ORK,LIW ORK, \(\mathbb{N} F O\) )

CHARACTER * 1 JOBZ, UPLO
D OUBLE COM PLEXAB (LDAB, *), Z (LDZ, \({ }^{*}\) ), W ORK ( \({ }^{*}\) )
\(\mathbb{N} T E G E R N, K D, L D A B, L D Z, L W\) ORK,LRW ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{I N}\) ORK (*)
DOUBLE PRECISION W (*), RW ORK (*)
SU BROUTINE ZHBEVD_64 (JOBZ,UPLO ,N,KD,AB,LDAB,W,Z,LDZ,WORK, LW ORK,RW ORK,LRW ORK, \(\mathbb{I W} O R K, L \mathbb{I} O R K, \mathbb{N} F O\) )

CHARACTER * 1 JOBZ, UPLO

\(\mathbb{N}\) TEGER*8N,KD,LDAB,LDZ,LW ORK,LRW ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER*8 \(\mathbb{I N}\) ORK (*)
DOUBLE PRECISION W (*),RWORK (*)

\section*{F95 INTERFACE}

SUBROUTINE HBEVD (OBZ,UPLO, \(\mathbb{N}], K D, A B,[L D A B], W, Z,[L D Z],[W\) ORK], \([\) [LW ORK ], \(\mathbb{R W}\) ORK ], [LRW ORK ], [ \(\mathbb{W}\) ORK ], [LIN ORK ], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1)::JOBZ,UPLO
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COMPLEX (8), D \(\mathbb{I M}\) ENSION (:,:) ::AB, Z
\(\mathbb{N} T E G E R:: N, K D, L D A B, L D Z, L W O R K, L R W O R K, L \mathbb{I N} O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I}\) ENSION (:) :: \(\mathbb{I N}\) ORK

SU BROUTINE HBEVD_64 (JOBZ, UPLO, \(\mathbb{N}\) ],KD, AB, [LDAB],W , Z, [LD Z], \([\mathbb{W}\) ORK ], [LW ORK ], [RW ORK ], [LRW ORK ], [IW ORK ], [LIW ORK ], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1)::JOBZ,UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COMPLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::AB, Z
\(\mathbb{N}\) TEGER (8) :: N, KD,LDAB,LD Z,LW ORK,LRW ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{I W}\) ORK
REAL (8), D IM ENSION (:) ::W ,RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zhbevd (char jobz, charuplo, intn, int kd, doublecom plex *ab, intldab, double *w, doublecom plex *z, int ldz, int*info);
void zhbevd_64 (char jobz, charuplo, long n, long kd, doublecom plex *ab, long ldab, double *w , doublecom plex *z, long Idz, long *info);

\section*{PURPOSE}
zhbevd com putes all the eigenvalues and, optionally, eigenvectors of a com plex H erm titian band \(m\) atrix A. Ifeigenvectors are desired, ituses a divide and conquer algorithm .

The divide and conquer algorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on \(m\) achines w ith a guard digit in add/subtract, or on those binary \(m\) achines \(w\) thout guard digits w hich subtract like the C ray \(X-M P, C\) ray Y M P , C ray C-90, or C ray-2. It could conceivably fail on hexadecim al or decim al machines w thout guard digits, butw e know of none.

\section*{ARGUMENTS}

JOBZ (input)
\(=N^{\prime}:\) C om pute eigenvalues only;
\(=\mathrm{V}\) ': C om pute eigenvalues and eigenvectors.
UPLO (input)
= U ': U pper triangle ofA is stored;
= L' : Low ertriangle ofA is stored.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
\(K D\) (input)
The num ber of superdiagonals of the \(m\) atrix A if UPLO \(=\mathrm{U}\) ', orthe num berof subdiagonals if UPLO \(=L^{\prime} . K D>=0\).

AB (input/output)
O \(n\) entry, the upper or low er triangle of the Her\(m\) itian band \(m\) atrix \(A\), stored in the firstKD +1 row sof the array. The \(j\) th colum n of \(A\) is stored in the \(j\) th colum \(n\) of the array \(A B\) as follow \(s\) : if \(\mathrm{UPLO}=\mathrm{U}, \mathrm{AB}(\mathrm{kd}+1+\mathrm{i}-j)=\mathrm{j}(i, 1)\) for \(\mathrm{max}(1, j\) \(\mathrm{kd})<=i<=\dot{j}\) if \(\mathrm{UPLO}=\mathrm{L}, \mathrm{AB}(1+i-j, 7)=A(i, j)\) for \(j=i<=m\) in \((n, j+k d)\).
On exit, AB is overw rilten by values generated during the reduction to tridiagonal form. IfU PLO \(=\mathrm{U}\) ', the first superdiagonal and the diagonal of the tridiagonal m atrix T are retumed in row sKD and \(K D+1\) of \(A B\), and if \(U P L O=L\) ', the diagonal and first subdiagonal of \(T\) are retumed in the first tw o row sofA B .

LDAB (input)
The leading dim ension of the array AB. LD A B > \(>=K D\) +1 .

W (output) If \(\mathbb{N} F O=0\), the eigenvalues in ascending order.
\(Z\) (input) If \(\mathcal{O B Z}=\mathrm{V}\) ', then if \(\mathbb{N} F O=0, \mathrm{Z}\) contains the orthonorm aleigenvectors of the \(m\) atrix \(A, w\) ith the i-th colum \(n\) of \(Z\) holding the eigenvector associated \(w\) th \(W\) (i). If \(\mathrm{JOBZ}=\mathrm{N}\) ', then Z is not referenced.

LD \(Z\) (input)
The leading din ension of the array \(\mathrm{Z} . \operatorname{LD} \mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD Z \(>=\mathrm{max}(1, N)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F} F=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array W ORK. If \(\mathrm{N}<=1\), LW ORK must be at least1. If \(J 0 B Z=N\) 'and \(N>\) 1, LW ORK m ust.be at leastN. If JO BZ \(=\mathrm{V}\) 'and N > 1, LW ORK m ustbe at least \(2 * N\) **2.

IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

RW ORK (w orkspace)
dim ension (LRW ORK) On exit, if \(\mathbb{I N} F O=0, R W O R K(1)\)
retums the optim allRW ORK.
LRW ORK (input)
The dimension of array RW ORK. If \(\mathrm{N}<=1\), LRW ORK mustibe at least 1. If \(\mathrm{JOBZ}=\mathrm{N}\) 'and \(\mathrm{N}>\) 1, LRW ORK m ustbe at leastN. If JO B Z \(=\mathrm{V}\) 'and N > 1,LRW ORK m ust.be at least1 + 5*N + 2*N **2.

If LRW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the RW ORK amray, retums this value as the first entry of the RW ORK amay, and no enorm essage related to LRW ORK is issued by X ERBLA.

IV ORK (w orkspace/output)
On exit, if \(\mathbb{N}\) FO \(=0, \mathbb{I N}\) ORK (1) retums the optim al LIN ORK.

LIN ORK (input)
The dim ension of array \(\mathbb{I N}\) ORK. If JOBZ = N 'or N \(<=1, \mathrm{LIV} O R K\) mustbe at least1. If \(\mathrm{JOBZ}=\mathrm{V}^{\prime}\) and \(N>1, L \mathbb{I}\) ORK m ustbe at least \(3+5{ }^{*} N\).

If \(L \mathbb{I V} O R K=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the \(\mathbb{I W}\) ORK array, retums this value as the first entry of the \(\mathbb{I W} O R K\) array, and no errorm essage related to LIIN ORK is issued by X ERBLA.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit.
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvahue. > 0 : if \(\mathbb{N} F O=\) i, the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhbevx - com pute selected eigenvalues and, optionally, eigenvectors of a com plex Herm itian band m atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHBEVX (JOBZ,RANGE,UPLO,N,KD,A,LDA,Q,LDQ,VL,}
VU,\mathbb{L},\mathbb{U},ABTOL,NFOUND,W,Z,LDZ,W ORK,W ORK2,\mathbb{IN ORK 3, FFA IL,}
\mathbb{NFO)}
CHARACTER * 1 JOBZ,RANGE,UPLO
DOUBLE COM PLEX A (LDA,*),Q (LDQ,*),Z (LD Z,*),W ORK (*)
INTEGERN,KD,LDA,LDQ,\mathbb{I},\mathbb{U},NFOUND,LDZ,\mathbb{NFO}
\mathbb{NTEGER IN ORK3(*),\mathbb{FA L (*)}}\mathbf{(*)}
DOUBLE PRECISION VL,VU,ABTOL
DOUBLE PRECISION W (*),W ORK2 (*)
SUBROUT\mathbb{NE ZHBEVX_64(JOBZ,RANGE,UPLO,N,KD,A,LDA,Q,LDQ,VL,}
VU,\mathbb{L},\mathbb{U},ABTOL,NFOUND,W,Z,LD Z,W ORK,W ORK2,\mathbb{W}ORK 3, \mathbb{FA}\mathbb{I},
INFO)

```
CHARACTER * 1 JOBZ,RANGE,UPLO

\(\mathbb{N} T E G E R * 8 N, K D, L D A, L D Q, \mathbb{L}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{I N}\) ORK 3 (*), \(\mathbb{F A} \mathbb{I}\) (*)
DOUBLE PRECISION VL,VU,ABTOL
DOUBLE PRECISION W (*), W ORK 2 (*)

\section*{F95 INTERFACE}

SU BROUTINE HBEVX (JOBZ,RANGE,UPLO, \(\mathbb{N}\) ],KD, A, [LDA], \(\mathrm{Q},[\mathrm{LD} \mathrm{Q}]\), VL,VU, \(\mathbb{I}, \mathbb{I}, A B T O L, \mathbb{N F O U N D}], W, Z,[L D Z],[W O R K],[W O R K 2]\), [ \(\mathbb{I W}\) ORK3], \(\mathbb{F A} \mathbb{I},[\mathbb{N} F O]\) )

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, \(\mathrm{Q}, \mathrm{Z}\)
\(\mathbb{N} T E G E R:: N, K D, L D A, L D Q, \mathbb{L}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{I N}\) ORK 3, \(\mathbb{F} A \mathbb{I}\)
REAL (8) ::VL,VU,ABTOL
REAL (8),D IM ENSION (:) ::W ,W ORK2

SU BROUTINE HBEVX_64 (OOBZ,RANGE,UPLO, \(\mathbb{N}], K D, A,[L D A], Q,[L D Q]\), VL,VU, \(\mathbb{I}, \mathbb{U}, A B T O L, \mathbb{N} F O U N D], W, Z,[L D Z],[W O R K],[W O R K 2]\), [ \(\mathbb{V}\) ORK3], \(\mathbb{F A} \mathbb{L},[\mathbb{N} F O]\) )

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\operatorname{IM}\) ENSION (:,:) ::A, \(Q, Z\)
\(\mathbb{N}\) TEGER (8) :: N , KD, LDA, LDQ, \(\mathbb{I}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{I V}\) ORK 3, \(\mathbb{F A} \mathbb{I}\)
REAL (8) ::VL,VU, ABTOL
REAL (8),D IM ENSION (:) ::W ,W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zhbevx (char jobz, char range, charuplo, intn, int kd, doublecom plex *a, int lda, doublecom plex *q, int ldq, double vl, double vu, int il, int in, double abtol, int *nfound, double * \({ }_{\mathrm{w}}\), doublecom plex *z, intldz, int *ifail, int*info);
void zhbevx_64 (char jंbz, char range, char uplo, long n, long kd, doublecom plex *a, long lda, doublecom plex *q, long ldq, double vl, double vu, long il, long iu, double abtol, long *nfound, double *w , doublecom plex *z, long ldz, long *ifail, long *info);

\section*{PURPOSE}
zhbevx com putes selected eigenvalues and, optionally, eigenvectors of a com plex \(H\) erm itian band matrix A. Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

\section*{ARGUMENTS}

JOBZ (input)
\(=\mathrm{N}\) ': C om pute eigenvahues only;
\(=\mathrm{V}\) : C om pute eigenvalues and eigenvectors.

RANGE (input)
= A : alleigenvalues w illibe found;
\(=\mathrm{V}\) ::alleigenvalues in the half-open interval
(NL,VU] will be found; = 'I': the \(\mathbb{L}\)-th through
\(\mathbb{I J}\)-th eigenvaluesw illlbe found.

UPLO (input)
= U ': U ppertriangle ofA is stored;
\(=\mathrm{L}\) ': Low er triangle of A is stored.
N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
KD (input)
The num ber of superdiagonals of the \(m\) atrix A if UPLO = U', orthe num berof subdiagonals if UPLO \(=\mathbb{L}^{\prime} . \mathrm{KD}>=0\).

A (input/output)
On entry, the upper or low er triangle of the Her\(m\) itian band \(m\) atrix \(A\), stored in the firstKD +1 row sof the array. The \(j\) th colum n of A is stored in the \(j\) th colum \(n\) of the amay A as follows: if \(\mathrm{UPLO}=\mathrm{U}, \mathrm{A}(\mathrm{kd}+1+\mathrm{i}-j)=\mathrm{A}(i, j)\) for max \((1, j\) \(\mathrm{kd})<=i<=j\) if \(\mathrm{UPLO}=\mathrm{L}, \mathrm{A}(1+i-j, j)=A(i, j)\) for \(j=i<=m\) in \((n, j+k d)\).

On exit, A is overw rilten by values generated during the reduction to tridiagonal form .

LD A (input)
The leading dim ension of the amray A. LD A >=KD + 1.

Q (output)
If \(\mathrm{JOBZ}=\mathrm{V}\) ', the N -by-N unitary m atrix used in the reduction to tridiagonal form. If \(J 0 B Z=N\) ', the array Q is not referenced.

LD Q (input)
The leading dim ension of the array \(Q\). If \(\operatorname{JOBZ}=\) V ', then LD \(Q>=\max (1, N)\).

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU.
N ot referenced ifRANGE = A 'or I'.
VU (input)

IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRA N G E = A' 'or I'.

II (input)
IfRA N G E= I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{U}<=\mathbb{U}<=N\), if \(N>0\); \(\mathbb{L}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N otreferenced ifRANGE= A 'or V'.
UU (input)
IfRA N G E= I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}\), if \(\mathrm{N}>0 ; \mathbb{L}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N otreferenced ifRANGE= A 'or V'.

ABTOL (input)
The absolute error tolerance for the eigenvalues. A \(n\) approxim ate eigenvalue is accepted as converged when it is determ ined to \(l i e\) in an interval [a,b]
of w idth less than orequal to
ABTOL + EPS * \(\max (k|,||\),\() ,\)
where EPS is them achine precision. If ABTOL is less than or equal to zero, then EPS* \(\mid\) |w illbe used in its place, where \(T\) | is the 1 -norm of the tridiagonal m atrix obtained by reducing \(A\) to tridiagonalform .

E igenvalues w illbe com puted m ost accurately when
ABTOL is set to tw ice the underflow threshold \(2 * S L A M C H\) ( \({ }^{\eta}\) ), not zero. If this routine retums w ith \(\mathbb{N}\) FO \(>0\), indicating that som e eigenvectors did not converge, try setting ABTOL to 2 *SLAM CH (S ).

See "C om puting Sm allSingularV alues ofB idiagonal M atrices w ith G uaranteed H igh Relative A ccuracy," by D em m eland K ahan, LA PA CK W orking N ote \#3.

NFOUND (output)
The total num ber of eigenvalues found. \(0<=\)
NFOUND <= N. IfRANGE = A',NFOUND = N, and if RANGE = 'I',NFOUND = \(\mathbb{U}-\mathbb{L}+1\).

W (output)
The firstNFOUND elem ents contain the selected eigenvalues in ascending order.
\(Z\) (input) If \(J O B Z=V '\), then if \(\mathbb{N} F O=0\), the first \(N F O U N D\) colum ns of \(Z\) contain the orthonorm aleigenvectors of them atrix A corresponding to the selected eigenvalues, \(w\) ith the \(i\)-th colum \(n\) of \(Z\) holding the eigenvector associated w ith \(W\) (i). If an eigenvector fails to converge, then that colum n of \(Z\) contains the latest approxim ation to the eigenvector, and the index of the eigenvector is retumed in FAII. If \(J O B Z=N\) ', then \(Z\) is not referenced.
\(N\) ote: the user must ensure that at least
\(\max (1, N F O U N D)\) colum ns are supplied in the anray \(Z\); if RANGE = V', the exact value ofNFOUND is not know \(n\) in advance and an upperbound \(m\) ustbe used.

LD Z (input)
The leading \(d i m\) ension of the array \(Z\). LD \(Z>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{max}(1, \mathrm{~N})\).

W ORK (w orkspace)
dim ension (N)

W ORK 2 (w orkspace)
dim ension ( \(7 * \mathrm{~N}\) )

IV ORK 3 (w orkspace)
dim ension ( \(5 * \mathrm{~N}\) )
FFA II (output)
If \(\mathrm{JOBZ}=\mathrm{V}\) ', then if \(\mathbb{N F O}=0\), the first NFOUND
elem ents of \(\mathbb{F A} I I\) are zero. If \(\mathbb{N} F O>0\), then
FAII contains the indices of the eigenvectors
that failed to converge. If \(\mathrm{JOBZ}=\mathrm{N}\) ', then
\(\mathbb{F A} \mathbb{I L}\) is not referenced.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
<0: if \(\mathbb{I N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
> 0 : if \(\mathbb{N F O}=\) i, then ieigenvectors failed to converge. Their indioes are stored in array ㅍFAI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhbgst-reduce a com plex H erm itian-definite banded generalized eigenproblem \(\mathrm{A} * \mathrm{x}=\operatorname{lam}\) bda* \(\mathrm{B} * \mathrm{x}\) to standard form \(\mathrm{C} * \mathrm{y}=\) lam bda*y,

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHBGST NECT,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,X,LDX,}
W ORK,RW ORK,INFO)
CHARACTER * 1 VECT,UPLO
DOUBLE COM PLEX AB (LDAB,*),BB (LDBB,*),X (LDX,*),W ORK (*)
\mathbb{NTEGERN,KA,KB,LDAB,LDBB,LDX,}\mathbb{N}FO
DOUBLE PRECISION RW ORK (*)
SUBROUTINE ZHBGST_64NECT,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,X,
LDX,W ORK,RW ORK,\mathbb{NFO)}
CHARACTER * 1 VECT,UPLO
D OUBLE COM PLEX AB (LDAB,*),BB (LDBB,*),X (LDX,*),W ORK (*)
INTEGER*8N,KA,KB,LDAB,LDBB,LDX,INFO
DOUBLE PRECISION RW ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE HBGST \(N E C T, U P L O, ~ \mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B], X\), [LDX], [W ORK], [RW ORK], [NFO])

CHARACTER (LEN=1) ::VECT,UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COMPLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::AB, BB, X
\(\mathbb{N} T E G E R:: N, K A, K B, L D A B, L D B B, L D X, \mathbb{N F O}\)
REAL (8),D IM ENSION (:) ::RW ORK

SUBROUTINE HBGST_64 NECT,UPLO, \(\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B]\), \(\mathrm{X},[\mathrm{LD} \mathrm{X}],[\mathrm{W}\) ORK ], \(\mathbb{R W}\) ORK ], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::VECT,UPLO
COM PLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::AB,BB,X
\(\mathbb{N} T E G E R(8):: N, K A, K B, L D A B, L D B B, L D X, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zhbgst(charvect, charuplo, intn, int ka, int kb, doublecom plex *ab, int ldab, doublecom plex *bb, int ldbb, doublecom plex *x, int ldx, int *info);
void zhbgst 64 (charvect, charuplo, long n, long ka, long kb, doublecom plex *ab, long ldab, doublecom plex *bb, long ldbb, doublecom plex *x, long ldx, long *info);

\section*{PURPOSE}
zhbgst reduces a com plex H erm tian-definite banded generalized eigenproblem \(A\) * \(x=\operatorname{lam}\) bda*B *x to standard form \(C\) * \(y=\) lam bda*y, such that \(C\) has the sam e bandw idth as A .

B m usthave been previously factorized as \(S * * H\) *S by CPBSTF, using a splitCholesky factorization. A is overw ritten by C \(=\mathrm{X} * * \mathrm{H} * \mathrm{~A} * \mathrm{X}\), where \(\mathrm{X}=\mathrm{S} * *(-1) * \mathrm{Q}\) and Q is a unitary matrix chosen to preserve the bandw idth of A .

\section*{ARGUMENTS}

VECT (input)
\(=N^{\prime}\) : do not form the transform ation \(m\) atrix \(X\);
\(=\mathrm{V}\) ': form X .

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.
N (input) The order of the m atrices A and \(\mathrm{B} . \mathrm{N}>=0\).

KA (input)
The num ber of superdiagonals of the \(m\) atrix \(A\) if
UPLO = U',orthe num berof subdiagonals ifU PLO
= L'. KA >= 0 .

K B (input)
The num ber of superdiagonals of the \(m\) atrix \(B\) if UPLO = U', orthe num berof subdiagonals if UPLO \(=L^{\prime} . K A>=K B>=0\).

AB (input/output)
O \(n\) entry, the upper or low er triangle of the Her \(m\) tian band \(m\) atrix \(A\), stored in the firstka+1 row sof the array. The \(j\) th colum n of A is stored in the \(j\) th colum \(n\) of the array AB as follow s: if \(\mathrm{UPLO}=\mathrm{U}, \mathrm{AB}(\mathrm{ka}+1+\mathrm{i}-j)=\mathrm{A}(i, j)\) for \(\mathrm{max}(1, j\)
 for \(\dot{j}=i<=m\) in \((n, j+k a)\).
On exit, the transform ed \(m\) atrix \(X * * H * A * X\), stored in the sam e form atasA.

LDAB (input)
The leading dim ension of the array AB. LD AB >= \(K A+1\).

BB (input)
The banded factors from the split Cholesky factorization ofB, as retumed by CPBSTF, stored in the first kb +1 row sof the amay.

\section*{LD BB (input)}

The leading dim ension of the array BB. LD BB >= K B +1 .

\section*{X (output)}

IfVECT = V', the n-by-n matrix X. If VECT = N ', the array X is not referenced.

LD X (input)
The leading dim ension of the aray X . LD X >= \(\max (1, N)\) if VECT \(=V\) ';LD \(X=1\) otherw ise.

W ORK (w orkspace)
dim ension (N)

RW ORK (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the i-th argum enthad an illegalvahue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhbgv -com pute allthe eigenvalues, and optionally, the eigenvectors of a com plex generalized H erm itian-definite banded eigenproblem, of the form \(A * x=(\operatorname{lam} . b d a) * B * x\)

\section*{SYNOPSIS}
```

SUBROUTINE ZHBGV (OOBZ,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,W,Z,
LD Z,W ORK,RW ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEX AB (LDAB,*),BB (LDBB,*),Z (LD Z,*),W ORK (*)
INTEGERN,KA,KB,LDAB,LDBB,LD Z, INFO
DOUBLE PRECISION W (*),RW ORK (*)
SU BROUTINE ZHBGV_64 (JOBZ,UPLO,N,KA,KB,AB,LDAB,BB,LDBB,W,Z,
LD Z,W ORK,RWORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEX AB (LDAB,*),BB (LDBB,*),Z (LD Z,*),W ORK (*)
INTEGER*8N,KA,KB,LDAB,LDBB,LD Z, \mathbb{NFO}
DOUBLE PRECISION W (*),RW ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE HBGV (JOBZ,UPLO, \(\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B], W\), Z, [LD Z], [W ORK], RW ORK], [NFO])

CHARACTER (LEN=1)::JOBZ,UPLO
COM PLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::AB, BB, Z
\(\mathbb{N} T E G E R:: N, K A, K B, L D A B, L D B B, L D Z, \mathbb{N F O}\)
REAL (8),D IM ENSION (:) ::W ,RW ORK

SU BROUTINE HBGV_64 (JOBZ, UPLO, \(\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B]\),
W, Z, [LD Z], [W ORK], RW ORK], [ \(\mathbb{N} F O])\)
CHARACTER (LEN=1):: JOBZ, UPLO
COM PLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::AB,BB,Z
\(\mathbb{N} T E G E R(8):: N, K A, K B, L D A B, L D B B, L D Z, \mathbb{N} F O\)
REAL (8), D IM ENSION (:) ::W ,RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zhbgv (char jobz, charuplo, int n, int ka, int kb, doublecom plex *ab, int ldab, doublecom plex *bb, int ldbb, double *w, doublecom plex \({ }^{*}\) z, int ldz, int*info);
void zhbgv_64 (char jobzz, charuplo, long n, long ka, long kb, doublecom plex *ab, long ldab, doublecom plex *bb, long ldbb, double *w , doublecom plex *z, long ldz, long *info);

\section*{PURPOSE}
zhbgv com putes all the eigenvalues, and optionally, the eigenvectors of a com plex generalized H erm itian-definite banded eigenproblem, of the form \(A * x=(l a m . b d a) * B * x\). Here A and \(B\) are assum ed to be \(H\) erm itian and banded, and \(B\) is also positive definite.

\section*{ARGUMENTS}

JOBZ (input)
\(=\mathrm{N}\) ': C om pute eigenvalues only;
= V ': C om pute eigenvalues and eigenvectors.

UPLO (input)
= U ': U pper triangles of A and B are stored;
\(=\mathrm{L}\) ': Low ertriangles of \(A\) and \(B\) are stored.

N (input) The order of the m atrioes A and \(\mathrm{B} . \mathrm{N}>=0\).

KA (input)
The num berof superdiagonals of the \(m\) atrix \(A\) if UPLO = U',orthe num berof subdiagonals ifUPLO
\(=\mathrm{L} \cdot \mathrm{KA}>=0\).

KB (input)
The num ber of superdiagonals of the \(m\) atrix \(B\) if UPLO = U',or the num berof subdiagonals ifU PLO \(=\mathbb{L} \cdot \mathrm{KB}>=0\).

AB (input/output)
O \(n\) entry, the upper or low er triangle of the Her \(m\) tian band \(m\) atrix \(A\), stored in the firstka+1 row sof the array. The \(j\) th colum n of \(A\) is stored in the jth colum n of the amray AB as follows: if \(\mathrm{UPLO}=\mathrm{U}, \mathrm{AB}(k a+1+i-j)=A(i, j)\) for \(m a x(1, j\) \(\mathrm{ka})<=i<=j ;\) ifUPLO \(=\mathrm{L}, \mathrm{AB}(1+i-j, j)=A(i, 7)\) for \(j=i<=m\) in \((n, j+k a)\).
On exit, the contents of \(A B\) are destroyed.

LD A B (input)
The leading dim ension of the array AB. LD AB >= KA+1.

BB (input/output)
O \(n\) entry, the upper or low ertriangle of the Her \(m\) tian band \(m\) atrix \(B\), stored in the firstkb+1 row sof the array. The \(j\) th colum n of \(B\) is stored in the \(j\) th colum \(n\) of the amay BB as follow s: if \(\mathrm{UPLO}=\mathrm{U}, \mathrm{BB}(\mathrm{kb}+1+\mathrm{i}-j, j)=\mathrm{B}(1, j)\) for \(\mathrm{max}(1, j\) \(\mathrm{kb})<=\mathrm{i}<=\dot{j}\) if \(\mathrm{UPLO}=\mathrm{L}, \mathrm{BB}(1+i-j, j)=B(i, 7)\) for \(j<=i<=m\) in \((n, j+k b)\).

On exit, the factorS from the splitCholesky factorization \(\mathrm{B}=\mathrm{S} * * \mathrm{H} * \mathrm{~S}\), as retumed by CPBSTF .

LD BB (input)
The leading dim ension of the array \(\mathrm{BB} . \operatorname{LDBB}>=\) KB+1.

W (output)
If \(\mathbb{N F} F=0\), the eigenvalues in ascending order.

Z (input) If \(\mathrm{JOBZ}=\mathrm{V}\) ', then if \(\mathbb{N} F O=0, \mathrm{Z}\) contains the \(m\) atrix \(Z\) ofeigenvectors, \(w\) ith the \(i\)-th colum n of Z holding the eigenvector associated w ith W (i). The eigenvectors are norm alized so thatZ **H *B *Z = I. If \(\mathrm{JOBZ}=\mathrm{N}\) ', then Z is not referenced.

LD Z (input)
The leading din ension of the array Z . LD \(\mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', \(\mathrm{LD} \mathrm{Z}>=\mathrm{N}\).

W ORK (w orkspace)
dim ension \((\mathbb{N})\)

RW ORK (w orkspace)
dim ension ( \(3 * N\) )
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N} F O=-i\), the i-th argum enthad an ille-
galvalue
\(>0:\) if \(\mathbb{N} F O=i\), and iis:
\(<=\mathrm{N}\) : the algorithm failed to converge: i offdiagonal elem ents of an interm ediate tridiagonal form did not converge to zero; \(>\mathrm{N}\) : if \(\mathbb{N} F O=\mathrm{N}\) \(+i\), for \(1<=i<=N\), then CPBSTF
retumed \(\mathbb{N} F O=i: B\) is not positive definite. The factorization ofB could notbe com pleted and no eigenvalues oreigenvectors w ere com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zhbgvd - com pute allthe eigenvalues, and optionally, the eigenvectors of a com plex generalized H erm itian-definite banded eigenproblem, of the form \(A\) *x (lam bda)* \(B\) *x

\section*{SYNOPSIS}

SU BROUTINE ZHBGVD (JOBZ, UPLO,N,KA,KB,AB,LDAB,BB,LDBB,W,Z, LD \(Z, W\) ORK, LW ORK,RW ORK,LRW ORK, \(\mathbb{I N} O R K, L \mathbb{I W} O R K, \mathbb{N} F O\) )

CHARACTER * 1 JOBZ, UPLO
D OUBLE COM PLEX AB (LDAB,*), BB (LDBB,*), Z (LD Z,*), W ORK (*)
\(\mathbb{N}\) TEGER N, KA, KB,LDAB,LDBB,LDZ,LW ORK, LRW ORK, LIN ORK, \(\mathbb{N} \mathrm{FO}\)
\(\mathbb{N}\) TEGER \(\mathbb{I N}\) ORK (*)
DOUBLE PRECISION W (*),RWORK (*)
SU BROUTINE ZHBGVD_64 (JOBZ, UPLO, N, KA, KB, AB, LDAB, BB, LD BB, \(W, Z\), LD \(Z, W\) ORK,LW ORK,RW ORK,LRW ORK, \(\mathbb{I N} O R K, L \mathbb{I W} O R K, \mathbb{N F O})\)

CHARACTER * 1 JOBZ, UPLO
D OUBLE COM PLEX AB (LDAB,*), BB (LDBB,*), Z (LD Z,*), W ORK (*)
\(\mathbb{N}\) TEGER* \(8 \mathrm{~N}, \mathrm{KA}, \mathrm{KB}, \mathrm{LDAB}, L D B B, L D Z, L W\) ORK,LRW ORK,LIN ORK,
\(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER*8 \(\mathbb{I N}\) ORK (*)
DOUBLE PRECISION W (*),RW ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE HBGVD (JOBZ, UPLO, \(\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B], W\), Z, [LD Z], [W ORK ], [LW ORK], RW ORK ], [LRW ORK ], [WW ORK ], [LIW ORK ], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1): : JOBZ, UPLO
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: :) ::AB, BB , Z
\(\mathbb{N}\) TEGER ::N, KA, KB, LDAB, LDBB, LD Z, LW ORK, LRW ORK,
LIW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{I W}\) ORK
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ,RW ORK

SU BROUTINE HBGVD_64 (OBZ, UPLO, \(\mathbb{N}], K A, K B, A B,[L D A B], B B,[L D B B]\), W , Z, [LDZ], [W ORK], [LW ORK], [RW ORK], [LRW ORK], [ \(\mathbb{W}\) ORK], [LIW ORK], [ \(\mathbb{N} \mathrm{FO}]\) )

CHARACTER (LEN=1): : J B Z , UPLO
COM PLEX (8), D \(\mathbb{M} E N S I O N(:):: W O R K\)
COM PLEX (8), D \(\mathbb{M}\) ENSION (: :) ::AB, BB, Z
\(\mathbb{N}\) TEGER (8) :: N , KA , KB, LDAB, LDBB, LD Z, LW ORK, LRW ORK,
L \(\mathbb{W}\) ORK, \(\mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}\) (:) :: IV ORK
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ,RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zhbgvd (char jobz, charuplo, intn, int ka, int kb, doublecom plex *ab, int ldab, doublecom plex *bb, int ldble, double *w, doublecom plex *z, int ldz, int*info);
void zhbovv_ 64 (char jojbz, char uplo, long n, long ka, long kb , doublecom plex *ab, long ldab, doublecom plex *bb, long ldbb, double *w , doublecom plex *z, long ldz, long *info);

\section*{PURPOSE}
zhbgvd com putes all the eigenvalues, and optionally, the eigenvectors of a com plex generalized \(H\) erm itian-definite banded eigenproblem, of the form A *x=(lam bda)*B *x. H ere A and \(B\) are assum ed to be \(H\) erm titian and banded, and \(B\) is also positive definite. If eigenvectors are desired, it uses a divide and conquer algorithm .

The divide and conquer algorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on \(m\) achines with a guard digit in add/subtract, or on those binary \(m\) achines \(w\) ithout guard digits \(w\) hich subtract like the \(C\) ray \(\mathrm{X}-\mathrm{M} P, C\) ray \(Y \mathrm{M} P\), C ray \(\mathrm{C}-90\), or C ray -2 . It could conceivably fail on hexadecim al or decim al machines \(w\) ithout guard digits, butw e know of none.

\section*{ARGUMENTS}

30 BZ (input)
= N ': C om pute eigenvalues only;
\(=\mathrm{V}: \mathrm{C}\) om pute eigenvalues and eigenvectors.

UPLO (input)
\(=\mathrm{U}\) ': U ppertriangles of \(A\) and \(B\) are stored;
\(=L^{\prime}\) : Low er triangles of \(A\) and \(B\) are stored.
N (input) The order of the m atrices A and \(\mathrm{B} . \mathrm{N}>=0\).
KA (input)
The num ber of superdiagonals of the \(m\) atrix A if UPLO \(=\mathrm{U}\) ', or the num berof subdiagonals if UPLO \(=L^{\prime} \cdot \mathrm{KA}>=0\).

KB (input)
The num ber of superdiagonals of the \(m\) atrix \(B\) if UPLO \(=\mathrm{U}\) ', or the num berof subdiagonals ifUPLO \(=\mathbb{L} \cdot \mathrm{KB}>=0\).

AB (input/output)
O n entry, the upper or low ertriangle of the Her\(m\) itian band \(m\) atrix \(A\), stored in the firstka+1 row sof the array. The \(j\) th colum n of A is stored in the jth colum \(n\) of the array \(A B\) as follow \(s\) : if \(\mathrm{UPLO}=\mathrm{U}\) ', AB \((k a+1+i-j)=A(i, j)\) for \(m a x(1, j\) \(\mathrm{ka})<=\mathrm{i}<=j\) if \(\mathrm{HPLO}=\mathrm{L}{ }^{\prime}, \mathrm{AB}(1+i-j, j)=A(i, 7)\) for \(\mathfrak{j}=i<=m\) in \((n, j+k a)\).

On exit, the contents of \(A B\) are destroyed.
LDAB (input)
The leading dim ension of the array \(A B\). LD AB \(>=\) KA+1.

BB (input/output)
O \(n\) entry, the upper or low er triangle of the Her \(m\) itian band \(m\) atrix \(B\), stored in the first kb+1 row sof the array. The \(j\) th colum n of \(B\) is stored in the jth colum \(n\) of the array \(B B\) as follow s: if \(\mathrm{UPLO}=\mathrm{U}, \mathrm{BB}(k b+1+i-j)=\mathrm{B}(i, 7)\) for \(\mathrm{max}(1, j\) \(\mathrm{kb})<=\mathrm{i}<=\dot{j}\) if \(\mathrm{UPLO}=\mathrm{L}, \mathrm{BB}(1+i-j, j)=\mathrm{B}(i, 7)\) for \(\dot{j}=i<=m\) in \((n, j+k b)\).

On exit, the factors from the splitCholesky fac-
torization \(B=S * * H * S\), as retumed by CPBSTF .

LD BB (input)
The leading dim ension of the array BB. LD BB >= K B+1.

W (output)
If \(\mathbb{N} F O=0\), the eigenvalues in ascending order.
\(Z\) (input) If \(J O B Z=V\) ', then if \(\mathbb{N} F O=0, Z\) contains the \(m\) atrix \(Z\) ofeigenvectors, \(w\) ith the \(i\)-th colum n of Z holding the eigenvector associated w ith W (i). The eigenvectors are norm alized so that Z **H *B *Z = I. If \(\mathrm{JOBZ}=\mathrm{N}\) ', then Z is not referenced.

LD Z (input)
The leading dim ension of the array \(Z\). LD \(Z>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD \(\mathrm{Z}>=\mathrm{N}\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

\section*{LW ORK (input)}

The dim ension of the array W ORK. If \(\mathrm{N}<=1\), LW ORK >=1. If JOBZ = N'and \(N>1\),LW ORK >=N. If \(\mathrm{JOBZ}=\mathrm{V}\) 'and \(\mathrm{N}>1\), LW \(O\) RK \(>=2 * \mathrm{~N} * * 2\) 。

IfLW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{RW ORK (w orkspace)}

On exit, if \(\mathbb{N} F O=0, R W\) ORK (1) retums the optim al LRW ORK.

LRW ORK (input)
The dim ension of aray RW ORK. If \(\mathrm{N}<=1\), LRW ORK >= 1. If \(J O B Z=N\) 'and \(N>1\),LRW ORK \(>=\) N . If JOBZ \(=\mathrm{V}\) 'and \(\mathrm{N}>1\),LRW \(O R K>=1+5 * \mathrm{~N}+\) \(2 * N * * 2\) 。

IfLRW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the RW ORK array, retums this value as the first entry of the RW ORK array, and no errorm essage related to LRW ORK is issued by X ERBLA.

IV ORK (w orkspace/output)
On exit, if \(\mathbb{N F O}=0, \mathbb{I N}\) ORK (1) retums the optim al LIN ORK.

LIN ORK (input)
The dim ension of array \(\mathbb{I N}\) ORK. If JO BZ \(=\mathrm{N}\) 'or N <= 1 , LIN \(O R K>=1\). If \(J O B Z=V\) 'and \(N>1\), LIN ORK \(>=3+5{ }^{*} N\).

If LIV ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the IV ORK array, retums this value as the first entry of the \(\mathbb{I W}\) ORK anay, and no errorm essage related to LIN ORK is issued by X ERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N F O}=i\), and \(i\) is:
\(<=\mathrm{N}\) : the algorithm failed to converge: i offdiagonal elem ents of an interm ediate tridiagonal
form did not converge to zero; \(>\mathrm{N}\) : if \(\mathbb{N} F \mathrm{FO}=\mathrm{N}\)
\(+i\), for \(1<=i<=N\), then CPBSTF
retumed \(\mathbb{N} F O=i: B\) is not positive definite.
The factorization ofB could notbe com pleted and no eigenvalues oreigenvectors w ere com puted.

\section*{FURTHER DETAILS}

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv. of K entucky, USA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zhbgvx - com pute allthe eigenvalues, and optionally, the eigenvectors of a com plex generalized H erm itian-definite banded eigenproblem, of the form \(A\) *x (lam bda)* \(B\) *x

\section*{SYNOPSIS}
```

SUBROUTINE ZHBGVX (OBZ,RANGE,UPLO,N,KA,KB,AB,LDAB,BB,LDBB, Q,LDQ,VL,VU, $\mathbb{L}, \mathbb{I}, A B S T O L, M, W, Z, L D Z, W$ ORK,RW ORK, IN ORK, $\mathbb{F A} \mathbb{L}, \mathbb{N} F O)$

```
```

CHARACTER * 1 JOBZ,RANGE,UPLO

```
DOUBLE COM PLEX AB (LDAB,*), BB (LDBB,*), Q (LDQ ,*), Z (LD Z,*),
W ORK (*)
\(\mathbb{N}\) TEGER N,KA,KB,LDAB,LDBB,LDQ, \(\mathbb{L}, \mathbb{U}, M, L D Z, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER \(\mathbb{I N}\) ORK (*), \(\mathbb{F} A \mathbb{L}(*)\)
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION W (*),RWORK (*)
SU BROUTINE ZHBGVX_64 (JOBZ,RANGE,UPLO,N,KA,KB,AB,LDAB,BB,
    LD BB, \(Q, L D Q, V L, V U, \mathbb{L}, \mathbb{U}, A B S T O L, M, W, Z, L D Z, W\) ORK,RW ORK,
    \(\mathbb{I N} O R K, \mathbb{F A} \mathbb{L}, \mathbb{N} F O)\)
CHARACTER * 1 JOBZ,RANGE,UPLO
D OUBLE COM PLEX AB (LDAB,*), BB (LDBB,*), Q (LDQ,*), Z (LD Z,*),
W ORK (*)
\(\mathbb{N}\) TEGER*8N,KA,KB,LDAB,LDBB,LDQ, \(\mathbb{L}, \mathbb{U}, M, L D Z, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{I N}\) ORK ( \(\left.{ }^{( }\right)\), \(\mathbb{F A} \mathbb{L}(*)\)
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION W (*),RW ORK (*)

SUBROUTINE HBGVX (JOBZ,RANGE,UPLO, \(\mathbb{N}], K A, K B, A B,[L D A B], B B\),
 [RW ORK ], [IN ORK], \(\mathbb{F A} \mathbb{I},[\mathbb{N} F O\) ])

CHARACTER (LEN=1):: OBZ,RANGE,UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::AB, BB,Q,Z
\(\mathbb{N} T E G E R:: N, K A, K B, L D A B, L D B B, L D Q, \mathbb{L}, \mathbb{U}, M, L D Z, \mathbb{N F O}\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK, \(\mathbb{F} A \mathbb{I}\)
REAL (8) ::VL,VU,ABSTOL
REAL (8),D IM ENSION (:) ::W ,RW ORK

SU BROUTINE HBGVX_64 (JOBZ,RANGE,UPLO, \(\mathbb{N}], K A, K B, A B,[L D A B], B B\), \([\mathrm{LDBB}], \mathrm{Q},[\mathrm{LD} Q], \mathrm{VL}, \mathrm{VU}, \mathbb{I}, \mathbb{U}, \mathrm{ABSTOL}, \mathrm{M}, \mathrm{W}, \mathrm{Z},[\mathrm{LD} \mathrm{Z}],[\mathbb{W} \mathrm{ORK}]\), [RW ORK], [ \(\mathbb{W}\) ORK ], \(\mathbb{F} A \mathbb{I},[\mathbb{N} F O\) ])

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COM PLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) :: AB, BB, Q, Z
\(\mathbb{N} T E G E R(8):: N, K A, K B, L D A B, L D B B, L D Q, \mathbb{L}, \mathbb{U}, M, L D Z\),
\(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{I N}\) ORK, \(\mathbb{F A} \mathbb{I}\)
REAL (8) ::VL,VU,ABSTOL
REAL (8), D IM ENSION (:) ::W ,RW ORK

\section*{C INTERFACE}
\#include < sunperfh>
void zhbgvx (char jobz, char range, char uplo, intn, int ka, int kb, doublecom plex *ab, int ldab, doublecom plex *bb, int ldbb, doublecom plex *q, int ldq, double vl, double vu, int il, intiu, double abstol, int \({ }^{\mathrm{m}}\), double \({ }_{\mathrm{w}}\), doublecom plex \(\mathrm{*}_{\mathrm{z}}\), int ldz , int *ifail, int*info);
void zhbgvx_64 (char jobz, char range, char uplo, long n, long ka, long kb, doublecom plex *ab, long ldab, doublecom plex *bb, long labb, doublecom plex *q, long ldq, double vl, double vu, long il, long in, double abstol, long \({ }^{*}\), double \({ }^{*}\) w , doublecom plex *z, long ldz, long *ifail, long *info);

\section*{PURPOSE}
zhbgvx com putes all the eigenvalues, and optionally, the eigenvectors of a com plex generalized H erm tian-definite banded eigenproblem, of the form A *x= (lam bda)*B *x. H ere A and \(B\) are assum ed to be \(H\) erm itian and banded, and \(B\) is also positive definite. Eigenvalues and eigenvectors can be
selected by specifying either alleigenvalues, a range of values or a range of indices for the desired eigenvalues.

\section*{ARGUMENTS}

JOBZ (input)
\(=N^{\prime}:\) C om pute eigenvalues only;
\(=\mathrm{V}\) : C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvalues w ill be found;
= V \(:\) :alleigenvahues in the half-open interval
( \(\mathrm{VL}, \mathrm{VU}]\) will be found; = I': the \(\mathbb{I}\)-th through \(\mathbb{I U}\)-th eigenvaluesw illlbe found.

UPLO (input)
\(=\mathrm{U}\) ': U ppertriangles of \(A\) and \(B\) are stored;
= L': Low ertriangles of A and B are stored.
N (input) The order of the m atrices A and \(\mathrm{B} . \mathrm{N}>=0\).
KA (input)
The num berof superdiagonals of the \(m\) atrix \(A\) if UPLO = U',orthe num berof subdiagonals if UPLO = L '.KA \(>=0\).

K B (input)
The num ber of superdiagonals of the \(m\) atrix \(B\) if \(\mathrm{UPLO}=\mathrm{U}\) ', orthe num berof subdiagonals if P PLO \(=L^{\prime} \cdot \mathrm{KB}>=0\).

AB (input/output)
O \(n\) entry, the upper or low ertriangle of the Her\(m\) itian band \(m\) atrix \(A\), stored in the firstka+1 row sof the array. The \(j\) th colum n of A is stored in the \(j\) th colum \(n\) of the amray AB as follow s: if \(\mathrm{UPLO}=\mathrm{U}, \mathrm{AB}(\mathrm{ka}+1+\mathrm{i}-j)=\mathrm{A}(i, 7)\) for max \((1, j\) \(\mathrm{ka})<=i<=j\) ifUPLO \(=\mathrm{L} \prime, A B(1+i-j, j)=A(i, 1)\) for \(\dot{j}=i<=m\) in \((n, j+k a)\).

On exit, the contents of \(A B\) are destroyed.
LDAB (input)
The leading dim ension of the array AB. LD AB >= KA+1.

BB (input/output)
O n entry, the upper or low ertriangle of the Her -
\(m\) itian band \(m\) atrix \(B\), stored in the first \(k b+1\) row sof the anray. The \(j\) th colum \(n\) of \(B\) is stored in the jth colum \(n\) of the array \(B B\) as follow s: if \(\mathrm{UPLO}=\mathrm{U}, \mathrm{BB}(\mathrm{kb}+1+\mathrm{i}-j, \bar{j})=\mathrm{B}(1, j)\) for \(\mathrm{max}(1, j\) \(\mathrm{kb})<=\mathrm{i}<=\dot{j}\) if P PLO \(=\mathrm{L}, \mathrm{\prime}, \mathrm{BB}(1+i-j)=\mathrm{i}(i, 7)\) for \(j=i<=m\) in \((n, j+k b)\).

On exit, the factors from the splitCholesky factorization \(\mathrm{B}=\mathrm{S} * * \mathrm{H} * \mathrm{~S}\), as retumed by CPBSTF .

LD BB (input)
The leading dim ension of the array BB. LD BB >= K B+1.
Q (output)
If \(\mathrm{OBZ}=\mathrm{V}^{\prime}\), the \(\mathrm{n}-\mathrm{by}-\mathrm{n}\) matrix used in the reduction of \(A \mathrm{X}=\left(\mathrm{lam}\right.\) bda) \(\mathrm{B}_{\mathrm{B}}{ }^{\mathrm{x}} \mathrm{x}\) to standard form, i.e. \(C * x=\left(l a m\right.\) bda) \({ }^{*} \mathrm{X}\), and consequently C to tridiagonal form. If \(J O B Z=N\) ', the array \(Q\) is not referenced.

LD Q (input)
The leading dim ension of the array \(Q\). If \(J 0 B Z=\) \(N^{\prime}, \operatorname{LD} Q>=1\). If \(\mathrm{JOB}=\mathrm{V}\) ', LD \(Q>=\mathrm{max}(1, N)\).

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRA NGE = A 'or I'.

VU (input)
IfRANGE=V', the low er and upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE=A'or I'.

II (input)
IfRA N G E= I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{I}<=\mathbb{U}<=N\), if \(N>0\); \(\mathbb{H}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N ot referenced ifRANGE \(=\) A'or V'.

IU (input)
If RA NGE= 'I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}\), if \(\mathrm{N}>0\); \(\mathbb{H}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N ot referenced ifRANGE \(=\) A'or V'.

ABSTOL (input)
The absolute error tolerance for the eigenvalues.

A \(n\) approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b]
of w idth less than orequal to
ABSTOL + EPS * max ( \(\mid\) |, \(\mid\) |),
where EPS is them achine precision. IfA BSTOL is less than or equalto zero, then EPS* \(\mid\) | w illbe used in its place, where \(F \mid\) is the 1 -norm of the tridiagonalm atrix obtained by reducing AP to tridiagonal form .

E igenvalues w illbe com puted m ost accurately when ABSTOL is set to tw ine the underflow threshold \(2 \star\) SLAM CH ( \({ }^{\prime}\) ), notzero. If this routine retums w ith \(\mathbb{I N}\) FO \(>0\), indicating that som e eigenvectors did not converge, try setting ABSTO L to \(2 *\) SLAM CH (S ).

M (output)
The total num ber ofeigenvalues found. \(0<=\mathrm{M}\) <= N . IfRANGE \(=A{ }^{\prime}, \mathrm{M}=\mathrm{N}\), and ifRANGE \(=\mathrm{I}^{\prime}, \mathrm{M}=\) \(\mathbb{I U}-\mathbb{L}+1\).

W (output)
If \(\mathbb{N}\) FO \(=0\), the eigenvalues in ascending order.
\(Z\) (input) If \(\mathcal{O} B Z=V\) ', then if \(\mathbb{N} F O=0, Z\) contains the \(m\) atrix \(Z\) of eigenvectors, \(w\) ith the \(i\)-th colum \(n\) of Z holding the eigenvector associated with W (i). The eigenvectors are norm alized so that \(\mathrm{Z} * * \mathrm{H} * \mathrm{~B} * \mathrm{Z}=\) I. If \(\mathrm{JOBZ}=\mathrm{N}\) ', then Z is not referenced.

LD \(Z\) (input)
The leading \(d i m\) ension of the array \(Z . L D Z>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD Z >=N.

W ORK (w orkspace)
dim ension (N)
RW ORK (w orkspace)
dim ension ( \(7 * \mathrm{~N}\) )
IW ORK (w orkspace)
dim ension ( \(5 *\) N )
FAII (output)
If \(\mathcal{O B Z}=\mathrm{V}\) ', then if \(\mathbb{N F O}=0\), the firstM ele\(m\) ents of \(\mathbb{F} A \mathbb{I}\) are zero. If \(\mathbb{N} F O>0\), then \(\mathbb{F} A \mathbb{I}\) contains the indices of the eigenvectors that failed to converge. If \(J O B Z=N\) ', then \(\mathbb{F A} I I\) is
not referenced.
\(\mathbb{N}\) FO (output)
= 0 : successfulexit
<0: if \(\mathbb{N F O}=-\mathrm{i}\), the i th argum ent had an illegal value
\(>0\) : if \(\mathbb{N F O}=i\), and \(i\) is:
\(<=\mathrm{N}\) : then i eigenvectors failed to converge. Their indices are stored in array \(\mathbb{F A} I \mathrm{~L} .>\mathrm{N}\) : if \(\mathbb{N} F O=N+i\), for \(1<=i<=N\), then CPBSTF retumed \(\mathbb{N} F O=i: B\) is not positive definite. The factorization ofB could notbe com pleted and no eigenvalues oreigenvectors w ere com puted.

\section*{FURTHER DETAILS}

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv. of K entucky, USA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhbm v -perform the m atrix-vectoroperation \(\mathrm{y}:=\) alpha* \(\mathrm{A} * \mathrm{x}\) + beta*y

\section*{SYNOPSIS}
```

SUBROUTINE ZHBMV (UPLO,N,K,ALPHA,A,LDA,X,INCX,BETA,Y,
INCY)
CHARACTER * 1 UPLO
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (LDA,*),X (*),Y (*)
\mathbb{NTEGERN,K,LDA,INCX,INCY}
SU BROUTINE ZHBM V_64 (UPLO,N,K,A LPHA,A,LDA,X, INCX,BETA,Y,
INCY)
CHARACTER * 1 UPLO
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (LDA,*),X (*),Y (*)
\mathbb{NTEGER*8 N,K,LDA, NNCX,}\mathbb{N}CY

```

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E\) HBMV \(\mathbb{U} P L O, \mathbb{N}], K, A L P H A, A,[L D A], X,[\mathbb{N C X}], B E T A\), Y, [ \(\mathbb{N} C Y])\)

CHARACTER (LEN=1) ::UPLO
COMPLEX (8) ::ALPHA,BETA
COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::X,Y
COM PLEX (8),D \(\mathbb{M}\) ENSIDN (: : : : : A
\(\mathbb{N} T E G E R:: N, K, L D A, \mathbb{N C X}, \mathbb{N} C Y\)

SU BROUTINE HBMV_64 (UPLO, \(\mathbb{N}], K, A L P H A, A,[L D A], X,[\mathbb{N C X}]\),

BETA, Y, [ \(\mathbb{N} C Y])\)

CHARACTER (LEN=1) ::UPLO
COMPLEX (8) ::ALPHA,BETA
COM PLEX (8),D \(\mathbb{I}\) ENSION (:) ::X,Y
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathbb{N C X}, \mathbb{N} C Y\)

\section*{C INTERFACE}
\#include <sunperfh>
void zhbm v (charuple, intn, int k, doublecom plex *alpha, doublecom plex *a, int lda, doublecom plex *x, int incx, doublecom plex *beta, doublecom plex *y, int incy);
void zhbm v_64 (char uplo, long n, long k, doublecom plex *alpha, doublecom plex *a, long lda, doublecom plex
*x, long incx, doublecom plex *beta, doublecom plex
*y, long incy);

\section*{PURPOSE}
zhbm \(v\) perform \(s\) the \(m\) atrix-vector operation \(y:=a l p h a * A * x+\) beta* \(y\) where alpha and beta are scalars, \(x\) and \(y\) are \(n\) ele\(m\) ent vectors and \(A\) is an \(n\) by \(n\) herm itian band \(m\) atrix, \(w\) ith k super-diagonals.

\section*{ARGUMENTS}

UPLO (input)
O n entry, UPLO specifies whether the upper or low er triangular part of the band \(m\) atrix \(A\) is being supplied as follow s:

UPLO = U 'or U ' The uppertriangularpartofA is being supplied.

UPLO = L'or 1' The low ertriangularpartofA is being supplied.

U nchanged on exit.

N (input)
On entry, N specifies the order of the m atrix A .
\(\mathrm{N}>=0\). U nchanged on exit.

K (input)

On entry, \(K\) specifies the number of superdiagonals of them atrix A. K m ust satisfy 0 le. K. U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
Before entry w ith UPLO = U 'or G ', the leading ( \(\mathrm{k}+1\) ) by n partof the array A m ustcontain the upper triangular band part of the herm tian \(m\) atrix, supplied colum \(n\) by colum \(n\), \(w\) th the leading diagonal of the matrix in row ( \(k+1\) ) of the array, the first super-diagonal starting at position 2 in row \(k\), and so on. The top leftk by \(k\) triangle of the array \(A\) is not referenced. The follow ing program segm entw ill transfer the upper triangular part of a herm itian band \(m\) atrix from conventional fullm atrix storage to band storage:
\[
\begin{aligned}
& \text { DO } 20, \mathrm{~J}=1, \mathrm{~N} \\
& \mathrm{M}=\mathrm{K}+1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{MAX}(1, \mathrm{~J}-\mathrm{K}), \mathrm{J} \\
& \quad \text { A }(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \text { CONTINUE } \\
& 20 \text { CONTINUE }
\end{aligned}
\]

Before entry w ith UPLO = L 'or 1', the leading ( \(k+1\) ) by \(n\) part of the array A m ust contain the low er triangular band part of the herm itian \(m\) atrix, supplied colum \(n\) by colum \(n\), \(w\) th the leading diagonalof the \(m\) atrix in row 1 of the array, the first sub-diagonalstarting atposition 1 in row 2 , and so on. The bottom right \(k\) by \(k\) triangle of the anray \(A\) is not referenced. The follow ing program segm entw ill transfer the low ertriangular part of a herm itian band \(m\) atrix from conventional fullm atrix storage to band storage:
\[
\begin{aligned}
& \text { DO } 20, J=1, N \\
& \mathrm{M}=1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{J}, \mathrm{M} \mathbb{N}(\mathrm{~N}, \mathrm{~J}+\mathrm{K}) \\
& \quad \text { A }(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \text { CONTINUE } \\
& 20 \text { CONTINUE }
\end{aligned}
\]
\(N\) ote that the im aginary parts of the diagonalele\(m\) ents need notbe set and are assum ed to be zero. U nchanged on exit.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD A \(>=\) ( \(\mathrm{k}+1\) ). U nchanged on exit.

X (input)
\((1+(n-1) \star \operatorname{abs}(\mathbb{N} C X))\). Before entry, the increm ented anray \(X\) must contain the vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{I N C X ~ m ~ u s t n o t b e ~ z e r o . ~ U ~ n c h a n g e d ~}\) on exit.

BETA (input)
On entry, BETA specifies the scalar beta.
U nchanged on exit.
Y (input/output)
\((1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) mustcontain the vectory. On exit, \(Y\) is overw ritten by the updated vectory.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of Y. \(\mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zhbtrd -reduce a com plex H erm tian band \(m\) atrix \(A\) to real sym metric tridiagonal form \(T\) by a unitary sim ilarity transform ation

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHBTRD NECT,UPLO,N,KD,AB,LDAB,D,E,Q,LDQ,W ORK,}
INFO)
CHARACTER * 1VECT,UPLO
DOUBLE COM PLEX AB (LDAB,*),Q (LDQ,*),W ORK (*)
\mathbb{NTEGER N,KD,LDAB,LDQ,NNFO}
DOUBLE PRECISIOND (*),E (*)
SUBROUT\mathbb{NE ZHBTRD_64NECT,UPLO,N,KD,AB,LDAB,D,E,Q,LDQ,W ORK,}
\mathbb{NFO)}
CHARACTER * 1 VECT,UPLO
DOUBLE COM PLEX AB (LDAB,*),Q (LDQ,*),W ORK (*)
INTEGER*8N,KD,LDAB,LDQ,INFO
DOUBLE PRECISION D (*),E (*)

```
F95 INTERFACE
    SU BROUTINE HBTRD (NECT, UPLO, \(\mathbb{N}], K D, A B,[L D A B], D, E, Q,[L D Q]\),
        [ W ORK], [ \(\mathbb{N} F \mathrm{~F}\) ])
    CHARACTER (LEN=1)::VECT,UPLO
    COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::W ORK
    COM PLEX (8), D IM ENSION (:,:) ::AB,Q
    \(\mathbb{N} T E G E R:: N, K D, L D A B, L D Q, \mathbb{N} F O\)
    REAL (8),D IM ENSION (:) ::D , E

SU BROUTINE HBTRD_64 NECT, UPLO, \(\mathbb{N}], K D, A B,[L D A B], D, E, Q,[L D Q]\), [ W ORK], [ \(\mathbb{N} F \mathrm{O}\) ])

CHARACTER (LEN=1) ::VECT,UPLO
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D IM ENSION (:,:) ::AB, \(Q\)
\(\mathbb{N} T E G E R(8):: N, K D, L D A B, L D Q, \mathbb{N} F O\)
REAL (8),D IM ENSION (:) ::D,E

\section*{C INTERFACE}
\#include < sunperfh>
void zhbtrd (charvect, char uple, intn, int kd, doublecom plex *ab, int ldab, double *d, double *e, doublecom plex *q, int ldq, int*info);
void zhbtrol_64 (char vect, charuplo, long n, long kd, doublecom plex *ab, long ldab, double *d, double *e, doublecom plex *q, long ldq, long *info);

\section*{PURPOSE}
zhbtrod reduces a com plex H erm itian band m atrix A to real symmetric tridiagonal form \(T\) by a unitary similarity transform ation: \(\mathrm{Q} * * \mathrm{H} * \mathrm{~A} * \mathrm{Q}=\mathrm{T}\).

\section*{ARGUMENTS}

VECT (input)
\(=\mathrm{N}\) : do not form \(Q\);
\(=\mathrm{V}\) : form Q ;
\(=\mathrm{U}\) : update a m atrix X , by form ing \(\mathrm{X} * \mathrm{Q}\).

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low er triangle of A is stored.

N (input) The order of the matrix A. \(\mathrm{N}>=0\).
KD (input)
The num ber of superdiagonals of the \(m\) atrix \(A\) if UPLO \(=\mathrm{U}\) ', or the num berof subdiagonals ifUPLO \(=\mathbb{L}^{\prime} . \mathrm{KD}>=0\) 。

AB (input/output)
O \(n\) entry, the upper or low er triangle of the \(H\) er\(m\) itian band \(m\) atrix \(A\), stored in the firstK \(D+1\)
row sof the anray. The \(j\) th colum n of \(A\) is stored in the \(j\) th colum \(n\) of the array AB as follow s: if \(\mathrm{UPLO}=\mathrm{U}, \mathrm{AB}(\mathrm{kd}+1+\mathrm{i}-j)=\mathrm{j}(i, 1)\) for \(\mathrm{max}(1, j\) \(\mathrm{kd})<=i<=j\) if \(\mathrm{UPLO}=\mathrm{L}, \mathrm{AB}(1+i-j)=A(i, 7)\) for \(\dot{j}=i<=m\) in \((n, \dot{j}+\mathrm{kd})\). O \(n\) exit, the diagonalele\(m\) ents of AB are overw ritten by the diagonalele\(m\) ents of the tridiagonalm atrix T ; if K D > 0, the elem ents on the first superdiagonal (if UPLO \(=\) U ) orthe first subdiagonal (ifU PLO = L ) are overw rilten by the off-diagonalelem ents of \(T\); the rest of A B is overw ritten by values generated during the reduction.

LDAB (input)
The leading dim ension of the array AB. LDAB >= K D +1 .

D (output)
The diagonalelem ents of the tridiagonalm atrix T .
E (output)
The off-diagonal elem ents of the tridiagonal m atrix \(\mathrm{T}: \mathrm{E}(\mathrm{i})=\mathrm{T}(\mathrm{i}, \mathrm{i}+1)\) if \(\mathrm{PLO}=\mathrm{U} ; \mathrm{E}(\mathrm{i})=\) \(T(i+1, i)\) if \(\mathrm{PLLO}=\mathrm{L}^{\prime}\) 。

Q (input/output)
Onentry, ifVECT = U', then \(Q \mathrm{must}\) contain an N by -N m atrix X ; if VECT \(=\mathrm{N}\) 'or V ', then Q need notbe set.

On exit: if \(\mathrm{VECT}=\mathrm{V}\) ', Q contains the N -by -N unitary matrix \(Q\); ifVECT = U', \(Q\) contains the product \(X\) * \(Q\); ifVECT \(=N\) ', the array \(Q\) is not referenced.

LD Q (input)
The leading \(d i m\) ension of the array \(Q . L D Q>=1\), and LDQ >= N ifVECT = V'or U'.

W ORK (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{FURTHER DETAILS}

M odified by Linda K aufn an, Bell Labs.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhecon -estim ate the reciprocal of the condition num ber of a complex Herm tian \(m\) atrix \(A\) using the factorization \(A=\) \(\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\) com puted by CHETRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHECON(UPLO,N,A,LDA,\mathbb{PIVOT,ANORM,RCOND,WORK,INFO)}}\mathbf{N}\mathrm{ (N,N}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)
INTEGERN,LDA,}\mathbb{N}F
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION ANORM ,RCOND
SUBROUT\mathbb{NE ZHECON_64(UPLO,N,A,LDA, \mathbb{PIVOT,ANORM,RCOND,W ORK,}}\mathbf{N},\textrm{N},\textrm{N}
\mathbb{NFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGER*8N,LDA,INFO}
INTEGER*8 \mathbb{PIVOT (*)}
DOUBLE PRECISION ANORM,RCOND
F95 INTERFACE

```

```

    [\mathbb{NFO ])}
    CHARACTER (LEN=1)::UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D IM ENSION (:,:) ::A
\mathbb{NTEGER::N,LDA,}\mathbb{N}FO
\mathbb{NTEGER,D IM ENSION (:)::\mathbb{PIVOT}}\mathbf{T}\mathrm{ (: }

```

SU BROUTINE HECON_64 (UPLO, \(\mathbb{N}\) ],A, [LDA ], \(\mathbb{P} \mathbb{I V O T , A N O R M , R C O N D , [ W O R K ] , ~}\) [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1)::UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSIDN (: : : : ::A
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} A, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V O T}\)
REAL (8) ::ANORM,RCOND

\section*{C INTERFACE}
\#include < sunperfh>
void zhecon (charuplo, intn, doublecom plex *a, int lda, int *ịívot, double anorm , double *rcond, int *info);
void zhecon_64 (charuplo, long n, doublecom plex *a, long lda, long *ípi̇ot, double anorm , double *roond, long *info);

\section*{PURPOSE}
zhecon estim ates the reciprocal of the condition num ber of a complex Herm itian matrix A using the factorization A = \(\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\) com puted by CHETRF.

A n estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND = 1 / ANORM * norm (inv (A))).

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) ': U ppertriangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\);
\(=\mathrm{L}\) ': Lowertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\).

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
A (input) The block diagonalm atrix D and the multipliers
used to obtain the factorU orL as com puted by
CHETRF.
LD A (input)
The leading dim ension of the array A. LDA >=
\(\max (1, \mathbb{N})\).
PIVOT (input)
D etails of the interchanges and the block structure ofD as determ ined by CHETRF.

ANORM (input)
The 1-norm of the originalm atrix A.
RCOND (output)
The reciprocal of the condition number of the
\(m\) atrix \(A\), com puted as RCOND \(=1 /(A N O R M * A \mathbb{N} V N M)\), where \(A \mathbb{N} V N M\) is an estim ate of the 1 -norm of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension \((2 * N)\)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zheev - com pute alleigenvalues and, optionally, eigenvectors of a com plex H erm itian m atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHEEV (OOBZ,UPLO,N,A,LDA,W ,W ORK,LDW ORK,W ORK 2, INFO)}
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)
INTEGERN,LDA,LDW ORK,\mathbb{NFO}
DOUBLE PRECISION W (*),W ORK2 (*)
SUBROUT\mathbb{NE ZHEEV_64(JOBZ,UPLO,N,A,LDA,W ,W ORK,LDW ORK,W ORK2,}
INFO)
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGER*8N,LDA,LDW ORK,INFO}
DOUBLE PRECISION W (*),W ORK2 (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE HEEV (OBB,UPLO, $\mathbb{N}], A,[L D A], W,[\mathbb{W} O R K],[L D W$ ORK], [W ORK 2], [ $\mathbb{N} F \mathrm{~F}$ ])
CHARACTER (LEN=1): : JOBZ, UPLO
COMPLEX (8), D IM ENSION (:) ::W ORK
COM PLEX (8),D IM ENSION (:,:) ::A
$\mathbb{N} T E G E R:: N, L D A, L D W$ ORK, $\mathbb{N} F O$
REAL (8), D IM ENSION (:) ::W ,W ORK2
SU BROUTINE HEEV_64 ( $\mathbb{O B Z}, \mathrm{UPLO}, \mathbb{N}], A,[L D A], W,[\mathbb{W}$ ORK ], [LDW ORK ], [W ORK2], [ $\mathbb{N} F \mathrm{O}$ ])

```

CHARACTER (〔EN=1):: OBZ, UPLO
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: \(:\) :) :: A
\(\mathbb{N} T E G E R(8):: N, L D A, L D W O R K, \mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ,W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zheev (char jobz, charuplo, int n, doublecom plex *a, int lda, double *w , int *info);
void zheev_64 (char jobz, charuplo, long n, doublecom plex *a, long lda, double *w , long *info);

\section*{PURPOSE}
zheev com putes alleigenvalues and, optionally, eigenvectors of a com plex \(H\) erm itian \(m\) atrix A .

\section*{ARGUMENTS}

JO B Z (input)
\(=\mathrm{N}\) : C om pute eigenvalues only;
\(=\mathrm{V}\) ': C om pute eigenvalues and eigenvectors.

UPLO (input)
\(=U^{\prime}:\) U ppertriangle of \(A\) is stored;
\(=\mathrm{L}\) ': Low er triangle ofA is stored.

N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the H erm itian m atrix A. If UPLO = U', the leading N łoy N uppertriangularpart of A contains the upper triangularpart of the \(m\) atrix \(A\). If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading N toy-N low er triangular part of A contains the low er triangular part of the \(m\) atrix \(A\). On exit, if \(J O B Z=V\) ', then if \(\mathbb{N} F O=0, A\) contains the orthonorm al eigenvectors of them atrix \(A\). If \(O B Z=N\) ', then on exit the low er triangle (if \(\mathrm{U} \mathrm{PLO}=\mathrm{L}\) ) or the upper triangle (if \(\mathrm{UPLO}=\mathrm{U}\) ) of A , including the diagonal, is destroyed.

The leading dim ension of the array A. LDA >= \(\max (1, N)\).

W (output)
If \(\mathbb{N} F O=0\), the eigenvalues in ascending order.
W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al
LDW ORK.
LDW ORK (input)
The length of the array W ORK. LDW ORK >= max \((1,2 * \mathrm{~N}-1)\). For optim alefficiency, LD W ORK >= \((N B+1) * N\), where \(N B\) is the blocksize for CHETRD retumed by ILAENV.

If LD W ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 2 (w orkspace)
dim ension (max (1,3*N-2))
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N} F O=\) i, the algonithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zheevd -com pute alleigenvalues and, optionally, eigenvectors of a com plex H erm itian m atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHEEVD (OOBZ,UPLO,N,A,LDA,W ,W ORK,LW ORK,RW ORK,}
LRW ORK,IN ORK,LIN ORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEXA (LDA,*),W ORK (*)
\mathbb{NTEGER N,LDA,LW ORK,LRW ORK,LIN ORK,\mathbb{NFO}}\mathbf{N}\mathrm{ (IN}
INTEGER IV ORK (*)
DOUBLE PRECISION W (*),RW ORK (*)
SU BROUTINE ZHEEVD_64 (JOBZ,UPLO,N,A,LDA,W ,W ORK,LW ORK,RW ORK,
LRW ORK,INORK,L\mathbb{N ORK,INFO)}
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)

```

```

INTEGER*8 IN ORK (*)
DOUBLE PRECISION W (*),RW ORK (*)

```

\section*{F95 INTERFACE}
```

SUBROUTINE HEEVD ( $\mathbb{O B Z}, \mathrm{UPLO}, \mathbb{N}], A,[L D A], W,[\mathbb{O}$ ORK], [LW ORK], [RW ORK ], [LRW ORK], [ $\mathbb{W}$ ORK ], [LIN ORK], [ $\mathbb{N} F O]$ )
CHARACTER (LEN=1): : JOBZ,UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8), D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N}$ TEGER :: N,LDA,LW ORK,LRW ORK,LIN ORK, $\mathbb{N} F O$

```
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{I W}\) ORK REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ,RW ORK

SU BROUTINE HEEVD_64 (OOBZ,UPLO, \(\mathbb{N}], A,[L D A], W,[W O R K],[L W O R K]\), \([R W\) ORK ], [LRW ORK], [ \(\mathbb{W}\) ORK ], [ \(\mathbb{I W}\) ORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1): : JOBZ, UPLO
COMPLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : : A
\(\mathbb{N} T E G E R(8):: N, L D A, L W O R K, L R W O R K, L \mathbb{I}\) ORK, \(\mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{I W}\) ORK
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ,RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zheevd (char j̣bz, char uplo, intn, doublecom plex *a, int lda, double *W , int *info);
void zheevd_64 (char jobz, charuplo, long n, doublecom plex *a, long lda, double *W , long *info);

\section*{PURPOSE}
zheevd com putes alleigenvalues and, optionally, eigenvectors of a com plex H erm titian m atrix A. If eigenvectors are desired, ituses a divide and conquer algorithm .

The divide and conquer algorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on m achines w ith a guard digit in add/subtract, or on those binary \(m\) achines \(w\) thout guard digits which subtract like the \(C\) ray \(\mathrm{X}-\mathrm{M} P\), C ray \(Y \mathrm{M} \mathrm{P}\), C ray \(\mathrm{C}-90\), or C ray-2. It could conceivably fail on hexadecim al or decim al machines \(w\) ithout guard digits, butw e know of none.

\section*{ARGUMENTS}
```

JO B Z (input)
= N ': C om pute eigenvalues only;
= V ': C om pute eigenvalues and eigenvectors.
UPLO (input)
= U ': U pper triangle ofA is stored;
= IL ': Low er triangle ofA is stored.
N (input) The order of them atrix A. N >=0.

```

A (input/output)
On entry, the \(H\) em itian m atrix A. If P PLO = U', the leading N -by -N uppertriangularpartofA contains the uppertriangularpart of the \(m\) atrix \(A\). If U PLO = L', the leading \(N\)-by N low er triangular part ofA contains the low er triangular part of the \(m\) atrix \(A\). On exit, if \(J O B Z=V\) ', then if \(\mathbb{N} F O=0, A\) contains the orthonorm al eigenvectors of the \(m\) atrix \(A\). If \(J O B Z=N\) ', then on exit the low er triangle (if \(\mathrm{PLLO}=\mathrm{L}\) ) or the upper triangle (if \(U P L O=U\) ) of \(A\), including the diagonal, is destroyed.

LDA (input)
The leading dim ension of the array A. LD A >= \(\max (1, \mathbb{N})\).

W (output)
If \(\mathbb{N} F O=0\), the eigenvalues in ascending order.

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, \mathrm{~W}\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The length of the array \(\mathrm{W} O R K\). If \(\mathrm{N}<=1\), LW ORK mustibe at least 1. If \(\mathrm{JOBZ}=\mathrm{N}\) 'and \(\mathrm{N}>\) 1, LW ORK m ustbe at least \(\mathrm{N}+1\). If \(\mathrm{JOBZ}=\mathrm{V}^{\prime}\) and \(\mathrm{N}>1\), LW ORK m ustbe at least2*N + N **2.

If LW O RK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of theW ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

RW ORK (w orkspace)
dim ension (LRW ORK) On ex止, if \(\mathbb{N F F O}=0\), RW ORK (1) retums the optim alLRW ORK .

LRW ORK (input)
The dim ension of the aray RW ORK. If \(\mathrm{N}<=1\), LRW ORK m ustbe at least1. If \(\mathrm{JOBZ}=\mathrm{N}\) 'and \(\mathrm{N}>\) 1,LRW ORK m ust.be at leastN. If JOBZ \(=V^{\prime}\) and \(\mathrm{N}>1\),LRW ORK m ust.be at least1 + 5*N + 2*N **2.

If LRW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the RW ORK array, retums this value as the first entry of the RW ORK aray, and no enrorm essage
related to LRW ORK is issued by XERBLA.
IN ORK (w orkspace/output)
On exit, if \(\mathbb{N} F O=0, \mathbb{I}\) O RK (1) retums the optim al LIN ORK.

LIV ORK (input)
The dim ension of the array \(\mathbb{I N}\) ORK. If \(\mathrm{N}<=1\), LIN ORK must.be at least1. If \(J 0 \mathrm{BZ}=\mathrm{N}\) 'and \(\mathrm{N}>\) \(1, \mathrm{LIV}\) ORK m ustbe at least 1. If \(\mathrm{JOBZ}=\mathrm{V}^{\prime}\) and \(\mathrm{N}>1, \mathrm{~L} \mathbb{I} \mathrm{O}\) ORK mustbe at least \(3+5 * \mathrm{~N}\).

If LIV ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the \(\mathbb{I V}\) ORK array, retums this value as the first entry of the \(\mathbb{I W}\) ORK array, and no errorm essage related to \(L \mathbb{I N} O R K\) is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N F O}=\) i, the algonithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

\section*{FURTHER DETAILS}

B ased on contributions by JeffR utter, C om puter Science D ívision, U niversity of C alifomia at B erkeley, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zheevr - com pute selected eigenvalues and, optionally, eigenvectors of a com plex H erm itian tridiagonalm atrix T

\section*{SYNOPSIS}

```

    ABSTOL,M,W ,Z,LD Z,ISUPPZ,W ORK,LW ORK,RW ORK,LRW ORK,IN ORK,
    L\mathbb{IN ORK, \mathbb{NFO)}}\mathbf{~}=()
    CHARACTER * 1 JOBZ,RANGE,UPLO
DOUBLE COM PLEX A (LDA,*),Z (LD Z,*),W ORK (*)
INTEGER N,LDA,\mathbb{L},\mathbb{U},M,LD Z,LW ORK,LRW ORK,LIN ORK,\mathbb{NFO}
INTEGER ISUPPZ (*), IN ORK (*)
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION W (*),RW ORK (*)
SUBROUTINE ZHEEVR_64 (JOBZ,RANGE,UPLO,N,A,LDA,VL,VU,\mathbb{L},\mathbb{U},
ABSTOL,M,W ,Z,LD Z,ISUPPZ,W ORK,LW ORK,RW ORK,LRW ORK,IN ORK,
L\mathbb{IN ORK, INFO)}

```
CHARACTER * 1 JOBZ,RANGE, UPLO
DOUBLE COM PLEX A (LDA, \(\left.{ }^{\star}\right)\), Z (LD Z,*), W ORK (*)
\(\mathbb{N} T E G E R * 8 N, L D A, \mathbb{I}, \mathbb{U}, M, L D Z, L W O R K, L R W O R K, L \mathbb{I N} O R K\),
\(\mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \operatorname{ISUPPZ}(*), \mathbb{I N}\) ORK (*)
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION W (*), RW ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE HEEVR (JOBZ,RANGE, UPLO, \(\mathbb{N}], A,[L D A], V L, V U, \mathbb{L}, \mathbb{I}\), ABSTOL,M,W,Z,[LD Z], ISU PPZ, [W ORK], [LW ORK ], RW ORK ], [LRW ORK ],
[ \(\mathbb{I N}\) ORK], [LIN ORK ], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COM PLEX (8), D \(\mathbb{M} \operatorname{ENSION}(:):: W\) ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: \(:\) ) : : A , Z
\(\mathbb{N} T E G E R:: N, L D A, \mathbb{I}, \mathbb{I}, M, L D Z, L W O R K, L R W O R K, L \mathbb{I W}\) ORK,
\(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: ISU PPZ, \(\mathbb{I N}\) ORK
REAL (8) :: VL,VU,ABSTOL
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ,RW ORK

SU BROUTINE HEEVR_64 (DOBZ,RANGE,UPLO, \(\mathbb{N}], A,[L D A], V L, V U, \mathbb{I}, \mathbb{U}\), ABSTOL,M,W,Z, [LDZ], ISUPPZ, [W ORK], [LW ORK], [RW ORK], [LRW ORK], \([\mathbb{I W}\) ORK \(],[\llbracket \mathbb{W}\) ORK \(],[\mathbb{N} F O])\)

CHARACTER (LEN=1) :: JOBZ, RANGE, UPLO
COM PLEX (8), D \(\mathbb{M} E N S I O N(:):\) W ORK
COMPLEX (8), D \(\mathbb{M}\) ENSION (: : : : : A , Z
\(\mathbb{N} \operatorname{TEGER}\) (8) :: \(\mathrm{N}, \mathrm{LD} A, \mathbb{Z}, \mathbb{U}, \mathrm{M}, \mathrm{LD} \mathrm{Z}, \mathrm{LW}\) ORK,LRWORK,LIWORK,
\(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER (8), D \(\mathbb{M}\) ENSION (:) :: ISUPPZ, \(\mathbb{I N}\) ORK
REAL (8) :: VL,VU,ABSTOL
REAL (8), D \(\mathbb{M} E N S I O N(:):: W, R W O R K\)

\section*{C INTERFACE}
\#include <sunperfh>
void zheevr(char jobz, char range, charuplo, int n, doublecom plex *a, intlda, double vl, double vu, int il, intiu, double abstol, int *m, double \({ }^{\prime}\), , doublecom plex \({ }^{*} z\), int \(l d z\), int *isuppz, int *info);
void zheevr_64 (char jंbz, charrange, char uplo, long n, doublecom plex *a, long lda, double vl, double vu, long il, long iu, double abstol, long *m, double \({ }^{*}\) w, doublecom plex *z, long ldz, long *isuppz, long *info);

\section*{PURPOSE}
zheevr com putes selected eigenvalues and, optionally, eigenvectors of a com plex H erm itian tridiagonalm atrix T. Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices forthe desired eigenvalues.

W heneverpossible, CHEEVR calls CSTEGR to com pute the eigenspectrum using Relatively Robust Representations.

CSTEGR com putes eigenvalues by the dqds algorithm, while orthogonaleigenvectors are com puted from various "good" L D \(L^{\wedge} T\) representations (also known as Relatively Robust Representations). G ram -Schm idtorthogonalization is avoided as far as possible. M ore specifically, the various steps of the algorithm are as follow s.For the i-th unreduced block oft,
(a) C om pute \(T\)-sigm a_i= L_iD _iL_i^T, such that L_i D_iL_i^T
is a relatively robust representation,
(b) C om pute the eigenvalues, lam bda_jof L_i D_i L_i^T to high
relative accuracy by the dqds algorithm,
(c) If there is a cluster of close eigenvalues, "choose"
sigm a_i
close to the cluster, and go to step (a),
(d) G iven the approxim ate eigenvalue lam bda_jofL_i D _i L_i^T,
com pute the corresponding eigenvectorby form ing a rank-revealing tw isted factorization. The desired accuracy of the output can be specified by the inputparam eterA BSTOL.

Form ore details, see "A new O ( \(n^{\wedge} 2\) ) algorithm for the sym \(m\) etric tridiagonal eigenvalue/eigenvector problem ", by Inder前D hillon, C om puterScience D ivision TechnicalR epont N o. U CB //C SD -97-971, U C B erkeley, M ay 1997.

N ote 1 : CHEEVR calls CSTEGR when the fill spectrum is requested on \(m\) achines \(w\) hich conform to the ieee-754 floating pointstandard. CHEEVR calls SSTEBZ and CSTE IN on non-ieee \(m\) achines and
w hen partialspectrum requests arem ade.

N orm alexecution of CSTEG R m ay create \(\mathrm{NaN} s\) and infinities and hence \(m\) ay abort due to a floating pointexception in environm ents which do nothandle N aN s and infinities in the ieee standard defaultm anner.

\section*{ARGUMENTS}

JO B Z (input)
\(=\mathrm{N}^{\prime}:\) C om pute eigenvalues only;
\(=\mathrm{V}\) ': C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvahues w illbe found.
\(=\mathrm{V}\) : alleigenvalues in the half-open interval
( \(\mathrm{LL}, \mathrm{VU}]\) w ill be found. = \(I^{\prime}\) : the \(\mathbb{I}\)-th through \(\mathbb{I U}\)-th eigenvaluesw illlbe found.

UPLO (input)
\(=\mathrm{U}:\) : U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.
\(N\) (input) The order of the matrix A. \(N>=0\).
A (input/output)
\(\mathrm{O} n\) entry, the \(H\) erm itian \(m\) atrix \(A\). If \(\mathrm{UPLO}=\mathrm{U}\) ', the leading \(\mathrm{N}-\) by -N upper triangularpartofA contains the upper triangular part of the \(m\) atrix \(A\). If UPLO \(=\mathrm{L}\) ', the leading N -by N low er triangular partofA contains the low er triangular part of the matrix A. On exit, the low ertriangle (if
\(\mathrm{UPLO}=\mathrm{L}\) ) or the upper triangle (if \(\mathrm{UPLO}=\mathrm{U}\) ) of A , including the diagonal, is destroyed.

LDA (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

VL (input)
IfRA N GE=V ', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE = A ' or I '.

VU (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU. N ot referenced ifRANGE=A 'or I'.

II (input)
If RA N GE= 'I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{H}<=\mathbb{U}<=N\), if \(N>0\); \(\mathbb{H}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N ot referenced ifRANGE \(=\) A 'or V'.
\(\mathbb{I U}\) (input)
If RA N G E= I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{U}<=\mathbb{U}<=N\), if \(N>0\); \(\mathbb{H}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N otreferenced ifRANGE = A'or V'.

ABSTOL (input)
The absolute error tolerance for the eigenvalues. A \(n\) approxim ate eigenvalue is accepted as converged
when it is determ ined to lie in an interval [a,b] of w idth less than orequal to

ABSTOL + EPS * max ( \(\mid\) |, b|),
\(w\) here EPS is the m achine precision. IfA BSTOL is less than orequal to zero, then EPS* \(\mid\) | w illbe used in its place, where \(F \mid\) is the 1 -norm of the tridiagonal m atrix obtained by reducing \(A\) to tridiagonal form.

See "C om puting Sm allSingularV ahues of B idiagonal \(M\) atrices w ith G uaranteed H igh Relative A ccuracy," by D em m eland \(K\) ahan, LA PA CK W orking \(N\) ote \#3.

If high relative accuracy is im portant, setA BSTO L to SLAM CH (Safe minimum '). D oing so will guarantee thateigenvalues are com puted to high relative accuracy when possible in future releases. The current code does not \(m\) ake any guarantees abouthigh relative accuracy, but funutre releases will. See J.Barlow and J. Demmel, "C om puting A ccurate Eigensystem s of Scaled D iagonally D om inantM atrioes", LA PA CK W orking N ote \#7, for a discussion of which \(m\) atrices define their eigenvalues to high relative accuracy.

M (output)
The totalnum ber ofeigenvalues found. \(0<=\mathrm{M}\) <= N . IfRANGE \(=\mathrm{A}^{\prime}, \mathrm{M}=\mathrm{N}\), and ifRANGE \(=\mathrm{I}^{\prime}, \mathrm{M}=\) \(\mathbb{U}-\mathbb{L}+1\).

W (output)
The firstM elem ents contain the selected eigenvalues in ascending order.
\(Z\) (input) If \(J O B Z=V^{\prime}\), then if \(\mathbb{N F O}=0\), the first \(M\) colum ns of \(Z\) contain the orthonorm aleigenvectors of them atrix A comesponding to the selected eigenvalues, \(w\) ith the \(i\)-th colum \(n\) of \(Z\) holding the eigenvector associated w ith W (i). If \(\mathrm{JO} \mathrm{BZ}=\mathrm{N}\) ', then \(Z\) is not referenced. N ote: the userm ust ensure that at leastm ax ( \(1, M\) ) colum ns are supplied in the array \(Z\); ifRANGE = \(V\) ', the exact value of M is not know n in advance and an upperbound \(m\) ust be used.

LD Z (input)
The leading dim ension of the array \(\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{max}(1, N)\).

ISU PPZ (output)
The support of the eigenvectors in \(Z, i . e .\), the indices indicating the nonzero elem ents in \(Z\). The i-th eigenvector is nonzero only in elem ents ISU PPZ (2*i-1 ) through ISU PPZ (2*i).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The length of the array WORK. LW ORK >= \(\max (1,2 \star \mathrm{~N})\). For optim al efficiency, LW ORK \(>=\) ( \(\mathrm{N} B+1)^{*} \mathrm{~N}\), where \(N B\) is the \(m\) ax of the blocksize for CHETRD and forCUNM TR as retumed by HAENV.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of theW ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

RW ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, R W\) ORK (1) retums the optim al (andm inim al) LRW ORK .

LRW ORK (input)
The length of the array RW ORK. LRW ORK >= \(\max (1,24 * N)\).

If LRW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the RW ORK array, retums this value as the first entry of the RW ORK aray, and no enrorm essage related to LRW ORK is issued by XERBLA.

IV ORK (w orkspace/output)
On exit, if \(\mathbb{N} F O=0, \mathbb{I N}\) ORK (1) retums the optim al (and \(m\) inim al) \(L \mathbb{I N} O R K\).

LIN ORK (input)
The dim ension of the array \(\mathbb{I W} O R K\). LIN ORK >= \(\max (1,10 * \mathrm{~N})\).

If \(L \mathbb{I V}\) ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the \(\mathbb{I W}\) ORK array, retums this value as the first entry of the \(\mathbb{I W}\) ORK array, and no errorm essage related to \(L \mathbb{I N} O R K\) is issued by XERBLA.
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an illegalvalue
> 0: Intemalemor

\section*{FURTHER DETAILS}

B ased on contributions by
Inder\#̈̈tD hillon, IBM A 1 m aden, U SA
O sniM arques, LBN L NERSC , U SA
K en Stanley, C om puterScience D ivision, U niversity of C alifomia atB erkeley, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zheevx - com pute selected eigenvalues and, optionally, eigenvectors of a com plex H erm itian m atrix A

\section*{SYNOPSIS}

```

    ABTOL,NFOUND,W ,Z,LDZ,W ORK,LDW ORK,W ORK2,IN ORK 3, FFA IL,
    INFO)
    CHARACTER * 1 JOBZ,RANGE,UPLO
DOUBLE COM PLEX A (LDA,*),Z (LD Z,*),W ORK (*)
\mathbb{N}TEGERN,LDA,\mathbb{I},\mathbb{U},NFOUND,LDZ,LDW ORK,\mathbb{NFO}
\mathbb{NTEGER IN ORK 3 (*),\mathbb{FA IL (*)}}\mathbf{(*)}
DOUBLE PRECISION VL,VU,ABTOL
DOUBLE PRECISION W (*),W ORK2 (*)

```

```

    ABTOL,NFOUND,W,Z,LDZ,W ORK,LDW ORK,W ORK2,\mathbb{W}ORK3,\mathbb{FA}\mathbb{I},
    \mathbb{NFO)}
    ```
CHARACTER * 1 JOBZ,RANGE,UPLO
DOUBLE COM PLEX A (LDA, \(\left.{ }^{\star}\right)\), \(\mathrm{Z}(\mathrm{LD} \mathrm{Z}, \star), \mathrm{W}\) ORK (*)
\(\mathbb{N}\) TEGER*8N,LDA, \(\mathbb{L}, \mathbb{U}, N F O U N D, L D Z, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{I N}\) ORK 3 (*), \(\mathbb{F A} \mathbb{H}\) (*)
DOUBLE PRECISION VL,VU,ABTOL
DOUBLE PRECISION W (*), W ORK 2 (*)

\section*{F95 INTERFACE}

SU BROUTINE HEEVX (JOBZ,RANGE, UPLO, \(\mathbb{N}], A,[L D A], V L, V U, \mathbb{L}, \mathbb{I U}\), ABTOL, \(\mathbb{N} F O U N D], W, Z,[L D Z],[W\) ORK ], [LDW ORK ], [W ORK 2], [W ORK 3], \(\mathbb{F A} \mathbb{I},[\mathbb{N} F O])\)

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A , Z
\(\mathbb{N} T E G E R:: N, L D A, \mathbb{L}, \mathbb{U}, N F O U N D, L D Z, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{I W}\) ORK 3, \(\mathbb{F A} \mathbb{I}\)
REAL (8) ::VL,VU,ABTOL
REAL (8),D IM ENSION (:) ::W ,W ORK2

SU BROUTINE HEEVX_64 (JOBZ,RANGE,UPLO, \(\mathbb{N}\) ],A, [LDA ],VL,VU, IL, \(\mathbb{U}\), ABTOL, \(\mathbb{N} F O U N D], W, Z,[L D Z],[W\) ORK ], [LDW ORK], \(\mathbb{W}\) ORK2], [IW ORK 3], \(\mathbb{F A} \mathbb{I},[\mathbb{N F O}])\)

CHARACTER (LEN=1):: OBZ,RANGE,UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, Z
\(\mathbb{N} T E G E R(8):: N, L D A, \mathbb{L}, \mathbb{U}, N F O U N D, L D Z, L D W\) ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{I N} O R K 3, \mathbb{F A} \mathbb{L}\)
REAL (8) ::VL,VU,ABTOL
REAL (8),D IM ENSIO N (:) ::W ,W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zheevx (char jobz, char range, charuplo, int n, doublecom plex *a, int lda, double vl, double vu, int il, int iu, double abtol, int *nfound, double *w, doublecom plex *z, int ldz, int *ifail, int*info);
void zheevx_64 (char jobz, char range, char uplo, long n, doublecom plex *a, long lda, double vl, double vu, long il, long iu, double abtol, long *nfound, double \({ }^{*}\), doublecom plex *z, long ldz, long *ifail, long *info);

\section*{PURPOSE}
zheevx com putes selected eigenvalues and, optionally, eigenvectors of a com plex Hem itian matrix A. Eigenvahues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

\section*{ARGUMENTS}

JOBZ (input)
= N ': C om pute eigenvalues only;
\(=\mathrm{V}\) : C om pute eigenvalues and eigenvectors.

\section*{RANGE (input)}
= A ': alleigenvalues willbe found.
= V ':alleigenvalues in the half-open interval
( \(\mathrm{L}, \mathrm{VU}]\) w ill be found. = \(I\) ': the \(\mathbb{I}\)-th through \(\mathbb{I U}\)-th eigenvaluesw illlbe found.

UPLO (input)
= U ': U pper triangle ofA is stored;
= LL': Low ertriangle ofA is stored.
N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the \(H\) em tian m atrix \(A\). If U PLO \(=\mathrm{U}\) ', the leading N -by -N upper triangular partofA contains the upper triangularpart of the \(m\) atrix A . If UPLO \(=\mathrm{L}\) ', the leading N -by -N low ertriangularpartofA contains the lower triangular part of them atrix A. On exit, the low ertriangle (if \(\mathrm{UPLO}=\mathrm{L}\) ) or the uppertriangle (if \(\mathrm{UPLO}=\mathrm{U}\) ) of A, including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the anay A. LDA >= \(\max (1, N)\).

VL (input)
IfRANGE=V ', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE=A'or I'.

VU (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE = A' 'or I'.

II (input)
If RA N G E= I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}\), if \(\mathrm{N}>0 ; \mathbb{L}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N ot referenced ifRANGE= A 'or V'.

IU (input)
If RA N G E= I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}\), if \(\mathrm{N}>0\); \(\mathbb{L}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N otreferenced ifRANGE= A 'or V'.

ABTOL (input)
The absolute error tolerance for the eigenvalues.
A \(n\) approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b]
of w idth less than orequal to
ABTOL + EPS * max (k|, \(\mathrm{b} \mid)\),
where EPS is them achine precision. If ABTOL is less than orequal to zero, then EPS* \(|\mathbb{F}|\) w illbe used in its place, where \(F \mid\) is the 1 -norm of the tridiagonal m atrix obtained by reducing \(A\) to tridiagonal form .

E igenvalues w illbe com puted m ostaccurately when ABTOL is set to tw ice the underflow threshold \(2 \star\) SLAM CH ( \({ }^{\prime}\) ), notzero. If this routine retums w ith \(\mathbb{N} F O>0\), indicating that som e eigenvectors did not converge, try setting ABTOL to \(2 *\) SLAM CH (S ).

See "C om puting Sm allSingularV ahues of B idiagonal \(M\) atrices \(w\) ith \(G\) uaranteed \(H\) igh Relative A ccuracy," by D em m eland K ahan, LA PA CK W orking N ote \#3.

\section*{NFOUND (output)}

The total num ber of eigenvalues found. \(0<=\) NFOUND <= N. IfRANGE = A',NFOUND =N, and if RANGE \(=I^{\prime}\) ', NFOUND \(=\mathbb{U}-\mathbb{U}+1\).

W (output)
On norm alexit, the firstN FOUND elem ents contain the selected eigenvalues in ascending order.
\(Z\) (input) If \(\mathrm{OOBZ}=\mathrm{V}^{\prime}\), then if \(\mathbb{N} F O=0\), the first NFOUND colum ns of \(Z\) contain the orthonorm aleigenvectors of them atrix A corresponding to the selected eigenvalues, \(w\) ith the \(i\)-th colum \(n\) of \(Z\) holding the eigenvector associated w ith W (i). If an eigenvector fails to converge, then that colum n of \(Z\) contains the latest approxim ation to the eigenvector, and the index of the eigenvector is retumed in FAII. If \(J O B Z=N\) ', then \(Z\) is not referenced. \(N\) ote: the user must ensure that at least \(m\) ax ( 1, NFO UND ) colum ns are supplied in the aray \(Z\); if RANGE = V', the exactvalue ofNFOUND is not know \(n\) in advance and an upperbound \(m\) ustbe used.

LD Z (input)
The leading dim ension of the array \(\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}^{\prime}, \mathrm{LD} Z>=\mathrm{max}(1, \mathrm{~N})\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK ( 1 ) retums the optim al LDW ORK.

LDW ORK (input)
The length of the array \(W\) ORK. LDW ORK >= \(\max (1,2 \star \mathrm{~N})\). For optim al efficiency, LDW ORK \(>=\) \((N B+1) * N\), where \(N B\) is the \(m\) ax of the blocksize for CHETRD and forCUNM TR as retumed by HAENV.

IfLDW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of theW ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 2 (w orkspace)
dim ension ( \(7 * \mathrm{~N}\) )
IV ORK 3 (w orkspace)
dim ension ( \(5 * \mathrm{~N}\) )
FAII (output)
If \(\mathrm{OBZ}=\mathrm{V}\) ', then if \(\mathbb{N F O}=0\), the first NFOUND
elem ents of \(\mathbb{F A} I I\) are zero. If \(\mathbb{N F O}>0\), then
FFA IL contains the indices of the eigenvectors that failed to converge. If \(\mathrm{JOBZ}=\mathrm{N}\) ', then FA \(\mathbb{I L}\) is not referenced.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue
> 0 : if \(\mathbb{N F O}=\) i, then ieigenvectors failed to converge. Their indioes are stored in array
ㅍFAI.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhegs2 - reduce a com plex Herm tian-definite generalized eigenproblem to standard form

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHEGS2(TTYPE,UPLO,N,A,LDA,B,LDB,INFO )}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGER ITYPE,N,LDA,LDB,INFO
SU BROUT\mathbb{NE ZHEGS2_64(ITYPE,UPLO,N,A ,LDA,B,LDB, INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGER*8 ITYPE,N,LDA,LDB,INFO
F95 INTERFACE

```

```

    CHARACTER (LEN=1) ::UPLO
    COMPLEX (8),D IM ENSION(:,:)::A,B
    \mathbb{NTEGER ::TTYPE,N,LDA,LDB,INFO}
    SUBROUT\mathbb{NE HEGS2_64 (TTYPE,UPLO,N ,A , [LDA ],B, [LD B ], [N FO ])}
    CHARACTER (LEN=1) ::UPLO
    COM PLEX (8),D IM ENSION (:,:) ::A,B
    ```

```

C INTERFACE
\#include <sunperfh>

```
void zhegs2 (int ilype, charuplo, intn, doublecom plex *a, int lda, doublecom plex *b, int ldb, int *info);
void zhegs2_64 (long itype, charuplo, long n, doublecom plex
*a, long lda, doublecom plex *b, long ldb, long
*info);

\section*{PURPOSE}
zhegs2 reduces a com plex Herm itian-definite generalized eigenproblem to standard form .

If ITYPE \(=1\), the problem is \(A * x=\operatorname{lam}\) bda \({ }^{*}{ }^{*} x_{\mathrm{x}}\), and \(A\) is overw ritten by inv (U)*A *inv (U) orinv (L)*A *inv (L) If ITYPE \(=2\) or 3 , the problem is \(A * B * x=l a m\) bda* \(x\) or \(B * A * X=\operatorname{lam}\) bda* \(X\), and \(A\) is overw ritten by \(U * A * U\) ` or \(L\) * \(A * L\).

B m usthave been previously factorized as U *U or L*L' by CPOTRF.

\section*{ARGUMENTS}

ITYPE (input)
\(=1\) : com pute inv (U) \()\) A *inv (U) orinv (L)*A *inv (L) ;
\(=2\) or 3 : com pute \(U * A * U\) 'orL \({ }^{*} A * L\).
UPLO (input)
Specifies w hether the upper or low er triangular part of the \(H\) erm itian \(m\) atrix \(A\) is stored, and how B has been factorized. = U': U pper triangular = L ': Low ertriangular

N (input) The order of the m atrices A and \(\mathrm{B} . \mathrm{N}>=0\).
A (input/output)
On entry, the \(H\) erm itian \(m\) atrix A. If UPLO = U', the leading \(n\) by \(n\) upper triangular part of \(A\) contains the upper triangular part of the \(m\) atrix \(A\), and the strictly low ertriangularpartofA is not referenced. IfUPLO = L', the leading n by n low er triangularpartofA contains the low ertriangularpart of the m atrix A, and the strictly upper triangular part ofA is not referenced.

Onexit, if \(\mathbb{N F O}=0\), the transform ed \(m\) atrix, stored in the sam e form at as A.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).
\(B\) (input) The triangular factor from the Cholesky factorization ofB, as retumed by CPO TRF .

LD B (input)
The leading dim ension of the anay \(B\). LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0: successfulexit.
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvahue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhegst-reduce a complex Herm itian-definite generalized eigenproblem to standard form

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHEGST (TTYPE,UPLO,N,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGER ITYPE,N,LDA,LDB,INFO
SUBROUT\mathbb{NE ZHEGST_64(TTYPE,UPLO,N,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEXA (LDA,*),B (LDB,*)
INTEGER*8 \mathbb{TYPE,N,LDA,LDB,INFO}
F95 INTERFACE
SUBROUT\mathbb{NE HEGST (TTYPE,UPLO,N,A , [LDA ],B, [LDB],[INFO])}
CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D IM ENSION (:,:) ::A,B
\mathbb{NTEGER ::\mathbb{TYPE,N,LDA,LDB,INFO}}\mathbf{N}=,L
SUBROUT\mathbb{NE HEGST_64 (TTYPE,UPLO,N,A,[LDA ],B, [LDB],[NNFO])}
CHARACTER (LEN=1)::UPLO
COM PLEX (8),D IM ENSION (:,:) ::A,B
INTEGER (8) :: \mathbb{Y PE,N,LDA,LDB, NNFO}

```
C INTERFACE
    \#include < sunperfh>
void zhegst(int iype, charuple, intn, doublecom plex *a, int lda, doublecom plex *b, int ldb, int *info);
void zhegst 64 (long itype, charuplo, long n, doublecom plex
*a, long lda, doublecom plex *b, long ldb, long
*info);

\section*{PURPOSE}
zhegst reduces a com plex Herm itian-definite generalized eigenproblem to standard form .

If ITYPE \(=1\), the problem is \(A * x=\operatorname{lam}\) bda \({ }^{*} \mathrm{~B} * \mathrm{X}\), and \(A\) is overw rilten by inv \((U * * H) * A * i n v(U)\) or \(\operatorname{inv}(\mathbb{L}) * A * \operatorname{inv}(\mathbb{L} * *)\)

If ITYPE \(=2\) or 3 , the problem is A *B * \(\mathrm{x}=\operatorname{lam}\) bda* x or \(\mathrm{B} * \mathrm{~A} * \mathrm{x}=\operatorname{lam}\) bda* x , and A is overw ritten by \(\mathrm{U} * \mathrm{~A} * \mathrm{U} * * \mathrm{H}\) or \(\mathrm{L} * *_{\mathrm{H}}{ }^{*} \mathrm{~A} * \mathrm{~L}\).

B m usthave been previously factorized as \(\mathrm{U} * \mathrm{~A}_{\mathrm{H}} \mathrm{*U}_{\mathrm{U}}\) or \(\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) by CPO TRF.

\section*{ARGUMENTS}

ITYPE (input)
\(=1\) : compute \(\quad \operatorname{inv}(U * * H) * A * \operatorname{inv}(U)\) or inv (L) \({ }^{\star} \mathrm{A}\) *inv ( \(\left(\mathrm{L}^{* *} \mathrm{H}\right)\); \(=2\) or 3 : com pute \(U{ }^{*} A * U * * H\) or \(L * * H * A * L\).

UPLO (input)
\(=\mathrm{U}\) ': Uppertriangle of \(A\) is stored and \(B\) is factored as U ** H * U ; = L': Low ertriangle ofA is stored and \(B\) is factored as \(L * L * * H\).

N (input) The order of the m atrioes A and \(\mathrm{B} . \mathrm{N}>=0\).
A (input/output)
O n entry, the H erm itian m atrix A . If \(\mathrm{U} P \mathrm{O}=\mathrm{U}\) ', the leading \(\mathrm{N}-\) by -N uppertriangularpartofA contains the upper triangular part of the \(m\) atrix \(A\), and the strictly low ertriangularpart of A is not referenced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading N -by -N low er triangularpart ofA contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpartofA is not referenced.

On exit, if \(\mathbb{N F O}=0\), the transform ed matrix, stored in the sam e form at as A.

\section*{LD A (input)}

The leading dim ension of the array A. LD A >= \(\max (1, N)\).
\(B\) (input) The triangular factor from the C holesky factorization ofB, as retumed by CPO TRF .

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhegv - com pute all the eigenvalues, and optionally, the eigenvectors of a com plex generalized H erm itian-definite eigenproblem, of the form \(A * x=(l a m . b d a) * B * x, A * B x=(l a m . b d a){ }^{*} x\), or \(B * A * x=(l a m, b d a){ }^{*} x\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHEGV (TTYPE,JOBZ,UPLO,N,A,LDA,B,LDB,W ,W ORK,}
LDW ORK,W ORK2,INFO)
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEXA (LDA,*),B (LDB,*),W ORK (*)
\mathbb{NTEGER ITYPE,N,LDA,LDB,LDW ORK,INFO}
DOUBLE PRECISION W (*),W ORK2 (*)
SU BROUT\mathbb{NE ZHEGV_64 (TTYPE,NOBZ,UPLO,N,A,LDA,B,LDB,W,W ORK,}
LDW ORK,W ORK2,INFO)
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*),W ORK (*)
\mathbb{NTEGER*8 ITYPE,N,LDA,LDB,LDW ORK,INFO}
DOUBLE PRECISION W (*),W ORK2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE HEGV (TTYPE, \(\operatorname{JoB} \mathrm{Z}, \mathrm{U} P L O, N, A,[L D A], B,[L D B], W, \mathbb{W} O R K]\),
[LDW ORK], [WORK2], [ \(\mathbb{N F O}]\) )
CHARACTER (LEN=1)::JOBZ,UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{I}\) ENSION (:,:) ::A,B
\(\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D A, L D B, L D W O R K, \mathbb{N} F O\)
REAL (8), D IM ENSION (:) ::W ,W ORK2

SU BROUTINE HEGV_64 (TTYPE, \(\operatorname{OOBZ}, \mathrm{UPLO}, \mathrm{N}, \mathrm{A},[\mathrm{LDA}], \mathrm{B},[\mathrm{LDB}], \mathrm{W},[\mathbb{W} O R K]\), [LDW ORK], [W ORK2], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1)::JOBZ,UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COMPLEX (8),D \(\mathbb{D}\) ENSION (:,:) ::A, B
\(\mathbb{N} T E G E R(8):: \mathbb{T} Y P E, N, L D A, L D B, L D W O R K, \mathbb{N} F O\)
REAL (8),D IM ENSION (:) ::W ,W ORK2

\section*{C INTERFACE}
\#include < sunperfh>
void zhegv (int itype, char jobz, char uplo, int \(n\), doublecom plex *a, int lda, doublecom plex *b, int ldb, double \({ }_{\text {W }}\), int *info);
void zhegv_64 (long itype, char j̀bz, char uplo, long n, doublecom plex *a, long lda, doublecom plex *b, long ldb, double *w , long *info);

\section*{PURPOSE}
zhegv com putes all the eigenvalues, and optionally, the eigenvectors of a com plex generalized H erm itian-definite eigenproblem, of the form \(A * x=(l a m ~ b d a) * B * x\), \(\mathrm{A} * \mathrm{~B} \mathrm{x}=(\operatorname{lam} . \mathrm{bda}){ }^{\star} \mathrm{x}\), or \(\mathrm{B} * \mathrm{~A} * \mathrm{X}=\left(\mathrm{lam}\right.\) bda) \({ }^{*} \mathrm{x}\). H ere A and B are assum ed to be \(H\) erm itian and \(B\) is also positive definite.

\section*{ARGUMENTS}

ITYPE (input)
Specifies the problem type to be solved:
\(=1: A{ }^{*} \mathrm{x}=\left(\mathrm{lam}\right.\) bda) \({ }^{\mathrm{B}}{ }^{*}{ }_{\mathrm{x}}\)
\(=2: \mathrm{A} * \mathrm{~B} * \mathrm{X}=\left(\mathrm{lam}\right.\) bda) \({ }^{*} \mathrm{X}\)
\(=3: B * A * X=(l a m ~ b d a){ }^{*} \mathrm{x}\)
\(J O B Z\) (input)
\(=N^{\prime}:\) C om pute eigenvalues only;
\(=\mathrm{V}:\) : C om pute eigenvalues and eigenvectors.

UPLO (input)
\(=\mathrm{U}\) ': U ppertriangles of \(A\) and \(B\) are stored;
\(=\mathrm{L}\) ': Low ertriangles of \(A\) and \(B\) are stored.

N (input) The order of the m atrices A and \(\mathrm{B} . \mathrm{N}>=0\).

A (input/output)
On entry, the \(H\) erm itian \(m\) atrix A. If UPLO = U', the leading N -by N upper triangularpartof A contains the upper triangularpart of the \(m\) atrix \(A\). If UPLO = L', the leading N -by-N low er triangular partofA contains the low er triangular part of the m atrix A.

On exit, if \(J \mathrm{OBZ}=\mathrm{V}^{\prime}\), then if \(\mathbb{N} F O=0\), A contains the \(m\) atrix \(Z\) ofeigenvectors. The eigenvectors are norm alized as follow s: if ITYPE = 1 or \(2, Z * * H * B * Z=I ;\) if \(I T Y P E=3, Z * * H * i n v(B) * Z=I\). If \(\mathrm{JOBZ}=\mathrm{N}\) ', then on exit the upper triangle (if \(\mathrm{U} P L O=\mathrm{U}\) ) or the low er triangle (if \(\mathrm{U} P L O=\mathrm{L}\) ) of A , including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

B (input/output)
On entry, the \(H\) erm itian positive definite \(m\) atrix \(B\). If \(U P L O=U\) ', the leading \(N\) by \(N\) uppertriangularpartofB contains the upper triangular part of them atrix B. IfU PLO = L', the leading N toy -N low er triangularpart of B contains the low er triangularpart of the \(m\) atrix B .

On exit, if \(\mathbb{N} F O<=N\), the part of \(B\) containing the \(m\) atrix is overw rilten by the triangular factor U orL from the Cholesky factorization \(\mathrm{B}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(B=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\).

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).

W (output)
If \(\mathbb{N}\) FO \(=0\), the eigenvalues in ascending order.

\section*{W ORK (w orkspace)}

On exit, if \(\mathbb{N} F O=0, W\) ORK ( 1 ) retums the optim al LDW ORK.

LDW ORK (input)
The length of the array \(W\) ORK. LDW ORK >= max \(\left(1,2^{\star} \mathrm{N}-1\right)\). For optim alefficiency, LD W ORK >= \((\mathrm{NB}+1) * \mathrm{~N}\), where NB is the blocksize for CHETRD retumed by LLAENV.

IfLD W ORK \(=-1\), then aw orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK anray, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 2 (w orkspace)
dim ension (max (1,3*N-2))
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
< \(0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
> 0: CPOTRF orCH EEV retumed an errorcode:
\(<=\mathrm{N}\) : if \(\mathbb{N F O}=\mathrm{i}, \mathrm{CHEEV}\) failed to converge; i off-diagonalelem ents of an interm ediate tridiagonal form did notconverge to zero; \(>N\) : if \(\mathbb{N F O}\)
\(=N+i\), for \(1<=i<=N\), then the leading \(m\) inor oforderiofB is not positive definite. The factorization of B could notbe com pleted and no eigenvahues or eigenvectors w ere com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zhegvd - com pute all the eigenvalues, and optionally, the eigenvectors of a com plex generalized H erm itian-definite eigenproblem, of the form \(A * x=(l a m . b d a) * B * x, A * B x=(l a m ~ b d a) * x\), or \(B{ }^{*} A * X=\left(l a m\right.\) bda) \({ }^{*} X\)

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZHEGVD (ITYPE,JOBZ,UPLO,N,A,LDA,B,LDB,W ,W ORK,}
LW ORK,RW ORK,LRW ORK,IN ORK,LIW ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*),W ORK (*)
\mathbb{NTEGER ITYPE,N,LDA,LDB,LW ORK,LRW ORK,LIN ORK,\mathbb{NFO}}\mathbf{N},\textrm{L}
INTEGER IN ORK (*)
DOUBLE PRECISION W (*),RW ORK (*)
SU BROUT\mathbb{NE ZHEGVD_64 (TTYPE, OOBZ,UPLO,N,A,LDA,B,LDB,W ,W ORK,}
LW ORK,RW ORK,LRW ORK,INORK,LIN ORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*),W ORK (*)
INTEGER*8 ITYPE,N,LDA,LDB,LW ORK,LRW ORK,LIN ORK,INFO
INTEGER*8 代ORK (*)
DOUBLE PRECISION W (*),RW ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE HEGVD (TTYPE, JOBZ, UPLO, \(\mathbb{N}], A,[L D A], B,[L D B], W,[W O R K]\), \([\) [W ORK ], RW ORK ], [LRW ORK ], [IW ORK], [LINORK], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1)::JOBZ,UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK

COMPLEX (8), D \(\mathbb{M}\) ENSION (: : : : : A , B
\(\mathbb{N}\) TEGER :: ITYPE,N,LDA,LDB,LW ORK,LRW ORK,LIN ORK, \(\mathbb{N} F\) FO
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL (8),D IM ENSION (:) ::W ,RW ORK

SU BROUTINE HEGVD_64 (TTYPE, JOBZ, UPLO, \(\mathbb{N}], A,[L D A], B,[L D B], W\), \([\mathbb{W}\) ORK ], [LW ORK ], RW ORK ], [LRW ORK ], [IW ORK ], [LIN ORK ], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1)::JOBZ,UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{D}\) ENSION (:,:) ::A, B
\(\mathbb{N}\) TEGER (8) :: \(\mathbb{T} Y\) PE,N,LDA, LDB, LW ORK, LRW ORK, LIN ORK,
\(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL (8),D IM ENSIO N (:) ::W ,RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zhegvd (int itype, char jobz, charuplo, int n, doublecom plex *a, int lda, doublecom plex *b, int ldb, double *w , int *info);
void zhegvd_64 (long itype, char jobz, char uplo, long n, doublecom plex *a, long lda, doublecom plex *b, long ldb, double *w , long *info);

\section*{PURPOSE}
zhegvd com putes allthe eigenvalues, and optionally, the eigenvectors of a com plex generalized \(H\) erm itian-definite eigenproblem, of the form \(A * x=(l a m ~ b d a) * B * x\), \(\mathrm{A} * \mathrm{~B} \mathrm{x}=(\operatorname{lam} \mathrm{bda}){ }^{*} \mathrm{x}\), or \(\mathrm{B} * \mathrm{~A} * \mathrm{X}=(\mathrm{lam} . \mathrm{bda}){ }^{*} \mathrm{x}\). H ere A and B are assum ed to be \(H\) erm itian and \(B\) is also positive definite. If eigenvectors are desired, ituses a divide and conquer algorithm .

The divide and conquer algorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on \(m\) achines w ith a guard digit in add/subtract, or on those binary \(m\) achines \(w\) thout guard digits w hich subtract like the \(C\) ray X M P , C ray Y M P , C ray C -90, orC ray-2. It could conceivably fail on hexadecim al or decim al machines without guard digits, butw e know of none.

\section*{ARGUMENTS}

ITYPE (input)

Specifies the problem type to be solved:
\(=1: A{ }^{*}=\left(\operatorname{lam}\right.\) bda) \({ }^{\mathrm{B}} \mathrm{A}^{*} \mathrm{x}\)
\(=2: A * B * x=\left(l a m\right.\) bda) \({ }^{*} \mathrm{x}\)
\(=3: B{ }^{*}{ }^{*}{ }^{x}=\left(\operatorname{lam}\right.\) bda) \({ }^{*} \mathrm{X}\)

JOBZ (input)
\(=\mathrm{N}\) ': C om pute eigenvalues only;
\(=\mathrm{V}\) :: C om pute eigenvalues and eigenvectors.
UPLO (input)
\(=\mathrm{U}:\) : U pper triangles of \(A\) and \(B\) are stored;
= L': Low ertriangles of \(A\) and \(B\) are stored.
N (input) The order of the m atrices A and \(\mathrm{B} . \mathrm{N}>=0\).
A (input/output)
On entry, the \(H\) erm itian m atrix A. If UPLO = U', the leading N -by N uppertriangularpart of A contains the upper triangularpart of the \(m\) atrix \(A\). If UPLO = L', the leading N -by N low er triangular partofA contains the low er triangular part of them atrix \(A\).

Onexit, if \(J O B Z=V^{\prime}\), then if \(\mathbb{N} F O=0, A\) contains the \(m\) atrix \(Z\) of eigenvectors. The eigenvectors are norm alized as follow s: if ITYPE = 1 or \(2, Z * * H * B * Z=I ;\) if \(I T Y P E=3, Z * * H * i n v(B) * Z=I\). If \(J O B Z=N\) ', then on exit the upper triangle (if \(\mathrm{UPLO}=\mathrm{U}\) ) or the low er triangle (if \(\mathrm{UPLO}=\mathrm{L}\) ) of A , including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, N)\).

B (input/output)
O n entry, the \(H\) erm itian m atrix B. If UPLO = U', the leading N -by N uppertriangularpart of B contains the upper triangularpart of the \(m\) atrix \(B\). If UPLO = L', the leading N by N low er triangular part ofB contains the low er triangular part of them atrix B .

Onexit, if \(\mathbb{N} F O<=N\), the part of \(B\) containing the \(m\) atrix is overw rilten by the triangular factor U orL from the Cholesky factorization \(\mathrm{B}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(B=L * L^{* *} \mathrm{H}\).

LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, N)\).

W (output)
If \(\mathbb{N} F O=0\), the eigenvalues in ascending order.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The length of the amay \(W\) ORK. If \(N<=1\), LW ORK \(>=1\). If \(O B Z=N\) 'and \(N>1, L W O R K>=N\) +1 . If \(O B Z=V\) 'andN \(>1\), LW ORK \(>=2 * N+\) \(\mathrm{N} * * 2\) 。

If LW O RK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK aray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

RW ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0\), RW ORK (1) retums the optim al LRW ORK.

LRW ORK (input)
The dim ension of the aray RW ORK. If \(\mathrm{N}<=1\), LRW ORK \(>=1\). If \(\mathrm{OBZ}=\mathrm{N}\) 'and \(\mathrm{N}>1\),LRW ORK \(>=\) N . If \(\mathrm{OBB}=\mathrm{V}^{\prime}\) and \(\mathrm{N}>1\),LRW ORK \(>=1+5 \star \mathrm{~N}+\) \(2 * N * * 2\) 。

If LRW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the RW ORK aray, retums this value as the first entry of the RW ORK array, and no errorm essage related to LRW ORK is issued by X ERBLA.

IW ORK (w orkspace/output)
On exit, if \(\mathbb{N} F O=0\), \(\mathbb{I N}\) O RK (1) retums the optim al
LIW ORK.

LIW ORK (input)
The dim ension of the anay \(\mathbb{I N} O R K\). If \(\mathrm{N}<=1\), LIW ORK \(>=1\). If \(O B Z=N\) 'andN \(>1, L \mathbb{W} O R K>=\) 1. If \(\mathrm{OBBZ}=\mathrm{V}^{\prime}\) and \(\mathrm{N}>1, \mathrm{~L} \mathbb{I} \mathrm{ORK}>=3+5 * \mathrm{~N}\).

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\) th argum enthad an illegalvalue
\(>0\) : CPOTRF orCHEEVD retumed an error code:
<= N: if \(\mathbb{N} F O=i, C H E E V D\) failed to converge; i off-diagonalelem ents of an interm ediate tridiagonal form did notconverge to zero; > N : if \(\mathbb{N}\) FO \(=N+i\), for \(1<=i<=N\), then the leading \(m\) inor oforderiofB is not positive definite. The factorization of B could notbe com pleted and no eigenvalues or eigenvectors w ere com puted.

\section*{FURTHER DETAILS}

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv. of K entucky, USA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zhegvx - com pute selected eigenvalues, and optionally, eigenvectors of a com plex generalized H erm itian-definite eigenproblem, of the form \(\mathrm{A} * \mathrm{x}=\left(\mathrm{lam}\right.\) bda) \({ }^{\mathrm{B}} \mathrm{B}\) x, \(\mathrm{A} * \mathrm{~B} x=\left(\operatorname{lam}\right.\) bda) \({ }^{\mathrm{x}}\), or \(B * A * X=\left(l a m\right.\) bda) \({ }^{*} X\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHEGVX (TTYPE,JOBZ,RANGE,UPLO,N,A,LDA,B,LDB,VL,}
VU,\mathbb{L,IU,ABSTOL,M ,W ,Z,LD Z,W ORK,LW ORK,RW ORK,IN ORK,}
\mathbb{FA}\mathbb{I},\mathbb{NNFO)}
CHARACTER * 1 JOBZ,RANGE,UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*),Z (LD Z,*),W ORK (*)
\mathbb{NTEGER ITYPE,N,LDA,LDB,\mathbb{L},\mathbb{U},M,LD Z,LW ORK,\mathbb{NFO}}\mathbf{N},\textrm{L}
\mathbb{NTEGER IN ORK (*), \mathbb{FA}\mathbb{L}(*)}
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION W (*),RW ORK (*)
SU BROUT\mathbb{NE ZHEGVX_64 (TTYPE,NOBZ,RANGE,UPLO,N,A,LDA,B,LDB,VL,}
VU,\mathbb{L},\mathbb{U},ABSTOL,M,W,Z,LD Z,W ORK,LW ORK,RW ORK,IV ORK,
FA[|,\mathbb{NFO)}
CHARACTER * 1 JOBZ,RANGE,UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*),Z (LD Z,*),W ORK (*)

```

```

INTEGER*8 IN ORK (*), \mathbb{FA IL (*)}
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION W (*),RW ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE HEGVX (TTYPE, \(\operatorname{OOBZ}, R A N G E, U P L O, \mathbb{N}], A,[L D A], B,[L D B]\),
\(\mathrm{VL}, \mathrm{VU}, \mathbb{I}, \mathbb{I}, \mathrm{ABSTOL}, \mathrm{M}, \mathrm{W}, \mathrm{Z},[\mathrm{LD} \mathrm{Z}],[\mathrm{W}\) ORK ], [LW ORK], RW ORK], [ \(\mathbb{I N}\) ORK], \(\mathbb{F A} \mathbb{H},[\mathbb{N} F O]\) )

CHARACTER (LEN=1):: JBZ,RANGE,UPLO
COM PLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, B,Z
\(\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D A, L D B, \mathbb{L}, \mathbb{U}, M, L D Z, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{I N}\) ORK, \(\mathbb{F} A \mathbb{I}\)
REAL (8) ::VL,VU,ABSTOL
REAL (8),D IM ENSION (:) ::W ,RW ORK

SU BROUTINE HEGVX_64 (TTYPE, JOBZ,RANGE,UPLO, \(\mathbb{N}]\) ], A, [LDA ], B, [LDB], VL,VU, \(\mathbb{I}, \mathbb{I}, A B S T O L, M, W, Z,[L D Z],[W O R K],[L W O R K], \mathbb{R W}\) ORK ], [ \(\mathbb{I N}\) ORK], \(\mathbb{F A} \mathbb{I},[\mathbb{N} F O]\) )

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A, B, Z
\(\mathbb{N}\) TEGER (8) :: ITYPE,N,LDA,LDB, \(\mathbb{L}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W\) ORK,
\(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER (8),D \(\mathbb{I M} \operatorname{ENSION}(:):: \mathbb{I N}\) ORK, \(\mathbb{F} A \mathbb{I}\)
REAL (8) ::VL,VU,ABSTOL
REAL (8),D IM ENSION (:) ::W ,RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zhegvx (int itype, char j̀bz, char range, charuplo, int
n, doublecom plex *a, int lda, doublecom plex *b, int ldb, double vl, double vu, int il, int iu, double abstol, int \({ }_{\mathrm{m}}\), double \({ }_{\mathrm{w}}\), doublecom plex *z, int ldz, int *ifail, int *info);
void zhegvx_64 (long itype, char jobz, char range, char uplo, long n, doublecom plex *a, long lda, doublecom plex *b, long ldb, double vl, double vu, long il, long iu, double abstol, long *m, double \({ }^{\mathrm{w}} \mathrm{w}\), doublecom plex *z, long ldz, long *ifail, long *info);

\section*{PURPOSE}
zhegvx com putes selected eigenvalues, and optionally, eigenvectors of a com plex generalized \(H\) erm itian-definite eigenproblem, of the form \(A * x=(\operatorname{lam} . b d a) * B * x, A * B x=(l a m ~ b d a){ }^{*} x\), or \(B{ }^{A} A * X=\left(\operatorname{lam}\right.\) bda) \({ }^{*} X\). H ere \(A\) and \(B\) are assum ed to be \(H\) erm titian and \(B\) is also positive definite. Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

\section*{ARGUMENTS}

ITYPE (input)
Specifies the problem type to be solved:
\(=1: A{ }^{*} \mathrm{x}=\left(\operatorname{lam}\right.\) bda)\({ }^{\mathrm{B}}{ }^{*} \mathrm{x}\)
\(=2: \mathrm{A} * \mathrm{~B} * \mathrm{x}=\left(\operatorname{lam}\right.\) bda) \({ }^{*} \mathrm{x}\)
\(=3: B * A * x=\left(l a m\right.\) bda) \({ }^{*} \mathrm{x}\)
JOBZ (input)
\(=\mathrm{N}\) ': C om pute eigenvalues only;
= V ': C om pute eigenvalues and eigenvectors.
RANGE (input)
= A : alleigenvalues w illbe found.
= V ': alleigenvalues in the half-open interval ( \(\mathrm{L}, \mathrm{VU}]\) w ill be found. = ' I ': the I -th through \(\mathbb{I U}\)-th eigenvaluesw illlbe found.

UPLO (input)
\(=\mathrm{U}\) ': U pper triangles of \(A\) and \(B\) are stored;
\(=\mathrm{L}\) ': Low ertriangles of \(A\) and \(B\) are stored.
\(N\) (input) The order of the \(m\) atrices \(A\) and \(B . N>=0\).
A (input/output)
O n entry, the \(H\) erm itian \(m\) atrix \(A\). If \(U P L O=U '\), the leading N -by N uppertriangular partofA contains the upper triangular part of the \(m\) atrix \(A\). If UPLO = L', the leading N by -N low er triangular part ofA contains the low er triangular part of them atrix \(A\).

On exit, the low er triangle (if \(\mathrm{UPLO}=\mathrm{L}\) ) or the upper triangle (if \(\mathrm{U} P \mathrm{O}=\mathrm{U}\) ) of , including the diagonal, is destroyed.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, \mathbb{N})\).

B (input/output)
O n entry, the \(H\) erm tian \(m\) atrix \(B\). If \(U P L O=U '\), the leading N -by N uppertriangular partofB contains the upper triangular part of the \(m\) atrix \(B\). If UPLO \(=\mathrm{L}\) ', the leading N -by -N low er triangular partofB contains the low er triangular part of the \(m\) atrix \(B\).

On exit, if \(\mathbb{N} F O<=N\), the part of \(B\) containing the \(m\) atrix is overw rilten by the triangular factor
U orL from the Cholesky factorization \(\mathrm{B}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(B=L{ }^{*} \mathrm{~L}^{* *} \mathrm{H}\).

LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, N)\).

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRA NGE = A 'or I'.
VU (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU. N ot referenced ifRANGE=A 'or I '.

II (input)
IfRA N G E= I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}\), ifN \(>0\); \(\mathbb{H}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N ot referenced ifRANGE \(=\) A'or V'.

IU (input)
If RA NGE= 'I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{Z}<=\mathbb{U}<=\mathrm{N}\), if \(\mathrm{N}>0\); \(\mathbb{H}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N otreferenced ifRANGE \(=\) A'or V'.

ABSTOL (input)
The absolute error tolerance for the eigenvalues.
A \(n\) approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b]
of w idth less than orequal to
ABSTOL + EPS * max ( \(k|\),\(| b|),\)
where EPS is the m achine precision. IfA BSTOL is less than or equalto zero, then EPS* \(\mid\) | w illbe used in its place, where \(F \mid\) is the 1 -norm of the tridiagonal \(m\) atrix obtained by reducing \(A\) to tridiagonal form .

E igenvalues w illbe com puted m ost accurately when ABSTOL is set to tw ioe the underflow threshold \(2 *\) SLAM CH ( \(\mathrm{S}^{\prime}\) ), not zero. If this routine retums w ith \(\mathbb{N}\) FO \(>0\), indicating that som e eigenvectors did
not converge, try setting ABSTOL to \(2 *\) SLAM CH (S ).
M (output)
The total num ber ofeigenvalues found. \(0<=\mathrm{M}<=\)
N . IfRANGE \(=\mathrm{A}^{\prime}, \mathrm{M}=\mathrm{N}\), and ifRANGE \(=\mathrm{I}^{\prime}, \mathrm{M}=\) \(\mathbb{U}-\mathbb{L}+1\).

W (output)
The firstM elem ents contain the selected eigenvalues in ascending order.
Z (output)
If \(\mathrm{JOBZ}=\mathrm{N}\) ', then Z is not referenced. If JOBZ \(=\mathrm{V}\) ', then if \(\mathbb{N} F O=0\), the firstM columns of \(Z\) contain the orthonorm aleigenvectors of the \(m\) atrix A comesponding to the selected eigenvalues, \(w\) ith the i-th colum \(n\) of \(Z\) holding the eigenvectorassociated \(w\) ith \(W\) (i). The eigenvectors are norm alized as follow s: if \(I T Y P E=1\) or \(2, Z * * T * B * Z=I\); if TTYPE \(=3, Z * * T * \operatorname{inv}(B) * Z=I\).

If an eigenvector fails to converge, then that colum \(n\) of \(Z\) contains the latestapproxim ation to the eigenvector, and the index of the eigenvector is retumed in \(\mathbb{F A} \mathbb{I}\). N ote: the userm ustensure that at leastm ax \((1, M)\) colum ns are supplied in the aray \(Z\); ifRANGE = V', the exactvalue of \(M\) is notknow \(n\) in advance and an upper bound \(m\) ust be used.

LD Z (input)
The leading dim ension of the array \(Z . L D Z>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD \(Z>=\mathrm{max}(1, N)\).

W ORK (w orkspace/output)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The length of the array \(W\) ORK. LW ORK >= \(\max \left(1,2{ }^{*} \mathrm{~N}-1\right)\). For optim al efficiency, LW ORK \(>=\) \((N B+1) * N\), where \(N B\) is the blocksize for CHETRD retumed by ILAENV.

If LW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

RW ORK (w orkspace)
dim ension ( \(7 * \mathrm{~N}\) )

IV ORK (w orkspace)
dim ension ( \(5 * N\) )

FAII (output)
If \(J O B Z=V^{\prime}\), then if \(\mathbb{N F O}=0\), the firstM ele\(m\) ents of \(\mathbb{F} A \mathbb{I}\) are zero. If \(\mathbb{N} F O>0\), then \(\mathbb{F} A \mathbb{H}\) contains the indices of the eigenvectors that failed to converge. If \(\mathrm{OBB}=\mathrm{N}\) ', then \(\mathbb{F} A \mathbb{I}\) is notreferenced.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i-\) th argum enthad an illegalvalue
> 0: CPO TRF orCHEEVX retumed an errorcode:
\(<=\mathrm{N}:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{CHEEVX}\) failed to converge; i eigenvectors failed to converge. Their indices are stored in amay \(\mathbb{F A} \mathbb{I} .>N:\) if \(\mathbb{N F O}=\mathrm{N}+\) \(i\), for \(1<=i<=N\), then the leading \(m\) inor of orderiofB is notpositive definite. The factorization of \(B\) could not be com pleted and no eigenvalues or eigenvectors w ere com puted.

\section*{FURTHER DETAILS}

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv. of K entucky, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhem \(m\)-perform one of the \(m\) atrix-m atrix operations \(C:=\) alpha*A *B + beta*C orC : alpha*B *A + beta*C

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHEMM (SDE,UPLO,M,N,ALPHA,A,LDA,B,LDB,BETA,C,}
LD C )
CHARACTER * 1SDEEUPLO
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (LDA,*),B (LDB ,*),C (LD C ,*)
INTEGERM,N,LDA,LDB,LDC
SU BROUTINE ZHEMM _64 (S\mathbb{DE,UPLO ,M ,N,ALPHA,A,LDA,B,LDB,BETA C,}
LD C)
CHARACTER * 1 SDEEUPLO
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (LDA,*),B (LDB,*),C (LD C ,*)
INTEGER*8M,N,LDA,LDB,LDC

```

\section*{F95 INTERFACE}

SU BROUTINE HEMM (SDE, UPLO, \(\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A], B,[L D B]\), BETA, C, [LDC])

CHARACTER (LEN=1) ::SDE,UPLO
COMPLEX (8) ::ALPHA,BETA
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) :: A, B, C
\(\mathbb{I N}\) TEGER ::M , N,LDA,LDB,LDC
SUBROUTINE HEMM _64 (SDE, UPLO, \(\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A], B,[L D B]\), BETA, C, [LDC])

CHARACTER ( \([E N=1):: S D E, U P L O\)
COM PLEX (8) ::ALPHA,BETA
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : ) :: A , B , C
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{LD} A, L D B, L D C\)

\section*{C INTERFACE}
\#include <sunperfh>
void zhem m (charside, char uplo, intm, intn, doublecom plex *alpha, doublecom plex *a, int lda, doublecom plex
*b, int ldlo, doublecom plex *beta, doublecom plex \({ }^{*} \mathrm{C}\), int ldc);
void zhem m _64 (charside, charuplo, long m , long n, doublecomplex *alpha, doublecom plex *a, long lda, doublecom plex *b, long ldl, doublecom plex *beta, doublecom plex \({ }^{\star}\) c, long ldc);

\section*{PURPOSE}
zhem \(m\) perform sone of the \(m\) atrix \(m\) atrix operations \(C:=\) alpha*A *B + beta*C orC := alpha*B *A + beta*C where alpha and beta are scalars, \(A\) is an herm itian \(m\) atrix and \(B\) and \(C\) are \(m\) by \(n m\) atrices.

\section*{ARGUMENTS}

SID E (input)
On entry, SIDE specifiesw hether the herm itian \(m\) atrix A appears on the leftor right in the operation as follow s:
\(S \mathbb{D} E=\) L'or \(\mathrm{I}^{\prime} \mathrm{C}:=\) alpha*A *B + beta*C,
\(S \mathbb{D E}=\mathrm{R}\) 'or \(\mathrm{r}^{\prime} \mathrm{C}:=\) alpha*B*A + beta* C ,

U nchanged on exit.

UPLO (input)
On entry, UPLO specifies whether the upper
or lower triangular part of the herm itian
\(m\) atrix \(A\) is to be referenced as follow \(s\) :
\(\mathrm{UPLO}=\mathrm{U}\) 'or L ' Only the upper triangularpart of the herm itian m atrix is to be referenced.

UPLO = 'L 'or I' O nly the low er triangularpart
of the herm itian \(m\) atrix is to be referenced.

U nchanged on exit.

M (input)
O \(n\) entry, \(M\) specifies the num ber of row sof the \(m\) atrix \(C . M>=0\). U nchanged on exit.

N (input)
O n entry, N specifies the num ber of colum ns of the \(m\) atrix \(C . N>=0\). U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
COM PLEX *16 array ofD \(\mathbb{I M}\) ENSION (LDA, ka ), where ka ism when \(S \mathbb{D E}=\mathrm{L}\) 'or \(\mathrm{I}^{\prime}\) and is n otherw ise.

Before entry with SIDE = L'or I', the m by m partof the array A mustcontain the herm itian \(m\) atrix, such thatw hen UPLO \(=U\) ' or \(L\) ', the leading \(m\) by \(m\) uppertriangularpart of the array A mustcontain the upper triangular part of the herm itian \(m\) atrix and the strictly low er triangularpart of A is not referenced, and \(w\) hen UPLO = L' or \({ }^{\prime}\) ', the leading \(m\) by \(m\) low er triangularpart of the array A must contain the lowertriangularpart of the herm itian \(m\) atrix and the strictly upper triangular part of A is not referenced.

Before entry w ith \(S \mathbb{D E}=\mathrm{R}\) 'or r ', the n by \(n\) partof the amay A mustcontain the herm itian \(m\) atrix, such thatw hen UPLO \(=U\) ' or \(L^{\prime}\) ', the leading \(n\) by \(n\) uppertriangularpart of the array A mustcontain the upper triangular part of the herm titian \(m\) atrix and the strictly low er triangularpart of A is not referenced, and when UPLO = L' or 1 ', the leading \(n\) by \(n\) low er triangularpart of the array A must contain the lowertriangularpart of the herm itian \(m\) atrix and the strictly upper triangular part of A is not referenced.
\(N\) ote that the im aginary parts of the diagonal elem ents need notbe set, they are assum ed to be zero. U nchanged on exit.

LDA (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen \(S \mathbb{D} E=\mathbb{L}\) 'or \(\mathrm{I}^{\prime}\) then LD \(A>=m a x(1, m)\), otherw ise LD A \(>=\max (1, \mathrm{n})\). U nchanged on exit.

B (input)
COM PLEX *16 array ofD \(\mathbb{I M}\) ENSION (LD \(\mathrm{B}, \mathrm{n}\) ). Before
entry, the leading \(m\) by \(n\) partof the array \(B\) m ustcontain the \(m\) atrix \(B\). Unchanged on exit.
LD B (input)
On entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program.
LD B must be at leastmax ( \(1, \mathrm{~m}\) ). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then \(C\) need not.be set on input. U nchanged on exit.

C (input/output)
COM PLEX *16 aray ofD \(\mathbb{I M}\) ENSION (LD C, n ).
Before entry, the leading \(m\) by \(n\) part of the aray \(C\) mustcontain the \(m\) atrix \(C\), exceptw hen beta is zero, in which case \(C\) need notbe set on entry.

On exit, the array \(C\) is overw ritten by the \(m\) by n updated m atrix.

LD C (input)
O n entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program. LD C must be at leastmax ( \(1, m\) ). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhem \(v\)-perform the \(m\) atrix-vectoroperation \(y:=a l p h a * A * x\) + beta*y

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHEMV (UPLO,N,ALPHA,A,LDA,X,INCX,BETA,Y,INCY)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (LDA,*),X (*),Y (*)
\mathbb{NTEGER N,LDA, INCX,}\mathbb{NCY}
SUBROUT\mathbb{NE ZHEM V_64(UPLO,N,ALPHA,A,LDA,X, INCX,BETA,Y,INCY)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (LDA,*),X (*),Y (*)
\mathbb{NTEGER*8N,LDA, INCX,}\mathbb{N}CY

```

\section*{F95 INTERFACE}

SU BROUTINE HEMV (UPLO, \(\mathbb{N}]\),A LPHA, A, [LDA ], X, [ \(\mathbb{N C X}], B E T A, Y,[\mathbb{N C Y}])\)
CHARACTER (LEN=1) ::UPLO
COMPLEX (8) ::ALPHA,BETA
COMPLEX (8),D IM ENSION (:) ::X,Y
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER ::N,LDA, \(\mathbb{N C X}, \mathbb{N} C Y\)
SU BROUTINE HEMV_64 (UPLO, \(\mathbb{N}], A L P H A, A,[L D A], X,[\mathbb{N} C X], B E T A, Y\), [ \(\mathbb{N} C Y\) ])

CHARACTER (LEN=1)::UPLO

COM PLEX (8) :: ALPHA,BETA
COM PLEX (8), D \(\mathbb{I M} E N S \mathbb{O N}(:):: X, Y\)
COM PLEX (8), D \(\mathbb{I M} E N S I O N(:,:):: A\)
\(\mathbb{N}\) TEGER (8) :: \(N\), LDA \(, \mathbb{N} C X, \mathbb{N} C Y\)

\section*{C INTERFACE}
\#include <sunperfh>
void zhem v (charuple, int n, doublecom plex *alpha, doublecom plex *a, int lda, doublecom plex *x, intincx, doublecom plex *beta, doublecom plex *y, intincy);
void zhem v_64 (charuple, long n, doublecom plex *alpha, doublecom plex *a, long lda, doublecom plex *x, long incx, doublecom plex *beta, doublecom plex *y, long incy);

\section*{PURPOSE}
zhem v perform sthe \(m\) atrix-vector operation \(y:=\) alpha* \(A * x+\) beta*y w here alpha and beta are scalars, \(x\) and \(y\) are \(n\) ele\(m\) ent vectors and \(A\) is an \(n\) by \(n\) herm itian \(m\) atrix.

\section*{ARGUMENTS}

UPLO (input)
O \(n\) entry, UPLO specifies whether the upper or low er triangular part of the array \(A\) is to be referenced as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or L ' Only the upper triangularpart of \(A\) is to be referenced.

UPLO = L 'or I' O nly the low ertriangularpart of A is to be referenced.

U nchanged on exit.

N (input)
O n entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)

Before entry w ith UPLO = U 'or L ', the leading \(n\) by \(n\) upper triangularpart of the array A \(m\) ust contain the upper triangular part of the herm itian \(m\) atrix and the strictly low er triangularpartofA is not referenced. Before entry \(w\) ith UPLO \(=\mathrm{L}^{\prime}\) or \({ }^{\prime}\) ', the leading \(n\) by \(n\) low er triangular part of the array A m ustcontain the low er triangular part of the herm itian \(m\) atrix and the strictly upper triangularpart of \(A\) is not referenced. N ote that the im aginary parts of the diagonalele\(m\) ents need notbe set and are assum ed to be zero. U nchanged on exit.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A \(>=\) \(m a x(1, n)\). U nchanged on exit.
\(X\) (input)
\((1+(n-1) * a b s(\mathbb{N} C X))\). Before entry, the increm ented array \(X\) must contain the \(n\) elem ent vectorx. U nchanged on exit.
\(\mathbb{N} C X\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X<>0\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then \(Y\) need notbe set on input. U nchanged on exit.

Y (input/output)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) m ust contain the \(n\) elem ent vectory. On exit, \(Y\) is overw rilten by the updated vectory.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents ofY. \(\mathbb{N C Y}\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zher-perform the herm itian rank 1 operation \(A:=\) alpha*x*conjg ( \(x^{\prime}\) ) + A

\section*{SYNOPSIS}
```

SUBROUTINE ZHER (UPLO,N,ALPHA,X,\mathbb{NCX,A,LDA)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX X (*),A (LDA,*)
INTEGERN,\mathbb{NCX,LDA}
DOUBLE PRECISION ALPHA
SUBROUT\mathbb{NE ZHER_64(UPLO,N,ALPHA,X,INCX,A,LDA)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX X (*),A (LDA,*)
\mathbb{NTEGER*8N,}\mathbb{N}CX,LDA
DOUBLE PRECISION ALPHA

```

\section*{F95 INTERFACE}

```

CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D IM ENSION (:) ::X
COM PLEX (8),D IM ENSION (:,:) ::A
\mathbb{NTEGER ::N,}\mathbb{NCX,LDA}
REAL (8) ::A LPHA
SU BROUT\mathbb{NE HER_64 (UPLO, N ],ALPHA ,X ,[NCX ],A,[LDA ])}
CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D IM ENSION (:) ::X

```

\section*{C INTERFACE}
\#include <sunperfh>
void zher(char uplo, intn, double alpha, doublecom plex *x, intincx, doublecom plex *a, int lda);
void zher_64 (charuplo, long n, double alpha, doublecom plex
*x, long incx, doublecom plex *a, long lda);

\section*{PURPOSE}
zher perform s the herm titian rank 1 operation \(A:=\) alpha* \(x^{\star}\) con \(\dot{j}\left(x^{\prime}\right)+A\) where alpha is a realscalar, \(x\) is an \(n\) elem ent vector and \(A\) is an \(n\) by \(n\) herm titian \(m\) atrix.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the array \(A\) is to be referenced as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or L ' Only the upper triangularpant of \(A\) is to be referenced.
\(\mathrm{UPLO}=\mathrm{L}\) 'or l' O nly the low er triangularpart of \(A\) is to be referenced.

U nchanged on exit.

N (input)
O n entry, \(N\) specifies the order of the \(m\) atrix \(A\). \(N>=0\). U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C X))\). Before entry, the increm ented array \(X\) must contain the \(n\) elem ent vectorx. U nchanged on exit.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X<>0\). U nchanged on exit.

A (input/output)
Before entry w ith UPLO = U 'or 4 ', the leading \(n\) by \(n\) upper triangular part of the array A m ust contain the upper triangular part of the herm itian \(m\) atrix and the strictly low er triangularpart of A is not referenced. O n exit, the upper triangular part of the array A is overw ritten by the upper triangularpart of the updated \(m\) atrix. Before entry w ith UPLO = L 'or I', the leading \(n\) by \(n\) low er triangularpart of the anray A m ust contain the low er triangularpart of the herm itian \(m\) atrix and the strictly upper triangularpart of A is not referenced. On exit, the low er triangularpart of the array \(A\) is overw rilten by the low er triangular part of the updated \(m\) atrix. N ote that the im aginary parts of the diagonalelem ents need not be set, they are assum ed to be zero, and on exit they are setto zero.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD A >= \(\max (1, n)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zher2 -perform the herm itian rank 2 operation \(A:=\) alpha*x*conjg( \(y^{\prime}\) ) + conjg (alpha ) \({ }^{\star} y^{\star}\) conjg \(\left(x^{\prime}\right)+\) A

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZHER2(UPLO,N,ALPHA,X, INCX,Y, INCY,A,LDA )}
CHARACTER * 1 UPLO
DOUBLE COM PLEX ALPHA
DOUBLE COM PLEX X (*),Y (*),A (LDA,*)
\mathbb{NTEGER N, INCX,INCY,LDA}
SU BROUT\mathbb{NE ZHER2_64 (UPLO,N,ALPHA,X, NNCX,Y,INCY,A,LDA )}
CHARACTER * 1 UPLO
DOUBLE COM PLEX ALPHA
DOUBLE COM PLEX X (*),Y (*),A (LDA,*)
INTEGER*8N,\mathbb{NCX,INCY,LDA}

```

\section*{F95 INTERFACE}

SU BROUTINE HER2 (UPLO, \(\mathbb{N}], A L P H A, X,[\mathbb{N C X}], Y,[\mathbb{N C Y}], A,[L D A])\)
CHARACTER (LEN=1) ::UPLO
COM PLEX (8) ::ALPHA
COMPLEX (8),D IM ENSION (:) ::X,Y
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: N, \mathbb{N C X}, \mathbb{I N C Y , L D A}\)
SU BROUTINE HER2_64 (UPLO, \(\mathbb{N}\) ],ALPHA, \(\mathrm{X},[\mathbb{N} C X], Y,[\mathbb{N} C Y], A,[L D A])\)
CHARACTER (LEN=1) ::UPLO
COM PLEX (8) ::ALPHA

\section*{C INTERFACE}
\#include <sunperfh>
void zher2 (charuplo, int n, doublecom plex *alpha, doublecomplex *x, int incx, doublecom plex *y, int incy, doublecom plex *a, int lda);
void zher2_64 (charuplo, long n, doublecom plex *alpha, doublecom plex *x, long incx, doublecom plex *y, long incy, doublecom plex *a, long lda);

\section*{PURPOSE}
zher2 performs the herm tian rank 2 operation \(\mathrm{A}:=\) alpha*x*conjg ( \(\mathrm{y}^{\prime}\) ) + con jg (alpha ) \(\mathrm{y}^{\star}\) con \(\dot{g}\left(\mathrm{x}^{\prime}\right)\) + A where alpha is a scalar, \(x\) and \(y\) are \(n\) elem entvectors and \(A\) is an n by n hem itian m atrix.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the array \(A\) is to be referenced as follow s:

U PLO = U 'or L ' Only the upper triangularpart ofA is to be referenced.

UPLO = L'or I' O nly the low ertriangularpart ofA is to be referenced.

U nchanged on exit.

N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha.
U nchanged on exit.

X (input)
\((1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C X))\). Before entry, the
increm ented array \(X \mathrm{~m}\) ust contain the n elem ent vectorx. Unchanged on exit.
\(\mathbb{N} C X\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X\) <> 0. U nchanged on exit.
\(Y\) (input)
\((1+(n-1) * a b s(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y \mathrm{~m}\) ust contain the n elem ent vectory. U nchanged on exit.
\(\mathbb{N C Y}\) (input)
O n entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y\). \(\mathbb{N C Y}\) <> 0. U nchanged on exit.

A (input/output)
Before entry w ith UPLO = U 'or L ', the leading \(n\) by \(n\) upper triangularpart of the array A \(m\) ust contain the upper triangular partof the herm itian \(m\) atrix and the strictly low er triangularpartofA is not referenced. On exit, the upper triangular part of the array A is overw rilten by the upper triangularpart of the updated \(m\) atrix. Before entry with UPLO = L'or I', the leading \(n\) by \(n\) low er triangularpart of the amay A m ust contain the low er triangularpart of the herm itian \(m\) atrix and the strictly uppertriangularpartofA is not referenced. On exit, the low er triangularpart of the array \(A\) is overw ritten by the low er triangular part of the updated \(m\) atrix. N ote that the im aginary parts of the diagonalelem ents need not be set, they are assum ed to be zero, and on exit they are set to zero.

LDA (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A \(>=\) \(\max (1, n)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zher2k -perform one of the Herm titian rank 2 k operations C \(:=\) alpha*A*conjg ( \(\mathrm{B}^{\prime}\) ) + con \(\dot{g}(\) alpha \(){ }^{*} \mathrm{~B} * \operatorname{con} \dot{g}\left(\mathrm{~A}^{\prime}\right)+\) beta*C orC \(:=\) alpha*conjg ( \(A^{\prime}\) ) \({ }^{*}\) B + cong ( alpha \(){ }^{\star}\) conjg ( \(\left.B^{\prime}\right)^{\star A}+\) beta*C

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHER2K (UPLO,TRANSA,N,K,ALPHA,A,LDA,B,LDB,BETA,C,}
LD C )
CHARACTER * 1 UPLO,TRANSA
DOUBLE COM PLEX ALPHA
DOUBLE COM PLEX A (LDA,*), B (LDB,*),C (LDC ,*)
IN TEGER N,K,LDA,LDB,LDC
DOUBLE PRECISION BETA
SUBROUTINE ZHER2K_64 (UPLO,TRANSA,N,K,ALPHA,A,LDA,B,LDB,BETA,
C,LDC)
CHARACTER * 1 UPLO,TRANSA
DOUBLE COM PLEX ALPHA
DOUBLE COM PLEX A (LDA,*),B (LDB,*),C (LDC,*)
INTEGER*8N,K,LDA,LD B ,LD C
DOUBLE PRECISION BETA

```

\section*{F95 INTERFACE}
```

SU BROUTINE HER2K (UPLO, [TRANSA], $\mathbb{N}],[K], A L P H A, A,[L D A], B,[L D B]$, BETA, C, [LDC])
CHARACTER (LEN=1) ::UPLO,TRANSA
COM PLEX (8) ::ALPHA
COM PLEX (8),D $\mathbb{M}$ ENSION (: : : ) :: A, B, C

```
\(\mathbb{N}\) TEGER : : N , K , LDA \(, \operatorname{LDB}, L D C\)
REAL (8) :: BETA

SU BROUTINE HER2K_64 (UPLO, [TRANSA ], \(\mathbb{N}],[K], A L P H A, A,[L D A], B\), [LDB],BETA, C , [LDC])

CHARACTER (LEN=1) :: UPLO, TRANSA
COM PLEX (8) ::ALPHA
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : A , B, C
\(\mathbb{N} T E G E R(8):: N, K, L D A, L D B, L D C\)
REAL (8) :: BETA

\section*{C INTERFACE}
\#include <sunperfh>
void zher2k (charuplo, chartransa, intn, intk, doublecom plex *alpha, doublecom plex *a, int lda, doublecom plex *b, int ldb, double beta, doublecom plex \({ }^{*}\) c, int Idc);
void zher2k_64 (charuplo, chartransa, long n, long k, doublecom plex *alpha, doublecomplex *a, long lda, doublecom plex *b, long ldb, double beta, doublecom plex *c, long ldc);

\section*{PURPOSE}
zher2k perform s one of the H erm titian rank 2 k operations \(\mathrm{C}:=\) alpha*A *con \(\dot{g}\left(\mathrm{~B}^{\prime}\right)+\) con \(\dot{g}(a l p h a) * B * c o n \dot{g}\left(A^{\prime}\right)+\) beta*C orC \(:=\) alpha*con \(\dot{g}\left(A^{\prime}\right) \star B+\) con \(\dot{g}(a l p h a) \star c o n \dot{g}\left(B^{\prime}\right) \star A+\) beta*C w here alpha and beta are scalarsw ith beta real, \(C\) is an \(n\) by \(n H\) erm itian \(m\) atrix and \(A\) and \(B\) are \(n\) by \(k\) \(m\) atrices in the first case and \(k\) by \(n m\) atrioes in the second case.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangular part of the array \(C\) is to be referenced as follow s:

UPLO = U'or L' Only the upper triangular partof \(C\) is to be referenced.

UPLO = L'or I' Only the lower triangular partof \(C\) is to be referenced.

U nchanged on exit.

TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) 'or \(h^{\prime} \quad C:=a l p h a * A * \infty n g\left(B^{\prime}\right)\)
+ cong (alpha \(){ }^{\star} \mathrm{B}^{*}\) con \(\dot{j}\left(\mathrm{~A}^{\prime}\right)+\) beta* C .
TRANSA = C'ort' \(C=a l p h a * \operatorname{conjg}\left(A^{\prime}\right) \star B\)
\(+\operatorname{con} \dot{g}(\) alpha \(){ }^{\star}\) oon \(\dot{g}\left(B^{\prime}\right) \star A+\) beta \({ }^{\star} C\).

U nchanged on exit.
TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE. N (input)

O n entry, \(N\) specifies the order of the \(m\) atrix \(C\). N m ustbe at least zero. U nchanged on exit.

K (input)
On entry w ith TRANSA \(=N\) 'or \(h\) ', \(K\) specifies the num ber of colum ns of the \(m\) atrices \(A\) and \(B\), and on entry \(w\) th TRANSA \(=C^{\prime}\) or \(\mathrm{t}^{\prime}\), K specifies the num ber of row sof the \(m\) atrices \(A\) and B. K m ustbe at least zero. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha.
U nchanged on exit.
A (input)
COM PLEX *16 anay ofD \(\mathbb{I M} E N S I O N\) (LDA, ka ), where ka isk when TRANSA \(=\mathrm{N}\) 'or h ', and is
n otherw ise. Before entry with TRANSA \(=\mathrm{N}\) ' or
h ', the leading n by k partof the array \(A\)
\(m\) ustcontain the \(m\) atrix \(A\), otherw ise the leading
k by n partof the array A mustcontain the \(m\) atrix A. U nchanged on exit.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program.
W hen TRANSA \(=N\) 'or \(h\) 'then LDA must be at least \(\max (1, n)\), otherw ise LDA m ust.be at least \(\max (1, k)\). U nchanged on exit.

B (input)
COM PLEX *16 aray ofD \(\mathbb{I M}\) ENSION (LD B, kb ), where kb isk when TRANSA \(=\mathrm{N}\) 'or h ', and is
n otherw ise. Before entry w ith TRANSA \(=\mathrm{N}^{\prime}\) or \(h\) ', the leading \(n\) by k part of the array \(B\) \(m\) ust contain the \(m\) atrix \(B\), otherw ise the leading k by n part of the aray \(B \mathrm{~m}\) ustcontain the \(m\) atrix \(B\). U nchanged on exit.

LD B (input)
O n entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program. W hen TRANSA \(=N\) 'or \(h\) 'then LDB must be at least \(\max (1, n)\), otherw ise LD B m ustbe at least \(\max (1, k)\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. U nchanged on exit.

C (input/output)
COMPLEX*16 amay ofD \(\mathbb{I M} E N S I O N(L D C, n)\).

Before entry w th UPLO = U 'or L', the leading \(n\) by \(n\) upper triangular part of the array \(C\) \(m\) ustcontain the upper triangular part of the \(H\) erm itian \(m\) atrix and the strictly low ertriangularpart of C is not referenced. On exit, the upper triangularpart of the array \(C\) is overw ritten by the upper triangularpart of the updated \(m\) atrix.

Before entry w th UPLO = L'or I', the leading \(n\) by \(n\) low er triangular part of the amray \(C\) m ustcontain the low er triangular part of the \(H\) erm itian \(m\) atrix and the strictly upper triangularpartof C is not referenced. On exit, the low er triangularpart of the array \(C\) is overw ritten by the low er triangularpart of the updated \(m\) atrix.

N ote that the im aginary parts of the diagonalele\(m\) ents need not be set, they are assum ed to be zero, and on exit they are set to zero.

LD C (input)
O n entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program.
LD C m ust be at leastm ax ( \(1, \mathrm{n}\) ). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zherfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is \(H\) erm tian indefinite, and provides errorbounds and backw ard enror estim ates for the solution

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHERFS (UPLO,N,NRHS,A,LDA,AF,LDAF,\mathbb{P}\mathbb{IVOT,B,LDB,X,}}\mathbf{N},\textrm{L},\textrm{L}
LD X,FERR,BERR,WORK,W ORK2, INFO)
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
IN TEGER N,NRHS,LDA,LDAF,LDB,LDX , IN FO
\mathbb{NTEGER \mathbb{PIVOT (*)}}\mathbf{(*)}
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)
SUBROUT\mathbb{NE ZHERFS_64 (UPLO,N,NRHS,A,LDA,AF,LDAF, IPIVOT,B,LDB,}
X,LDX,FERR,BERR,W ORK,WORK2, INFO)
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,\star),
W ORK (*)
\mathbb{N TEGER*8 N,NRHS,LDA,LDAF,LDB,LDX , IN FO}
\mathbb{NTEGER*8 \mathbb{P IVOT (*)}}\mathbf{(*)}
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE HERFS (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F], \mathbb{P} \mathbb{I V O T}, B\), [LD B], X, [LDX ],FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1)::UPLO

COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{I M}\) ENSION (:,:) ::A,AF,B,X
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{V} O T\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK2
SUBROUTINE HERFS_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F], \mathbb{P} \mathbb{I} O T\), B, [LDB], X, [LDX],FERR,BERR, [W ORK], [W ORK2], [NFO])

CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{I M}\) ENSION (:,:) ::A,AF,B,X
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void zherfs (char uplo, intn, int nrhs, doublecom plex *a, int lda, doublecom plex *af, int ldaf, int *ipivot, doublecom plex *b, int ldb, doublecom plex *x, int ldx, double * ferr, double *berr, int *info);
void zherfs_64 (charuplo, long n, long nrhs, doublecom plex
*a, long lda, doublecom plex *af, long ldaf, long
*ịívot, doublecom plex *b, long ldb, doublecom plex
*x, long ldx, double *ferr, double *berr, long *info);

\section*{PURPOSE}
zherfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is Herm itian indefintite, and provides errorbounds and backw ard error estim ates forthe solution.

\section*{ARGUMENTS}

UPLO (input)
= U ': U ppertriangle ofA is stored;
= LL': Low ertriangle ofA is stored.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber
of colum ns of the \(m\) atrices \(B\) and \(X . N R H S>=0\).

A (input) The H erm itian m atrix A . If \(\mathrm{UPLO}=\mathrm{U}\) ', the leading \(N-b y-N\) uppertriangularpartofA contains the upper triangularpart of the \(m\) atrix \(A\), and the strictly low er triangular partofA is not referenced. IfU PLO = L', the leading N -by-N lower triangularpartofA contains the low er triangular part of the \(m\) atrix \(A\), and the strictly upper triangularpartofA is not referenced.

LD A (input)
The leading dim ension of the array A. LDA >= max (1,N).
AF (input)
The factored form of them atrix A. AF contains the block diagonal matrix D and themultipliers used to obtain the factor \(U\) orl from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\) as com puted by CHETRF.

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, N)\).

PIVOT (input)
D etails of the interchanges and the block structure ofD as determ ined by CHETRF.
\(B\) (input) The righthand side m atrix \(B\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by CHETRS. On exit, the im proved solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the amay X . LD \(\mathrm{X}>=\) \(\max (1, N)\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X(\mathcal{)}\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{O}), \operatorname{FERR}(\mathcal{)})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(\mathrm{X}(\mathcal{j})\)-XTRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{)}\). The estim ate is as
reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

BERR (output)
The com ponentw ise relative backw ard error of each solution vectorX ( \(j\) ) (ie., the sm allest relative change in any elem entof \(A\) or \(B\) thatm akes \(X(\mathcal{J})\) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

zherk -perform one of the H erm itian rank k operations C
:= alpha*A*conjg(A') + beta*C orC := alpha*conjg(A')*A

+ beta*C

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHERK (UPLO,TRANSA,N,K,ALPHA,A,LDA,BETA,C,LDC)}
CHARACTER * 1 UPLO,TRANSA
DOUBLE COM PLEX A (LDA,*),C (LDC ,*)
INTEGER N,K,LDA,LDC
DOUBLE PRECISION ALPHA,BETA
SUBROUTINE ZHERK_64(UPLO,TRANSA,N,K,ALPHA,A,LDA,BETA,C,LDC)
CHARACTER * 1 UPLO,TRANSA
DOUBLE COM PLEX A (LDA,*),C (LDC,*)
INTEGER*8N,K,LDA,LDC
DOUBLE PRECISION ALPHA,BETA

```
F95 INTERFACE
    SU BROUTINE HERK (UPLO, [TRANSA], \(\mathbb{N}],[K], A L P H A, A,[L D A], B E T A, C\),
        [LD C ])
    CHARACTER (LEN=1) :: UPLO,TRANSA
    COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : ) :: A, C
    \(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathrm{K}, \mathrm{LDA}, \mathrm{LD} \mathrm{C}\)
    REAL (8) ::A LPHA,BETA
    SU BROUTINE HERK_64 (UPLO, [TRANSA ], \(\mathbb{N}],[K], A L P H A, A,[L D A], B E T A\),
        C, [LD C ])

CHARACTER (LEN=1) ::UPLO,TRANSA
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER (8) :: N , K, LDA ,LDC
REAL (8) ::A LPHA,BETA

\section*{C INTERFACE}
\#include <sunperfh>
void zherk (charuplo, chartransa, int n, int \(k\), double alpha, doublecom plex *a, int lda, double beta, doublecom plex * \({ }^{\text {c }}\), int ldc);
void zherk_64 (charuplo, chartransa, long n, long k, double alpha, doublecom plex *a, long lda, double beta, doublecom plex *c, long ldc);

\section*{PURPOSE}
zherk penform s one of the \(H\) erm itian rank \(k\) operations \(C:=\) alpha*A *conjg ( \(A^{\prime}\) ) + beta*C orC := alpha*cong (A')*A + beta*C where alpha and beta are realscalars, C is an n by \(n\) Herm itian \(m\) atrix and \(A\) is an \(n\) by \(k\) matrix in the first case and \(\mathrm{a} k\) by n atrix in the second case.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the upper or lower triangular part of the array \(C\) is to be referenced as follow s:

UPLO = U'or L' Only the upper triangular partof \(C\) is to be referenced.

UPLO = L'or I' Only the lower triangular partof \(C\) is to be referenced.

U nchanged on exit.

TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=\) N 'or h' \(\mathrm{C}:=\) alpha*A *oonjg (A') + beta*C .

TRANSA = C'or \(\boldsymbol{E}^{\prime} \mathrm{C}:=\) alpha*cong (A')*A +
beta*C .

U nchanged on exit.
TRANSA is defaulted to N 'forF95 \(\mathbb{I N}\) TERFACE.

N (input)
On entry, \(N\) specifies the order of the \(m\) atrix \(C\).
N m ustbe at least zero. U nchanged on exit.
K (input)
On entry with TRANSA = N 'or h', K specifies the number of columns of the matrix \(A\), and on entry \(w\) ith TRANSA \(=C^{\prime}\) or \(\mathrm{C}^{\prime}\), \(K\) specifies the num ber of row sof the \(m\) atrix A. K m ustbe at least zero. U nchanged on exit.
ALPHA (input)
On entry, ALPHA specifies the scalar alpha.
U nchanged on exit.

A (input)
COM PLEX *16 aray ofD \(\mathbb{I M}\) ENSION (LDA, ka ), where ka isk when TRANSA = N 'or h', and is
n otherw ise. Before entry w th TRANSA \(=\mathrm{N}\) ' or h ', the leading n by k part of the array \(A\) \(m\) ust contain the \(m\) atrix \(A\), otherw ise the leading k by n partof the array A mustcontain the \(m\) atrix A. U nchanged on exit.

LDA (input)
O \(n\) entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen TRANSA \(=\mathrm{N}\) 'or h 'then LDA must be at least \(\max (1, n)\), otherw ise LDA m ust.be at least \(\max (1, k)\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. U nchanged on exit.

C (input/output)
COM PLEX *16 anay ofD \(\mathbb{I M}\) ENSION (LD C , n ) .
Before entry w ith UPLO = U 'or L ', the leading \(n\) by \(n\) upper triangularpart of the array \(C\) \(m\) ust contain the upper triangular part of the H erm itian \(m\) atrix and the strictly low ertriangularpartofC is not referenced. On exit, the upper triangularpart of the array \(C\) is overw ritten by the upper triangularpart of the updated
m atrix.

Before entry w ith UPLO = L 'or I', the leading \(n\) by \(n\) low er triangular part of the array \(C\) \(m\) ust contain the low er triangular part of the H erm itian \(m\) atrix and the strictly upper triangularpartof \(C\) is not referenced. On exit, the low er triangularpart of the amay \(C\) is overw ritten by the low er triangularpart of the updated \(m\) atrix.

N ote that the im aginary parts of the diagonalele\(m\) ents need not be set, they are assum ed to be zero, and on exit they are set to zero.
LD C (input)
O n entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program. LD C must be at leastm ax ( \(1, \mathrm{n}\) ). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhesv -com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHESV (UPLO,N,NRHS,A,LDA, \mathbb{PIVOT,B,LDB,W ORK,LDW ORK,}}\mathbf{N},\textrm{N},\textrm{N}
\mathbb{NFO)}
CHARACTER * 1 UPLO
D OUBLE COM PLEX A (LDA,*),B (LDB,*),W ORK (*)
INTEGERN,NRHS,LDA,LDB,LDW ORK,INFO
INTEGER \mathbb{PIVOT (*)}
SU BROUT\mathbb{NE ZHESV_64 (UPLO,N,NRHS,A,LDA, \mathbb{PIVOT,B,LDB,W ORK,}}\mathbf{N},
LDW ORK,\mathbb{NFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*),W ORK (*)
INTEGER*8N,NRHS,LDA,LDB,LDW ORK,INFO
INTEGER*8 \mathbb{PIVOT (*)}

```

\section*{F95 INTERFACE}

SU BROUTINE HESV (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{V} O T, B,[L D B],[W O R K]\), [LDW ORK], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A,B
\(\mathbb{N}\) TEGER :: N,NRHS,LDA,LDB,LDW ORK, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{V} O T\)
SU BROUTINE HESV_64 (UPLO, \(\mathbb{N}], \mathbb{N R H S}], A,[L D A], \mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LDB}]\),
[W ORK], [LDW ORK], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1) :: UPLO
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: \(:\) :) : : A , B
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDA,LDB,LDW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{I M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zhesv (char uplo, intn, intnrhs, doublecom plex *a, int lda, int *ipivot, doublecom plex *b, int ldb, int *info);
void zhesv_64 (charuplo, long n, long nrhs, doublecom plex *a, long lda, long *ípivot, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
zhesv com putes the solution to a com plex system of linear equations
\(A * X=B, w h e r e A\) is an \(N\) boy- \(N\) H erm tian \(m\) atrix and \(X\) and \(B\) are N -by-N RH S m atrices.

The diagonalpivoting \(m\) ethod is used to factorA as
\[
\begin{aligned}
& A=U * D * U * * H, \text { if } U P L O=U ' \text {, or } \\
& A=L * D * L * * H, \text { if } U P L O=L^{\prime}
\end{aligned}
\]
where \(U\) (orL) is a productofperm utation and unit upper (low er) triangular matrioes, and D is H erm itian and block diagonalw ith 1 boy-1 and 2 -by-2 diagonalblocks. The factored form of \(A\) is then used to solve the system of equations \(\mathrm{A} * \mathrm{X}=\mathrm{B}\).

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) : U pper triangle ofA is stored;
\(=L^{\prime}\) : Low er triangle ofA is stored.

N (input) The num ber of linearequations, ie., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of the m atrix B. NRH S \(>=0\).

A (input/output)
On entry, the \(H\) erm itian \(m\) atrix A. If UPLO = U', the leading \(\mathrm{N}-\) by -N uppertriangularpartofA contains the upper triangular part of the \(m\) atrix \(A\), and the strictly low er triangularpart ofA is not referenced. If UPLO = L', the leading N -by-N low er triangularpartof \(A\) contains the low ertriangularpartof them atrix A, and the strictly upper triangular part of A is not referenced.

On exit, if \(\mathbb{N F O}=0\), the block diagonalm atrix \(D\) and the \(m\) ultipliers used to obtain the factor \(U\) or L from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\) or \(\mathrm{A}=\) L*D *L**H as com puted by CHETRF.
LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

IPIVOT (output)
D etails of the interchanges and the block structure ofD, as determ ined by CHETRF. If \(\mathbb{P}\) IVOT (k) \(>0\), then row sand colum nsk and \(\mathbb{P} \mathbb{V} O T(k)\) were interchanged, and \(\mathrm{D}(\mathrm{k}, \mathrm{k})\) is a 1 -by- 1 diagonal block. If UPLO \(=\mathrm{U}\) 'and \(\mathbb{P} \mathbb{V} O T(k)=\mathbb{P} \mathbb{V} O T(k-1)\) \(<0\), then row s and colum nsk-1 and - \(\mathbb{P}\) IV OT (k) were interchanged and \(D(k-1 k, k-1 k)\) is a 2 -by-2 diagonal block. If UPLO \(=\mathbb{L}\) ' and \(\mathbb{P} \mathbb{I V O T}(k)=\) IP IV O \((k+1)<0\), then row s and colum ns \(k+1\) and \(-\mathbb{P}\) IV O T ( \(k\) ) w ere interchanged and \(D(k *+1, k k+1)\) is a 2 -by-2 diagonalblock.

B (input/output)
On entry, the N -by-NRH \(S\) righthand side \(m\) atrix \(B\). On exi, if \(\mathbb{N} F O=0\), the \(N\) by \(N\) RH solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the anay B . LD B >= \(\max (1, N)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, \mathrm{~W} O R K(1)\) retums the optim al LDW ORK.

LD W ORK (input)
The length of ORK . LDW ORK >= 1, and for best perform ance LDW ORK \(>=N * N B\), where \(N B\) is the optim alblocksize forCHETRF.

If LD W ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{I N} F O\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N} F O=\mathrm{i}, \mathrm{D}(i, i)\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, so the solution could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhesvx - use the diagonalpivoting factorization to com pute the solution to a com plex system of linearequations A * \(\mathrm{X}=\) B,

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHESVX (FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,B,}}\mathbf{N},
LDB,X,LDX,RCOND,FERR,BERR,W ORK,LDW ORK,W ORK2,INFO)
CHARACTER * 1FACT,UPLO
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
INTEGER N,NRHS,LDA,LDAF,LDB,LDX,LDW ORK,\mathbb{NFO}
\mathbb{NTEGER \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION RCOND
D OUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)
SUBROUT\mathbb{NE ZHESVX_64(FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,}}\mathbf{N},\mp@code{N},
B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,LDW ORK,W ORK 2, \mathbb{NFO)}
CHARACTER * 1FACT,UPLO
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
INTEGER*8N,NRHS,LDA,LDAF,LDB,LDX,LDW ORK,INFO
INTEGER*8 \mathbb{P IVOT (*)}
DOUBLE PRECISION RCOND
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE HESVX \(\mathbb{F} A C T, U P L O, ~ \mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F]\), \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LD} \mathrm{B}], \mathrm{X},[\operatorname{LD} \mathrm{X}], R C O N D, F E R R, B E R R,[W\) ORK], [LDW ORK], [W ORK2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::FACT, UPLO
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{I M} E N S I O N(:,:):: A, A F, B, X\)
\(\mathbb{N}\) TEGER ::N,NRHS,LDA,LDAF,LDB,LDX,LDWORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL (8) :: RCOND
REAL (8), D \(\mathbb{I}\) ENSION (:) :: FERR,BERR,W ORK2

SUBROUTINE HESVX_64 (FACT, UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F]\), \(\mathbb{P} I V O T, B,[L D B], X,[L D X], R C O N D, F E R R, B E R R,[W O R K],[L D W O R K]\), [ W ORK2], \([\mathbb{N} \mathrm{FO}]\) )

CHARACTER (LEN=1) ::FACT, UPLO
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: \(:\) :) : : A, AF, B, X
\(\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D A F, L D B, L D X, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT}\)
REAL (8) :: RCOND
REAL (8), D \(\mathbb{I}\) ENSION (:) :: FERR, BERR,W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void zhesvx (char fact, charuplo, int n, int nrhs, doublecom plex *a, int lda, doublecom plex *af, int ldaf, int *ipivot, doublecom plex *b, int ldb, doublecom plex *x, int ldx, double *rcond, double *ferr, double *berr, int *info);
void zhesvx_64 (char fact, char uplo, long n, long nrhs, doublecom plex *a, long lda, doublecom plex *af, long ldaf, long *ípivot, doublecom plex *b, long ldb, doublecom plex *x, long ldx, double *roond, double * ferr, double *berr, long *info);

\section*{PURPOSE}
zhesvx uses the diagonalpivoting factorization to com pute the solution to a com plex system of linear equations \(\mathrm{A} * \mathrm{X}=\) \(B\), where \(A\) is an \(N\) by \(-N\) H erm itian \(m\) atrix and \(X\) and \(B\) are \(N-\) by-N R H S m atrices.

E rrorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT \(=N\) ', the diagonalpivoting \(m\) ethod is used to
factorA.
The form of the factorization is
\[
\begin{aligned}
& A=U * D * U * * H, \text { if } U P L O=U ' \text {, or } \\
& A=L * D * L * * H, \text { if } U P L O=L \prime
\end{aligned}
\]
where \(U\) (orL) is a product of perm utation and unit upper (low er)
triangularm atrices, and \(D\) is Herm titian and block diagonalw ith
1-by-1 and 2-by-2 diagonalblocks.
2. If som eD \((i, i)=0\), so thatD is exactly singular, then the routine
retums w ith \(\mathbb{N} F O=\) i. O therw ise, the factored form of \(A\) is used
to estim ate the condition num ber of the \(m\) atrix \(A\). If the
reciprocal of the condition num ber is less than \(m\) achine precision,
\(\mathbb{N} F O=N+1\) is retumed as a waming, but the routine stillgoes on
to solve for \(X\) and com pute enror bounds as described below .
3.The system ofequations is solved for \(X\) using the factored form of A.
4. Herative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate emror bounds and backw ard enror estim ates
for it.

\section*{ARGUMENTS}

FACT (input)
Specifies w hether ornot the factored form of A has been supplied on entry. = F ': On entry, A F and \(\mathbb{P}\) IV OT contain the factored form of A. A, AF and \(\mathbb{P} \mathbb{V O T}\) will not be modified. = N : The \(m\) atrix A w illlbe copied to A F and factored.

UPLO (input)
\(=\mathrm{U}:\) : U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The num ber of linearequations, ie., the order of them atrix A. \(N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atriges B and X . NRHS \(>=0\).

A (input) The \(H\) erm itian \(m\) atrix \(A\). IfUPLO \(=U\) ', the leading N -by- N uppertriangularpartofA contains the uppertriangularpart of the \(m\) atrix \(A\), and the strictly low ertriangularpart of A is notreferenced. IfUPLO = L', the leading N -by-N lower triangularpart ofA contains the low er triangular part of the m atrix A, and the strictly upper triangularpart ofA is notreferenced.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).
AF (input/output)
IfFACT \(=F^{\prime}\), then \(A F\) is an inputargum entand on entry contains the block diagonalm atrix D and the m ultipliers used to obtain the factor U orL from the factorization \(A=U * D * U * * H\) orA \(=L * D * L * * H\) as com puted by CHETRF .

IfFACT \(=N\) ', then \(A F\) is an output argum ent and on exit retums the block diagonalm atrix D and the m ultipliers used to obtain the factorU or L from the factorization \(A=U * D * U * * H\) or \(A=\) L*D*L**H.

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, N)\).

IPIVOT (inputoroutput)
IfFACT = \(\mathrm{F}^{\prime}\), then \(\mathbb{P} \mathbb{I V O T}\) is an input argum ent and on entry contains details of the interchanges and the block structure of D , as determ ined by CHETRF. If \(\mathbb{P} \mathbb{I V O T}(k)>0\), then row \(s\) and colum nsk and \(\mathbb{P} \mathbb{I V} O T(k)\) w ere interchanged and \(D(k, k)\) is a 1 boy-1 diagonal block. If \(\mathrm{UPLO}=\mathrm{U}\) ' and \(\mathbb{P} \operatorname{IV} \circ T(\mathrm{k})=\mathbb{P} \mathbb{I} \circ \mathrm{T}(\mathrm{k}-1)<0\), then row s and colum ns \(\mathrm{k}-1\) and \(-\mathbb{P} \mathbb{I V O T}(\mathrm{k})\) were interchanged and \(\mathrm{D}(\mathrm{k}-\) \(1 \mathrm{k}, \mathrm{k}-1 \mathrm{k}\) ) is a 2-by-2 diagonalblock. If \(\mathrm{PLO}=\) \(\mathbb{L}\) 'and \(\mathbb{P} \mathbb{I V O T}(k)=\mathbb{P} \mathbb{V} O T(k+1)<0\), then row sand colum nsk+1 and -TP IVOT (k) w ere interchanged and D \((k: k+1, k: k+1)\) is a 2 -by-2 diagonalblock.

If \(\mathrm{FACT}=\mathrm{N}\) ', then \(\mathbb{P} \mathbb{I V O T}\) is an output argum ent and on exitcontains details of the interchanges and the block structure of D , as determ ined by

CHETRF.

B (input) The N Hoy-N RHS righthand side m atrix B .

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N}\) FO \(=N+1\), the N -by -N RH S solution
m atrix \(X\).
LD X (input)
The leading dim ension of the array X. LD X >= \(\max (1, N)\).

RCOND (output)
The estim ate of the reciprocal condition num ber of the matrix A. IfRCOND is less than them achine precision (in particular, if RCOND \(=0\) ), the \(m\) atrix is singular to working precision. This condition is indicated by a retum code of \(\mathbb{N F O}>\) 0 .

\section*{FERR (output)}

The estim ated forw ard errorbound for each solution vector \(X\) ( \()\) ) (the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(i), F E R R(i)\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) entin ( \(X(\mathcal{I})-X\) TRUE) divided by the \(m\) agninude of the largestelem entin X ( 7 ). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slightoverestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vectorX (i) (ie., the sm allest relative change in any elem entofA orB thatm akes X (i) an exactsolution).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length ofW ORK. LDW ORK \(>=2 * N\), and for best perform ance LDW ORK \(>=N * N B\), where \(N B\) is the optim alblocksize forCHETRF .

IfLDW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{N N F O}=-\) i, the \(i\)-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N F O}=i\), and \(i\) is
\(<=\mathrm{N}: \mathrm{D}(i, i)\) is exactly zero. The factorization
has been completed but the factorD is exactly singular, so the solution and error bounds could not be com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1\) : D is nonsingular, butRCOND is less than machine precision, \(m\) eaning that the \(m\) atrix is singular to w orking precision. Nevertheless, the solution and error bounds are com puted because there are a num berof situationsw here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zhetf2 -com pute the factorization of a com plex Herm itian \(m\) atrix A using the Bunch \(-K\) aufn an diagonalpivoting \(m\) ethod

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
D OUBLE COM PLEX A (LDA,*)
\mathbb{NTEGERN,LDA,}\mathbb{NNFO}
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(*)}
SUBROUT\mathbb{NE ZHETF2_64(UPLO,N,A,LDA, \mathbb{PIV , INFO)}}\mathbf{N}\mathrm{ (N)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*)
INTEGER*8N,LDA,INFO
\mathbb{NTEGER*8 \mathbb{PIV (*)}}\mathbf{*}\mathrm{ ( }

```

\section*{F95 INTERFACE}
```

SU BROUTINE HETF2 (UPLO, $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I},[\mathbb{N} F O])$
CHARACTER (LEN=1)::UPLO
COMPLEX (8), D IM ENSION (:,:) ::A
$\mathbb{N}$ TEGER ::N,LDA, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}$
SU BROUTINE HETF2_64 (UPLO, $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I},[\mathbb{N} F O])$
CHARACTER (LEN=1)::UPLO
COM PLEX (8),D $\mathbb{M}$ ENSION (:,:) ::A

```
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} A, \mathbb{N}\) FO
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zhetf2 (charuple, intn, doublecom plex *a, int lda, int *ị̆̇v, int *info);
void zhetf2_64 (charuplo, long n, doublecom plex *a, long lda, long *ịiv, long *info);

\section*{PURPOSE}
zhetf2 com putes the factorization of a com plex H erm itian \(m\) atrix A using the Bunch-K aufm an diagonalpivoting \(m\) ethod:
\[
A=U * D * U^{\prime} \text { or } A=L * D * L^{\prime}
\]
where \(U\) (orL) is a product of perm utation and unit upper (low er) triangular m atrices, \(U\) 'is the conjugate transpose of \(U\), and \(D\) is \(H\) erm itian and block diagonalw ith 1 boy- 1 and 2-by-2 diagonalblocks.

This is the unblocked version of the algorithm, calling Level2 BLAS.

\section*{ARGUMENTS}

\section*{UPLO (input)}

Specifies w hether the upper or low er triangular part of the \(H\) erm itian \(m\) atrix \(A\) is stored:
\(=\mathrm{U}\) ': Upper triangular
\(=\mathrm{L}\) ': Low ertriangular

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the \(H\) erm itian \(m\) atrix A. If \(\mathrm{U} P L O=\mathrm{U}\) ', the leading \(n-b y-n\) uppertriangularpartofA contains the upper triangular part of the \(m\) atrix \(A\), and the strictly low er triangularpart of A is not referenced. If UPLO \(=\mathbb{L}\) ', the leading \(n-b y-n\) low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of \(A\) is notreferenced.

On exit, the block diagonalm atrix D and the mul
tipliers used to obtain the factorU orl (see below for further details).

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, \mathbb{N})\).

IPIV (output)
D etails of the interchanges and the block structure ofD. If \(\mathbb{P} \mathbb{I V}(k)>0\), then row \(s\) and colum ns \(k\) and \(\mathbb{P} \mathbb{I V}(k)\) were interchanged and \(D(k, k)\) is a 1 -by-1 diagonalblock. IfUPLO \(=U\) 'and \(\mathbb{P} \mathbb{I V}(k)\) \(=\mathbb{P} \mathbb{I V}(k-1)<0\), then row \(s\) and colum ns \(k-1\) and \(-\mathbb{P} \mathbb{I V}(k)\) w ere interchanged and \(D(k-1 *, k-1 k)\) is a 2 -by-2 diagonalblock. IfUPLO \(=\mathbb{L}\) 'and \(\mathbb{P} \mathbb{I V}(k)\)
\(=\mathbb{P} \mathbb{I}(k+1)<0\), then row sand colum ns \(k+1\) and \(-\mathbb{P} \mathbb{V}(k)\) w ere interchanged and \(D(k, k+1, k \mathrm{k}+1)\) is a 2-by-2 diagonalblock.
\(\mathbb{N F O}\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\mathrm{k}\), the k -th argum enthad an illegalvalue \(>0:\) if \(\mathbb{N} F O=k, D(k, k)\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, and division by zero w ill occur if it is used to solve a system of equations.

\section*{FURTHER DETAILS}

1-96-B ased on m odifications by
J. Lew is, B oeing C om puter Servioes C om pany
A. Petitet, C om puterScience D ept, U niv . of Tenn., K noxville, U SA

If \(U P L O=U\) ', then \(A=U * D * U '\), where
\(U=P(n) \star U(n) * \ldots\)... \((k) U(k) * \ldots\),
i.e., \(U\) is a product of term \(s P(k) * U(k)\), where \(k\) decreases from \(n\) to 1 in steps of 1 or 2 , and \(D\) is a block diagonal \(m\) atrix \(w\) th 1 -by -1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I}(k)\), and \(U(k)\) is a unituppertriangularm atrix, such that if the diagonal block D (k) is of orders ( \(s=1\) or 2 ), then
```

    ( I v 0 ) k-s
    U(k)=(0 I 0 ) s
( 0 0 I ) n-k
k-s s n-k

```

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(1 k-\) \(1, k)\). If \(s=2\), the uppertriangle of \((k)\) overw rites \(A(k-\) \(1, k-1), A(k-1, k)\), and \(A(k, k)\), and \(v\) overw rites \(A(1 k-2, k-\) 1 k).

IfUPLO \(=\mathrm{L}\) ', then \(A=\mathrm{L} * \mathrm{D} * \mathrm{~L}\) ', where \(L=P(1) \star L(1)^{\star} \ldots * P(k) \star L(k) \star \ldots\)
ie., \(L\) is a productofterm \(s P(k) * L(k)\), where \(k\) increases from 1 to n in steps of 1 or 2 , and D is a block diagonal \(m\) atrix w ith 1 -by -1 and 2 -by -2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{V}(k)\), and \(L(k)\) is a unitlow ertriangularm atrix, such that if the diagonal
block D (k) is of orders ( \(s=1\) or 2 ), then
\(\left(\begin{array}{llll}I & 0 & 0\end{array}\right) k-1\)
\(L(k)=\left(\begin{array}{lll}0 & I & 0\end{array}\right) s\)
( 0 v I ) n-k-s+1
\(\mathrm{k}-1\) s n-k-s+1

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(k+1 m, k)\). If \(s=2\), the low ertriangle ofD \((k)\) overw rites \(A(k, k), A(k+1, k)\), and \(A(k+1, k+1)\), and \(V\) overw rites A \((k+2 m, k: k+1)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zhetrd - reduce a com plex H erm itian matrix A to real sym \(m\) etric tridiagonal form \(T\) by a unitary sim ilarity transfor\(m\) ation

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHETRD (UPLO,N,A,LDA,D,E,TAU,W ORK,LW ORK, INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
\mathbb{NTEGER N,LDA,LW ORK,INFO}
DOUBLE PRECISION D (*),E (*)
SUBROUT\mathbb{NE ZHETRD_64(UPLO,N,A,LDA,D,E,TAU,W ORK,LW ORK,INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
\mathbb{NTEGER*8N,LDA,LW ORK,INFO}
DOUBLE PRECISION D (*),E (*)

```

\section*{F95 INTERFACE}

SU BROUTINE HETRD (UPLO, \(\mathbb{N}], A,[L D A], D, E, T A U,[W O R K],[L W ~ O R K]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1)::UPLO
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU ,W ORK
COM PLEX (8),D IM ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: N, L D A, L W O R K, \mathbb{N} F O\)
REAL (8),D IM ENSION (:) ::D ,E

SUBROUTINE HETRD_64 (UPLO, \(\mathbb{N}], A,[L D A], D, E, T A U,[W O R K],[L W O R K]\),
[ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) :: UPLO
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::TAU ,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : A
\(\mathbb{N} T E G E R(8):: N, L D A, L W O R K, \mathbb{N F O}\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D , E

\section*{C INTERFACE}
\#include <sunperfh>
void zhetrd (charuplo, intn, doublecom plex *a, int lda, double *d, double *e, doublecom plex *tau, int *info);
void zhetrd_64 (charuple, long n, doublecom plex *a, long lda, double *d, double *e, doublecom plex *tau, long *info);

\section*{PURPOSE}
zhetrd reduces a com plex \(H\) erm itian \(m\) atrix A to real sym \(m\) etric tridiagonal form \(T\) by a unitary sim ilarity transformation: Q ** \(\mathrm{H} * \mathrm{~A} * \mathrm{Q}=\mathrm{T}\).

\section*{ARGUMENTS}

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low er triangle ofA is stored.

N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input) \(O n\) entry, the \(H\) erm itian \(m\) atrix \(A\). If \(U P L O=U '\),
the leading \(\mathrm{N}-\) by -N uppertriangularpartofA con-
tains the upper triangular part of the \(m\) atrix \(A\), and the strictly low ertriangularpartofA is not referenced. If UPLO \(=\mathrm{L}\) ', the leading N -by -N
low er triangularpartofA contains the low ertriangularpart of the matrix A, and the strictly upper triangular part ofA is not referenced. On exit, if UPLO = U ', the diagonal and firstsuperdiagonalofA are overw rilten by the comesponding elem ents of the tridiagonalm atrix T , and the ele\(m\) ents above the first superdiagonal, \(w\) ith the array TAU, represent the unitary \(m\) atrix \(Q\) as a product of elem entary reflectors; if UPLO \(=\mathrm{L}\) ', the diagonaland firstsubdiagonalofA are over-
w ritten by the comesponding elem ents of the tridiagonalm atrix \(T\), and the elem ents below the first subdiagonal, with the array TAU, represent the unitary \(m\) atrix \(Q\) as a product of elem entary reflectors. See FurtherD etails.

LD A (input)
The leading dim ension of the amay A. LDA >= \(\max (1, N)\).

D (output)
The diagonalelem ents of the tridiagonalm atrix T :
D (i) \(=A(i, i)\).
E (output)
The off-diagonal elem ents of the tridiagonal matrix \(T: E(i)=A(i, i+1)\) if \(U P L O=U^{\prime}, E(i)=\) A \((i+1, i)\) if \(U P L O=L\).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails).

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, W\) ORK (1) retums the optim al
LW ORK.
LW ORK (input)
The dim ension of the array \(W\) ORK. LW ORK >=1. For optim um perform ance LW ORK >=N *NB, where NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

If \(\mathrm{PLO}=\mathrm{U}\) ', the \(m\) atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(n-1) \ldots H(2) H(1) .
\]

Each H (i) has the form
\(H(i)=I-\tan * V^{*} V^{\prime}\)
\(w\) here tau is a com plex scalar, and \(v\) is a com plex vector w ith \(\mathrm{v}(\mathrm{i}+1 \mathrm{n})=0\) and \(\mathrm{v}(\mathrm{i})=1\); \(\mathrm{v}(1: i-1)\) is stored on exit in
A ( \(1: i-1, i+1\) ), and tau in TAU (i).

If U PLO \(=\mathbb{L}\) ', them atrix Q is represented as a product of elem entary reflectors
\[
Q=H(1) H(2) \ldots H(n-1)
\]

Each H (i) has the form
\[
H(i)=I-\tan { }^{\star} v^{*} v^{\prime}
\]
where tau is a com plex scalar, and \(v\) is a com plex vector w ith \(\mathrm{v}(1: i)=0\) and \(\mathrm{v}(\mathrm{i}+1)=1 ; \mathrm{v}(\mathrm{i}+2 \mathrm{n})\) is stored on exit in \(A(i+2 m, i)\), and tau in TAU (i).

The contents ofA on exitare illustrated by the follow ing exam ples \(w\) ith \(n=5\) :
```

ifUPLO = U ': ifUPLO = L ':

```
    ( d e v2 v3 v4 ) (d
)
    ( d e v3 v4 ) ( e d
)
    ( d e v4 ) (v1 e d
)
    ( d e ) ( v1 v2 e d
)
    ( d ) (v1 v2 v3 e d
)
where d and e denote diagonal and off-diagonal elem ents of \(T\), and videnotes an elem ent of the vectordefining \(H\) (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zhetrf-com pute the factorization of a com plex Herm tian \(m\) atrix A using the Bunch-K aufn an diagonalpivoting \(m\) ethod

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)
INTEGERN,LDA,LDWORK,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}

```

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)
INTEGER*8N,LDA,LDW ORK, INFO
INTEGER*8 \mathbb{PIVOT (*)}

```

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E\) HETRF (UPLO, \(\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V O T},[\mathbb{W}\) ORK ], [LDW ORK ], [ \(\mathbb{N F O}])\)
CHARACTER (LEN=1)::UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER ::N,LDA,LDW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
SU BROUTINE HETRF_64 (UPLO, \(\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I} O T,[\mathbb{W}\) ORK ], [LDW ORK ], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER (8) ::N,LDA,LDW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V O T}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zhetrf(charuplo, intn, doublecom plex *a, int lda, int
*ịívot, int*info);
void zhetrf_64 (charuplo, long n, doublecom plex *a, long lda, long *ịíivot, long *info);

\section*{PURPOSE}
zhetrf com putes the factorization of a complex Herm itian \(m\) atrix A using the B unch-K aufn an diagonalpivoting \(m\) ethod. The form of the factorization is
\[
\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H} \text { or } \mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}
\]
where \(U\) (orL) is a productof perm utation and unit upper (low er) triangular matrioes, and D is H erm itian and block diagonalw ith 1 -by-1 and 2 -by-2 diagonalblocks.

This is the blocked version of the algorithm, calling Level 3 BLAS.

\section*{ARGUMENTS}

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrix A. N >=0.

A (input/output)
On entry, the H erm itian matrix A . If UPLO = \(\mathrm{U}^{\prime}\), the leading \(\mathrm{N}-\) by -N uppertriangularpartofA contains the upper triangular part of the \(m\) atrix \(A\), and the strictly low er triangularpart ofA is not referenced. If UPLO = L', the leading N -by-N low er triangularpart of \(A\) contains the low ertriangularpartof them atrix A, and the strictly upper triangular part of A is not referenced.

On exit, the block diagonalm atrix D and the multipliers used to obtain the factorU orL (see below for further details).

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

\section*{IPIVOT (output)}

D etails of the interchanges and the block structure of D. If \(\mathbb{P I V O T}(k)>0\), then row sand columnsk and \(\mathbb{P I V O T}(k)\) were interchanged and \(D(k, k)\) is a \(1-b y-1\) diagonalblock. If \(U P L O=U^{\prime}\) and \(\mathbb{P} \mathbb{V} O T(k)=\mathbb{P} \mathbb{V} O T(k-1)<0\), then row \(s\) and colum ns \(k-1\) and - \(\mathbb{P I V O T}(k)\) were interchanged and D ( \(k-1 * k, k-1 k)\) is a \(2-b y-2\) diagonal block. If UPLO \(=\mathrm{L}\) 'and \(\mathbb{P I V O T}(k)=\mathbb{P} \mathbb{I V O T}(k+1)<0\), then row sand colum ns \(k+1\) and \(-\mathbb{P} \mathbb{V} O T(k)\) were interchanged and \(D(k: k+1, k k+1)\) is a \(2-b y-2\) diagonal block.

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, \mathrm{~W}\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length ofW ORK. LDW ORK >=1. Forbestperfor\(m\) ance LDW ORK >=N *NB, where NB is the block size retumed by ㅍAENV .
\(\mathbb{I N}\) FO (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N} F O=\mathrm{i}, \mathrm{D}(\mathrm{i}, \mathrm{i})\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, and division by zero w illoccur if it is used to solve a system ofequations.

\section*{FURTHER DETAILS}

If U PLO \(=\mathrm{U}\) ', then \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}\) ', where
\(U=P(n) \star U(n)^{\star} \ldots * P(k) U(k)^{\star} . . .\),
i.e., \(U\) is a product of term \(s P(k) * U(k)\), where \(k\) decreases
from \(n\) to 1 in steps of 1 or 2 , and \(D\) is a block diagonal \(m\) atrix \(w\) ith 1 -by- 1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I V O T}(k)\), and \(U(k)\) is a unit uppertriangularm atrix, such that if the diagonal
block \(D(k)\) is of orders ( \(s=1\) or 2 ), then
\[
\begin{aligned}
& \text { ( I v 0 ) k-s } \\
& U(k)=\left(\begin{array}{lll}
0 & I
\end{array}\right) s \\
& \text { ( } 0 \text { O I ) } n-k \\
& \mathrm{k}-\mathrm{s} \text { s } \mathrm{n}-\mathrm{k}
\end{aligned}
\]

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(1 k-\) \(1, k)\). If \(s=2\), the uppertriangle of \((k)\) overw rites \(A(k-\) \(1, k-1), A(k-1, k)\), and \(A(k, k)\), and \(v\) overw rites \(A(1 k-2, k-\) 1 k).

If \(U P L O=L\) ', then \(A=L * D * L '\), where
\(L=P(1) \star L(1)^{\star} \ldots * P(k) \star L(k)^{\star} \ldots\),
ie., \(L\) is a product ofterm \(s P(k) * L(k)\), where \(k\) increases from 1 to \(n\) in steps of 1 or 2 , and \(D\) is a block diagonal \(m\) atrix \(w\) ith 1 -by -1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I V} O T(k)\), and \(L(k)\) is a unitlow ertriangularm atrix, such that if the diagonal
block D (k) is of orders ( \(s=1\) or 2 ), then
\[
\begin{aligned}
& \text { ( I 0 0 ) k-1 } \\
& L(k)=\left(\begin{array}{lll}
0 & I & 0
\end{array}\right) s \\
& \text { ( } 0 \text { v I ) n-k-s+1 } \\
& \mathrm{k}-1 \text { s } \mathrm{n}-\mathrm{k}-\mathrm{s}+1
\end{aligned}
\]

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(k+1 m, k)\). If \(s=2\), the low ertriangle ofD \((k)\) overw rites \(A(k, k), A(k+1, k)\), and \(A(k+1, k+1)\), and \(v\) overw rites \(A(k+2 m, k: k+1)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhetri-com pute the inverse of a com plex H erm itian indefinthe \(m\) atrix \(A\) using the factorization \(A=U * D * U * * H\) orA \(=\) L*D*L**H com puted by CHETRF

\section*{SYNOPSIS}


CHARACTER * 1 UPLO
D OUBLE COM PLEX A (LDA, \(\left.{ }^{*}\right)\), W ORK (*)
\(\mathbb{N}\) TEGER \(N, L D A, \mathbb{N} F O\)
\(\mathbb{N T E G E R} \mathbb{P} \mathbb{V} O T\left({ }^{( }\right)\)

SU BROUTINE ZHETRI_64 (UPLO,N,A,LDA, \(\mathbb{P} \mathbb{I V O T}, \mathrm{W} O R K, \mathbb{N} F O)\)
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA, \({ }^{*}\) ), W ORK (*)
\(\mathbb{N} T E G E R * 8 N, L D A, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{P} \mathbb{I V O T}(\star)\)

\section*{F95 INTERFACE}

SUBROUTINE HETRI(UPLO, \(\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{N} O T,[\mathbb{W} O R K],[\mathbb{N F O}])\)

CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER ::N,LDA, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)

SU BROUTINE HETRI_64 (UPLO, \(\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I V O T},[\mathbb{W} O R K],[\mathbb{N} F O])\)
CHARACTER (LEN=1)::UPLO

COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{LD} \mathrm{A}, \mathbb{N} F \mathrm{O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{V} O T\)

\section*{C INTERFACE}
\#include <sunperfh>
void zhetri(charuplo, intn, doublecom plex *a, int lda, int *ipivot, int*info);
void zhetri_ 64 (charuplo, long n, doublecom plex *a, long lda, long *ịíivot, long *info);

\section*{PURPOSE}
zhetricom putes the inverse of a com plex \(H\) erm inian indefinite \(m\) atrix A using the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\) orA \(=\) L*D *L**H com puted by CHETRF .

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) ': Uppertriangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\);
= L ': Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D}\) * \(\mathrm{L} * * \mathrm{H}\).

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the block diagonalm atrix D and the \(m\) ultipliers used to obtain the factorU orL as com puted by CHETRF.

On exit, if \(\mathbb{N} F O=0\), the ( H erm titian) inverse of the original m atrix. If \(\mathrm{UPLO}=\mathrm{U}\) ', the upper triangularpart of the inverse is form ed and the partofA below the diagonal is not referenced; if UPLO = L' the lower triangular part of the inverse is formed and the partofA above the diagonal is not referenced.

LD A (input)
The leading din ension of the array A. LD A >= \(\max (1, N)\).
\(\mathbb{P I V O T}\) (input)
D etails of the interchanges and the block structure ofD as determ ined by CHETRF.

W ORK (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthad an illegalvahue
\(>0:\) if \(\mathbb{N} F O=i, D(i, i)=0\); the \(m\) atrix is singular and its inverse could not.be com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhetrs-solve a system of linearequations \(A * X=B\) w th a complex Herm itian matrix A using the factorization \(A=\) \(\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\) com puted by CHETRF

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGERN,NRHS,LDA,LDB,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}
SUBROUT\mathbb{NE ZHETRS_64(UPLO,N,NRHS,A,LDA,\mathbb{PIVOT,B,LDB,INFO)}}\mathbf{(N,N}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGER*8N,NRHS,LDA,LDB,INFO
INTEGER*8 \mathbb{PIVOT (*)}

```
F95 INTERFACE
    SUBROUTINE HETRS (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I} O T, B,[L D B],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO
    COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : ) ::A, B
    \(\mathbb{N}\) TEGER ::N,NRHS,LDA,LDB, \(\mathbb{N} F O\)
    \(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION(:)::\mathbb {P}\mathbb {O}OT}\)
    SU BROUTINE HETRS_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I V O T}, B,[L D B]\),
        [ \(\mathbb{N} F O\) ])
    CHARACTER (LEN=1)::UPLO

COM PLEX (8), D IM ENSION (: : : : : A , B
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDA,LDB, \(\mathbb{N}\) FO
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zhetrs (charuplo, intn, int nrhs, doublecom plex *a, int lda, int*ipivot, doublecom plex *b, intldb, int*info);
void zhetrs_64 (charuplo, long n, long nrhs, doublecom plex
*a, long lda, long *ípivot, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
zhetrs solves a system of linearequations \(A * X=B\) with a complex Herm itian matrix A using the factorization A = \(U * D * U * * H\) or \(A=L * D * L * * H\) com puted by CHETRF.

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) : : Uppertriangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}\) ** H ;
\(=L^{\prime}:\) Low ertriangular, form is \(A=L * D * L * * H\).

N (input) The order of them atrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRH S \(>=0\).

A (input) The block diagonalm atrix \(D\) and the m ultipliers used to obtain the factorU orL as com puted by CHETRF.

LD A (input)
The leading dim ension of the amay A. LDA >= \(\max (1, N)\).

IPIVOT (input)
D etails of the interchanges and the block structure ofD as determ ined by CHETRF.

B (input/output)
On entry, the righthand side m atrix B. On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the anay B . LD B >= \(\max (1, N)\).
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zhgeqz -im plem enta single-shiftversion of the \(Q Z\) method
for finding the generalized eigenvalues w (i)=ALPHA (i)BETA (i) of the equation \(\operatorname{det}(\mathrm{A}-\mathrm{w}\) (i) B\()=0\) If \(J 0 B=S\) ', then the pair \((A, B)\) is sim ultaneously reduced to Schurform (i.e., \(A\) and \(B\) are both upper triangular) by applying one unitary tranform ation (usually called Q) on the left and another (usually called \(Z\) ) on the right

\section*{SYNOPSIS}

```

    A LPHA,BETA,Q,LDQ,Z,LD Z,W ORK,LW ORK,RW ORK, INFO)
    ```
CHARACTER * \(1 \mathrm{JOB}, \mathrm{COMPQ}, \mathrm{COMPZ}\)
DOUBLE COM PLEX A (LDA,*), B (LDB,*), ALPHA (*), BETA (*),
Q (LDQ, \(), \mathrm{Z}(\mathrm{LD} Z, \star), \mathrm{W} O R K(*)\)
\(\mathbb{N}\) TEGER \(N, \mathbb{L O}, \mathbb{H} I, L D A, L D B, L D Q, L D Z, L W O R K, \mathbb{N} F O\)
DOUBLE PRECISION RW ORK (*)
SU BROUTINE ZHGEQZ_64 (JOB,COMPQ,COMPZ,N, ILO, \(\mathbb{H} I, A, L D A, B, L D B\),
    A LPHA, BETA, \(Q, L D Q, Z, L D Z, W\) ORK, LW ORK,RWORK, \(\mathbb{N} F O\) )
CHARACTER * \(1 \mathrm{JOB}, \mathrm{COMPQ}, \mathrm{COMPZ}\)
DOUBLE COM PLEX A (LDA,*), B (LDB,*), ALPHA (*), BETA (*),
Q (LDQ , \(), \mathrm{Z}(\mathrm{LD} \mathrm{Z}, \star), \mathrm{W} O R K(*)\)
\(\mathbb{N} T E G E R * 8 N, \mathbb{L} O, \mathbb{H} I, L D A, L D B, L D Q, L D Z, L W O R K, \mathbb{N} F O\)
DOUBLE PRECISION RW ORK (*)

\section*{F95 INTERFACE}

SU BROUTINE HGEQZ (JOB,COMPQ,COMPZ, \(\mathbb{N}], \mathbb{L} O, \mathbb{H} I, A,[L D A], B,[L D B]\), ALPHA,BETA, \(Q\), [LDQ], \(Z\), [LD Z], [W ORK], [LW ORK], [RW ORK], [ \(\mathbb{N F O}])\)

CHARACTER ( \(L E N=1\) ) :: OBB,COMPQ, COMPZ
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::ALPHA,BETA,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : A , B , Q , Z
\(\mathbb{N} T E G E R:: N, \Pi O, \mathbb{H} I, L D A, L D B, L D Q, L D Z, L W O R K, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::RW ORK

SU BROUT \(\mathbb{N} E\) H GEQ Z_64 (OB , COMPQ, COMPZ, \(\mathbb{N}], \mathbb{I} O, \mathbb{H} I, A,[L D A], B\), \([L D B], A L P H A, B E T A, Q,[L D Q], Z,[L D Z],[W O R K],[L W O R K],[R W O R K]\), [ \(\mathbb{N}\) FO ])

CHARACTER ( \(L E N=1)::\) OB, COMPQ, COMPZ
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::ALPHA,BETA, W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : A , B, Q , Z
\(\mathbb{N}\) TEGER (8) :: N , \(\mathbb{L} O, \mathbb{H} I, L D A, L D B, L D Q, L D Z, L W O R K, \mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zhgeqz (char j.b, charcom pq, char com pz, int n, int ilo, int ini, doublecom plex *a, int lda, doublecom plex *b, int ldb, doublecom plex *alpha, doublecom plex *beta, doublecom plex *q, int ldq, doublecom plex * \(z\), int ldz, int *info);
void zhgeqz_64 (char job, charcom pq, char com pz, long n, long 10 , long ihi, doublecom plex *a, long lda, doublecom plex *b, long ldb, doublecom plex *alpha, doublecom plex *beta, doublecom plex *q, long ldq, doublecom plex *z, long ldz, long *info);

\section*{PURPOSE}
zhgeqz im plem ents a single-shiftversion of the Q Z m ethod
for finding the generalized eigenvalues
w ( \(i=A L P H A\) (i) \(B E T A\) ( \((\mathbf{i})\) of the equation \(A\) are then
\(A \operatorname{LPHA}(1), \ldots, A \operatorname{LPHA}(\mathbb{N})\), and ofB areBETA \((1), \ldots, B E T A(\mathbb{N})\).

If \(\mathrm{JOB}=\mathrm{S}\) 'and COMPQ and COMPZ are V 'or 'I', then the unitary transform ations used to reduce ( \(\mathrm{A}, \mathrm{B}\) ) are accum ulated into the arays \(Q\) and \(Z\) s.t.:
(in) A (in) Z (in) \({ }^{*}=\mathrm{Q}\) (out) A (out) Z (out)*

Ref: C B.M oler \& G W . .Stew art, "A n A lgorithm for G eneralized M atrixigenvalue Problem s", SIAM J. Num er. A nal, 10 (1973) р. 241-256.

\section*{ARGUMENTS}

JOB (input)
\(=\mathrm{E}\) ': com pute only A LPHA andBETA. A and B w ill
not necessarily be put into generalized Schur
form . = S':putA and B into generalized Schur form, asw ell as com puting A LPHA and BETA.

COMPQ (input)
\(=\mathrm{N}\) : do notm odify Q .
\(=\mathrm{V}\) ': m ultiply the array Q on the right by the conjugate transpose of the unitary tranform ation that is applied to the left side of \(A\) and \(B\) to reduce them to Schur form . = I': like COMPQ=V', except thatQ w illbe initialized to the identity first.

COMPZ (input)
\(=\mathrm{N}\) : do notm odify Z.
\(=\mathrm{V}\) ': m ultiply the array Z on the right by the unitary tranform ation that is applied to the right side ofA and B to reduce them to Schur form . = 'I': like COM PZ=V', except thatZ w illbe initialized to the identity first.

N (input) The order of the \(m\) atrices \(A, B, Q\), and \(Z . N>=0\).

IIO (input)
It is assum ed thatA is already upper triangular
in row s and colum ns \(1: \mathbb{H O}-1\) and \(\mathrm{H} \mathrm{I}+1 \mathrm{~N} .1<=\Pi 0\)
\(<=\mathbb{H} I<=N\), if \(N>0 ; \Pi \mathrm{O}=1\) and \(\mathbb{H} \mathrm{I}=0\), if \(\mathrm{N}=0\).

IH I (input)
It is assum ed thatA is already upper triangular in row s and colum ns \(1: \mathbb{L O}-1\) and \(\mathrm{H} \mathrm{I}+1 \mathbb{N} .1<=\Pi 0\) \(<=\mathbb{H} I<=N\), if \(N>0 ; \Pi O=1\) and \(\mathbb{H} \mathrm{I}=0\), if \(\mathrm{N}=0\).

A (input) On entry, the \(N\) boy-N upper \(H\) essenberg \(m\) atrix \(A\).
Elem ents below the subdiagonalm ustbe zero. If
\(J O B=S\) ', then on exit A and B w ill have been sim ultaneously reduced to upper triangular form. If OB = E', then on exitA will have been destroyed.

LD A (input)
The leading dim ension of the anray A.LDA >= \(\max\) ( \(1, N\) ).
\(B\) (input) On entry, the \(N\) boy N upper triangular \(m\) atrix \(B\).
Elem ents below the diagonal must be zero. If

JO \(B=S\) ', then on exit \(A\) and \(B\) will have been sim ultaneously reduced to upper triangular form. If \(\mathrm{JOB}=\mathrm{E}\) ', then on exit B will have been destroyed.

LD B (input)
The leading dim ension of the array \(B . L D B>=m a x(\) \(1, N\) ).

ALPHA (output)
The diagonalelem ents of \(A\) when the pair \((A, B)\) has been reduced to Schur form. ALPHA (i) BETA (i) \(i=1, \ldots, N\) are the generalized eigenvalues.

\section*{BETA (output)}

The diagonalelem ents of \(B\) w hen the pair \((A, B)\) has been reduced to Schur form. ALPHA (i) BETA (i) \(i=1, \ldots, N\) are the generalized eigenvalues. A and B are norm alized so thatBETA (1),..,BETA \((\mathbb{N})\) are non-negative realnum bers.

Q (input/output)
If \(C O M P Q=N\) ', then \(Q\) willnotbe referenced. If \(C O M P Q=V\) ' or \(I\) ', then the conjugate transpose of the unitary transform ations w hich are applied to A and \(B\) on the leftw ill.be applied to the array \(Q\) on the right.

LD Q (input)
The leading dim ension of the array \(Q . L D Q>=1\). If \(C O M P Q=V\) 'or \(I\) ', then \(L D Q>=N\).

Z (input/output)
If \(C O M P Z=N\) ', then \(Z \mathrm{w}\) ill notbe referenced. If
COM PZ=V 'or 'I', then the unitary transform ations which are applied to \(A\) and \(B\) on the rightw ill be applied to the amay \(Z\) on the right.

LD Z (input)
The leading dim ension of the array \(\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1\). If \(\mathrm{COM} \mathrm{MZ}=\mathrm{V}\) 'or I ', then LD \(\mathrm{Z}>=\mathrm{N}\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O>=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the array W ORK. LW ORK >= \(\max (1, N)\).

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK aray, retums this value as the first entry of the W ORK array, and no emorm essage related to LW ORK is issued by XERBLA.
RW ORK (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the \(i\)-th argum enthad an illegalvałue
\(=1, \ldots, N\) : the \(\mathrm{Q} Z\) teration did not converge.
\((A, B)\) is not in Schur form, butALPHA (i) and \(\mathrm{BETA}(\mathrm{i}), \mathrm{F}=\mathbb{N F O}+1, \ldots, \mathrm{~N}\) should be correct. \(=\) \(\mathrm{N}+1, \ldots, 2 * \mathrm{~N}\) : the shift calculation failed. (A , B ) is notin Schur form, but ALPHA (i) and BETA (i), \(i=\mathbb{N}\) FO \(-\mathrm{N}+1, \ldots, \mathrm{~N}\) should be comect. \(>2 \star \mathrm{~N}\) : various "im possible" errors.

\section*{FURTHER DETAILS}

W e assum e that com plex A BS w orks as long as its value is less than overflow .

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhpcon -estim ate the reciprocal of the condition num ber of a com plex H em itian packed \(m\) atrix A using the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\) com puted by CHPTRF

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*),W ORK (*)
\mathbb{NTEGER N,}\mathbb{N}FO
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION ANORM,RCOND

```

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*),W ORK (*)
INTEGER*8N,\mathbb{NFO}
NNTEGER*8 \mathbb{PIVOT (*)}
DOUBLE PRECISION ANORM,RCOND

```

\section*{F95 INTERFACE}

SU BROUTINE HPCON (UPLO,N,A, \(\mathbb{P} \mathbb{I V O T}, A N O R M, R C O N D,[W O R K],[\mathbb{N} F O]\) )
CHARACTER (LEN=1)::UPLO
COM PLEX (8),D IM ENSION (:) ::A,W ORK
\(\mathbb{N}\) TEGER :: N, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V} O T\)
REAL (8) ::ANORM,RCOND


CHARACTER (LEN=1) :: UPLO
COM PLEX (8), D \(\mathbb{M} E N S I O N\) (:) ::A,W ORK
\(\mathbb{N} \operatorname{TEGER}\) (8) :: N , \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL (8) ::ANORM,RCOND

\section*{C INTERFACE}
\#include <sunperfh>
void zhpcon (charuplo, intn, doublecom plex *a, int*ipivot, double anorm, double *rcond, int *info);
void zhpcon_64 (charuplo, long n, doublecom plex *a, long *ipívot, double anorm, double *rcond, long *info);

\section*{PURPOSE}
zhpcon estim ates the reciprocal of the condition num ber of a com plex \(H\) erm itian packed \(m\) atrix A using the factorization A \(=U * D * U * * H\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\) com puted by CHPTRF.

A \(n\) estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND \(=1 /\) (ANORM * norm (inv (A )) ).

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) : : U ppertriangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}\) ** H ;
\(=\mathrm{L}:\) : Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\).

N (input) The order of them atrix A. \(\mathrm{N}>=0\).

A (input) The block diagonalm atrix \(D\) and the \(m\) ultipliers used to obtain the factorU orL as com puted by CH PTRF, stored as a packed triangularm atrix.

IPIVOT (input)
D etails of the interchanges and the block structure ofD as determ ined by CHPTRF.

ANORM (input)
The 1 -norm of the originalm atrix A.

\section*{RCOND (output)}

The reciprocal of the condition num ber of the
\(m\) atrix \(A\), com puted as RCOND = 1/(ANORM *A \(\mathbb{N} V N M)\),
where \(A \mathbb{N} V N M\) is an estim ate of the 1 -nom of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhpev - com pute all the eigenvalues and, optionally, eigenvectors of a com plex H erm Hian m atrix in packed storage

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZHPEV (JOBZ,UPLO,N,A,W ,Z,LD Z,W ORK,W ORK 2, INFO)}
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEX A (*), Z (LD Z ,*),W ORK (*)
\mathbb{NTEGER N,LD Z,INFO}
DOUBLE PRECISION W (*),W ORK2 (*)
SUBROUT\mathbb{NE ZHPEV_64(OOBZ,UPLO,N,A,W ,Z,LD Z,W ORK,W ORK2,INFO)}
CHARACTER * 1 OOBZ,UPLO
DOUBLE COM PLEX A (*), Z (LD Z ,*),W ORK (*)
\mathbb{NTEGER*8N,LDZ,INFO}
DOUBLE PRECISION W (*),W ORK2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE HPEV ( \(\mathrm{O} \mathrm{BZ}, \mathrm{UPLO}, \mathrm{N}, \mathrm{A}, \mathrm{W}, \mathrm{Z},[\operatorname{LD} Z],[\mathbb{W}\) ORK ], [W ORK 2], [ \(\mathbb{N} F \mathrm{~F}\) ])

CHARACTER (LEN=1) :: JOBZ, UPLO
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::A,W ORK
COM PLEX (8),D IM ENSION (:,:) :: Z
\(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathrm{LD} \mathrm{Z}, \mathbb{I N F O}\)
REAL (8), D IM ENSION (:) ::W ,W ORK2

SU BROUTINE HPEV_64 (JOBZ, UPLO ,N,A,W,Z, [LD Z], [W ORK ], [W ORK2], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1)::JOBZ,UPLO
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::A,W ORK
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::Z
\(\mathbb{N} T E G E R(8):: N, L D Z, \mathbb{N} F O\)
REAL (8),D IM ENSION (:) ::W ,W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zhpev (char jobz, charuplo, int n, doublecom plex *a, double *w, doublecom plex *z, int ldz, int *info);
void zhpev_64 (char jobz, char uplo, long n, doublecom plex
*a, double \({ }_{\mathrm{w}}\), doublecom plex *z, long ldz, long
*info);

\section*{PURPOSE}
zhpev com putes all the eigenvalues and, optionally, eigenvectors of a com plex H erm tiian m atrix in packed storage.

\section*{ARGUMENTS}

JO BZ (input)
= N ': C om pute eigenvalues only;
\(=\mathrm{V}:\) : om pute eigenvalues and eigenvectors.

UPLO (input)
\(=\mathrm{U}\) ': U ppertriangle of A is stored;
= L': Low ertriangle ofA is stored.
N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O \(n\) entry, the upper or low ertriangle of the Her \(m\) tian \(m\) atrix \(A\), packed collm nw ise in a linear array. The jth column of A is stored in the array A as follows: if UPLO = U',A (i+ (j 1) \(\mathrm{j}^{2} 2\) ) \(=\mathrm{A}(i, 7)\) for \(1<=\mathrm{i}<=\dot{j}\) if UPLO \(=\mathrm{L}\) ', \(\mathrm{A}(i+\) \((j-1)^{*}(2 * n-j / 2)=A(i, 7)\) for \(j=i<=n\).

On exit, A is overw rilten by values generated during the reduction to tridiagonal form. If \(\mathrm{PLO}=\) U', the diagonal and firstsuperdiagonal of the tridiagonal \(m\) atrix \(T\) overw rite the corresponding elem ents ofA, and if UPLO \(=\mathrm{L}\) ', the diagonal and first subdiagonal of \(T\) overw rite the corresponding
elem ents ofA .

W (output)
If \(\mathbb{N} F O=0\), the eigenvalues in ascending order.

Z (input) If \(\operatorname{JOBZ}=\mathrm{V}^{\prime}\), then if \(\mathbb{N F O}=0, \mathrm{Z}\) contains the orthonorm aleigenvectors of the \(m\) atrix \(A, w\) ith the \(i\)-th colum \(n\) of \(Z\) holding the eigenvector associated w ith \(W\) (i). If \(J O B Z=N\) ', then \(Z\) is not referenced.

LD Z (input)
The leading din ension of the array \(\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD \(Z>=\mathrm{max}(1, \mathrm{~N})\).
W ORK (w orkspace)
dim ension M AX ( \(1,2 * \mathrm{~N}-1)\) )
W ORK 2 (w orkspace)
dim ension (max (1,3*N-2))
\(\mathbb{N} F O\) (output)
= 0: successfulexit.
<0: if \(\mathbb{N N}\) FO = -i, the i-th argum enthad an illegalvahue.
> 0 : if \(\mathbb{N} F O=\) i, the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did not converge to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhpevd -com pute all the eigenvalues and, optionally, eigenvectors of a com plex H erm itian m atrix A in packed storage

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZHPEVD (JOBZ,UPLO,N,AP,W ,Z,LD Z,W ORK,LW ORK,RW ORK,}
LRW ORK,INORK,LIN ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEX AP (*),Z (LD Z,*),W ORK (*)
\mathbb{N TEGER N,LDZ,LW ORK,LRW ORK,LIN ORK,INFO}
INTEGER \mathbb{IN ORK (*)}
DOUBLE PRECISION W (*),RW ORK (*)
SUBROUTINE ZHPEVD_64(JOBZ,UPLO,N,AP,W ,Z,LD Z,W ORK,LW ORK,
RW ORK,LRW ORK,INORK,LIN ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEX AP (*),Z (LD Z,*),W ORK (*)
INTEGER*8N,LD Z,LW ORK,LRW ORK,LIN ORK,INFO
INTEGER*8 \mathbb{IN ORK (*)}
DOUBLE PRECISION W (*),RW ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE HPEVD (JOBZ,UPLO,N,AP,W,Z, [LD Z], [W ORK ], [LW ORK], \([R W\) ORK ], [LRW ORK], [ \(\mathbb{W}\) ORK ], [ \(\mathbb{L} \mathbb{W}\) ORK ], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1)::JOBZ, UPLO
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::AP,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::Z
\(\mathbb{N} T E G E R:: N, L D Z, L W\) ORK,LRW ORK,LIN ORK, \(\mathbb{N} F O\)


SU BROUTINE HPEVD_64 (JOBZ, UPLO ,N ,AP, W , Z, [LD Z], [W ORK ], [LW ORK ], \([R W\) ORK ], [LRW ORK], [ \(\mathbb{W}\) ORK ], [LIN ORK], [ \(\mathbb{N} F O])\)

CHARACTER (LEN=1) :: JOBZ,UPLO
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::AP,W ORK
COM PLEX (8), D IM ENSION (:,:) :: Z
\(\mathbb{N}\) TEGER (8) ::N,LD Z,LW ORK,LRW ORK,LIN ORK, \(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER (8),D \(\mathbb{I M}\) ENSION (:) :: IN ORK
REAL (8), D IM ENSION (:) ::W ,RW ORK

\section*{C INTERFACE}
\#include < sunperfh>
void zhpevd (char jobz, charuplo, intn, doublecom plex *ap, double \({ }_{\mathrm{w}}\), doublecom plex *z, int ldz, int *info);
void zhpevd_64 (char jobz, charuplo, long n, doublecom plex *ap, double *w, doublecom plex *z, long ldz, long *info);

\section*{PURPOSE}
zhpevd com putes all the eigenvalues and, optionally, eigenvectors of a com plex \(H\) erm itian \(m\) atrix \(A\) in packed storage. Ifeigenvectors are desired, ituses a divide and conquer algorithm .

The divide and conquer algorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on \(m\) achines w ith a guard digit in add/subtract, or on those binary \(m\) achines \(w\) ithout guard digits w hich subtract like the C ray X M P , C ray Y M P , C ray C-90, orC ray-2. It could conceivably fail on hexadecim al or decim al \(m\) achines \(w\) ithout guard digits, butw e know of none.

\section*{ARGUMENTS}

JOBZ (input)
\(=N^{\prime}:\) C om pute eigenvalues only;
\(=\mathrm{V}\) ': C om pute eigenvalues and eigenvectors.
UPLO (input)
= U :: U ppertriangle ofA is stored;
= L' ': Low er triangle ofA is stored.
N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

\section*{AP (input/output)}

O \(n\) entry, the upper or low ertriangle of the Her \(m\) itian matrix \(A\), packed colum nw ise in a linear array. The jth colum \(n\) of \(A\) is stored in the array AP as follows: ifUPLO = U',AP (i+ (j
 \(\left.+(j-1)^{\star}(2 * n-j) / 2\right)=A(i,>)\) for \(j=i<=n\).

On exit, AP is overw ritten by values generated during the reduction to tridiagonal form. IfU PLO = U', the diagonal and first superdiagonal of the tridiagonal m atrix T overw rite the comesponding elem ents ofA, and if UPLO \(=\mathrm{L}\) ', the diagonal and first subdiagonal of \(T\) overw rite the corresponding elem ents ofA.

W (output)
If \(\mathbb{N} F O=0\), the eigenvalues in ascending order.
Z (input) If \(\mathrm{OOBZ}=\mathrm{V}^{\prime}\), then if \(\mathbb{N} F O=0, \mathrm{Z}\) contains the orthonom aleigenvectors of the \(m\) atrix \(A, w\) ith the i-th colum \(n\) of \(Z\) holding the eigenvector associated w th \(W\) (i). If \(J O B Z=N\) ', then \(Z\) is not referenced.

LD Z (input)
The leading \(\mathrm{d} m\) ension of the array Z . LD \(\mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD Z \(>=\mathrm{max}(1, N)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of array \(\mathrm{W} O R \mathrm{~K}\). If \(\mathrm{N}<=1\), LW ORK must be at least 1. If \(J 0 B Z=N\) 'and \(N>\) 1, LW ORK m ust.be at least N . If \(\mathrm{JOBZ}=\mathrm{V}\) 'and N > 1, LW ORK m ustbe at least \(2{ }^{\star} \mathrm{N}\).

If LW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

RW ORK (w orkspace)
dim ension (LRW ORK) On exit, if \(\mathbb{N} F O=0\), RW ORK (1) retums the optim allRW ORK.

LRW ORK (input)
The dim ension of aray RW ORK. If \(\mathrm{N}<=1\), LRW ORK m ustbe atleast1. If OBZ \(=\mathrm{N}\) 'and \(N>\) 1, LRW ORK m ustbe atleast \(N\). If \(\mathrm{OBB}=\mathrm{V}\) 'and N \(>1\), LRW ORK m ustbe at least \(1+5 * \mathrm{~N}+2 \star \mathrm{~N} * * 2\).

IfLRW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the RW ORK aray, retums this value as the first entry of the RW ORK anray, and no errorm essage related to LRW ORK is issued by XERBLA.

IW ORK (w orkspace/output)
O n exit, if \(\mathbb{N} F O=0, \mathbb{I N}\) ORK (1) retums the optim al LIW ORK.

LIW ORK (input)
The dim ension of array \(\mathbb{I W}\) ORK. If JOBZ \(=\mathrm{N}\) 'orN \(<=1\), LIN ORK m ustbe at least1. If \(\mathrm{OBZ}=\mathrm{V}^{\prime}\) and \(N>1, L \mathbb{W}\) ORK m ustbe at least \(3+5 * N\).

If LIV ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the \(\mathbb{I W}\) ORK array, retums this value as the first entry of the \(\mathbb{I W}\) ORK amay, and no errorm essage related to \(L \mathbb{I W} O R K\) is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i-\) th argum ent had an illegalvalue.
\(>0:\) if \(\mathbb{N F O}=i\), the algorithm failed to converge; ioff-diagonalelem ents of an interm ediate tridiagonal form did notconverge to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhpevx - com pute selected eigenvalues and, optionally, eigenvectors of a complex Herm tian matrix A in packed storage

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHPEVX (ODBZ,RANGE,UPLO,N,A,VL,VU,\mathbb{I},\mathbb{U},ABTOL,}
NFOUND,W,Z,LD Z,W ORK,W ORK2,IN ORK 3, \mathbb{FA}\mathbb{I},\mathbb{NNFO)}

```
CHARACTER * 1 JOBZ,RANGE,UPLO
D OUBLE COM PLEX A (*), Z (LD Z, *), W ORK (*)
\(\mathbb{N}\) TEGERN, \(\mathbb{I}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{I}\) ORK 3 (*), \(\mathbb{F} A \mathbb{L}\) (*)
DOUBLE PRECISION VL,VU,ABTOL
DOUBLE PRECISION W (*),W ORK2 (*)
SU BROUTINE ZHPEVX_64 (JOBZ,RANGE,UPLO,N,A,VL,VU, IL, \(\mathbb{I}, A B T O L\),
    NFOUND,W,Z,LD Z,W ORK,WORK2, IN ORK \(3, \mathbb{F} A \mathbb{I}, \mathbb{N} F O)\)
CHARACTER * 1 JOBZ,RANGE, UPLO
DOUBLE COM PLEX A (*), Z (LD Z, \({ }^{\star}\) ), W ORK (*)
\(\mathbb{N}\) TEGER* \(8 \mathrm{~N}, \mathbb{I}, \mathbb{I}, N F O U N D, L D Z, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{I V}\) ORK 3 (*), \(\mathbb{F A} \mathbb{L}(*)\)
DOUBLE PRECISION VL,VU,ABTOL
DOUBLE PRECISION W (*),W ORK2 (*)

\section*{F95 INTERFACE}

SU BROUTINE HPEVX (JOBZ,RANGE,UPLO,N,A,VL,VU, \(\mathbb{I}, \mathbb{I}, A B T O L\), \(\mathbb{N} F O U N D], W, Z,[L D Z], \mathbb{W}\) ORK \(],[\mathbb{W} O R K 2],[\mathbb{N} O R K 3], \mathbb{F A} \mathbb{I},[\mathbb{N} F O])\)

CHARACTER (LEN=1):: OBZ,RANGE,UPLO
COM PLEX (8),D \(\mathbb{I}\) ENSION (:) ::A,W ORK

COM PLEX (8), D \(\mathbb{I M} E N S I O N(:,:):\) Z
\(\mathbb{N}\) TEGER :: N, \(\mathbb{I}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{I N} O R K 3, \mathbb{F A} \mathbb{I}\)
REAL (8) ::VL, VU, ABTOL
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ,W ORK2

SU BROUTINE HPEVX_64 (OBZ,RANGE, UPLO,N,A,VL,VU, \(\mathbb{I}, \mathbb{U}, A B T O L\), \([\mathbb{N} F O U N D], W, Z,[L D Z],[W O R K],[W O R K 2],[\mathbb{N} O R K 3], \mathbb{F} A \mathbb{H},[\mathbb{N} F O])\)

CHARACTER ( \(几 E N=1)::\) OBZ,RANGE, UPLO
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::A,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) :: Z
\(\mathbb{N}\) TEGER (8) :: N , \(\mathbb{I}, \mathbb{U}, N F O U N D, L D Z, \mathbb{N F O}\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I N} O R K 3, \mathbb{F} A \mathbb{I}\)
REAL (8) :: VL, VU, ABTOL
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ,W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zhpevx (char jobz, char range, charuplo, int n, doublecom plex *a, double vl, double vu, int il, int iu, double abtol, int *nfound, double *w, doublecom plex *z, int ldz, int *ifail, int *info);
void zhpevx_64 (char j̀bz, charrange, char uplo, long n, doublecom plex *a, double vl, double vu, long il, long in, double abtol, long *nfound, double *w, doublecom plex *z, long ldz, long *ifail, long *info);

\section*{PURPOSE}
zhpevx com putes selected eigenvalues and, optionally, eigenvectors of a com plex H erm itian m atrix A in packed storage. E igenvalues/vectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

\section*{ARGUMENTS}
```

JO B Z (input)
= N ': C om pute eigenvalues only;
= V ': C om pute eigenvalues and eigenvectors.

```

RANGE (input)
= A ': alleigenvalues w illbe found;
\(=\) V : alleigenvalues in the half-open interval

NL,VU] will be found; = I': the \(\mathbb{I}\)-th through \(\mathbb{I U}\)-th eigenvaluesw illbe found.

UPLO (input)
\(=\mathrm{U}:\) : U pper triangle ofA is stored;
= L': Low ertriangle ofA is stored.
N (input) The order of the matrix \(A . N>=0\).
A (input/output)
O \(n\) entry, the upper or low er triangle of the H erm itian matrix A, packed colum nw ise in a linear array. The jth colum \(n\) of \(A\) is stored in the array A as follows: if UPLO \(=U^{\prime}\), A (i+ ( \(j\) 1) \(* j 2\) ) \(=A(i, j)\) for \(1<=i<=j\) ifUPLO \(=L '\), A ( \(i+\) \((j-1) \star(2 \star n-y / 2)=A(i, 7)\) for \(j=i<=n\).
On exit, A is overw ritten by values generated during the reduction to tridiagonal form. If \(\mathrm{PLO}=\) U ', the diagonal and firstsuperdiagonal of the tridiagonal \(m\) atrix \(T\) overw rite the comesponding elem ents of A, and if U PLO = 5 ', the diagonal and first subdiagonal of \(T\) overw rite the coresponding elem ents ofA.

VL (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU. N ot referenced ifRANGE=A 'or I''.

VU (input)
IfRANGE=V', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU .
N ot referenced ifRANGE=A 'or I''.
II (input)
IfRA N G E= I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{U}<=\mathbb{U}<=N\), if \(N>0\); \(\mathbb{H}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N otreferenced ifRANGE= A'or V'.

IU (input)
IfRANGE= 'I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{U}<=\mathbb{U}<=N\), if \(N>0\); \(\mathbb{H}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N otreferenced ifRANGE \(=\) A'or V'.

ABTOL (input)
The absolute error tolerance for the eigenvalues.

A \(n\) approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b]
ofw idth less than orequal to
ABTOL + EPS * max ( \(\mathfrak{a}|\),\(| b|),\)
\(w\) here EPS is them achine precision. If ABTOL is less than orequal to zero, then EPS* \(\mid\) | w illibe used in its place, where \(F \mid\) is the 1 -norm of the tridiagonal \(m\) atrix obtained by reducing \(A\) to tridiagonal form .

E igenvalues w illlbe com puted m ostaccurately when ABTOL is set to tw ice the underflow threshold \(2 *\) SLAM CH (S ), notzero. If this routine retums w ith \(\mathbb{N}\) FO \(>0\), indicating that som e eigenvectors did not converge, try setting ABTOL to \(2 *\) SLAM CH (S ).

See "C om puting Sm allSingularV ahues of B idiagonal M atrices with G uaranteed H igh Relative A ccuracy," by Dem m eland \(K\) ahan, LA PA CK W orking \(N\) ote \#3.

\section*{NFOUND (output)}

The total num ber of eigenvalues found. \(0<=\) NFOUND <= N. IfRANGE = A',NFOUND = N, and if RANGE = 'I',NFOUND = \(\mathbb{U}-\mathbb{L}+1\).

W (output)
If \(\mathbb{N}\) FO \(=0\), the selected eigenvalues in ascending order.
\(Z\) (input) If \(\mathrm{JOBZ}=\mathrm{V}\) ', then if \(\mathbb{N} F O=0\), the first NFOUND colum ns of \(Z\) contain the orthonorm aleigenvectors of them atrix A comesponding to the selected eigenvalues, \(w\) ith the i-th colum \(n\) of \(Z\) holding the eigenvector associated w ith W (i). If an eigenvector fails to converge, then that colum n of \(Z\) contains the latest approxim ation to the eigenvector, and the index of the eigenvector is retumed in \(\mathbb{F} A \mathbb{I}\). If \(J O B Z=N\) ', then \(Z\) is not referenced.
\(N\) ote: the user must ensure that at least
\(\max (1, N F O U N D)\) colum ns are supplied in the array \(Z\); if RANGE = V', the exactvalue ofNFOUND is not know \(n\) in advance and an upperbound \(m\) ustbe used.

LD Z (input)
The leading dim ension of the array \(Z\). LD \(Z>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD \(Z>=\mathrm{max}(1, N)\).

W ORK (w orkspace)
dim ension (2*N)

W ORK 2 (w orkspace) dim ension ( \(7 * \mathrm{~N}\) )

IV ORK 3 (w orkspace) dim ension ( \(5 * N\) )
تAII (output)
If \(\mathrm{OBZ}=\mathrm{V}\) ', then if \(\mathbb{N F O}=0\), the first NFOUND elem ents of \(\mathbb{F} A \mathbb{I}\) are zero. If \(\mathbb{N} F O>0\), then IFA II contains the indices of the eigenvectors that failed to converge. If \(\mathrm{OBZ}=\mathrm{N}\) ', then IFA II is notreferenced.
\(\mathbb{N}\) FO (output)
= 0 : successfiulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=\) i, then ieigenvectors failed to converge. Their indiges are stored in amay \(\mathbb{F}\) III.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhpgst-reduce a com plex Herm itian-definite generalized eigenproblem to standard form, using packed storage

\section*{SYNOPSIS}

SUBROUTINE ZHPGST (TTYPE, UPLO,N,AP,BP, \(\mathbb{N} F O\) )
CHARACTER * 1 UPLO
DOUBLE COM PLEX AP (*), BP (*)
\(\mathbb{N} T E G E R \mathbb{T T Y P E}, N, \mathbb{N F O}\)
SU BROUTINE ZHPGST_64 (TTYPE,UPLO,N,AP,BP, \(\mathbb{N} F O\) )

CHARACTER * 1 UPLO
D OUBLE COM PLEXAP (*) , BP (*)
\(\mathbb{N}\) TEGER*8 \(\mathbb{T} Y \mathrm{PE}, \mathrm{N}, \mathbb{N} F O\)

\section*{F95 INTERFACE}

SU BROUTINE HPGST (TTYPE, UPLO ,N,AP, BP, [ \(\mathbb{N} F O]\) )
CHARACTER (LEN=1)::UPLO
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::AP,BP
\(\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, \mathbb{N} F O\)
SUBROUTINE HPGST_64 (TTYPE, UPLO ,N,AP,BP, [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D IM ENSION (:) ::AP,BP
\(\mathbb{N} T E G E R(8):: \mathbb{T Y P E}, N, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zhpgst(int itype, charuplo, intn, doublecom plex *ap, doublecom plex *bp, int *info);
void zhpgst_64 (long itype, charuplo, long n, doublecom plex *ap, doublecom plex *bp, long *info);

\section*{PURPOSE}
zhpgst reduces a complex Herm itian-definite generalized eigenproblem to standard form, using packed storage.

If ITYPE \(=1\), the problem is \(A * x=\operatorname{lam}\) bda* \(\mathrm{B} * \mathrm{x}\), and \(A\) is overw ritten by \(\operatorname{inv}(U * * H) * A * \operatorname{inv}(U)\) or \(\operatorname{inv}(\mathbb{L}) * A * \operatorname{inv}(\mathbb{L} * * H)\)
If ITYPE \(=2\) or 3 , the problem is \(A * B * x=\) lam bda* \(x\) or \(\mathrm{B} * \mathrm{~A} * \mathrm{X}=\operatorname{lam}\) bda* x , and A is overw rilten by \(\mathrm{U} * \mathrm{~A} * \mathrm{U} * * \mathrm{H}\) or \(\mathrm{L} * *_{\mathrm{H}} *_{\mathrm{A}}{ }^{\mathrm{L}} \mathrm{L}\).

B m usthave been previously factorized as U **H *U or L \({ }^{*}\) L** H by CPPTRF.

\section*{ARGUMENTS}

ITYPE (input)
\(=1\) : compute \(\quad \operatorname{inv}(U * * H) * A * \operatorname{inv}(U)\) or
inv (L) \({ }^{\star} \mathrm{A}\) *inv ( \((\mathrm{L} * * \mathrm{H})\);
\(=2\) or 3 :com pute \(U * A * U * * H\) or \(L * * H * A * L\).
UPLO (input)
= U ': Uppertriangle ofA is stored and B is factored as U ** \(\mathrm{H} * \mathrm{U}\); = L ': Low er triangle ofA is stored and \(B\) is factored as \(L \star L * * H\).

N (input) The order of the m atrices A and \(\mathrm{B} . \mathrm{N}>=0\).
AP (input/output)
O \(n\) entry, the upper or low er triangle of the H er\(m\) itian \(m\) atrix \(A\), packed colum nw ise in a linear array. The jth colum n of A is stored in the array AP as follows: ifUPLO = U',AP (i+ \((j\)
 \(+(j-1)^{\star}(2 n-j / 2)=A(i, j)\) for \(j=i<=n\).

Onexit, if \(\mathbb{N F O}=0\), the transform ed \(m\) atrix, stored in the sam e form at as A.
\(B P\) (input)
The triangular factor from the \(C\) holesky factoriza-
tion of B, stored in the sam e form at as A, as retumed by CPPTRF.
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

zhpgv -com pute allthe eigenvalues and, optionally, the
eigenvectors of a com plex generalized H erm itian-definite
eigenproblem, of the form $A * x=(l a m . b d a) * B * x, A * B x=(l a m ~ b d a){ }^{*} x$,
or * $^{\mathrm{A}}$ * $\mathrm{x}=(\operatorname{lam} . \mathrm{bda}){ }^{\mathrm{x}} \mathrm{x}$

```

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZHPGV (TTYPE,NOBZ,UPLO,N,A,B,W ,Z,LD Z,W ORK,W ORK2,}
INFO)
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEX A (*),B (*), Z (LD Z ,*),W ORK (*)
\mathbb{NTEGER ITYPE,N,LD Z,INFO}
DOUBLE PRECISION W (*),W ORK2 (*)
SUBROUT\mathbb{NE ZHPGV_64(ITYPE,JOBZ,UPLO,N,A,B,W ,Z,LD Z,W ORK,}
W ORK2,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEX A (*),B (*),Z (LD Z,*),W ORK (*)
\mathbb{NTEGER*8 ITYPE,N,LD Z,INFO}
DOUBLE PRECISION W (*),W ORK2 (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE HPGV (TTYPE, JOBZ, UPLO ,N,A,B,W,Z,[LD Z], [W ORK ], [W ORK2], [ $\mathbb{N F O}$ ])
CHARACTER (LEN=1): : JOBZ, UPLO
COM PLEX (8),D $\mathbb{M}$ ENSION (:) ::A,B,W ORK
COM PLEX (8), D $\mathbb{M}$ ENSION (:,:) :: Z
$\mathbb{N} T E G E R:: \mathbb{T} Y P E, N, L D Z, \mathbb{N} F O$
REAL (8), D IM ENSION (:) ::W ,W ORK2

```

SU BROUTINE HPGV_64 (TTYPE, \(\mathcal{O}\) BZ, UPLO , N, A, B, W, Z, [LD Z], [W ORK ], [W ORK 2], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1):: JOBZ, UPLO
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::A,B,W ORK
COM PLEX (8), D IM ENSION (: : : : : Z
\(\mathbb{N}\) TEGER (8) :: \(\mathbb{T} Y\) PE, N,LD \(\mathrm{Z}, \mathbb{N} F O\)
REAL (8),D IM ENSION (:) ::W ,W ORK2

\section*{C INTERFACE}
\#include < sunperfh>
void zhpgv (int itype, char jobz, char uplo, int \(n\), doublecomplex *a, doublecomplex *b, double *w, doublecom plex *z, int ldz, int*info);
void zhpgv_64 (long itype, char jobz, char uplo, long n, doublecom plex *a, doublecom plex *b, double *w, doublecom plex *z, long ldz, long *info);

\section*{PURPOSE}
zhpgv com putes all the eigenvalues and, optionally, the eigenvectors of a com plex generalized H erm itian-definite eigenproblem, of the form \(A * x=(l a m ~ b d a) * B * x\), \(\mathrm{A} * \mathrm{~B} \mathrm{X}=(\operatorname{lam} . \mathrm{bda}){ }^{\star} \mathrm{X}\), or \(\mathrm{B}{ }^{\mathrm{A}} \mathrm{A}^{*} \mathrm{X}=\left(\mathrm{lam}\right.\) bda)\({ }^{*} \mathrm{X}\). Here A and B are assum ed to be \(H\) erm tian, stored in packed form at, and B is also positive definite.

\section*{ARGUMENTS}

ITYPE (input)
Specifies the problem type to be solved:
\(=1: A{ }^{*} \mathrm{x}=\left(\mathrm{lam}\right.\) bda) \({ }^{\mathrm{B}}{ }^{*}{ }_{\mathrm{x}}\)
\(=2: \mathrm{A} * \mathrm{~B} * \mathrm{X}=\left(\mathrm{lam}\right.\) bda) \({ }^{*} \mathrm{X}\)
\(=3: B * A * X=(l a m ~ b d a){ }^{*} \mathrm{x}\)
\(J O B Z\) (input)
\(=N^{\prime}:\) C om pute eigenvalues only;
\(=\mathrm{V}\) ': C om pute eigenvalues and eigenvectors.

UPLO (input)
\(=\mathrm{U}\) : : U pper triangles of \(A\) and \(B\) are stored;
\(=\mathrm{L}\) ': Low ertriangles of \(A\) and \(B\) are stored.
N (input) The order of the m atrices A and \(\mathrm{B} . \mathrm{N}>=0\).

A (input/output)
O \(n\) entry, the upper or low er triangle of the Her \(m\) itian \(m\) atrix \(A\), packed colum nw ise in a linear aray. The jth column of A is stored in the array A as follows: if UPLO \(=\mathrm{U}^{\prime}, \mathrm{A}(i+(j\)
 \((j-1)^{\star}(2 \star n-j / 2)=A(i, 7)\) for \(j=i<=n\).

On exit, the contents ofA are destroyed.
B (input/output)
O n entry, the upper or low er triangle of the Her \(m\) tian \(m\) atrix \(B\), packed colum nw ise in a linear anray. The jth column of \(B\) is stored in the array B as follows: if UPLO \(=\mathrm{U}\) ', \(\mathrm{B}(\mathrm{i}+(j\) 1) \(\star j 2\) ) \(=\mathrm{B}(i, 7)\) for \(1<=i<=j\) if \(U P L O=L\) ', \(B(i+\) \((j-1) *(2 * n-j / 2)=B(i, 7)\) for \(j=i<=n\).

On exit, the triangular factor \(U\) or \(L\) from the Cholesky factorization \(B=U * * H * U\) or \(B=L * L * * H\), in the sam e storage form atas \(B\).

W (output)
If \(\mathbb{N}\) FO \(=0\), the eigenvalues in ascending order.
\(Z\) (input) If \(J O B Z=V\) ', then if \(\mathbb{N F O}=0, Z\) contains the m atrix Z of eigenvectors. The eigenvectors are norm alized as follow s : if ITYPE \(=1\) or 2 , \(Z * * H * B * Z=I\); if IT \(Y P E=3, Z * * H * i n v(B) * Z=I\). If JO BZ \(=N\) ', then \(Z\) is not referenced.

\section*{LD \(Z\) (input)}

The leading \(d i m\) ension of the array \(Z . L D Z>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD \(Z>=\max (1, N)\).

W ORK (w orkspace)
dim ension M AX ( \(1,2 * \mathrm{~N}-1)\) )
W ORK 2 (w orkspace)
dim ension MAX ( \(1,3 * N-2)\) )
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
<0: if \(\mathbb{N N}\) FO = -i, the i-th argum ent had an illegalvahue
> 0: CPPTRF orCHPEV retumed an error code:
\(<=\mathrm{N}:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{CH}\) PEV failed to converge; i
off-diagonalelem ents of an interm ediate tridiagonal form did not convergeto zero; > N : if \(\mathbb{N} F O=\)
\(\mathrm{N}+\mathrm{i}\), for \(1<=\mathrm{i}<=\mathrm{n}\), then the leading m inorof orderiofB is notpositive definite. The factorization of B could not be com pleted and no eigenvalues or eigenvectors w ere com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zhpgvd - com pute allthe eigenvalues and, optionally, the eigenvectors of a com plex generalized H erm itian-definite eigenproblem, of the form \(\mathrm{A} * \mathrm{x}=\left(\mathrm{lam}\right.\) bda) \({ }^{\mathrm{B}} \mathrm{B}\) x, \(\mathrm{A} * \mathrm{~B} x=\left(\operatorname{lam}\right.\) bda) \({ }^{\mathrm{x}}\), or \(B{ }^{*} A * X=\left(l a m\right.\) bda) \({ }^{*} X\)

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZHPGVD (TTYPE,JOBZ,UPLO,N,AP,BP,W,Z,LD Z,W ORK,}
LW ORK,RW ORK,LRW ORK,IN ORK,LIN ORK,INFO)
CHARACTER * 1 JOBZ,UPLO
DOUBLE COM PLEX AP (*),BP (*),Z (LD Z ,*),W ORK (*)

```

```

INTEGER IN ORK (*)
DOUBLE PRECISION W (*),RW ORK (*)
SU BROUT\mathbb{NE ZHPGVD_64 (TTYPE,NOBZ,UPLO,N,AP,BP,W ,Z,LDZ,W ORK,}
LW ORK,RW ORK,LRW ORK,IN ORK,LIN ORK,\mathbb{NFO)}
CHARACTER * 1 JOBZ,UPLO
D OU BLE COM PLEX AP (*),BP (*),Z (LD Z,*),W ORK (*)
NNTEGER*8 ITYPE,N,LDZ,LW ORK,LRW ORK,LIW ORK,INFO
INTEGER*8 代ORK (*)
DOUBLE PRECISION W (*),RW ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE HPGVD (TTYPE, JOBZ, UPLO, N,AP,BP, W, Z, [LD Z], [W ORK], [LW ORK], RW ORK ], [LRW ORK], [IW ORK], [LINORK], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) :: JOBZ, UPLO
COMPLEX (8),D \(\mathbb{I M} E N S \mathbb{O N}(:):: A P, B P, W\) ORK

COM PLEX (8), D IM ENSION (:,:) ::Z
\(\mathbb{N}\) TEGER :: \(\mathbb{T} Y\) PE, \(N, L D Z, L W\) ORK,LRW ORK,LIN ORK, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL (8),D IM ENSION (:) ::W ,RW ORK

SU BROUTINE HPGVD_64 (TTYPE, JOBZ, UPLO ,N,AP,BP,W,Z, [LD Z], \(\mathbb{W}\) ORK ], [LW ORK ], RW ORK ], [LRW ORK ], [IW ORK ], [LIN ORK ], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1)::JOBZ,UPLO
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::AP,BP,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : Z
\(\mathbb{N}\) TEGER (8) :: ITYPE,N,LD Z,LW ORK,LRW ORK,LIN ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL (8), D IM ENSION (:) ::W ,RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zhpgvd (int iype, char jobz, charuplo, int n, doublecom plex *ap, doublecom plex *bp, double *w , doublecom plex *z, int ldz, int*info);
void zhpgvd_64 (long itype, char jobz, char uplo, long n, doublecom plex *ap, doublecom plex *bp, double *w, doublecom plex *z, long ldz, long *info);

\section*{PURPOSE}
zhpgrd com putes all the eigenvalues and, optionally, the eigenvectors of a com plex generalized H erm itian-definite eigenproblem, of the form \(A * x=(l a m ~ b d a) * B * x\), \(\mathrm{A} * \mathrm{~B} x=(\operatorname{lam} \operatorname{bda}) * \mathrm{x}\), or \(\mathrm{B} * \mathrm{~A} * \mathrm{x}=(\mathrm{lam} . \mathrm{bda}) * \mathrm{x}\). H ere A and B are assum ed to be \(H\) erm itian, stored in packed form at, and \(B\) is also positive definite.
Ifeigenvectors are desired, ituses a divide and conquer algorithm .

The divide and conquer algorithm \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. Itw illw ork on \(m\) achines w th a guard digit in add/subtract, or on those binary \(m\) achines \(w\) thout guard digits w hich subtract like the \(C\) ray X M P , C ray Y M P , C ray C -90, orC ray-2. It could conceivably fail on hexadecim al or decim al machines w thout guard digits, butw e know of none.

\section*{ARGUMENTS}

ITYPE (input)

Specifies the problem type to be solved:
\(=1: \mathrm{A}{ }^{*} \mathrm{x}=\left(\mathrm{lam}\right.\) bda) \({ }^{\mathrm{B}} \mathrm{B}_{\mathrm{x}}\)
\(=2: \mathrm{A} * \mathrm{~B} * \mathrm{x}=\left(\mathrm{lam}\right.\) bda) \({ }^{*} \mathrm{x}\)
\(=3: B{ }^{*} \mathrm{~A} * \mathrm{X}=\left(\right.\) lam bda) \({ }^{*} \mathrm{X}\)

JOBZ (input)
\(=\mathrm{N}^{\prime}\) : C om pute eigenvalues only;
\(=\mathrm{V}\) : C om pute eigenvalues and eigenvectors.
UPLO (input)
\(=U\) ': U ppertriangles of \(A\) and \(B\) are stored;
\(=\mathrm{L}\) ': Low ertriangles of \(A\) and \(B\) are stored.
N (input) The order of the \(m\) atrices \(A\) and \(B . N>=0\).
AP (input/output)
O \(n\) entry, the upper or low er triangle of the H er-
\(m\) itian matrix A, packed colum nw ise in a linear
array. The jth colum n of A is stored in the
array AP as follows: ifUPLO = U', AP (i+ (j

\(\left.+(j-1)^{\star}(2 \star n-j) / 2\right)=A(i, 7)\) for \(j=i<=n\).
On exit, the contents of AP are destroyed.

BP (input/output)
O \(n\) entry, the upper or low ertriangle of the Her \(m\) tian \(m\) atrix \(B\), packed colum nw ise in a linear array. The jth column of \(B\) is stored in the array BP as follows: if UPLO \(=\mathrm{U}^{\prime}, \mathrm{BP}(i+(j\)
 \(+(j-1)^{\star}(2 \star n-j / 2)=B(i, 7)\) for \(j=i<=n\).

On exit, the triangular factor \(U\) or \(L\) from the Cholesky factorization \(B=U * * H * U\) or \(B=L * L * * H\), in the sam e storage form at as \(B\).

W (output)
If \(\mathbb{N}\) FO \(=0\), the eigenvalues in ascending order.
\(Z\) (input) If \(\mathcal{O D B Z}=\mathrm{V}^{\prime}\), then if \(\mathbb{N} F O=0, \mathrm{Z}\) contains the m atrix Z of eigenvectors. The eigenvectors are norm alized as follows: if ITYPE \(=1\) or 2 , \(Z * * H * B * Z=I\); if IT \(Y P E=3, Z * * H * i n v(B) * Z=I\). If Jo \(\mathrm{BZ}=\mathrm{N}\) ', then Z is not referenced.

LD \(Z\) (input)
The leading \(d i m\) ension of the array \(Z . L D Z>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD \(Z>=\mathrm{max}(1, N)\).

W ORK (w orkspace)

On exit, if \(\mathbb{N F} F=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of array \(\mathrm{W} O R \mathrm{~K}\). If \(\mathrm{N}<=1\), LW ORK \(>=1\). If \(\mathrm{JOBZ}=\mathrm{N}\) 'and \(N>1\), LW \(O R K>=N\). If \(\mathrm{OOBZ}=\mathrm{V}\) 'and \(\mathrm{N}>1\), LW ORK \(>=2 \star \mathrm{~N}\).

IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of theW ORK ancay, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

RW ORK (w orkspace)
On exit, if \(\mathbb{N}\) FO \(=0, R W\) ORK (1) retums the optim al LRW ORK.

LRW ORK (input)
The dimension of array RW ORK. If \(\mathrm{N}<=1\), LRW ORK >=1. IfJOBZ \(=N\) 'and \(N>1\),LRW ORK \(>=\) N. If JOBZ \(=\mathrm{V}\) 'and \(\mathrm{N}>1\),LRW ORK >=1 + 5*N + \(2 * N * * 2\) 。

If LRW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
the RW ORK array, retums this value as the first entry of the RW ORK aray, and no enrorm essage related to LRW ORK is issued by X ERBLA.

IN ORK (w orkspace/output)
On exit, if \(\mathbb{N} F O=0, \mathbb{I N}\) ORK (1) retums the optim al
LIN ORK.

LIV ORK (input)
The dim ension of array \(\mathbb{I N}\) ORK. If JOBZ \(=\mathrm{N}\) 'orN
\(<=1\), LIV ORK >= 1. If JOBZ \(=V\) 'and \(N>1\), LIN ORK >= \(3+5 * N\) 。

If \(L \mathbb{I V} O R K=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the \(\mathbb{I V}\) ORK array, retums this value as the first entry of the \(\mathbb{I W} O R K\) aray, and no errorm essage related to \(L \mathbb{I N} O R K\) is issued by X ERBLA.
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue
>0: CPPTRF orCHPEVD retumed an enrorcode:
<= N: if \(\mathbb{N F F O}=\mathrm{i}, \mathrm{CHPEVD}\) failed to converge; i off-diagonalelem ents of an interm ediate tridiagonal form did notconvergeto zero; > N : if \(\mathbb{N} F O=\) \(\mathrm{N}+\mathrm{i}\), for \(1<=\mathrm{i}<=\mathrm{n}\), then the leading m inorof orderiofB is notpositive definite. The factorization of B could not be com pleted and no eigenvalues or eigenvectors w ere com puted.

\section*{FURTHER DETAILS}

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv. of K entucky, USA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
```

zhpgvx - com pute selected eigenvalues and, optionally,
eigenvectors of a com plex generalized H erm itian-definite
eigenproblem, of the form A *x= (lam bda)*B *x,A *B x= (lam bda)*x,
orB *A *x= (lam bda)*x

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHPGVX (TTYPE,JOBZ,RANGE,UPLO,N,AP,BP,VL,VU,IL,}
\mathbb{U},ABSTOL,M,W ,Z,LDZ,W ORK,RW ORK,IN ORK,\mathbb{FA}\mathbb{I},\mathbb{N}FO)

```
CHARACTER * 1 JOBZ,RANGE,UPLO
DOUBLE COM PLEX AP (*), BP (*), Z (LD Z,*), W ORK (*)
\(\mathbb{N} T E G E R \mathbb{I T Y} P E, N, \mathbb{I}, \mathbb{U}, \mathrm{M}, L D Z, \mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{I N}\) ORK (*), \(\mathbb{F A} \mathbb{I}\left({ }^{*}\right)\)
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION W (*),RW ORK (*)
SU BROUTINE ZHPGVX_64 (TTYPE, JOBZ,RANGE,UPLO,N,AP,BP,VL,VU, IL,
    \(\mathbb{I U}, A B S T O L, M, W, Z, L D Z, W\) ORK,RW ORK, \(\mathbb{I N} O R K, \mathbb{F} A \mathbb{I}, \mathbb{N} F O)\)
CHARACTER * 1 JOBZ,RANGE,UPLO
DOUBLE COM PLEX AP (*), BP (*), Z (LD Z,*), W ORK (*)
\(\mathbb{N} T E G E R * 8 \mathbb{I T} Y P E, N, \mathbb{L}, \mathbb{U}, M, L D Z, \mathbb{N} F O\)
\(\mathbb{N} T E G E R * 8 \mathbb{I N} O R K(*), \mathbb{F A} \mathbb{I}(*)\)
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISION W (*),RW ORK (*)

F95 INTERFACE
SU BROUTINE HPGVX (TTYPE, JOBZ,RANGE,UPLO,N,AP,BP,VL,VU, IL, \(\mathbb{U}, A B S T O L, M, W, Z,[L D Z],[W O R K],[R W O R K],[\mathbb{W} O R K], \mathbb{F} A \mathbb{I}\), [ \(\mathbb{N}\) FO ])

CHARACTER ( \(L E N=1\) ) : : JOBZ,RANGE, UPLO
COMPLEX (8), D \(\mathbb{M}\) ENSION (:) ::AP, BP,W ORK
COM PLEX (8), D \(\mathbb{I M} E N S \mathbb{O N}(:,:\) ) ::Z
\(\mathbb{N}\) TEGER :: \(\mathbb{T} Y P E, N, \mathbb{I}, \mathbb{Z}, M, L D Z, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{I W}\) ORK, \(\mathbb{F} A \mathbb{I}\)
REAL (8) ::VL,VU,ABSTOL
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ,RW ORK

SU BROUTINE HPGVX_64 (TTYPE, JOBZ,RANGE, UPLO, N, AP, BP, VL, VU, \(\mathbb{H}, \mathbb{U}, A B S T O L, M, W, Z,[L D Z],[W O R K],[R W\) ORK \(],[\mathbb{W} O R K], \mathbb{F} A \mathbb{L}\), [ \(\mathbb{N} \mathrm{FO}]\) )

CHARACTER (LEN=1) :: JOBZ,RANGE,UPLO
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::AP,BP,W ORK
COM PLEX (8), D \(\mathbb{I}\) ENSION (: : : : : Z
\(\mathbb{N} T E G E R(8):: \mathbb{I T Y} P E, N, \mathbb{H}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, \mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{I W} O R K, \mathbb{F A} \mathbb{I}\)
REAL (8) ::VL,VU,ABSTOL
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ,RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zhpgvx (int itype, char jंbz, char range, charuplo, int n, doublecom plex *ap, doublecom plex *bp, double Vl , double vu, int il, intiu, double abstol, int \({ }_{\mathrm{m}}\), double \({ }^{*} \mathrm{~W}\), doublecom plex \({ }^{*} \mathrm{z}_{\text {, int }}\) int z , int *ifail, int*info);
void zhpgvx_64 (long type, char jobz, char range, charuplo, long n, doublecom plex *ap, doublecom plex *bp, double vl, double vu, long il, long iu, double abstol, long *m, double *w , doublecom plex *z, long ldz, long *ifail, long *info);

\section*{PURPOSE}
zhpgvx com putes selected eigenvalues and, optionally, eigenvectors of a com plex generalized \(H\) erm itian-definite eigenproblem, of the form \(A * x=(\operatorname{lam} . b d a) * B * x, A * B x=(\operatorname{lam} b d a) * x\), or \(B * A * x=(l a m\) boda \() * x\). \(H\) ere \(A\) and \(B\) are assum ed to be \(H\) erm itian, stored in packed form \(a t\), and \(B\) is also positive definite.
E igenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

\section*{ARGUMENTS}

ITYPE (input)
Specifies the problem type to be solved:
\(=1: A{ }_{\mathrm{X}}=\left(\operatorname{lam}\right.\) bda) \({ }^{*} \mathrm{~B}{ }^{*} \mathrm{X}\)
\(=2: A * B *_{x}=\left(\operatorname{lam}\right.\) bda) \({ }^{*} \mathrm{x}\)
\(=3: B * A{ }^{*} \mathrm{X}=(\operatorname{lam} \mathrm{bda}){ }^{\star} \mathrm{x}\)

\section*{JO BZ (input)}
\(=\mathrm{N}^{\prime}:\) C om pute eigenvalues only;
\(=\mathrm{V}^{\prime}:\) C om pute eigenvalues and eigenvectors.

RANGE (input)
= A ': alleigenvalues w illbe found;
\(=V\) : alleigenvalues in the half-open interval
( \(\mathrm{L}, \mathrm{VU}] \mathrm{w}\) ill be found; = I': the \(I \mathrm{l}\)-th through
\(\mathbb{I U}\)-th eigenvalues w ill.be found.
UPLO (input)
\(=U\) ': U ppertriangles of \(A\) and \(B\) are stored;
\(=\mathbb{L}\) ': Low ertriangles of \(A\) and \(B\) are stored.

N (input) The order of the m atriges A and \(\mathrm{B} . \mathrm{N}>=0\).

AP (input/output)
O n entry, the upper or low er triangle of the H er\(m\) itian \(m\) atrix A, packed colum nw ise in a linear amay. The \(j\) th colum n of \(A\) is stored in the
anay AP as follow s: ifUPLO = U',AP (i+ (j
 \(\left.+(j-1)^{\star}(2 \star n-j) / 2\right)=A(i, j)\) for \(j=i<=n\).

On exit, the contents ofAP are destroyed.

BP (input/output)
O n entry, the upper or low ertriangle of the H er\(m\) titian \(m\) atrix \(B\), packed colum nw ise in a linear aray. The \(j\) th colum \(n\) of \(B\) is stored in the anay BP as follows: if UPLO \(=\mathrm{U}\) ', BP (i+ \((\ddagger\) \(1) \star j 2)=B(i, \gamma)\) for \(1<=i<=j ;\) if \(U P L O=L ', B P(i\) \(+(j-1) \star(2 * n-j / 2)=B(i, j)\) for \(j<=i<=n\).

On exit, the triangular factor U or L from the Cholesky factorization \(\mathrm{B}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{B}=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\), in the sam e storage form at as B.

VL (input)
IfRANGE=V', the low er and upper bounds of the interval to be searched foreigenvalues. \(\mathrm{VL}<\mathrm{VU}\).
N otreferenced ifRANGE=A 'or 'I'.

VU (input)
IfRANGE=V ', the low erand upper bounds of the interval to be searched foreigenvalues. VL < VU . N ot referenced ifRANGE=A'or I'.

II (input)
If RA NGE= 'I', the indioes (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{L}<=\mathbb{U}<=N\), if \(\mathrm{N}>0\); \(\mathbb{L}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N ot referenced ifRANGE= A 'or V'.
IU (input)
IfRA N G E= 'I', the indices (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{L}<=\mathbb{U}<=N\), if \(N>0\); \(\mathbb{L}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N otreferenced ifRANGE= A 'or V'.

ABSTOL (input)
The absolute error tolerance for the eigenvalues. A \(n\) approxim ate eigenvalue is accepted as converged when it is determ ined to lie in an interval [a,b] of w idth less than orequal to

ABSTOL + EPS * max ( \(\mid\) |, \(\mid\) |),
where EPS is them achine precision. IfA BSTOL is less than or equalto zero, then EPS* \(\mid\) | w illbe used in its place, w here \(T\) | is the 1 -norm of the tridiagonalm atrix obtained by reducing A \(P\) to tridiagonal form .

E igenvalues w illbe com puted m ostaccurately when ABSTOL is set to tw ige the underflow threshold 2*SLAM CH ( S ), not zero. If this routine retums w ith \(\mathbb{N}\) FO \(>0\), indicating that som e eigenvectors did not converge, try setting ABSTO L to \(2 \star\) SLAM CH (S ).

M (output)
The total num berofeigenvalues found. \(0<=\mathrm{M}<=\) N . IfRANGE = \(\mathrm{A}^{\prime}, \mathrm{M}=\mathrm{N}\), and ifRANGE= \(\mathrm{I}^{\prime}, \mathrm{M}=\) \(\mathbb{U}-\mathbb{L}+1\).

W (output)
O n norm alexit, the firstM elem ents contain the selected eigenvalues in ascending order.

Z (input) If \(\mathrm{JO} \mathrm{BZ}=\mathrm{N}\) ', then Z is not referenced. If OOB Z \(=V '\), then if \(\mathbb{N} F O=0\), the firstM colum ns of \(Z\)
contain the orthonorm aleigenvectors of the \(m\) atrix

A comesponding to the selected eigenvalues, with the i-th colum \(n\) of \(Z\) holding the eigenvector associated w ith W (i). The eigenvectors are norm alized as follow s: if ITYPE \(=1\) or \(2, \mathrm{Z} * * \mathrm{H} * \mathrm{~B} * \mathrm{Z}=\mathrm{I}\); if \(\operatorname{ITYPE}=3, Z * * H * \operatorname{inv}(B) * Z=I\).

If an eigenvector fails to converge, then that colum \(n\) of \(Z\) contains the latestapproxim ation to the eigenvector, and the index of the eigenvector is retumed in \(\mathbb{F} A \mathbb{I}\). N ote: the userm ustensure that at leastm ax \((1, M)\) colum ns are supplied in the array \(Z\); ifRANGE = \(V\) ', the exactvalue of \(M\) is notknow \(n\) in advance and an upper bound \(m\) ust be used.

\section*{LD \(Z\) (input)}

The leading dim ension of the array \(\mathrm{Z} . \mathrm{LD} \mathrm{Z}>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD \(Z>=\mathrm{max}(1, \mathrm{~N})\).

W ORK (w orkspace)
dim ension \((2 * N)\)
RW ORK (w orkspace)
dim ension ( \(7 * \mathrm{~N}\) )
IN ORK (w orkspace)
dim ension ( \(5 * \mathrm{~N}\) )
FAII (output)
If \(\mathrm{JOBZ}=\mathrm{V}\) ', then if \(\mathbb{N F O}=0\), the firstM ele\(m\) ents of \(\mathbb{F A} \mathbb{I}\) are zero. If \(\mathbb{N} F O>0\), then \(\mathbb{F A} \mathbb{I}\) contains the indices of the eigenvectors that failed to converge. If \(J O B Z=N\) ', then \(\mathbb{F A} I\) is not referenced.
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO = -i, the i-th argum enthad an illegalvalue
> 0: CPPTRF orCHPEVX retumed an emorcode:
\(<=\mathrm{N}:\) if \(\mathbb{N} F O=\mathrm{i}, \mathrm{CH}\) PEVX failed to converge; i eigenvectors failed to converge. Their indices are stored in array \(\mathbb{F A} \mathbb{I} .>N\) : if \(\mathbb{N} F O=N+\) \(i\), for \(1<=i<=n\), then the leading \(m\) inorof orderiofB is notpositive definite. The factorization of B could not be com pleted and no eigenvalues or eigenvectors w ere com puted.

\section*{FURTHER DETAILS}

B ased on contributions by
M ark Fahey, D epartm entofM athem atics, U niv . of K entucky, USA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhpm v -perform the m atrix-vector operation \(\mathrm{y}:=\) alpha* \(\mathrm{A} * \mathrm{x}\) + beta*y

\section*{SYNOPSIS}
```

SUBROUTINE ZHPMV(UPLO,N,ALPHA,A,X, INCX,BETA,Y, INCY)
CHARACTER * 1 UPLO
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (*),X (*),Y (*)

```

```

SU BROUT\mathbb{NE ZHPM V_64 (UPLO ,N,ALPHA,A,X, INCX,BETA,Y, INCY)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (*),X (*),Y (*)
INTEGER*8N,\mathbb{NCX,INCY}

```

\section*{F95 INTERFACE}


```

CHARACTER (LEN=1)::UPLO
COM PLEX (8) ::ALPHA,BETA
COM PLEX (8),D IM ENSION (:) ::A,X,Y
\mathbb{NTEGER ::N,}\mathbb{NNCX,}\mathbb{N}CY

```

```

CHARACTER (LEN=1) ::UPLO
COM PLEX (8) ::ALPHA,BETA
COMPLEX (8),D IM ENSION (:) ::A,X,Y

```
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{I N C X}, \mathbb{N} C Y\)

\section*{C INTERFACE}
\#include <sunperfh>
void zhpm v (charuplo, int n, doublecom plex *alpha, doublecomplex *a, doublecom plex *x, intincx, doublecom plex *beta, doublecom plex *y, intincy);
void zhpm v_64 (char uplo, long n, doublecom plex *alpha, doublecom plex *a, doublecom plex *x, long incx, doublecom plex *beta, doublecom plex *y, long incy);

\section*{PURPOSE}
zhpm \(v\) perform \(s\) the \(m\) atrix-vector operation \(y:=a l p h a * A * x+\) beta* \(y\) where alpha and beta are scalars, \(x\) and \(y\) are \(n\) ele\(m\) entvectors and \(A\) is an \(n\) by \(n\) hem itian \(m\) atrix, supplied in packed form.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangularpart of the \(m\) atrix A is supplied in the packed array A as follow s:

UPLO = U'or L ' The uppertriangularpartofA is supplied in \(A\).

UPLO = L'or 1' The low ertriangularpartofA is supplied in A.

U nchanged on exit.

N (input)
On entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
\(\left(\left(\mathrm{n}^{*}(\mathrm{n}+1)\right) / 2\right)\). Before entry w ith UPLO \(=\) \(U\) ' or G ', the anay A m ustcontain the upper triangularpart of the herm itian matrix packed
sequentially, column by colum \(n\), so that A (1) containsa(1,1), A (2) and A (3) contain a( 1,2 ) and a( 2,2 ) respectively, and so on. Before entry w ith UPLO = L'or I', the array A m ust contain the low er triangularpart of the her\(m\) tiian matrix packed sequentially, column by colum n, so thatA (1) contains a (1,1), A (2) and \(A(3)\) contain a \((2,1)\) and a \((3,1)\) respectively, and so on. N ote that the im aginary parts of the diagonalelem ents need notbe set and are assum ed to be zero. U nchanged on exit.

X (input)
\((1+(n-1) * a b s(\mathbb{N} C X))\). Before entry, the increm ented array \(X \mathrm{~m}\) ust contain the n elem ent vectorx. U nchanged on exit.
\(\mathbb{N} C X\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N C X}\) <> 0 . U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen
BETA is supplied as zero then \(Y\) need notbe seton input. U nchanged on exit.

Y (input/output)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) must contain the \(n\) elem ent vectory. On exit, Y is overw rilten by the updated vectory.
\(\mathbb{N} C Y\) (input)
O n entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents ofY. \(\mathbb{N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhpr-perform the herm tian rank 1 operation \(\mathrm{A}:=\) alpha* \(x^{*}\) con \(j\left(x^{\prime}\right)+A\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHPR (UPLO,N,ALPHA,X,NNCX,A)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX X (*),A (*)
\mathbb{NTEGER N,}\mathbb{N}CX
DOUBLE PRECISION ALPHA
SUBROUTINE ZHPR_64(UPLO,N,ALPHA,X, INCX,A)
CHARACTER * 1 UPLO
DOUBLE COM PLEX X (*),A (*)
INTEGER*8N,\mathbb{NCX}
DOUBLE PRECISION ALPHA

```

\section*{F95 INTERFACE}
```

SU BROUTINE HPR (UPLO, $\mathbb{N}], A L P H A, X,[\mathbb{N} C X], A)$
CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D IM ENSION (:) ::X,A
$\mathbb{N} T E G E R:: N, \mathbb{N} C X$
REAL (8) :: A LPHA
SU BROUTINE HPR_64 (UPLO, $\mathbb{N}], A L P H A, X,[\mathbb{N} C X], A)$
CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D $\mathbb{M}$ ENSION (:) ::X,A
$\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{N C X}$

```

\section*{C INTERFACE}
\#include <sunperfh>
void zhpr(charuple, intn, double alpha, doublecom plex *x, int incx, doublecom plex *a);
void zhpr_64 (charuple, long n, double alpha, doublecom plex *x, long incx, doublecom plex *a);

\section*{PURPOSE}
zhpr perform s the herm titian rank 1 operation \(A:=\) alpha* \(x^{*}\) con \(\dot{g}\left(x^{\prime}\right)+A\) where alpha is a realscalar, \(x\) is an \(n\) elem entvector and \(A\) is an \(n\) by \(n\) herm itian \(m\) atrix, supplied in packed form .

\section*{ARGUMENTS}

UPLO (input)
Onentry,UPLO specifies whether the upper or low er triangular part of the \(m\) atrix A is supplied in the packed array A as follow s:

UPLO = U'or U ' The uppertriangularpartof \(A\) is supplied in A.

UPLO = L'or 1' The low ertriangularpart of A is supplied in A.

U nchanged on exit.

N (input)
O n entry, N specifies the order of the m atrix A . \(\mathrm{N}>=0\). U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
\((1+(n-1) * a b s(\mathbb{N} C X))\). Before entry, the increm ented array X must contain the n elem ent vectorx. U nchanged on exit.

On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X<>0\). U nchanged on exit.

A (input/output)
\(\left(\left(n^{*}(n+1)\right) / 2\right)\). Before entry \(w\) ith UPLO \(=\) U' or L', the array A mustcontain the upper triangularpart of the herm itian matrix packed sequentially, column by colum n, so thatA (1) containsa(1,1), A (2) and A (3) contain a( 1,2 ) and a ( 2,2 ) respectively, and so on. On exit, the amay A is overw ritten by the uppertriangular partof the updated \(m\) atrix. Before entry w ith UPLO = L 'or I', the array A m ust contain the low er triangularpart of the herm itian \(m\) atrix packed sequentially, colum \(n\) by collm \(n\), so that A ( 1 ) contains a ( 1,1 ), A (2) and A (3) contain \(a(2,1)\) and \(a(3,1)\) respectively, and so on. On exit, the array A is overw rilten by the low er triangularpart of the updated \(m\) atrix. N ote that the im aginary parts of the diagonalelem ents need notbe set, they are assum ed to be zero, and on exit they are set to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhpr2 -perform the Herm tian rank 2 operation \(A:=\) alpha*x*conjg( \(y^{\prime}\) ) + conjg (alpha ) \({ }^{\star} y^{\star}\) conjg \(\left(x^{\prime}\right)+\) A

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX ALPHA
DOUBLE COM PLEX X (*),Y (*),AP (*)

```


```

CHARACTER * 1 UPLO
DOUBLE COM PLEX ALPHA
DOUBLE COM PLEX X (*),Y (*),AP (*)
INTEGER*8N,\mathbb{NCX,INCY}

```

\section*{F95 INTERFACE}
```

SU BROUTINE HPR2 (UPLO, $\mathbb{N}], A L P H A, X,[\mathbb{N} C X], Y,[\mathbb{N C Y}], A P)$
CHARACTER (LEN=1) ::UPLO
COM PLEX (8) ::ALPHA
COMPLEX (8),D $\mathbb{M}$ ENSION (:) ::X,Y,AP
$\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N C Y}$
SU BROUTINE HPR2_64 (UPLO, $\mathbb{N}], A L P H A, X,[\mathbb{N} C X], Y,[\mathbb{N} C Y], A P)$
CHARACTER (LEN=1) ::UPLO
COM PLEX (8) ::ALPHA
COM PLEX (8),D $\mathbb{M}$ ENSION (:) ::X,Y,AP

```
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{I N C X}, \mathbb{N} C Y\)

\section*{C INTERFACE}
\#include <sunperfh>
void zhpr2 (char uplo, int n, doublecom plex *alpha, doublecom plex *x, int incx, doublecom plex *y, int incy, doublecom plex *ap);
void zhpr2_64 (charuplo, long n, doublecom plex *alpha, doublecom plex *x, long incx, doublecom plex *y, long incy, doublecom plex *ap);

\section*{PURPOSE}
zhpr2 performs the Herm tian rank 2 operation \(A:=\) alpha*x*conjg ( \(\mathrm{y}^{\prime}\) ) + con jg (alpha ) \(\mathrm{y}^{\star}\) con \(\dot{g}\left(\mathrm{x}^{\prime}\right)+\mathrm{A}\) where alpha is a scalar, \(x\) and \(y\) are \(n\) elem ent vectors and \(A\) is an \(n\) by \(n\) herm itian \(m\) atrix, supplied in packed form .

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the upper or low er triangularpart of the \(m\) atrix A is supplied in the packed anray AP as follow s:

UPLO = U 'or L ' The uppertriangularpartofA is supplied in AP .

UPLO = L'or '1' The low ertriangularpart of A is supplied in AP .

U nchanged on exit.

N (input)
O n entry, N specifies the order of the m atrix A . \(\mathrm{N}>=0\). U nchanged on exit.

A LPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

X (input)
\((1+(n-1) * a b s(\mathbb{N} C X))\). Before entry, the increm ented array X m ust contain the n elem ent vectorx. Unchanged on exit.
\(\mathbb{I N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents ofX. \(\mathbb{N} C X\) <> 0 . U nchanged on exit.

Y (input)
\((1+(n-1) * a b s(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y m\) ust contain the \(n\) elem ent vectory. U nchanged on exit.
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents ofY. \(\mathbb{N C Y}\) <> 0 . U nchanged on exit.

\section*{AP (input/output)}
( ( \(\mathrm{n} *(\mathrm{n}+1)) / 2\) ). Before entry w ith UPLO \(=\) \(U\) ' or U ', the array A P m ust contain the upper triangularpartof the hem tian \(m\) atrix packed sequentially, colum \(n\) by colum \(n\), so thatA \(P(1)\) contains a (1,1), AP (2) and AP (3) contain a ( 1,2 ) and a (2,2) respectively, and so on. On exit, the array AP is overw ritten by the upper triangular part of the updated \(m\) atrix. Before entry w ith UPLO = L'or 1 ', the aray AP must contain the low er triangular part of the herm itian \(m\) atrix packed sequentially, colum \(n\) by colum \(n\), so thatAP (1) contains a (1, 1), AP (2) and AP (3 ) contain a ( 2,1 ) and a ( 3,1 ) respectively, and so on.On exit, the array AP is overw rilten by the low er triangular part of the updated \(m\) atrix. \(N\) ote that the im aginary parts of the diagonalele\(m\) ents need notbe set, they are assum ed to be zero, and on exit they are set to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}

> zhprfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is H erm itian indefinite and packed, and provides emorbounds and backw ard enror estim ates for the solution

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHPRFS (UPLO,N,NRHS,A,AF,\mathbb{PIVOT,B,LDB,X,LDX,FERR,}}\mathbf{N},\textrm{N},\textrm{N}
BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEXA (*),AF (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGER N,NRHS,LDB,LDX,INFO
\mathbb{NTEGER \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)
SU BROUT\mathbb{NE ZHPRFS_64 (UPLO,N,NRHS,A,AF, PPIVOT,B,LDB,X,LDX,}
FERR,BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*),AF (*),B (LD B ,*),X (LDX,*),W ORK (*)
INTEGER*8N,NRHS,LDB,LDX,INFO
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE HPRFS (UPLO,N, NRHS],A,AF, \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LD} \mathrm{B}], \mathrm{X},[\mathrm{LDX}]\), FERR, BERR, [W ORK], [W ORK 2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::A,AF,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : ) :: B, X
\(\mathbb{N}\) TEGER : \(: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} \mathrm{B}, \mathrm{LD} \mathrm{X}, \mathbb{N} F \mathrm{O}\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{I} O T\)
REAL (8),D \(\mathbb{M}\) ENSION (:) :: FERR,BERR,W ORK2

SU BROUT \(\mathbb{N} E\) HPRFS_64 (UPLO, N, \(\mathbb{N} R H S], A, A F, \mathbb{P} \mathbb{I V O T}, B,[L D B], X\), [ \(L\) D X ], FERR, BERR, [WORK], [WORK2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::UPLO
COM PLEX (8), D \(\mathbb{I M} E N S I O N\) (:) ::A,AF,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : B , X
\(\mathbb{N}\) TEGER (8) ::N , NRHS,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL (8), D \(\mathbb{I M} E N S I O N(:):: F E R R, B E R R, W\) ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zhprfs (charuple, intn, int nhs, doublecom plex *a, doublecom plex *af, int*ipivot, doublecom plex *b, int ldb, doublecom plex *x, int ldx, double *ferr, double *berr, int *info);
void zhprfs_64 (charuplo, long n, long nrhs, doublecom plex
*a, doublecom plex *af, long *ipivot, doublecom plex
*b, long ldb, doublecom plex *x, long ldx, double
* ferr, double *berr, long *info);

\section*{PURPOSE}
zhprfs im proves the com puted solution to a system of linear equations w hen the coefficientm atrix is \(H\) erm itian indefintie and packed, and provides errorbounds and backw ard error estim ates for the solution.

\section*{ARGUMENTS}

UPLO (input)
\(=U\) ': U pper triangle of A is stored;
= LL': Low ertriangle ofA is stored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber
of colum ns of the m atrioes B and X. NRHS >=0.

A (input) The upper or low er triangle of the \(H\) erm titian \(m\) atrix \(A\), packed colum nw ise in a linear array.

The jth column of A is stored in the array A as follow s: if UPLO \(=U^{\prime}, A(i+(j-1) * j 2)=A(i, 7)\) for \(1<=i<=j\) if \(\mathrm{UPLO}=\mathrm{L}\) ', \(\mathrm{A}(i+(j-1) *(2 * \mathrm{n}-\mathrm{j} / 2)\) \(=A(i, j)\) for \(j=i<=n\).

\section*{AF (input)}

The factored form of them atrix A. AF contains the block diagonal matrix D and themultipliers used to obtain the factor \(U\) orL from the factorization \(A=U * D * U * * H\) orA \(=L * D * L * * H\) as com puted by CHPTRF, stored as a packed triangularm atrix.
\(\mathbb{P I V O T}\) (input)
D etails of the interchanges and the block structure ofD as determ ined by CHPTRF.
\(B\) (input) The righthand side m atrix \(B\).
LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by CHPTRS. On exit, the im proved solution \(m\) atrix \(X\).

\section*{LD X (input)}

The leading dim ension of the array X . LD X >= \(\max (1, N)\).

\section*{FERR (output)}

The estim ated forw ard errorbound for each solution vector \(X(\mathcal{)}\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution comesponding to \(X(\mathcal{O})\), FERR ( \()\) ) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{i})-X\) TRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{)}\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each
solution vector \(X\) ( \(j\) ) (ie., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the \(i\)-th argum ent had an illegal value

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zhpsv - com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*),B (LDB,*)
\mathbb{N TEGER N,NRHS,LDB, INFO}
INTEGER \mathbb{PIVOT (*)}

```

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*),B (LDB,*)
INTEGER*8 N,NRHS,LDB, IN FO
INTEGER *8 \mathbb{P IVOT (*)}

```

\section*{F95 INTERFACE}

SU BROUTINE HPSV (UPLO ,N, \(\mathbb{N} R H S], A, \mathbb{P} \mathbb{I V} O T, B,[L D B],[\mathbb{N} F O])\)
CHARACTER (LEN=1)::UPLO
COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::A
COM PLEX (8), D IM ENSION (:,:) ::B
\(\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT}\)
SUBROUTINE HPSV_64 (UPLO,N, \(\mathbb{N} R H S], A, \mathbb{P} \mathbb{I V O T}, \mathrm{~B},[\mathrm{LDB}],[\mathbb{N} F O])\)
CHARACTER (LEN=1)::UPLO

COMPLEX (8), D \(\mathbb{M}\) ENSION (:) ::A
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::B
\(\mathbb{N}\) TEGER (8) :: N,NRHS,LD B, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zhpsv (charuplo, intn, intnrhs, doublecom plex *a, int *ipivot, doublecom plex *b, int ldb, int *info);
void zhpsv_64 (char uplo, long n, long nrhs, doublecom plex
*a, long *ìipivot, doublecom plex *b, long ldb, long
*info);

\section*{PURPOSE}
zhpsv com putes the solution to a com plex system of linear equations
\(\mathrm{A} * \mathrm{X}=\mathrm{B}, \mathrm{w}\) here A is an N -by -N H erm itian m atrix stored in packed form at and X and B are N -by-N R H S m atrices.

The diagonal pivoting \(m\) ethod is used to factorA as
\[
A=U * D * U * * H \text {, if } U P L O=U \text { ', or }
\]
\[
\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}, \text { if } \mathrm{UPLO}=\mathrm{L},
\]
where \(U\) (orL) is a productof perm utation and unit upper (low er) triangularm atrices, D is H erm itian and block diagonalw ith 1-by-1 and 2-by-2 diagonal blocks. The factored form ofA is then used to solve the system of equations A * \(X=B\).

\section*{ARGUMENTS}

UPLO (input)
= U ': U pper triangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The num ber of linearequations, ie., the order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS \(>=0\).

A (input/output)
On entry, the upper or low ertriangle of the Her \(m\) tian \(m\) atrix A, packed collm nw ise in a linear
array. The jth column of A is stored in the array A as follows: if UPLO = U',A (i+ (j)
 \((j-1) *(2 n-j / 2)=A(i, j)\) for \(j=i<=n\). See below for furtherdetails.

On exit, the block diagonalm atrix \(D\) and the \(m u l\) tipliers used to obtain the factor \(U\) orL from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\) as com puted by CHPTRF, stored as a packed triangular \(m\) atrix in the sam e storage form atasA.

IPIVOT (output)
D etails of the interchanges and the block structure ofD, as determ ined by CHPTRF. If \(\mathbb{P} \mathbb{I V}\) OT (k) \(>0\), then row sand colum nsk and \(\mathbb{P} \mathbb{V} O T(k)\) were interchanged, and \(\mathrm{D}(\mathrm{k}, \mathrm{k})\) is a 1 -by- 1 diagonal block. If UPLO \(=\mathrm{U}\) 'and \(\mathbb{P} \mathbb{I V O T}(\mathrm{k})=\mathbb{P} \mathbb{V} O T(\mathrm{k}-1)\) \(<0\), then row \(s\) and colum ns \(k-1\) and \(-\mathbb{P}\) IV O T (k) w ere interchanged and \(D(k-1 * k, k-1 k)\) is a 2 -by-2 diagonal block. If UPLO \(=\) L' and \(\mathbb{P} \mathbb{V} O T(k)=\) \(\operatorname{PIVOT}(k+1)<0\), then row s and colum ns \(k+1\) and \(-\mathbb{P}\) IV O T ( \(k\) ) w ere interchanged and \(D(k: k+1, k \cdot k+1)\) is a 2 -by-2 diagonalblock.

B (input/output)
On entry, the \(\mathrm{N}-\mathrm{by}-\mathrm{NRH} \mathrm{S}\) righthand side m atrix B . On exit, if \(\mathbb{N F O}=0\), the \(N\) by \(-\mathbb{N}\) RH \(S\) solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, \mathbb{N})\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{D}(\mathrm{i}, \mathrm{i})\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, so the solution could notbe com puted.

\section*{FURTHER DETAILS}

The packed storage schem e is illustrated by the follow ing exam plew hen \(\mathrm{N}=4, \mathrm{UPLO}=\mathrm{U}\) ':

Tw o-dim ensional storage of the H erm itian m atrix A :
a11 a12 a13 a14
a22 a23 a24
a33 a34 (aij=cong \(\bar{g}(a \ddot{j}))\)
a44

Packed storage of the upper triangle of A :
\(A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]\)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zhpsvx - use the diagonal pivoting factorization \(\mathrm{A}=\) \(U * D * U * * H\) or \(A=L * D * L * * H\) to com pute the solution to a com plex system of linearequations \(A * X=B\), where \(A\) is an \(N\) by -N H erm tian \(m\) atrix stored in packed form at and \(X\) and \(B\) are N -by-N R H S m atrices

\section*{SYNOPSIS}

```

    RCOND,FERR,BERR,W ORK,W ORK2, INFO)
    CHARACTER * 1FACT,UPLO
DOUBLE COM PLEX A (*),AF (*),B (LD B ,*),X (LDX ,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,INFO
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION RCOND
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)
SUBROUT\mathbb{NE ZHPSVX_64(FACT,UPLO,N,NRHS,A,AF, \mathbb{PIVOT,B,LDB,X,}}\mathbf{N},
LDX,RCOND,FERR,BERR,WORK,W ORK 2, INFO)
CHARACTER * 1FACT,UPLO
DOUBLE COM PLEXA (*),AF (*),B (LDB,*),X (LDX,*),W ORK (*)
\mathbb{NTEGER*8N,NRHS,LDB,LDX,INFO}
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION RCOND
D OUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE HPSVX (FACT,UPLO,N, \(\mathbb{N} R H S], A, A F, \mathbb{P} \mathbb{V} O T, B,[L D B], X\), [LDX],RCOND,FERR,BERR, [W ORK], [W ORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::FACT, UPLO
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::A, AF,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:): : B, X
\(\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8) ::RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

SUBROUTINE HPSVX_64 (FACT, UPLO,N, \(\mathbb{N} R H S], A, A F, \mathbb{P} \mathbb{I} O T, B,[L D B]\), \(X,[L D X], R C O N D, F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N F O}])\)

CHARACTER (LEN=1): :FACT, UPLO
COMPLEX (8), D \(\mathbb{M} E N S I O N(:):: A, A F, W\) ORK
COM PLEX (8), D IM ENSION (: ::) ::B,X
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION(:):: \(\mathbb{P} \mathbb{I V O T}\)
REAL (8) ::RCOND
REAL (8), D \(\mathbb{M} E N S I O N(:):: F E R R, B E R R, W\) ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void zhpsvx (char fact, charuplo, int n, int nrhs, doublecom plex *a, doublecom plex *af, int *ípivot, doublecom plex *b, intldb, doublecom plex *x, int ldx, double *roond, double *ferr, double *berr, int*info);
void zhpsvx_64 (char fact, char uplo, long n, long nihs, doublecom plex *a, doublecom plex *af, long *ipivot, doublecom plex *b, long ldb, doublecom plex *x, long \(l d x\), double *roond, double *ferr, double *berr, long *info);

\section*{PURPOSE}
zhpsvx uses the diagonalpivoting factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\) or \(A=L * D * L * * H\) to com pute the solution to a com plex system of linear equations \(A * X=B\), where \(A\) is an \(N\) boy \(N\) H erm itian \(m\) atrix stored in packed form atand \(X\) and \(B\) are \(N\)-byN R H S m atrices.

Emorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT \(=N\) ', the diagonalpivoting \(m\) ethod is used to
factorA as
\(A=U * D * U * * H\), if \(U P L O=U\) ', or
\(A=L * D * L * * H\), if \(U P L O=L \prime\),
where \(U\) (orL) is a productofperm utation and unitupper (low er)
triangularm atrioes and D is H erm itian and block diagonal w ith

1 -by-1 and 2 -by-2 diagonalblocks.
2. If som eD \((i, i)=0\), so thatD is exactly singular, then the routine
retums \(w\) ith \(\mathbb{N N F O}=\) i. O therw ise, the factored form of \(A\) is used
to estim ate the condition num ber of the m atrix A. If the reciprocal of the condition num ber is less than m achine precision,
\(\mathbb{N} F O=\mathrm{N}+1\) is retumed as a waming, but the routine stillgoes on
to solve for \(X\) and compute error bounds as described below.
3. The system ofequations is solved forX using the factored form of A.
4. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for 五.

\section*{ARGUMENTS}

\section*{FACT (input)}

Specifies w hether ornot the factored form of \(A\) has been supplied on entry. = \(\mathrm{F}^{\prime}\) : On entry, AF and IP IV O T contain the factored form of A. AF and \(\mathbb{P}\) IV O T w illnotbe m odified. \(=\mathrm{N}\) : Them atrix A w illbe copied to A F and factored.
```

UPLO (input)

```
\(=\mathrm{U}\) ': Uppertriangle of \(A\) is stored;
\(=\mathrm{L}\) ': Low er triangle of A is stored.

N (input) The num ber of linearequations, ie., the order of them atrix A. N >=0.

NRHS (input)

The num ber of righthand sides, ie., the num ber of collm ns of the \(m\) atrioes B and X. NRH S >=0.

A (input) The upper or low er triangle of the H erm itian \(m\) atrix \(A\), packed colum nw ise in a lineararray. The jth column of A is stored in the array A as follows: if UPLO \(=U\) ', A \((i+(j-1) * j 2)=A(i, 7)\) for \(1<=i<=j\) if \(U P L O=L ', A(i+(j-1) *(2 * n-7 / 2)\) \(=A(i, j)\) for \(j=i<=n\). See below forfurther details.

AF (input/output)
If \(F A C T=F '\), then \(A F\) is an input argum ent and on entry contains the block diagonalm atrix \(D\) and the m ultipliers used to obtain the factor U orL from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\) as com puted by CHPTRF, stored as a packed triangular \(m\) atrix in the sam e storage form atas A.

IfFACT \(=N\) ', then AF is an output argum ent and on exit contains the block diagonalm atrix \(D\) and the m ultipliers used to obtain the factorU or L from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\) or \(\mathrm{A}=\) L*D *L**H as com puted by CHPTRF, stored as a packed triangularm atrix in the sam e storage form at as A.

IPIVOT (inputoroutput)
IfFACT = \(\mathrm{F}^{\prime}\), then \(\mathbb{P I V O T}\) is an input argum ent and on entry contains details of the interchanges and the block structure of \(D\), as determ ined by CHPTRF. If \(\mathbb{P}\) IVOT \((k)>0\), then row s and colum nsk and \(\mathbb{P} \operatorname{IVOT}(k)\) w ere interchanged and \(D(k, k)\) is a 1 -by-1 diagonal block. If UPLO \(=U^{\prime}\) and \(\mathbb{P} \mathbb{V} O T(k)=\mathbb{P} \mathbb{I V} O T(k-1)<0\), then row sand colum ns \(\mathrm{k}-1\) and \(-\mathbb{P} \mathbb{I V O T}(\mathrm{k})\) were interchanged and \(\mathrm{D}(\mathrm{k}-\) \(1 k, k-1 k)\) is a \(2-b y-2\) diagonalblock. IfU PLO \(=\) L 'and \(\mathbb{P} \mathbb{I V} \circ \mathrm{T}(\mathrm{k})=\mathbb{P} \mathbb{I V} \circ \mathrm{T}(k+1)<0\), then row s and colum nsk+1 and - \(\mathbb{P}\) IV OT (k) w ere interchanged and D \((k k+1, k k+1)\) is a 2 -by-2 diagonalblock.

IfFACT = N ', then \(\mathbb{P I V O T}\) is an output argum ent and on exit contains details of the interchanges and the block structure of D, as determ ined by CHPTRF.
\(B\) (input) The \(N\)-by-N R H S righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the anay B . LD B >= \(\max (1, N)\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=N+1\), the \(N\) by -N RH \(S\) solution
\(m\) atrix \(X\).

LD X (input)
The leading dim ension of the anay \(\mathrm{X} . \mathrm{LD} \mathrm{X}\) >= \(\max (1, N)\).

RCOND (output)
The estim ate of the reciprocal condition num ber of the \(m\) atrix \(A\). IfRCOND is less than them achine precision (in particular, if RCOND \(=0\) ), the \(m\) atrix is singular to working precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO > 0.

FERR (output)
The estim ated forw ard enrorbound for each solution vector \(X()\) (the \(j\) th column of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(\mathrm{X}(\mathcal{)}, \mathrm{FERR}(\mathcal{)})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{\nu})-X\) TRU \(E\) ) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

BERR (output)
The com ponentw ise relative backw ard error of each solution vector X ( \()\) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )

W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{I N F O}=-\) i, the \(i\)-th argum enthad an illegalvalue
> 0 : if \(\mathbb{N F O}=i\), and \(i\) is
<= N : D (i,i) is exactly zero. The factorization has been completed but the factorD is exactly singular, so the solution and error bounds could not be com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1\) : D is nonsingular, butRCOND is less than \(m\) achine
precision, \(m\) eaning that the \(m\) atrix is singular to w orking precision. N evertheless, the solution and error bounds are com puted because there are a num ber of situations w here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{FURTHER DETAILS}

The packed storage schem e is illustrated by the follow ing exam ple when \(N=4\), UPLO \(=U\) ':
Tw o-dim ensionalstorage of the \(H\) erm itian \(m\) atrix A :
```

a11 a12 a13 a14
a22 a23 a24
a33 a34 (aij= conig (aji))
a44

```

Packed storage of the upper triangle ofA :
\(A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]\)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zhptrd-reduce a com plex Herm itian matrix A stored in packed form to realsym \(m\) etric tridiagonal form \(T\) by a unitary sim ilarity transform ation

\section*{SYNOPSIS}
```

SUBROUTINE ZHPTRD(UPLO,N,AP,D,E,TAU,INFO)
CHARACTER * 1 UPLO
DOUBLE COM PLEX AP (*),TAU (*)
INTEGER N,\mathbb{NFO}
DOUBLE PRECISION D (*),E (*)
SUBROUT\mathbb{NE ZHPTRD_64(UPLO,N,AP,D,E,TAU, INFO)}
CHARACTER * 1 UPLO
D OUBLE COM PLEXAP (*),TAU (*)
INTEGER*8N,\mathbb{NFO}
DOUBLE PRECISIOND (*),E (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE HPTRD (UPLO,N,AP,D,E,TAU, [ $\mathbb{N} F O$ ])
CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D $\mathbb{I M}$ ENSION (:) ::AP,TAU
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
REAL (8), D IM ENSION (:) ::D , E
SU BROUTINE HPTRD_64 (UPLO,N,AP,D ,E,TAU, [ $\mathbb{N} F \mathrm{FO}$ ])
CHARACTER (LEN=1)::UPLO

```

\section*{C INTERFACE}
\#include <sunperfh>
void zhptrd (charuplo, intn, doublecom plex *ap, double *d, double *e, doublecom plex *tau, int *info);
void zhptrd_64 (charuplo, long n, doublecom plex *ap, double *d, double *e, doublecom plex *tau, long *info);

\section*{PURPOSE}
zhptrd reduces a com plex H erm itian m atrix A stored in packed form to realsym \(m\) etric tridiagonal form \(T\) by a unitary sim ilarity transform ation: \(\mathrm{Q} * * \mathrm{H} * \mathrm{~A} * \mathrm{Q}=\mathrm{T}\).

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}:\) : U pper triangle of A is stored;
= L': Low ertriangle ofA is stored.
\(N\) (input) The order of the matrix \(A . N>=0\).

AP (input)
O n entry, the upperor low ertriangle of the H er\(m\) tian \(m\) atrix \(A\), packed colum nw ise in a linear array. The jth colum \(n\) of \(A\) is stored in the array AP as follows: ifUPLO = U',AP (i+ (j) \(\left.1)^{\star} \mathfrak{j} 2\right)=A(i, \gamma)\) for \(1<=i<=j\) ifUPLO \(=L^{\prime}\) ', AP ( \(i\) \(+(j-1) \star(2 * n-7 / 2)=A(i, j)\) for \(j<=i<=n\). On exit, if \(U P L O=U\) ', the diagonal and first superdiagonalofA are overw ritten by the comesponding ele\(m\) ents of the tridiagonalm atrix \(T\), and the ele\(m\) ents above the first superdiagonal, \(w\) ith the array TAU, represent the unitary matrix \(Q\) as a product of elem entary reflectors; if UPLO = L' ', the diagonaland firstsubdiagonalofA are overw ritten by the comresponding elem ents of the tridiagonalm atrix \(T\), and the elem ents below the first subdiagonal, \(w\) ith the array TA \(U\), represent the unitary \(m\) atrix \(Q\) as a product of elem entary reflectors. See FurtherD etails.

D (output)
The diagonalelem ents of the tridiagonalm atrix T :
\(D(i)=A(i, i)\).

E (output)
The off-diagonal elem ents of the tridiagonal \(m\) atrix \(T: E(i)=A(i, i+1)\) if \(U P L O=U^{\prime}, E(i)=\) A \((i+1, i)\) if \(\mathrm{UPLO}=\mathrm{L}\).

TAU (output)
The scalar factors of the elem entary reflectors (see FurtherD etails) .
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the \(i-\) th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

IfU PLO \(=U\) ', the m atrix \(Q\) is represented as a product of elem entary reflectors
\[
Q=H(n-1) \ldots H(2) H(1)
\]

Each H (i) has the form
\(H(i)=I-\tan { }^{*} V^{*} V^{\prime}\)
\(w\) here tau is a com plex scalar, and \(v\) is a com plex vector w ith \(\mathrm{v}(\mathrm{i}+1 \mathrm{n})=0\) and \(\mathrm{v}(\mathrm{i})=1 ; \mathrm{v}(1: i-1)\) is stored on exit in \(A P\), overw riting \(A(1: i-1, i+1)\), and tau is stored in TAU (i).

If U PLO \(=\mathrm{L}\) ', them atrix Q is represented as a product of elem entary reflectors
\[
Q=H(1) H(2) \ldots H(n-1)
\]

Each H (i) has the form
\[
H(i)=I-\tan * v^{\star} v^{\prime}
\]
\(w\) here tau is a com plex scalar, and \(v\) is a com plex vector w ith \(\mathrm{v}(1: i)=0\) and \(v(i+1)=1 ; v(i+2 \mathrm{~m})\) is stored on exit in AP, overw riting A (i+2 \(\mathrm{n}, \mathrm{i}\) ), and tau is stored in TA U (i).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zhptrf-com pute the factorization of a com plex Herm tian packed \(m\) atrix A using the Bunch \(-K\) aufn an diagonalpívoting m ethod

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZHPTRF(UPLO,N,A,\mathbb{PIVOT, INFO)}}\mathbf{N}=()
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*)
\mathbb{NTEGER N,}\mathbb{N}FO
INTEGER \mathbb{PIVOT (*)}

```

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*)
INTEGER*8N,\mathbb{NFO}
\mathbb{NTEGER *8 \mathbb{PIVOT (*)}}\mathbf{(})

```

\section*{F95 INTERFACE}
```

SU BROUTINE HPTRF (UPLO ,N,A, $\mathbb{P} \mathbb{I V O T},[\mathbb{N} F O]$ )
CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D $\mathbb{I M}$ ENSION (:) ::A
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T$
SU BROUTINE HPTRF_64 (UPLO, N, A, $\mathbb{P} \mathbb{I V O T},[\mathbb{N} F O]$ )
CHARACTER (LEN=1)::UPLO

```

\section*{C INTERFACE}
\#include <sunperfh>
void zhptrf(charuplo, intn, doublecom plex *a, int *ipivot, int*info);
void zhptrf_64 (charuplo, long n, doublecom plex *a, long *ipivot, long *info);

\section*{PURPOSE}
zhptrf com putes the factorization of a com plex Herm itian packed \(m\) atrix A using the Bunch \(K\) aufn an diagonalpivoting \(m\) ethod:
\[
\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H} \text { or } \mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}
\]
where \(U\) (orL) is a product of perm utation and unit upper (low er) triangular \(m\) atrices, and \(D\) is \(H\) em itian and block diagonalw ith 1 -by-1 and 2 -by- 2 diagonalblocks.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) : U ppertriangle of A is stored;
= L': Low er triangle ofA is stored.
N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O \(n\) entry, the upper or low er triangle of the Her m tian m atrix \(A\), packed colum nw ise in a linear array. The jth column of A is stored in the array A as follows: if UPLO = U', A (i+ (j 1) \(* j 2\) ) \(=A(i, 7)\) for \(1<=i<=j\) ifUPLO \(=L '\) ' A ( \(i+\) \((j-1)^{*}(2 n-7 / 2)=A(i, 7)\) for \(j=i<=n\).

On exit, the block diagonalm atrix D and the multipliers used to obtain the factorU orL, stored as a packed triangularm atrix overw riting A (see below for further details).

D etails of the interchanges and the block structure of \(D\). If \(\mathbb{P I V O T}(k)>0\), then row sand columnsk and \(\mathbb{P I V O T}(k)\) were interchanged and \(D(k, k)\) is a \(1-b y-1\) diagonalblock. If \(U P L O=U '\) and \(\mathbb{P} \mathbb{I V O T}(k)=\mathbb{P} \mathbb{V} O T(k-1)<0\), then row \(s\) and colum ns \(k-1\) and - \(\mathbb{P I V O T}(k)\) were interchanged and D \((k-1 k, k-1 k)\) is a \(2-b y-2\) diagonal block. If UPLO \(=\mathrm{L}\) 'and \(\mathbb{P I V O T}(k)=\mathbb{P} \mathbb{I V O T}(k+1)<0\), then row s and colum nsk+1 and - \(\mathbb{P}\) IV OT (k) were interchanged and \(D(k, k+1, k \cdot k+1)\) is a 2 -by -2 diagonal block.
\(\mathbb{I N} F O\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an illegalvalue \(>0:\) if \(\mathbb{N} F O=i, D(i, i)\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, and division by zero w ill occur if it is used to solve a system ofequations.

\section*{FURTHER DETAILS}

\section*{5-96-B ased on m odifications by J.Lew is, Boeing C om puter} Services

Com pany
If \(U P L O=U\) ', then \(A=U * D * U\) ', where
\(U=P(n) \star U(n) * \ldots{ }^{\star}(k) U(k)^{\star} \ldots\),
i.e., \(U\) is a product of term \(S P(k) * U(k)\), where \(k\) decreases from \(n\) to 1 in steps of 1 or 2 , and \(D\) is ablock diagonal \(m\) atrix \(w\) ith 1 -by -1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I V O T}(k)\), and \(U(k)\) is a unituppertriangularm atrix, such that if the diagonal block \(D(k)\) is of orders ( \(s=1\) or 2 ), then
\[
\begin{aligned}
& \text { ( I v 0 ) k-s } \\
& U(k)=(0 \quad 1 \quad 0) s \\
& \text { ( } 00 \text { I ) n-k } \\
& \mathrm{k}-\mathrm{s} \mathrm{~s} \mathrm{n}-\mathrm{k}
\end{aligned}
\]

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(1 k-\) \(1, k\) ). If \(s=2\), the upper triangle ofD ( \(k\) ) overw rites \(A(k-\) \(1, k-1)\), A \((k-1, k)\), and \(A(k, k)\), and \(v\) overw rites A \((1 k-2, k-\) 1 k).

If \(U P L O=L\) ', then \(A=L * D * L\) ', where
\[
L=P(\mathbb{1}) \star L(1) \star \ldots * P(k) \star L(k)^{\star} \ldots,
\]
i.e., \(L\) is a product of term \(s P(k) * L(k)\), where \(k\) increases
from 1 to \(n\) in steps of 1 or 2, and \(D\) is a block diagonal \(m\) atrix \(w\) ith 1 -by- 1 and 2 -by-2 diagonalblocks \(D(k)\). \(P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I V} O T(k)\), and \(L(k)\) is a unit low ertriangularm atrix, such that if the diagonal
block D (k) is of orders ( \(s=1\) or2), then
```

    ( I 0 0 ) k-1
    L (k)=( 0 I 0 ) s
( 0 v I ) n-k-s+1
k-1 s n-k-s+1

```

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites A ( \(k+1 n, k\) ). If \(s=2\), the low ertriangle ofD ( \(k\) ) overw rites \(A(k, k), A(k+1, k)\), and \(A(k+1, k+1)\), and \(v\) overw rites A ( \(k+2 n, k k+1\) ).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhptri-com pute the inverse of a com plex H erm itian indefinthe \(m\) atrix \(A\) in packed storage using the factorization \(A=\) \(\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\) com puted by CHPTRF

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*),W ORK (*)
\mathbb{NTEGER N,}\mathbb{N}FO
INTEGER \mathbb{PIVOT (*)}
SU BROUT\mathbb{NE ZHPTRI_64(UPLO,N,A,\mathbb{PIVOT,W ORK,INFO)}}\mathbf{(N,N}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*),W ORK (*)
INTEGER*8N,\mathbb{NFO}
\mathbb{NTEGER *8 \mathbb{PIVOT (*)}}\mathbf{(})
F95 INTERFACE

```

```

    CHARACTER (LEN=1) ::UPLO
    COMPLEX (8),D IM ENSION (:) ::A ,W ORK
    \mathbb{NTEGER ::N,\mathbb{NFO}}0=0
    \mathbb{NTEGER,D IM ENSION (:) ::\mathbb{PIVOT}}\mathbf{T}\mathrm{ (:}
    SUBROUTINE HPTRI_64 (UPLO,N,A,\mathbb{PIVOT, [W ORK ], [NFO ])}
    CHARACTER (LEN=1) ::UPLO
    COM PLEX (8),D IM ENSION (:) ::A,W ORK
    ```
\(\mathbb{N}\) TEGER (8) ::N, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8),D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V}\) OT

\section*{C INTERFACE}
\#include < sunperfh>
void zhptri(charuplo, intn, doublecom plex *a, int *ipìivot, int*info);
void zhptri_ 64 (char uplo, long n, doublecom plex *a, long *ịívot, long *info);

\section*{PURPOSE}
zhptri computes the inverse of a complex H erm itian indefinite \(m\) atrix \(A\) in packed storage using the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\) com puted by CHPTRF.

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) ': U ppertriangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\);
= L ': Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D}\) * \(\mathrm{L} * * \mathrm{H}\).

N (input) The order of the matrix A. \(\mathrm{N}>=0\).
A (input/output)
On entry, the block diagonalm atrix D and the m ultipliers used to obtain the factorU orL as com puted by CHPTRF, stored as a packed triangular \(m\) atrix.

On exit, if \(\mathbb{N} F O=0\), the ( H em itian) inverse of the originalm atrix, stored as a packed triangular \(m\) atrix. The \(j\) th colum \(n\) of inv ( \(A\) ) is stored in the array A as follows: if UPLO = U',A (i+ (j 1) \(* j 2)=\operatorname{inv}(A)(i, \gamma)\) for \(1<=i<=j\); ifUPLO \(=L^{\prime}\), A \((i+(j-1) *(2 n-1) / 2)=\operatorname{inv}(A)(i, 7)\) for \(j<=i<=n\).
\(\mathbb{P I V O T}\) (input)
D etails of the interchanges and the block structure ofD as determ ined by CHPTRF.

W ORK (w orkspace)
dim ension (N)
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i, D(i, i)=0\); the \(m\) atrix is singular and its inverse could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

zhptrs-solve a system of linear equationsA *X = B w ith a
com plex H erm itian m atrix A stored in packed form at using the
factorization A = U *D *U**H or A = L*D *L**H computed by
CHPTRF

```

\section*{SYNOPSIS}

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*), B (LDB,*)
\(\mathbb{N} T E G E R N, N R H S, L D B, \mathbb{N} F O\)
\(\mathbb{I N} T E G E R \mathbb{P} \mathbb{I V} O T\left({ }^{*}\right)\)
SU BROUTINE ZHPTRS_64 (UPLO,N,NRHS,A, \(\mathbb{P} \mathbb{I V O T}, \mathrm{B}, \mathrm{LD} B, \mathbb{N} F O)\)
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*), B (LDB,*)
\(\mathbb{N}\) TEGER*8N,NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER*8 \(\mathbb{P} \mathbb{I V O T}\) ( \({ }^{*}\) )

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E\) HPTRS (UPLO ,N, \(\mathbb{N} R H S], A, \mathbb{P} \mathbb{I V} O T, B,[L D B],[\mathbb{N} F O])\)
CHARACTER (LEN=1)::UPLO
COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::A
COM PLEX (8), D IM ENSION (:,:) ::B
\(\mathbb{N}\) TEGER :: N, NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT}\)
SU BROUTINE HPTRS_64 (UPLO,N, \(\mathbb{N} R H S], A, \mathbb{P} \mathbb{I} O T, B,[L D B],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::A
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) :: B
\(\mathbb{N}\) TEGER (8) :: N, NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8),D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V}\) OT

\section*{C INTERFACE}
\#include <sunperfh>
void zhptrs (charuplo, intn, int nihs, doublecom plex *a, int *ipivot, doublecom plex *b, int ldb, int *info);
void zhptrs_64 (charuple, long n, long nrhs, doublecom plex *a, long *ípivot, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
zhptrs solves a system of linearequations A *X \(=\mathrm{B}\) with a com plex \(H\) erm itian \(m\) atrix A stored in packed form at using the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\) computed by CHPTRF.

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) ': U ppertriangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{H}\);
= L ': Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\).

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS >=0.

A (input) The block diagonalm atrix \(D\) and the \(m\) ultipliers used to obtain the factorU orL as com puted by CH PTRF, stored as a packed triangularm atrix.

PIVOT (input)
D etails of the interchanges and the block structure ofD as determ ined by CHPTRF.

B (input/output)

O \(n\) entry, the right hand side \(m\) atrix \(B\). On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zhsein-use inverse iteration to find specified right and/or left eigenvectors of a com plex upperH essenberg matrix H

\section*{SYNOPSIS}

```

    LDVL,VR,LDVR,MM,M,W ORK,RW ORK,\mathbb{FA}\mathbb{LL},\mathbb{F}A\mathbb{L}R,\mathbb{N}FO)
    CHARACTER * 1S\mathbb{DE,EIGSRC, IN ITV}
DOUBLE COMPLEX H (LD H,*),W (*),VL (LDVL,*), VR (LDVR,*),
W ORK (*)
\mathbb{NTEGERN,LDD,LDVL,LDVR,MM,M, INFO}
\mathbb{NTEGER FFA|L (*),\mathbb{FA}|R(*)}
LOG ICAL SELECT (*)
DOUBLE PRECISION RW ORK (*)
SU BROUT\mathbb{NE ZHSE\mathbb{N_64 (SDE E,EIG SRC, IN ITV,SELECT,N,H,LDH,W,VL,}}\mathbf{N},\textrm{N},
LDVL,VR,LDVR,MM,M,W ORK,RW ORK,\mathbb{FA}|L,\mathbb{FA}|R,\mathbb{NFO)}
CHARACTER * 1 SDDE,EIG SRC, IN ITV
DOUBLE COMPLEX H (LDH,*),W (*),VL (LDVL,*), VR (LDVR,*),
W ORK (*)
INTEGER*8N,LDH,LDVL,LDVR,MM,M,INFO
\mathbb{N TEGER*8 \mathbb{FA}|L (*), \mathbb{FA}|R(*)}
LOG ICAL*8 SELECT (*)
DOUBLE PRECISION RW ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE HSEIN (STDE,EIGSRC, \(\mathbb{N} \mathbb{I T V}, \operatorname{SELECT}, \mathbb{N}], H,[L D H], W, V L\), [LDVL],VR, [LDVR], MM, M, [W ORK], \(\mathbb{R W}\) ORK], \(\mathbb{F A} \mathbb{L} L, \mathbb{F A} \mathbb{I} R,[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::SDE,EIG SRC, \(\mathbb{N} I T V\)
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
COM PLEX (8), D \(\mathbb{I M} \operatorname{ENSION}(:,:\) ) :: H , VL, VR
\(\mathbb{N}\) TEGER : : N, LD H, LDVL, LDVR, M M , M, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{F} A \mathbb{I}, \mathbb{F} A \mathbb{R}\)
LOGICAL, D \(\mathbb{I M} E N S I O N(:):\) SELECT
REAL (8), D \(\mathbb{M}\) ENSION (:) ::RW ORK

SU BROUTINE HSEIN_64 (SDE E, EIGSRC, \(\mathbb{N} \mathbb{I T V}, \operatorname{SELECT}, \mathbb{N}], H,[L D H], W\),
 [ \(\mathbb{N} \mathrm{FO}]\) )

CHARACTER (LEN=1) ::SIDE,EIG SRC, \(\mathbb{N}\) ITV
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : : H , VL, VR
\(\mathbb{N} T E G E R(8):: N, L D H, L D V L, L D V R, M M, M, \mathbb{N F O}\)

LOGICAL (8), D \(\mathbb{M}\) ENSION (:) :: SELECT
REAL (8), D \(\mathbb{M}\) ENSION (:) ::RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zhsein (charside, chareigsre, char initv, int *select, int \(n\), doublecom plex *h, int ldh, doublecom plex \({ }^{*}\) w , doublecom plex *vl, int ldvl, doublecom plex *Vr, int ldvr, intmm, int*m, int*ifaill, int *ifailr, int*info);
void zhsein_64 (charside, char eigsrc, char initv, long * select, long n, doublecom plex *h, long ldh, doublecom plex *W , doublecom plex *vl, long ldvl, doublecom plex *vr, long ldvr, long mm, long *m, long *ifaill, long *ifailr, long *info);

\section*{PURPOSE}
zhsein uses inverse iteration to find specified rightand/or lefteigenvectors of a com plex upper H essenberg m atrix H.

The righteigenvectorx and the lefteigenvector \(y\) of the \(m\) atrix \(H\) comesponding to an eigenvalue \(w\) are defined by:
\[
\mathrm{H}^{\star} \mathrm{x}=\mathrm{w}^{\star} \mathrm{x}, \quad \mathrm{y}^{\star * \mathrm{~h}} \star \mathrm{H}=\mathrm{w}^{\star} \mathrm{y}^{\star *} \mathrm{~h}
\]
where \(y^{\star *}\) h denotes the conjugate transpose of the vectory.

\section*{ARGUMENTS}
```

S\mathbb{DE (input)}
$=\mathrm{R}$ ': com pute righteigenvectors only;
= L ': com pute lefteigenvectors only;
= B ': com pute both right and lefteigenvectors.

```

E IG SRC (input)
Specifies the source of eigenvalues supplied in \(W\) :
= Q ': the eigenvalues were found using CHSEQR;
thus, if \(H\) has zero subdiagonalelem ents, and so is block-triangular, then the \(j\) th eigenvalue can be assum ed to be an eigenvalue of the block containing the jth row/collm \(n\). This property allow \(s\) CHSEIN to perform inverse iteration on justone diagonalblock. = N ': no assum ptions are m ade on the correspondence betw een eigenvalues and diagonalblocks. In this case, CHSE \(\mathbb{N}\) must alw ays perform inverse tieration using the whole \(m\) atrix H.

\section*{\(\mathbb{N}\) ITV (input)}
= N ': no initial vectors are supplied;
= U ': user-supplied initial vectors are stored in the arrays V L and/orV R .

\section*{SELECT (input)}

Specifies the eigenvectors to be com puted. To select the eigenvector comesponding to the eigenvalueW ( \(\boldsymbol{j}\) ) SELECT ( \(\boldsymbol{j}\) ) m ustbe set to TRUE..

N (input) The order of the m atrix \(\mathrm{H} . \mathrm{N}>=0\).

H (input) The upperH essenberg matrix H.

LD H (input)
The leading dim ension of the array \(H\). LD H >= \(\max (1, N)\).

W (input/output)
On entry, the eigenvalues of H . On exit, the real parts of \(W\) may have been altered since close eigenvalues are perturbed slightly in searching for independenteigenvectors.

VL (input/output)
On entry, if \(\mathbb{N} \mathbb{T T V}=\mathrm{U}\) 'and \(S \mathbb{D} E=\mathrm{L}\) 'or B ', VL \(m\) ust contain starting vectors for the inverse iteration for the lefteigenvectors; the starting
vector for each eigenvectorm ustbe in the sam e colum n in w hich the eigenvector w ill be stored. On exit, ifS \(\mathrm{D} E=\mathrm{L}\) 'or B ', the lefteigenvectors specified by SELECT w ill.be stored consecutively in the colum ns of V , in the sam e order as theireigenvalues. If \(S \mathbb{D E}=\mathrm{R}, \mathrm{VL}\) is not referenced.

LDVL (input)
The leading dim ension of the array VL . LDVL >= \(\max (1, N)\) if \(S \mathbb{D} E=L\) 'or \(B^{\prime} ; L D V L>=1\) otherwise.
VR (input/output)
On entry, if \(\mathbb{N}\) ITV \(=U\) 'and \(S \mathbb{D} E=R\) 'or \(B ', V R\) \(m\) ust contain starting vectors for the inverse iteration for the righteigenvectors; the starting vector for each eigenvectorm ustbe in the sam e colum \(n\) in \(w\) hich the eigenvector \(w i l l\) be stored. On exit, ifSIDE = R 'or B', the righteigenvectors specified by SELECT w ill.be stored consecutively in the colum ns ofVR, in the sam e order as theireigenvalues. If \(S \mathbb{D} E=L \prime, V R\) is not referenced.

LDVR (input)
The leading dim ension of the amay VR. LDVR >= \(\max (1, N)\) if \(S \mathbb{D} E=R^{\prime}\) or \(B^{\prime} ; L D V R>=1\) otherw ise.

M M (input)
The num ber of colum \(n s\) in the amays \(V L\) and/or \(V R\).
M \(M\) > \(=\).

M (output)
The num ber of colum ns in the arays \(V \mathrm{~L}\) and/or VR required to store the eigenvectors \(\vDash\) the num ber of TRUE. elem ents in SELECT).

W ORK (w orkspace)
dim ension \((\mathbb{N} * N)\)

RW ORK (w orkspace)
dim ension \((\mathbb{N})\)

FAILL (output)
IfSIDE = L'or B', 平A ILL ( \(\mathbf{i}\) ) \(=j>0\) if the left eigenvector in the ith column of VL (corresponding to the eigenvalue w ( 1 ) failed to converge; \(\mathbb{F} A \amalg L(i)=0\) ifthe eigenvectorconverged satisfactorily. If \(S \mathbb{D} E=R ', \mathbb{F A} \mathbb{I} L\) is
not referenced.

FA \(\mathbb{I} R\) (output)
If \(S \mathbb{D} E=R\) 'or \(B ', \mathbb{F A} \| R(i)=j>0\) if the right eigenvector in the \(i\)-th colum \(n\) of \(V R\) (comesponding to the eigenvalue \(w(\mathcal{J})\) ) failed to converge; \(\mathbb{F A} \mathbb{I} R(i)=0\) if the eigenvectorconverged satisfactorily. IfSDE \(E=L ;\) FA \(\mathbb{I} R\) is not referenced.
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue
>0: if \(\mathbb{N} F O=i\), \(i\) is the num ber of eigenvectors which failed to converge; see \(\mathbb{F} A \mathbb{L} L\) and \(\mathbb{F} A \mathbb{I L}\) for further details.

\section*{FURTHER DETAILS}

E ach eigenvector is nom alized so that the elem ent of largest \(m\) agnitude has \(m\) agnitude 1 ; here the \(m\) agnitude of a com plex num ber \((x, y)\) is taken to be \(|x|+t \mid\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zhseqr - com pute the eigenvalues of a com plex upper H essenberg \(m\) atrix \(H\), and, optionally, the \(m\) atrices \(T\) and \(Z\) from the Schurdecom position \(H=Z\) T \(\mathrm{Z} * * \mathrm{H}\), where T is an upper triangular \(m\) atrix (the Schur form ), and \(Z\) is the unitary \(m\) atrix of Schurvectors

\section*{SYNOPSIS}
```

SU BROUTINE ZHSEQR(JOB,COMPZ,N, LOO,\mathbb{HI,H,LDH,W,Z,LD Z,W ORK,}
LW ORK,\mathbb{NFO)}
CHARACTER * 1 JOB,COMPZ
DOUBLE COM PLEX H (LD H,*),W (*),Z (LD Z,*),W ORK (*)
\mathbb{NTEGER N,}\mathbb{HO},\mathbb{H}I,LDH,LDZ,LW ORK,INFO
SUBROUT\mathbb{NE ZHSEQR_64(JOB,COM PZ,N, HO,THI,H,LDH,W,Z,LD Z,}
W ORK,LW ORK,\mathbb{NFO)}
CHARACTER * 1 JOB,COMPZ
DOUBLE COM PLEX H (LDH,*),W (*),Z (LD Z,*),W ORK (*)

```


\section*{F95 INTERFACE}
 \(\mathbb{W}\) ORK ], LW ORK, \([\mathbb{N} F O])\)

CHARACTER (LEN=1) :: JOB ,COM PZ
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : ) : : H, Z
\(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathbb{\mathbb { L }}, \mathbb{H} \mathrm{I}, \mathrm{LD} \mathrm{H}, \mathrm{LD} \mathrm{Z}, \mathrm{LW}\) ORK, \(\mathbb{N} F \mathrm{O}\)
SU BROUTINE HSEQR_64 (JOB,COMPZ,N, \(\mathbb{H} O, \mathbb{H} I, H,[L D H], W, Z,[L D Z]\),
[W ORK],LW ORK, [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) :: JOB ,COM PZ
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
COM PLEX (8),D IM ENSION (:,:) ::H,Z
\(\mathbb{N}\) TEGER (8) ::N , \(\mathbb{L O}, \mathbb{H}\) I, LD H ,LD Z, LW ORK , \(\mathbb{N}\) FO

\section*{C INTERFACE}
\#include <sunperfh>
void zhseqr (char, char, int, int, int, doublecom plex*, int, doublecom plex*, doublecom plex*, int, int*);
void zhseqr_64 (char, char, long, long, long, doublecom plex*, long, doublecom plex*, doublecom plex*, long, long*);

\section*{PURPOSE}
zhseqr com putes the eigenvalues of a com plex upper \(H\) essenberg \(m\) atrix \(H\), and, optionally, the \(m\) atrices \(T\) and \(Z\) from the Schurdecom position \(\mathrm{H}=\mathrm{Z} \mathrm{T} \mathrm{Z**H}\),where T is an upper triangular \(m\) atrix (the \(S\) chur form ), and \(Z\) is the unitary \(m\) atrix of Schurvectors.

O ptionally Z m ay be postm ultiplied into an input unitary \(m\) atrix \(Q\), so that this routine can give the Schur factorization of a m atrix A which has been reduced to the \(H\) essenberg form \(H\) by the unitary \(m\) atrix \(Q: A=Q * H * Q * * H=\) (Q Z)*T* \(\mathrm{Q} Z)^{\star *} \mathrm{H}\).

\section*{ARGUMENTS}

JOB (input)
= E ': com pute eigenvahues only;
\(=S\) ': com pute eigenvalues and the Schur form \(T\).

COM PZ (input)
= N ': no Schurvectors are com puted;
= I': Z is in itialized to the unitm atrix and the
m atrix Z of Schurvectors of H is retumed; = V ':
\(Z \mathrm{~m}\) ustcontain an unitary m atrix Q on entry, and the product Q *Z is retumed.

N (input) The order of the m atrix \(\mathrm{H} . \mathrm{N}>=0\).

It is assum ed that H is already upper triangular in row s and colum ns 1: \(\mathrm{HO}-1\) and \(\mathbb{H} \mathrm{I}+1 \mathbb{N}\). \(\mathbb{}\) O and IH I are norm ally setby a previous call to C G EBA L, and then passed to CGEHRD when them atrix output by CGEBAL is reduced to \(H\) essenberg form . O therw ise HO and \(\mathbb{H}\) I should be set to 1 and N respectively. \(1<=\mathbb{H O}<=\mathbb{H} \mathrm{I}<=\mathrm{N}\), if \(\mathrm{N}>0\); \(\mathbb{H} \mathrm{O}=1\) and \(\mathbb{H} \mathrm{I}=0\), if \(\mathrm{N}=0\).

IH I (input)
See the description of IIO .

H (input/output)
On entry, the upperH essenberg \(m\) atrix \(H\). On exit, if \(\mathrm{JOB}=\mathrm{S}\) ', H contains the uppertriangular \(m\) atrix \(T\) from the Schurdecom position the Schur form). If \(\mathrm{OB}=\mathrm{E}\) ', the contents of H are unspecified on exit.

LD H (input)
The leading dim ension of the array H.LDH >= max (1,N).

W (output)
The com puted eigenvalues. If \(\mathrm{JOB}=\mathrm{S}^{\prime}\) ', the eigenvalues are stored in the sam e order as on the diagonal of the Schur form retumed in \(H\), w ith W (i) \(=\mathrm{H}(\mathrm{i}, \mathrm{i})\).

Z (input) If \(\mathrm{COMPZ}=\mathrm{N}^{\prime}: \mathrm{Z}\) is not referenced.
If COM PZ = I': on entry, Z need notbe set, and on exit, \(Z\) contains the unitary \(m\) atrix \(Z\) of the Schurvectors of H . IfCOM PZ = V : on entry Z \(m\) ust contain an \(N\) by -N m atrix Q , which is assum ed to be equal to the unitm atrix except for the sub-
 N orm ally Q is the unitary m atrix generated by CUNGHR after the call to CGEH RD which form ed the \(H\) essenberg \(m\) atrix \(H\).

LD \(Z\) (input)
The leading dim ension of the aray Z . LD \(\mathrm{Z}>=\) \(\max (1, N)\) if COMPZ = I'or V';LD Z >=1 otherw ise.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (output)

The dimension of the array \(W\) ORK. LW ORK >= \(\max (1, N)\).

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK amay, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N} F O=-i\), the i-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=\) i, CH SEQR failed to compute all the eigenvalues in a total of 30 * ( \(\mathbb{H} \mathrm{I}-\mathbb{H} \mathrm{O}+1\) ) tierations; elem ents 1 :ilo-1 and i+1 m of W contain those eigenvalues which have been successfully com puted.

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}

> zj̇dm m -Jagged diagonalm atrix-m atrix m ultioly (m odified E llpack)

\section*{SYNOPSIS}

SUBROUTINE ZJADMM (TRANSA, M, N, K,ALPHA,DESCRA,
* VAL, \(\mathbb{N} D \mathrm{X}, \mathrm{PNTR}, \mathrm{MAXNZ}, \mathbb{P} E R M\),
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
\(\mathbb{N} T E G E R\) TRANSA, M,N,K,DESCRA (5),MAXNZ,
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R \quad \mathbb{N} D X(N N Z), P N T R(M A X N Z+1), \mathbb{P E R M}(M)\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COMPLEXVAL (NNZ), B (LDB,*), C (LDC,\(\left.^{\star}\right), W\) ORK (LW ORK)
SUBROUTINE ZJADMM_64(TRANSA, M,N,K,ALPHA,DESCRA,
* VAL, \(\mathbb{N} D \mathrm{X}, \mathrm{PNTR}, \mathrm{MAXNZ}, \mathbb{P} E R M\),
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, M,N,K,DESCRA (5),MAXNZ,
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R * 8 \mathbb{N} D X(N N Z), P N T R(M A X N Z+1), \mathbb{P E R M}(M)\)
DOUBLE COM PLEX ALPHA, BETA
DOUBLE COM PLEX VAL (NNZ), B (LDB,*), C (LDC,*), WORK (LW ORK)
where NN Z=PN TR M A XN Z +1)-PN TR (1)+1 is the num berofnon-zero elem ents

\section*{F95 INTERFACE}

SUBROUTINE JADMM (TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N} D X\), * PNTR,MAXNZ, \(\mathbb{P} E R M, B,[L D B], B E T A, C,[L D C],[W\) ORK], [LW ORK]) \(\mathbb{N} T E G E R\) TRANSA, M, K, MAXNZ
\(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: DESCRA, \(\mathbb{N} D \mathrm{X}, \mathrm{PN} T \mathrm{R}, \mathbb{P} E R M\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COMPLEX,D \(\mathbb{I M} E N S I O N(:):\) VAL

DOUBLE COM PLEX,D \(\mathbb{I}\) ENSION (: : : :: B, C

SUBROUTINE JADMM_64(TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L, \mathbb{N} D X\), * PNTR,MAXNZ, \(\mathbb{P E R M}, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N} T E G E R * 8\) TRANSA, \(M, K, M A X N Z\)
\(\mathbb{N} T E G E R * 8, D \mathbb{M}\) ENSION (:) :: DESCRA, \(\mathbb{N} D X, P N T R, \mathbb{P E R M}\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D \(\mathbb{I}\) ENSION (:) :: VAL
DOUBLE COM PLEX,D \(\mathbb{I}\) ENSION (:, :) :: B,C

\section*{DESCRIPTION}
\[
C \text { <-alpha op (A) B + beta C }
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, \(A\) is a m atrix represented in jagged-diagonal form at and op (A) is one of \(\mathrm{op}(\mathrm{A})=\mathrm{A}\) or \(\mathrm{op}(\mathrm{A})=A^{\prime}\) or \(\mathrm{op}(\mathrm{A})=\operatorname{conj}\left(\mathrm{A}^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate w ith m atrix
1 : operate w th transpose m atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in \(m\) atrix A

N \(\quad N\) um berof colum ns in matrix C
K \(\quad\) Num berof colum ns in matrix A

A LPH A Scalarparam eter
DESCRA 0 D escriptor argum ent. Five elem ent integer amay
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(A=A\) )
2 : Herm Aian ( \(\mathrm{A}=\mathrm{CONJG}\) (A))
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJ}\) ( A ) )
D ESCRA (2) upper/low er triangular indicator
1: low er

2 :upper
DESCRA (3) main diagonal type
0 :non-unit
1 :unit
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
D ESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 :no repeated indices

VAL () array of length NN Z consisting of entries of A. VA L can be view ed as a colum \(n m\) ajorordering of a row perm utation of the Ellpack representation of , where the Ellpack representation is perm uted so that the row s are non-increasing in the num ber of nonzero entries. V alues added forpadding in E llpack are not included in the Jagged-D iagonal form at.
\(\mathbb{I N D X}\) () array of length NNZ consisting of the colum n indices of the comesponding entries in VAL.
PN TR () array of length M AXNZ+1, where PNTR (I)-PN TR (1)+1 points to the location in VAL of the firstelem ent in the row -perm uted E llpack represenation of \(A\).

MAXNZ max num berofnonzeros elem ents per row .
\(\mathbb{P E R M} 0\) integer array of length \(M\) such that \(I=\mathbb{P E R M}\) ( \(I\) ), where row I in the originalE llpack representation corresponds to row I' in the perm uted representation. If \(\operatorname{PERM}(1)=0\), it is assum ed by convention that \(\mathbb{P E R M}(\mathbb{I})=I . \mathbb{P E R M}\) is used to determ ine the order in which row s ofC are updated.

B 0 rectangular array w ith first dim ension LD B .
LD B leading dim ension of \(B\)
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension of \(C\)
W ORK 0 scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is notreferenced in the current version.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at: http://m ath nistgov/n cso/Staff/k Rem ington/Ispblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
zádrp - rightperm utation of a jagged diagonalm atrix

\section*{SYNOPSIS}

SUBROUT \(\mathbb{N} E Z J A D R P(T R A N S P, M, K, V A L, \mathbb{N D X , P N T R , M A X N Z , ~}\) * IPERM, WORK, LW ORK)
INTEGER TRANSP,M,K,MAXNZ,LWORK
\(\mathbb{N} T E G E R \quad \mathbb{N} D X(*), \operatorname{PNTR}(M A X N Z+1), \mathbb{P E R M}(\mathbb{K}), W\) ORK (LW ORK)
DOUBLE COM PLEXVAL (*)

SU BROUTINE Z \(\mathbb{A} A D R P \_64(T R A N S P, M, K, V A L, \mathbb{N} D X, P N T R, M A X N Z\), * \(\quad\) PERM, WORK, LW ORK)
\(\mathbb{N}\) TEGER*8 TRANSP, M, K, MAXNZ,LW ORK
\(\mathbb{N} T E G E R * 8 \mathbb{N D X}(*), \operatorname{PNTR}(M A X N Z+1), \mathbb{P E R M}(\mathbb{K}), W\) ORK (LW ORK)
DOUBLE COM PLEX VAL (*)

F95 INTERFACE

SUBROUTINE JADRP (TRANSP, M, K, VAL, \(\mathbb{N} D X, P N T R, M A X N Z\), * \(\quad \mathbb{P} E R M,[W O R K],[L W O R K])\)
\(\mathbb{N}\) TEGER TRANSP, M, K, MAXNZ
\(\mathbb{N}\) TEGER,D \(\mathbb{M} E N S I O N(:):: \mathbb{N D} X, P N T R, \mathbb{P E R M}\)
DOUBLE COM PLEX,D \(\mathbb{M}\) ENSION (:) ::VAL

SUBROUTINE JADRP_64 (TRANSP, M, K, VAL, \(\mathbb{N} D X, P N T R, M A X N Z\), * \(\quad \mathbb{P} E R M,[W O R K],[L W O R K])\)
\(\mathbb{N}\) TEGER*8 TRANSP, M, K, MAXNZ
\(\mathbb{N}\) TEGER*8, D \(\mathbb{M}\) ENS \(\mathbb{O} N(:):: \mathbb{N D}\), PN TR, \(\mathbb{P} E R M\)
DOUBLE COM PLEX,D \(\mathbb{M}\) ENSION (:) ::VAL

DESCRIPTION

A \(<-A P\)
\(A<-A P^{\prime}\)
( 'indicates m atrix transpose)
\(w\) here perm utation \(P\) is represented by an integervector \(\mathbb{P} E R M\), such that \(\mathbb{P E R M}(I)\) is equal to the position of the only nonzero elem entin row Iofperm utation \(m\) atrix \(P\).

N O TE : In orderto get a sym etrically perm uted jagged diagonal \(m\) atrix P A P', one can explicitly perm ute the colum ns P A by calling

SJADRP ( \(0, M, M, V A L, \mathbb{N} D X, P N T R, M A X N Z, \mathbb{P} E R M, W\) ORK,LW ORK)
where param eters \(V A L, \mathbb{N D X}, P N T R, M A X N Z, \mathbb{P E R M}\) are the representation of \(A\) in the jagged diagonal form at. The operation \(m\) akes sense if the originalm atrix \(A\) is square.

\section*{ARGUMENTS}

TRAN SP Indicates how to operate \(w\) ith the perm utation \(m\) atrix
0 : operate w ith m atrix
1 : operate \(w\) ith transpose \(m\) atrix

M \(\quad \mathrm{N}\) um berof row \(s\) in matrix A

K \(\quad \mathrm{N}\) um ber of colum ns in matrix A

VAL () amay of length PNTR MAXNZ+1)-PNTR (1) consisting of entries ofA. VA L can be view ed as a colum \(n m\) ajor ordering of a row perm utation of the E llpack representation of A, w here the Ellpack representation is perm uted so that the row \(s\) are non-increasing in the num ber of nonzero entries. \(V\) alues added for padding in Ellpack are not included in the Jagged - D iagonal form at.

INDX () array of length PN TR MAXNZ+1)-PNTR (1) consisting of the colum \(n\) indices of the corresponding entries in VAL.

PNTR () array of length M AXNZ+1, where PNTR (I) PNTR (1)+1 points to the location in VA L of the firstelem ent in the row -perm uted E lhpack represenation of .

M A X N Z max num ber ofnonzeros elem ents per row.
\(\mathbb{P} E R M\) ( integeramay of length \(K\) such that \(I=\mathbb{P} E R M\) ( \(I\) ).

A ray \(\mathbb{P} E R M\) represents a perm utation \(P\), such that \(\mathbb{P E R M}\) ( I ) is equal to the position of the only nonzero elem ent in row Iofperm utation \(m\) atrix \(P\).
Forexam ple, if
|001|
\(\mathrm{P}=\left|\begin{array}{lll}1 & 0 & 0\end{array}\right|\)
|010|
then \(\mathbb{P E R M}=(3,1,2)\).

W ORK () scratch array of length LW ORK. LW ORK should be at leastK.

LW ORK length ofW ORK aray

If LW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/m cso/Staff/K Rem ington/tspblas/
"D ocum ent for the B asic \(L\) inearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib org/utk/papers/sparse _ps

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

zädsm -Jagged-diagonal form at triangular solve

```

\section*{SYNOPSIS}
```

SUBROUTINE ZJADSM (TRANSA, M,N,UNITD,DV,ALPHA,DESCRA,

* VAL, $\mathbb{N} D X$, PNTR,MAXNZ, $\mathbb{P E R M}$,
* B,LDB,BETA,C,LDC,WORK,LWORK)
$\mathbb{N} T E G E R$ TRANSA,M,N,UNITD,DESCRA (5),MAXNZ,
* LDB,LDC,LWORK
$\mathbb{N} T E G E R \quad \mathbb{N} D X(N N Z), P N T R(M A X N Z+1), \mathbb{P E R M}(M)$
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX DV M),VAL (NNZ), B (LDB,*), C (LDC ,*), W ORK (LW ORK)

```
SUBROUTINE ZJAD SM_64 (TRANSA, M,N,UNITD,DV,ALPHA,DESCRA,
* VAL, \(\mathbb{N} D \mathrm{X}, \mathrm{PNTR}, \mathrm{MAXNZ}, \mathbb{P} E R M\),
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, M,N,UNITD,DESCRA (5), MAXNZ,
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R * 8 \mathbb{N} D X(N N Z), P N T R(M A X N Z+1), \mathbb{P E R M}(M)\)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX DV M),VAL NNZ), B (LDB,*), C (LDC ,*), W ORK (LW ORK)
where NN Z \(=\) PN TR \(M\) A XN Z +1 )-PN TR ( 1 )+1 is the num berofnon-zero elem ents

\section*{F95 INTERFACE}

SUBROUTINE JADSM (TRANSA, M, \(\mathbb{N}], U N \mathbb{I T D}, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X\), * PNTR,MAXNZ, \(\mathbb{P E R M}, \mathrm{B},[\mathrm{LD} \mathrm{B}], \mathrm{BETA}, \mathrm{C},[\mathrm{LD} \mathrm{C}],[W \mathrm{ORK}]\), [LW ORK ]) \(\mathbb{I N T E G E R}\) TRANSA, M, MAXNZ

SUBROUTINE JAD SM _64 (TRANSA, M, \(\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A, V A L, \mathbb{N} D X\),
* PNTR,MAXNZ, \(\mathbb{P E R M}, B,[L D B], B E T A, C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N} T E G E R * 8\) TRANSA, M , M AXNZ
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: D E S C R A, \mathbb{N} D X, P N T R, \mathbb{P} E R M\)
DOUBLECOMPLEX ALPHA,BETA
DOUBLE COM PLEX ,D \(\mathbb{I}\) ENSION (:) :: VAL, DV
DOUBLE COM PLEX ,D \(\mathbb{M}\) ENSION (:, :) :: B , C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA } C \quad C<-A L P H A D \text { Op (A) B + BETA C } \\
& C<-A L P H A \text { Op (A)D B + BETA } C
\end{aligned}
\]
where A LPH A and BETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense \(m\) atrices, \(D\) is a diagonalscaling \(m\) atrix, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix represented in jagged-diagonal form at and \(o p(A)\) is one of op (A) \()=\operatorname{inv}(A)\) or op (A \()=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}\left(\infty n \dot{g}\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRAN SA Indicates how to operate \(w\) th the sparse \(m\) atrix 0 : operate \(w\) ith \(m\) atrix 1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in \(m\) atrix A

N \(\quad \mathrm{N}\) um berof colum ns in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identity \(m\) atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum n scaling)
4 : A utom atic row scaling (see section N O TES for further details)

DV () A rray of length M containing the diagonalentries of the scaling diagonalm atrix D .

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem entinteger anay D ESCRA (1) m atrix structure

0 : general
1 : symm etric ( \(A=A\) )
2 : Herm ( \(\mathrm{A}=\mathrm{CONJG}\) (A))
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CON}\) J ( A ) )
N ote:For the routine, DESCRA (1)=3 is only supported.
DESCRA (2) upper/low er triangular indicator
1: low er
2 : upper
DESCRA (3) m ain diagonaltype
0 :non-unit
1 : unit
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{M}\) PLEM ENTED )
0 :C C + + com patible
1 :Fortran com patible
DESCRA (5) repeated indices? \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL 0 array of length NNZ consisting of entries of A. VA L can be view ed as a colum \(n m\) ajorordering of a row perm utation of the Ellpack representation of A, where the Ellpack representation is perm uted so that the row s are non-increasing in the num ber of nonzero entries. V alues added forpadding in E llpack are not included in the Jagged-D iagonal form at.
\(\mathbb{N} D \mathrm{X} 0 \quad\) array of length NNZ consisting of the colum n indices of the comesponding entries in VAL.

PNTR 0) array of length M AXNZ +1 , where PNTR ( 1 ) PNTR (1) +1 points to the location in VA L of the firstelem ent in the row -perm uted \(E\) lipack represenation of \(A\).

MAXNZ max num berofnonzeros elem ents per row .
\(\mathbb{P E R M}\) ) integer array of length M such that \(\mathrm{I}=\mathbb{P} E R M\) ( I ), where row I in the originalE llpack representation corresponds to row I' in the perm uted representation. If \(\operatorname{PERM}(\mathbb{1})=0\), its assum ed by convention that \(\mathbb{P E R M}(\mathbb{I})=I . \mathbb{P E R M}\) is used to determ ine the order in which row s ofC are updated.

B 0 rectangular array w th first dim ension LD B.

LD B leading dim ension ofB
BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .
LD C leading dim ension of \(C\)

W ORK () scratch array of length LW ORK.
On exit, ifLW ORK \(=-1, W\) ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK aray. LW ORK should be at least2*M.

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on multiple processors, LW ORK \(>=2 * \mathrm{M}\) *N_CPUS where N_CPUS is the maxim um num berof processors available to the program.

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

\section*{SEE ALSO}

N IST FORTRAN Sparse B las U ser's G uide available at:
http://m ath nistgov/m csd/Staff/k Rem ington/Ispblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. No test for singularity ornear-singularity is included in this routine. Such tests \(m\) ust.be perform ed before calling this routine.
2. If U N ITD \(=4\), the routine scales the row s ofA such that their 2 -norm s are one. The scaling \(m\) ay im prove the accuracy of the com puted solution. C orresponding entries ofV A L are changed only in the particular case. On retum DV m atrix stored as a vector contains the diagonalm atrix by w hich the row s have been scaled. U N ITD = 2 should be used for the next calls to the routine w ith overw ritten VAL and DV.

WORK (1)=0 on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here \(i\) is the row num berw hich 2 -norm is exactly zero.
3. If \(\operatorname{DESCRA}(3)=1\) and UN ITD < 4, the unitdiagonalelem ents \(m\) ightorm ightnotbe referenced in the JA D representation of a sparse m atrix. They are notused anyw ay in these cases. ButifUN ITD=4, the unit diagonalelem ents M U ST be referenced in the \(\sqrt{A} D\) representation.
4.The routine can be applied for solving triangular system \(s\) w hen the upper or low er triangle of the general sparse m atrix A is used. H ow ever DESCRA (1) m ust.be equal to 3 in this case.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zlarz -applie a com plex elem entary reflector \(H\) to a com plex M -by-N m atrix C , from either the left or the right

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZLARZ (SDEE,M,N,L,V, NNCV,TAU,C,LDC,WORK)}
CHARACTER*1SIDE
DOUBLE COM PLEX TAU
DOUBLE COM PLEXV (*),C (LDC ,*),W ORK (*)
INTEGERM,N,L, \mathbb{NCV,LDC}

```

```

CHARACTER*1SDE
DOUBLE COM PLEX TAU
DOUBLE COM PLEX V (*),C (LDC ,*),W ORK (*)
INTEGER*8 M ,N,L, \mathbb{NCV ,LDC}

```

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E\) LARZ (S \(\mathbb{D} E, \mathbb{M}], \mathbb{N}], L, V,[\mathbb{N} C V], T A U, C,[L D C],[W O R K])\)
CHARACTER (LEN=1)::SDE
COM PLEX (8) ::TAU
COM PLEX (8),D IM ENSION (:) ::V,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) :: C
\(\mathbb{N} T E G E R:: M, N, L, \mathbb{N} C V, L D C\)
SU BROUTINE LARZ_64 (SDE, \(\mathbb{M}], \mathbb{N}], L, V,[\mathbb{N} C V], T A U, C,[L D C],[W O R K])\)

CHARACTER (LEN=1) ::SDE

COM PLEX (8) ::TAU
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::V,W ORK
COM PLEX (8), D \(\mathbb{I}\) ENSION (:,:) ::C
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{L}, \mathbb{N} C V, L D C\)

\section*{C INTERFACE}
\#include <sunperfh>
void zlarz (char side, intm , intn, int l, doublecom plex *v, int incv, doublecom plex *tau, doublecom plex \({ }^{*} \mathrm{C}_{\text {r }}\) int ldc);
void zlarz_64 (charside, long m, long n, long l, doublecom plex *v, long incv, doublecom plex *tau, doublecom plex *c, long ldc);

\section*{PURPOSE}
zlarz applies a com plex elem entary reflector \(H\) to a com plex \(M\) boy \(-\mathrm{N} m\) atrix \(C\), from eitherthe leftorthe right. \(H\) is represented in the form
\[
H=I-\tan { }^{\star} V^{*} V^{\prime}
\]
\(w\) here tau is a com plex scalar and \(v\) is a com plex vector.

If tau \(=0\), then \(H\) is taken to be the unitm atrix.

To apply H' (the conjugate transpose of H), supply conjg (tau) instead tau.
\(H\) is a product ofk elem entary reflectors as retumed by CTZRZF.

\section*{ARGUMENTS}

SID E (input)
\(=\mathrm{L}\) : form H * C
\(=R\) : form \(C * H\)

M (input) The num ber of row s of the \(m\) atrix \(C\).

N (input) The num ber of colum ns of the \(m\) atrix C .

L (input) The num ber ofentries of the vector \(V\) containing the m eaningful part of the \(H\) ouseholdervectors. If \(S \mathbb{D} E=L \prime, M>=L>=0\), if \(S \mathbb{D} E=R \prime N>=L\)
\(>=0\).

V (input) The vector v in the representation of H as retumed by CTZRZF. \(V\) is notused ifTA \(U=0\).
\(\mathbb{N} C V\) (input)
The increm entbetw een elem ents of \(v . \mathbb{I N} C V<>0\).

TAU (input)
The value tau in the representation of H.

C (input/output)
On entry, the \(\mathrm{M}-b y-\mathrm{N} m\) atrix C . On exit, C is overw ritten by them atrix \(\mathrm{H} * \mathrm{C}\) if \(S \mathbb{D} E=L\) ', or \(C * H\) if \(S \mathbb{D} E=R\) '.

LD C (input)
The leading dim ension of the array \(C . L D C>=\) \(\max (1, M)\).

W ORK (w orkspace)
\((\mathbb{N})\) if \(S \mathbb{D} E=L^{\prime}\) or \(M\) ) if \(S \mathbb{D} E=R^{\prime}\)

\section*{FURTHER DETAILS}

B ased on contributions by
A. Petitet, C om puter Science D ept., U niv. of Tenn., K noxville, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zlarzb - applie a com plex block reflectort or its transpose \(\mathrm{H} * * \mathrm{H}\) to a com plex distributed M -by-N C from the leftor the right

\section*{SYNOPSIS}

SUBROUTINE ZLARZB (SDE,TRANS,D \(\mathbb{R E C T}, S T O R E V, M, N, K, L, V, L D V, T\), LDT, C,LDC,W ORK,LDWORK)

CHARACTER * \(1 \mathrm{~S} \mathbb{D} E, T R A N S, D \mathbb{R E C T}, S T O R E V\)

\(\mathbb{I N}\) TEGER M, N,K,L,LDV,LDT,LDC,LDW ORK
SU BROUTINE ZLARZB_64 (SDE,TRANS,D \(\mathbb{R E C T}, S T O R E V, M, N, K, L, V, L D V\), T,LDT,C,LDC,W ORK,LDW ORK)

CHARACTER * 1 SDE, TRANS,D \(\mathbb{R E C T}\), STOREV

\(\mathbb{N}\) TEGER*8M,N,K,L,LDV,LD T,LDC,LDW ORK

\section*{F95 INTERFACE}

SU BROUTINE LARZB (SDE,TRANS,D \(\mathbb{R E C T}, \operatorname{STOREV}, \mathbb{M}], \mathbb{N}], K, L, V,[L D V]\), T, [LDT], C, [LDC], [W ORK], [LDW ORK])

CHARACTER (LEN=1) ::SDE,TRANS,D \(\mathbb{R E C T}, S T O R E V\)
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::V,T,C,WORK
\(\mathbb{N} T E G E R:: M, N, K, L, L D V, L D T, L D C, L D W\) ORK
SU BROUTINE LARZB_64 (SDE,TRANS,D \(\mathbb{R E C T}, S T O R E V, \mathbb{M}], \mathbb{N}], K, L, V\), [LDV], T, [LD T], C, [LDC], [W ORK], [LDW ORK])

CHARACTER (LEN=1) ::SDE,TRANS,D \(\mathbb{R E C T}\), STOREV
COM PLEX (8),D \(\mathbb{I}\) ENSION (:,:) ::V,T,C,W ORK \(\mathbb{N}\) TEGER (8) ::M , N , K , L, LDV ,LD T,LD C ,LD W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zlarzb (char side, char trans, char direct, char storev, int \(m\), intn, int \(k\), int l, doublecom plex *v, int ldv, doublecom plex *t, int ldt, doublecom plex \({ }^{*} \mathrm{C}_{\text {, }}\) int ldc, int ldw ork);
void zlarzb_64 (charside, char trans, char direct, char storev, long \(m\), long \(n\), long \(k\), long 1 , doublecom plex *v, long ldv, doublecom plex *t, long ldt, doublecom plex *c, long ldc, long ldw ork);

\section*{PURPOSE}
zlarzb applies a com plex block reflector H or its transpose \(\mathrm{H} * * \mathrm{H}\) to a com plex distributed M -by -N C from the leftorthe right.

C umently, only STOREV = R'and D \(\mathbb{R E C T}=\mathrm{B}\) 'are supported.

\section*{ARGUMENTS}

SIDE (input)
= L': apply H orH 'from the Left
= R':apply H orH 'from the Right

TRANS (input)
= N ': apply H N o transpose)
= C ': apply H ' (C onjugate transpose)
D \(\mathbb{R E C T}\) (input)
Indicates how H is form ed from a product of elem entary reflectors = F ': H = H (1) H (2) . . . H (k) (Forw ard, not supported yet)
= B':H = H (k) ...H (2) H (1) (Backw ard)

STOREV (input)
Indicates how the vectors w hich define the elem en-
tary reflectors are stored:
= C':Colum nw ise (notsup-
ported yet)
= R ':Rowwise

M (input) The num ber of row s of the \(m\) atrix \(C\).

N (input) The num ber of colum ns of the m atrix C .

K (input) The order of the \(m\) atrix T ( \(=\) the num ber of elem entary reflectors whose product defines the block reflector).

L (input) The num berof colum ns of the \(m\) atrix \(V\) containing the \(m\) eaningfilpart of the \(H\) ouseholder reflectors. If \(S \mathbb{D} E=L ', M>=L>=0\), if \(S \mathbb{D} E=R \prime, N>=L\) \(>=0\) 。
V (input) If \(S T O R E V=C^{\prime}, N V=K\); if \(S T O R E V=R \prime, N V=L\).

LDV (input)
The leading dim ension of the aray \(V\). IfSTOREV = C',LDV >=L; ifSTOREV = R',LDV >=K .

T (input) The triangular K -by K m atrix T in the representation of the block reflector.

LD T (input)
The leading dim ension of the anay \(\mathrm{T} \cdot \mathrm{LD} \mathrm{T}>=\mathrm{K}\).

C (input/output)
On entry, the M -by -N m atrix C . On exit, C is overw ritten by H * C orH * C or \(\mathrm{C}^{*} \mathrm{H}\) orC *H '.

LD C (input)
The leading dim ension of the array \(C . \operatorname{LDC}>=\) \(\max (1, M)\).

W ORK (w orkspace)
dim ension ( \(\mathrm{M} A X(M, N), K)\)

LDW ORK (input)
The leading dim ension of the array \(W\) ORK. If \(S \mathbb{D} E\)
\(=\mathrm{L} \prime\), LDW ORK >= max \((1, N)\); ifS \(\mathbb{D} E=R \prime\) LDW ORK \(>=\max (1, M)\).

\section*{FURTHER DETAILS}

B ased on contributions by
A. Petitet, C om puter Science D ept., U niv . ofTenn ., K noxville, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zlarzt-form the triangular factor \(T\) of a com plex block
reflector \(H\) of order> \(n\), which is defined as a product ofk elem entary reflectors

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZLARZT(D\mathbb{RECT,STOREV,N,K,V,LDV,TAU,T,LDT)}}\mathbf{N}\mathrm{ , (T,}
CHARACTER * 1D IRECT,STOREV
DOUBLE COM PLEX V (LDV ,*),TAU (*),T (LDT,*)
INTEGERN,K,LDV,LDT
SUBROUT\mathbb{NE ZLARZT_64@ PRECT,STOREV,N,K,V,LDV,TAU,T,LDT)}
CHARACTER * 1D RECT,STOREV
DOUBLE COM PLEX V (LDV ,*),TAU (*),T (LDT,*)
INTEGER*8N,K,LDV,LDT

```
F95 INTERFACE
    SUBROUT \(\mathbb{N} E\) LARZT (D \(\mathbb{R E C T}, \operatorname{STOREV}, \mathrm{N}, \mathrm{K}, \mathrm{V},[\mathrm{LDV}], T A U, T,[L D T])\)
    CHARACTER (LEN=1) ::D RECT,STOREV
    COMPLEX (8), D IM ENSION (:) ::TAU
    COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::V,T
    \(\mathbb{I N}\) TEGER :: \(\mathrm{N}, \mathrm{K}\),LDV,LDT
    SU BROUTINE LARZT_64 D \(\mathbb{R E C T}, \operatorname{STOREV}, N, K, V,[L D V], T A U, T,[L D T])\)
    CHARACTER (LEN=1)::D \(\mathbb{R E C T}\), STOREV
    COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::V,T
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{K}, \mathrm{LD} V\),LD T

\section*{C INTERFACE}
\#include <sunperfh>
void zlarzt(chardirect, charstorev, intn, int \(k\), doublecom plex *V, int ldv, doublecom plex *tau, doublecom plex *t, int ldt);
void zlarzt_64 (chardirect, charstorev, long n, long k, doublecom plex *v, long ldv, doublecom plex *tau, doublecom plex *t, long ldt);

\section*{PURPOSE}
zlarzt form sthe triangular factor T of a com plex block reflectorH of order> n,which is defined as a product of \(k\) elem entary reflectors.

IfD \(\mathbb{R E C T}=\mathrm{F}^{\prime}, \mathrm{H}=\mathrm{H}(1) \mathrm{H}(2) \ldots \mathrm{H}(\mathrm{k})\) and T is upper triangular;

IfD \(\mathbb{R E C T}=\mathrm{B}^{\prime}, \mathrm{H}=\mathrm{H}(\mathrm{k}) \ldots \mathrm{H}(2) \mathrm{H}(1)\) and T is lower triangular.

IfSTOREV = \(C\) ', the vector which defines the elem entary reflector \(H\) (i) is stored in the i-th colum \(n\) of the array \(V\), and
\[
H=I-V \star T \star V^{\prime}
\]

If \(S T O R E V=R\) ', the vector which defines the elem entary reflectorH (i) is stored in the \(i\)-th row of the amay \(V\), and
\(\mathrm{H}=\mathrm{I}-\mathrm{V}^{\prime} \star \mathrm{T} * \mathrm{~V}\)

C urrently, only STOREV = R'and D \(\mathbb{R E C T}=\mathrm{B}\) 'are supported.

\section*{ARGUMENTS}

D \(\mathbb{R E C T}\) (input)
Specifies the order in which the elem entary
reflectors are multiplied to form the block
reflector:
\(=F \cdot: H=H(1) H(2) \ldots H(k)\) Forw ard, notsup-
ported yet)
\(=B^{\prime}: H=H(k) \ldots H(2) H(1)\) (Backw ard)

STOREV (input)
Specifies how the vectors w hich define the elem entary reflectors are stored (see also Further D etails):
= R ': row wise
N (input) The order of the block reflector \(\mathrm{H} . \mathrm{N}>=0\).
\(K\) (input) The order of the triangular factor \(T \vDash\) the num ber of elem entary reflectors). \(\mathrm{K}>=1\).

V (input) (LDV,K) ifSTOREV = C'(LDV,N) if STOREV = R' Them atrix \(V\). See furtherdetails.
LDV (input)
The leading dim ension of the array V . If \(\operatorname{STOREV}=\) C',LDV >=max (1,N); ifSTOREV = R',LDV >=K .

TAU (input)
TAU (i) must contain the scalar factor of the elem entary reflector H (i).
\(T\) (input) The \(k\) by \(k\) triangular factor \(T\) of the block reflector. If \(\mathrm{D} \mathbb{R E C T}=\mathrm{F}^{\prime}, \mathrm{T}\) is upper triangular; if \(\mathrm{D} \mathbb{R E C T}=\mathrm{B}\) ', T is low er triangular. The restof the anay is notused.

LD \(T\) (input)
The leading dim ension of the array T .LD T >= K .

\section*{FURTHER DETAILS}

B ased on contributions by
A. Petitet, C om puterScience D ept, U niv . of Tenn., K noxville, U SA

The shape of the \(m\) atrix \(V\) and the storage of the vectors which define the \(H\) (i) is bestillustrated by the follow ing exam ple w th \(\mathrm{n}=5\) and \(\mathrm{k}=3\). The elem ents equal to 1 are not stored; the comesponding array elem ents are m odified but restored on exit. The restof the array is not used.

D \(\mathbb{R E C T}=\mathrm{F}^{\prime}\) and STOREV \(=\mathrm{C}^{\prime}: \quad \mathrm{D} \mathbb{R E C T}=\mathrm{F}^{\prime}\) and STOREV = R :
```

                                    V
    ```

```

(v1 v2 v3) (v1 v1 v1 v1 v1 ···..1
)
V = (v1 v2 v3 ) (v2 v2 v2 v2 v2 .

```
```

..1 )
(v1 v2 v3 ) (v3 v3 v3 v3 v3 .
.1 )
(v1 v2 v3 )
. . .
1..
1.
1
D\mathbb{RECT}=\mp@subsup{B}{}{\prime}\mathrm{ 'andSTOREV = C': D PRECT = B' and}
STOREV = R':
1

```

```

    . 1
        (1 . . ..v1 v1 v1 v1 v1 )
        . . }
    v2 v2 v2 )
...
v3 v3 v3 )
•••
(v1 v2 v3 )
V = (v1 v2 v3)
(v1 v2 v3 )

```

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zlatzm -routine is deprecated and has been replaced by routine CUNMRZ

\section*{SYNOPSIS}

```

CHARACTER * 1 SDE
DOUBLE COM PLEX TAU
DOUBLE COM PLEX V (*),C1 (LD C ,*),C2 (LD C ,*),W ORK (*)
INTEGERM,N,\mathbb{NCV,LDC}
SUBROUT\mathbb{NE ZLATZM _64(S\mathbb{DE,M,N,V,INCV,TAU,C1,C2,LDC,W ORK)}}\mathbf{N},\textrm{N},\textrm{N}
CHARACTER * 1SDE
DOUBLE COM PLEX TAU
DOUBLE COM PLEX V (*),C1 (LDC ,*),C2 (LD C ,*),W ORK (*)
INTEGER*8M,N,INCV,LDC

```

\section*{F95 INTERFACE}
    SUBROUTINE LATZM (SDE, \(\mathbb{M}], \mathbb{N}], V,[\mathbb{N} C V], T A U, C 1, C 2,[L D C],[\mathbb{O R K}])\)
    CHARACTER (LEN=1) ::SDE
    COM PLEX (8) ::TAU
    COM PLEX (8),D IM ENSION (:) ::V,W ORK
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::C1,C2
    \(\mathbb{N} T E G E R:: M, N, \mathbb{N} C V, L D C\)
SU BROUT \(\mathbb{N} E\) LATZM_64 (SDE, \(\mathbb{M}], \mathbb{N}], V,[\mathbb{N C V}], T A U, C 1, C 2,[\operatorname{DC}]\),
    [ W ORK])
CHARACTER (LEN=1)::SDE

COM PLEX (8) ::TAU
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::V,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::C1,C2
\(\mathbb{N}\) TEGER (8) :: M , N , \(\mathbb{N} C V, L D C\)

\section*{C INTERFACE}
\#include <sunperfh>
void zlatzm (charside, intm, intn, doublecom plex *v, int incv, doublecom plex *tau, doublecom plex *c1, doublecom plex *c2, intldc);
void zlatzm _64 (charside, long m, long n, doublecom plex *v, long incv, doublecom plex *tau, doublecom plex *c1, doublecom plex *C2, long ldc);

\section*{PURPOSE}
zlatzm routine is deprecated and has been replaced by routine CUNMRZ.

CLA TZM applies a H ouseholderm atrix generated by CTZRQF to a matrix.

LetP = I-tau*u*u', u = (1),
(v)
where \(v\) is an \((m-1)\) vector if \(S \mathbb{D} E=\mathbb{L}\) ', ora ( \(n-1\) ) vector if \(S \mathbb{D} E=R\).

If \(S \mathbb{D} E\) equals \(\mathbb{L}\) ', let
C \(=[\mathrm{C} 1] 1\)
[C2]m-1
n
Then C is overw rilten by P * C .

If \(S \mathbb{D} E\) equals \(R\) ', let
\(\mathrm{C}=[\mathrm{C} 1, \mathrm{C} 2] \mathrm{m}\)
1 n-1
Then C is overw rilten by C *P.

\section*{ARGUMENTS}
```

S\mathbb{DE (input)}
= L': form P * C
= R':form C * P

```

M (input) The num ber of row s of the m atrix C .

N (input) The num ber of colum ns of the \(m\) atrix \(C\).
\(V\) (input) \((1+\mathbb{M}-1) * a b s(\mathbb{N} C V))\) if \(S \mathbb{D} E=L^{\prime}(1+\mathbb{N}-\) \(1) * \operatorname{abs}(\mathbb{N} C V))\) if \(S \mathbb{D} E=R\) 'The vectorv in the representation ofP. V is notused if \(\mathrm{TA} \mathrm{U}=0\).
\(\mathbb{I N C V}\) (input)
The increm entbetw een elem ents of \(v . \mathbb{I N} C V<>0\)

TAU (input)
The value tau in the representation ofP.
C1 (input/output)
\((L D C N)\) if \(\left.S \mathbb{D} E=L^{\prime} M, 1\right)\) if \(S \mathbb{D} E=R^{\prime} O n\) entry, the \(n\)-vector \(C 1\) if \(S \mathbb{D E}=\mathrm{L}\) ', orthemvectorC 1 if \(S \mathbb{D} E=R\).

On exit, the first row ofP*C ifS \(\mathbb{D} E=\) ' ', or the first colum \(n\) of \(C * P\) if \(S I D E=R\).

C2 (input/output)
\((\mathbb{L D} C, N)\) if \(S \mathbb{D} E=L^{\prime}(\mathbb{L D} C, N-1)\) if \(S \mathbb{D} E=R^{\prime}\) On entry, the \((m-1) x n m\) atrix \(C 2\) if \(S \mathbb{D} E=\mathbb{L}\) ', or them \(x(n-1) m\) atrix \(C 2\) if \(S D E=R\) '.

Onexit, rows 2 m ofP* C if \(S D E=\mathrm{L}\) ', orcolum ns 2 m of \(\mathrm{C} *\) if \(\mathrm{SID} E=R\).

LD C (input)
The leading dim ension of the arrays C1 and C2. LD C \(>=\max (1, M)\).

W ORK (w orkspace)
\((\mathbb{N})\) if \(S \mathbb{D} E=L^{\prime}(M)\) if \(S \mathbb{D} E=R^{\prime}\)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zpbcon -estim ate the reciprocal of the condition num ber (in the 1 -norm ) of a com plex Herm titian positive definite band \(m\) atrix using the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) L*L**H com puted by CPBTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPBCON (UPLO,N,KD,A,LDA,ANORM,RCOND,W ORK,W ORK2,}
\mathbb{NFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGER N,KD,LDA,}\mathbb{NFO}
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION W ORK2 (*)
SU BROUTINE ZPBCON_64 (UPLO,N,KD,A,LDA,ANORM,RCOND,W ORK,
W ORK2,\mathbb{NFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)
INTEGER*8N,KD,LDA,INFO
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION W ORK2 (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE PBCON $\mathbb{U} P L O, \mathbb{N}], K D, A,[L D A], A N O R M, R C O N D,[W O R K]$, [W ORK2], [ $\mathbb{N} F O$ ])
CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COMPLEX (8),D $\mathbb{M}$ ENSION (: : : ) ::A

```
\(\mathbb{N} T E G E R:: N, K D, L D A, \mathbb{N F O}\)
REAL (8) ::ANORM,RCOND
REAL (8),D \(\mathbb{I}\) ENSION (:) ::W ORK2

SU BROUTINE PBCON_64 (UPLO, \(\mathbb{N}], K D, A,[L D A], A N O R M, R C O N D,[W O R K]\), [ W ORK2], [ \(\mathbb{N} \mathrm{FO}\) ])

CHARACTER (LEN=1) :: UPLO
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{I M} E N S I O N(:,:):: A\)
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{KD}, \mathrm{LD} \mathrm{A}, \mathbb{N} \mathrm{FO}\)
REAL (8) ::ANORM,RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zpbcon (charuple, intn, intkd, doublecom plex *a, int lda, double anorm, double *roond, int *info);
void zpbcon_64 (char uplo, long n, long kd, doublecom plex *a, long lda, double anorm, double *roond, long *info);

\section*{PURPOSE}
zpbcon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a com plex H erm itian positive definite band \(m\) atrix using the Cholesky factorization \(A=U * * H * U\) or \(A=\) L*L**H com puted by CPBTRF .

A \(n\) estim ate is obtained fornorm (inv (A ) ), and the reciprocal of the condition num ber is com puted as RCOND \(=1 /(\operatorname{ANORM}\) * norm (inv (A))).

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) : U pper triangular factor stored in A ;
\(=\mathbb{L}\) ': Low er triangular factor stored in A .

N (input) The order of them atrix A. N \(>=0\).

KD (input)
The num ber of superdiagonals of the matrix A if U PLO = U', or the num ber ofsub-diagonals if UPLO
\(=\mathbb{L}\) '. KD \(>=0\) 。

A (input) The triangular factorU or L from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) of the band \(m\) atrix A, stored in the first KD +1 row s of the array. The jth colum n of \(\begin{aligned} & \text { orL is stored in the }\end{aligned}\) \(j\) th colum \(n\) of the array A as follows: if UPLO \(=U ', A(k d+1+i-j)=U(i, j)\) for \(m a x(1, j\) \(\mathrm{kd})<=i<=j\) ifUPLO \(=\mathrm{L}\) ', \(A(1+i-j, j)=\mathrm{L}(i,-j)\) for \(j<=i<=m\) in \((n, j+k d)\).

LD A (input)
The leading dim ension of the aray A. LDA >= K D +1.
ANORM (input)
The 1-norm (or infinity-norm) of the Herm tian band \(m\) atrix A.

RCOND (output)
The reciprocal of the condition number of the \(m\) atrix \(A\), com puted as RCOND = 1/(ANORM *A \(\mathbb{N} V N M)\), where \(A \mathbb{N} V N M\) is an estim ate of the 1 -norm of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zpbequ - com pute row and colum n scalings intended to equilibrate a \(H\) erm itian positive definite band \(m\) atrix \(A\) and reduce its condition num ber (w ith respect to the tw o-norm )

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPBEQU(UPLO,N,KD,A,LDA,SCALE,SCOND,AMAX,INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*)
\mathbb{NTEGER N,KD,LDA,}\mathbb{NFO}
DOUBLE PRECISION SCOND,AMAX
DOUBLE PRECISION SCALE (*)
SUBROUT\mathbb{NE ZPBEQU_64 (UPLO,N,KD,A,LDA,SCALE,SCOND,AMAX,}
\mathbb{NFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEXA (LDA,*)
\mathbb{NTEGER*8N,KD,LDA,}\mathbb{N}FO
DOUBLE PRECISION SCOND,AMAX
DOUBLE PRECISION SCALE (*)
F95 INTERFACE
SU BROUT\mathbb{NE PBEQU (UPLO, N ],KD,A,[LDA],SCA LE,SCOND,AMAX,}
[\mathbb{NFO])}
CHARACTER (LEN=1)::UPLO
COM PLEX (8),D IM ENSION (:,:) ::A
INTEGER ::N,KD,LDA, NNFO
REAL (8) ::SCOND,AMAX
REAL (8),D IM ENSION (:) ::SCALE

```

SU BROUTINE PBEQU_64 (UPLO, \(\mathbb{N}], K D, A,[L D A], S C A L E, S C O N D, A M A X\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) :: UPLO
COM PLEX (8), D \(\mathbb{I M} E N S I O N(:,:):: A\)
\(\mathbb{N} \operatorname{TEGER}(8):: N, K D, L D A, \mathbb{N} F O\)
REAL (8) :: SCOND,AMAX
REAL (8), D \(\mathbb{M} E N S I O N(:):: S C A L E\)

\section*{C INTERFACE}
\#include <sunperfh>
void zpbequ (charuplo, intn, intkd, doublecom plex *a, int lda, double *scale, double *scond, double *am ax, int*info);
void zpbequ_64 (char uplo, long n, long kd, doublecom plex *a, long lda, double *scale, double *scond, double *am ax, long *info);

\section*{PURPOSE}
zpbequ com putes row and colum n scalings intended to equilibrate a H erm itian positive definite band m atrix A and reduce its condition num ber ( \(w\) ith respect to the tw o-norm ) . S contains the scale factors, \(S(i)=1 /\) sqnt \((A(i, i))\), chosen so that the scaled matrix B w ith elem ents \(B(i, j)=\) \(S(i) \star A(i, j) * S(j)\) has ones on the diagonal. This choige of \(S\) puts the condition num berofB \(w\) ithin a factor \(N\) of the sm allest possible condition num ber over allpossible diagonalscalings.

\section*{ARGUMENTS}

\section*{UPLO (input)}
\(=\mathrm{U}\) ': Upper triangularofA is stored;
= L ': Low er triangular of A is stored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

KD (input)
The num ber of superdiagonals of the \(m\) atrix \(A\) if \(\mathrm{UPLO}=\mathrm{U}\) ', or the num ber of subdiagonals if U PLO \(=\mathbb{L}^{\prime} . \mathrm{KD}>=0\) 。

A (input) The upper or low er triangle of the \(H\) erm itian band
\(m\) atrix \(A\), stored in the firstK \(D+1\) row sof the array. The \(j\) th colum n of A is stored in the \(j\) th column of the amay A as follow s: if UPLO = U', A \((k d+1+i-j)=A(i, j)\) for \(\max (1, j k d)<=i<=j\) if UPLO \(=L \prime\) ', \(A(1+i-j)=A(i, 7)\) for \(\dot{j}=i<=m\) in \((n, \dot{j}+k d)\).

LD A (input)
The leading dim ension of the array A. LDA >= K D +1 .

SCALE (output)
If \(\mathbb{N} F O=0\), SCA LE contains the scale factors for A.

SCOND (output)
If \(\mathbb{N} F O=0, S C A\) LE contains the ratio of the sm allest SCA LE (i) to the largestSCA LE (i). If SCOND \(>=0.1\) and AMAX is neither too large nor too sm all, it is notw orth scaling by SCA LE .

AMAX (output)
A bsolute value of largestm atrix elem ent. IfA M AX is very close to overflow orvery close to underflow , the \(m\) atrix should be scaled.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvahue.
\(>0\) : if \(\mathbb{N F O}=\) i, the \(i\)-th diagonal elem ent is nompositive.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zpobrfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is H em itian positive definite and banded, and provides emrorbounds and backw ard errorestim ates for the solution

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPBRFS (UPLO,N,KD,NRHS,A,LDA,AF,LDAF,B,LDB,X,}
LDX,FERR,BERR,W ORK,W ORK 2, INFO)
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
INTEGER N,KD,NRHS,LDA,LDAF,LD B,LDX,}\mathbb{N}F
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)
SU BROUT\mathbb{NE ZPBRFS_64 (UPLO,N,KD,NRHS,A,LDA,AF,LDAF,B,LDB,}
X,LDX,FERR,BERR,W ORK,W ORK 2,INFO)
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
INTEGER*8 N,KD,NRHS,LDA,LDAF,LD B,LDX,}\mathbb{N}F
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE PBRFS (UPLO, $\mathbb{N}], K D, \mathbb{N R H S}], A,[L D A], A F,[L D A F], B$, [LD B], X, [LDX ],FERR,BERR, [W ORK], [W ORK 2], [ $\mathbb{N F O}$ ])
CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D $\mathbb{M}$ ENSION (: : : : : A, AF, B, X

```
\(\mathbb{N}\) TEGER : \(: \mathrm{N}, \mathrm{KD}, \mathrm{NRH} \mathrm{S}, \mathrm{LDA} \mathrm{A}, \mathrm{LD} A \mathrm{~F}, \mathrm{LD} \mathrm{B}, \mathrm{LD} \mathrm{X}, \mathbb{N} \mathrm{FO}\) REAL (8), D \(\mathbb{I M} E N S I O N\) (:) ::FERR,BERR,W ORK 2

SUBROUTINE PBRFS_64 (UPLO, \(\mathbb{N}], K D, \mathbb{N R H S}], A,[L D A], A F,[L D A F]\), \(B,[L D B], X,[L D X], F E R R, B E R R,[W O R K],[W\) ORK2], [ \(\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO
COM PLEX (8), D IM ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : A , AF, B , X
\(\mathbb{N}\) TEGER (8) :: N , KD, NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{I}\) ENSION (:) ::FERR,BERR,W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zpbrfs (charuplo, intn, intkd, int nhs, doublecom plex *a, int lda, doublecom plex *af, int ldaf, doublecom plex *b, intldl, doublecom plex *x, int ldx, double * ferr, double *berr, int *info);
void zpbrfs_64 (charuplo, long n, long kd, long nrhs, doublecom plex *a, long lda, doublecom plex *af, long ldaf, doublecom plex *b, long ldb, doublecom plex *x, long ldx, double *ferr, double *berr, long *info);

\section*{PURPOSE}
zpbrfs im proves the com puted solution to a system of linear equations w hen the coefficientm atrix is H erm itian positive definite and banded, and provides errorbounds and backw ard errorestim ates for the solution.

\section*{ARGUMENTS}

\section*{UPLO (input)}
\(=\mathrm{U}\) : U ppertriangle ofA is stored;
\(=\mathbb{L}\) ': Low ertriangle ofA is stored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

KD (input)
The num ber of superdiagonals of the \(m\) atrix \(A\) if
\(\mathrm{UPLO}=\mathrm{U}\) ', or the num ber of subdiagonals if \(\mathrm{U} P L O\)
\(=\mathbb{L} . \mathrm{KD}>=0\) 。

NRHS (input)
The num ber of righthand sides, ie., the num ber
of collm ns of the \(m\) atrices \(B\) and \(X . N R H S>=0\).

A (input) The upper or low er triangle of the \(H\) erm itian band \(m\) atrix \(A\), stored in the firstK \(D+1\) row s of the array. The \(j\) th colum \(n\) ofA is stored in the \(j\) th column of the array \(A\) as follow \(s:\) if \(U P L O=U\) ', \(A(k d+1+i-j, j)=A(i, 7)\) for \(m a x(1, j k d)<=i<=j\) if \(\mathrm{UPLO}=\mathrm{L}^{\prime}, \mathrm{A}(1+i-j)=A(i, 7)\) for j \(<=i<=m\) in \((n, j+k d)\).

LDA (input)
The leading dim ension of the array A. LDA >= K D+1.
AF (input)
The triangular factor \(U\) or \(L\) from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) of the band \(m\) atrix \(A\) as com puted by CPBTRF, in the same storage form at as A (see A).

LDAF (input)
The leading dim ension of the array AF. LDAF >= K D +1.
\(B\) (input) The righthand side m atrix \(B\).

LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, N)\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by CPBTRS. On exit, the im proved solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the array X . LD X >= \(\max (1, N)\).

FERR (output)
The estim ated forw ard emrorbound for each solution vector \(X(\mathcal{)}\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix \(X\) ). If XTRUE is the true solution comesponding to \(X(\mathcal{O}), \operatorname{FERR}(\mathcal{O})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{\nu})-X\) TRU \(E\) ) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

BERR (output)
The com ponentw ise relative backw ard error of each
solution vectorX (i) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{(})\) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2{ }^{*} \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
< 0 : if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zpbstf - com pute a split C holesky factorization of a com plex Herm itian positive definite band \(m\) atrix A

\section*{SYNOPSIS}
```

SUBROUTINE ZPBSTF (UPLO,N,KD,AB,LDAB, INFO)
CHARACTER * 1 UPLO
DOUBLE COM PLEX AB (LDAB,*)
INTEGERN,KD,LDAB,INFO
SUBROUT\mathbb{NE ZPBSTF_64(UPLO,N,KD,AB,LDAB,INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEXAB (LDAB,*)
INTEGER*8N,KD,LDAB,INFO

```

\section*{F95 INTERFACE}
```

    SUBROUT\mathbb{NE PBSTF (UPLO, N ],KD,AB,[LDAB], [NNO ])}
    CHARACTER (LEN=1)::UPLO
    COM PLEX (8),D IM ENSION (:,:) ::AB
    \mathbb{NTEGER ::N,KD,LDAB,INFO}
    SU BROUT\mathbb{NE PBSTF_64 (UPLO , N ],KD ,AB, [LDAB ], [N FO ])}
    CHARACTER (LEN=1)::UPLO
COM PLEX (8),D IM ENSION (:,:) ::AB
INTEGER (8)::N,KD,LDAB,\mathbb{NFO}

```
void zpbstf(char uple, intn, int kd, doublecom plex *ab, int ldab, int *info);
void zpbstf_64 (charuplo, long n, long kd, doublecom plex *ab, long ldab, long *info);

\section*{PURPOSE}
zpbstf com putes a split C holesky factorization of a com plex H erm itian positive definite band \(m\) atrix \(A\).

This routine is designed to be used in conjunction w th CHBGST.
The factorization has the form \(A=S * * H * S\) where \(S\) is a band \(m\) atrix of the sam e bandw idth as A and the follow ing structure:
\[
\begin{array}{r}
S=\left(\begin{array}{ll}
U & ) \\
(M \quad L)
\end{array}, ~\right.
\end{array}
\]
w here \(U\) is upper triangular of orderm \(=(n+k d) / 2\), and \(L\) is low er triangular of ordern-m.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': Upper triangle ofA is stored;
\(=\mathrm{L}\) ': Low er triangle ofA is stored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

KD (input)
The num ber of superdiagonals of the \(m\) atrix \(A\) if
\(\mathrm{UPLO}=\mathrm{U}\) ', or the num ber of subdiagonals if U PLO
\(=\mathbb{L}^{\prime} . \mathrm{KD}>=0\) 。

A B (input/output)
O n entry, the upper or low er triangle of the Her \(m\) titian band \(m\) atrix \(A\), stored in the first kd+1 row s of the amay. The \(j\) th colum n of A is stored in the \(j\) th colum \(n\) of the array A B as follow s: if \(\mathrm{UPLO}=\mathrm{U}^{\prime}, \mathrm{AB}(\mathrm{kd}+1+i-j, j)=A(i, j)\) for \(\mathrm{max}(1, j\) \(\mathrm{kd})<=\dot{i}<=\dot{j}\) ifUPLO \(=\mathrm{L}\) ', AB \((1+i-j, j=A(i, j)\) for \(\dot{j}=\dot{i}<=m\) in \((n, j+k d)\).

On exit, if \(\mathbb{N F O}=0\), the factors from the split Cholesky factorization \(A=S * * H * S\). See Further D etails.

LD A B (input)
The leading dim ension of the array AB. LD A B >= K D +1 .
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue \(>0:\) if \(\mathbb{N} F O=\) i, the factorization could not be com pleted, because the updated elem enta (i,i) w as negative; the m atrix \(A\) is notposilive definite.

\section*{FURTHER DETAILS}

The band storage schem e is illustrated by the follow ing exam ple, w hen \(N=7, K D=2\) :
\begin{tabular}{|c|c|}
\hline \multicolumn{2}{|l|}{\(\mathrm{S}=(\mathrm{s} 11 \mathrm{~s} 12 \mathrm{~s} 13\)} \\
\hline ( & s22 s23 s24 ) \\
\hline ( & s33 s34 ) \\
\hline ( & s44 ) \\
\hline ( & s53 s54 s55 \\
\hline ( & s64 s65 s66 ) \\
\hline ( & s75 s76 s77) \\
\hline
\end{tabular}

If U PLO \(=\mathrm{U}\) ', the amay A B holds:
on entry: on exit:
* * a13 a24 a35 a46 a57 * * s13 s24 s53' s64's75'
* a12 a23 a34 a45 a56 a67 * s12 s23 s34 s54' s65's76' a11 a22 a33 a44 a55 a66 a77 s11 s22 s33 s44 s55 s66 s77

IfU PLO = L', the anay AB holds:
on entry: on exit:
```

a11 a22 a33 a44 a55 a66 a77 s11 s22 s33 s44 s55
s66 s77 a21 a32 a43 a54 a65 a76 * s12's23's34'
s54 s65 s76 * a31 a42 a53 a64 a64 * * s13'
s24's53 s64 s75 * *

```

A ray elem entsm arked * are notused by the routine; s12' denotes con \(\mathfrak{j}(s 12)\); the diagonalelem ents of \(S\) are real.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zpbsv - com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPBSV (UPLO,N,ND IAG,NRHS,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGERN,NDIAG,NRHS,LDA,LDB,INFO

```

```

CHARACTER * 1 UPLO
D OUBLE COM PLEX A (LDA,*),B (LDB,*)
\mathbb{NTEGER*8N,ND IAG,NRHS,LDA,LDB,INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE PBSV (UPLO, \(\mathbb{N}], N D \mathbb{I A} G, \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N} F O])\)

CHARACTER (LEN=1)::UPLO
COM PLEX (8),D IM ENSION (:,:) ::A,B
\(\mathbb{N} T E G E R:: N, N D \mathbb{A} G, N R H S, L D A, L D B, \mathbb{N} F O\)
SU BROUTINE PBSV_64 (UPLO, \(\mathbb{N}], N D \mathbb{I} G, \mathbb{N} R H S], A,[L D A], B,[L D B]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1)::UPLO
COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : ) :: A, B
\(\mathbb{N} \operatorname{TEGER}(8):: N, N D \mathbb{I} G, N R H S, L D A, L D B, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zpbsv (charuplo, intn, intndiag, intnrhs, doublecom plex *a, int lda, doublecom plex *b, int ldb, int *info);
void zpbsv_64 (charuplo, long n, long ndiag, long nrhs, doublecom plex *a, long lda, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
zpbsv com putes the solution to a com plex system of linear equations
\(A * X=B\), where \(A\) is an \(N\) boy \(-N\) Herm itian positive definite band \(m\) atrix and \(X\) and \(B\) are \(N\) by \(-N\) RH S m atriges.

The Cholesky decom position is used to factorA as
\[
\begin{aligned}
& A=U * * H * U, \text { if } U P L O=U ', \text { or } \\
& A=L * L * * H, \text { if } U P L O=L '
\end{aligned}
\]
\(w\) here \(U\) is an uppertriangularband \(m\) atrix, and \(L\) is a low er triangular band \(m\) atrix, \(w\) ith the sam e num ber of superdiagonals or subdiagonals as A. The factored form of A is then used to solve the system ofequations \(A * X=B\).

\section*{ARGUMENTS}

\section*{UPLO (input)}
\(=\mathrm{U}\) ': Upper triangle ofA is stored;
\(=\mathbb{L}\) ': Low ertriangle of is stored.

N (input) The num ber of linearequations, ie., the order of them atrix A. N >=0.

ND IA G (input)
The num ber of superdiagonals of the \(m\) atrix \(A\) if U PLO \(=\mathrm{U}\) ', or the num ber of subdiagonals ifU PLO \(=\mathbb{L} \cdot \mathrm{NDIAG>}>=0\) 。

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS \(>=0\).

A (input/output)
O \(n\) entry, the upper or low ertriangle of the Her -
\(m\) itian band \(m\) atrix A, stored in the firstND IA G +1
row s of the anray. The \(j\) th colum n of A is stored in the \(j\) th colum \(n\) of the amay A as follow s: if \(\mathrm{UPLO}=\mathrm{U}\) ', \(A(\mathbb{N D I A G + 1 + i - j})=A(i, j\) for \(\max (1, j \mathrm{j} D \mathrm{IA} \mathrm{G})<=\dot{i}=\dot{j}\) if \(\mathrm{UPLO}=\mathrm{L}\) ', A ( \(1+i-j\) )
\(=A(i, j)\) for \(\dot{j}=\dot{i}=m\) in \((\mathbb{N}, j+N D\) IA G \()\). See below for furtherdetails.

On exit, if \(\mathbb{N} F O=0\), the triangular factor \(U\) orL from the Cholesky factorization \(A=U * * H * U\) or \(A=\) \(L \star L * * H\) of the band \(m\) atrix A, in the sam e storage form atasA.

LD A (input)
The leading dim ension of the aray A. LDA >= N D IA G +1.
B (input/output)
O n entry, the N -by-N RH S righthand side m atrix B. On exit, if \(\mathbb{N F O}=0\), the N -by-NRHS solution m atrix X .

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N F O}=-i\), the i-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N}\) FO \(=i\), the leading \(m\) inoroforder iof A is notpositive definite, so the factorization could notbe com pleted, and the solution has not been com puted.

\section*{FURTHER DETAILS}

The band storage schem e is illustrated by the follow ing exam ple, when \(N=6, N D I A G=2\), and \(U P L O=U:\)

On entry: On exit:
```

    * * a13 a24 a35 a46 * * u13 u24 u35
    u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55

```
u66

Sim ilarly, if UPLO = 'L 'the form atofA is as follow s:
```

On entry: On exit:

```
```

    a11 a22 a33 a44 a55 a66 111 122 133 144 155
    166
a21 a32 a43 a54 a65 * 121 132 143 154 165
*
a31 a42 a53 a64 * * 131 142 153 164 *

```
A may elem entsm arked * are notused by the routine.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zpbsvx -use the C holesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) to com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPBSVX FACT,UPLO,N,NDIAG,NRHS,A,LDA,AF,LDAF,}
EQUED,S,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK 2,
\mathbb{NFO)}

```
CHARACTER * 1 FACT, UPLO, EQUED
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
\(\mathbb{N}\) TEGERN,ND \(\mathbb{I} G, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
DOUBLE PRECISION RCOND
D OUBLE PRECISION S (*), FERR (*), BERR (*), W ORK 2 (*)
SU BROUTINE ZPBSVX_64 FACT, UPLO,N,NDIAG,NRHS,A,LDA,AF,LDAF,
    EQUED,S,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK2,
    \(\mathbb{N} F O\) )
CHARACTER * 1 FACT, UPLO, EQUED
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
\(\mathbb{N}\) TEGER*8N,ND \(\mathbb{I} G, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
DOUBLE PRECISION RCOND
D OUBLE PRECISION S (*), FERR (*), BERR (*), W ORK 2 (*)

\section*{F95 INTERFACE}

SU BROUTINE PBSVX (FACT,UPLO, \(\mathbb{N}], N D \mathbb{I} G, \mathbb{N R H S}], A,[L D A], A F,[L D A F]\), EQUED, S, B, [LDB],X, [LDX],RCOND,FERR,BERR, [W ORK],

CHARACTER (LEN=1) ::FACT,UPLO,EQUED
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D IM ENSION (: : : : : A, AF, B, X
\(\mathbb{N} T E G E R:: N, N D \mathbb{A} G, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
REAL (8) :: RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::S,FERR,BERR,W ORK 2
SUBROUTINE PBSVX_64 \(\mathbb{F} A C T, U P L O, \mathbb{N}], N D \mathbb{I} G, \mathbb{N R H S ]}, A,[L D A], A F\), [LDAF],EQUED,S,B,[LDB],X, [LDX],RCOND,FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::FACT,UPLO,EQUED
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D IM ENSION (: : : : : A, AF, B, X
\(\mathbb{N} T E G E R(8):: N, N D \mathbb{A} G, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
REAL (8) :: RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::S,FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void zpbsvx (char fact, charuplo, int n, int ndiag, int nrhs, doublecom plex *a, int lda, doublecom plex *af, int ldaf, char equed, double *s, doublecom plex *b, int ladb, doublecom plex *x, int ldx, double *roond, double *ferr, double *berr, int *info);
void zpbsvx_64 (char fact, char uplo, long n, long ndiag, long nihs, doublecom plex *a, long lda, doublecom plex *af, long ldaf, charequed, double *s, doublecom plex *b, long ldb, doublecom plex *x, long \(l d x\), double *rcond, double *ferr, double *berr, long *info);

\section*{PURPOSE}
zpbsvx uses the C holesky factorization \(A=U * * H * U\) or \(A=\) \(\mathrm{L} * \mathrm{~L}^{\star *} \mathrm{H}\) to com pute the solution to a com plex system of linear equations
\(A * X=B\), where \(A\) is an \(N\) by \(-N\) Herm itian positive definthe band \(m\) atrix and \(X\) and \(B\) are \(N\) boy \(-N\) RH \(S m\) atrices.

Errorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT = E', real scaling factors are com puted to equilibrate
the system :
\(\operatorname{diag}(S) * A * \operatorname{diag}(S) * \operatorname{inv}(\operatorname{diag}(S)) * X=\operatorname{diag}(S) * B\)
W hether or not the system w illbe equilibrated depends on the
scaling of them atrix A , but if equilibration is used, A is
overw ritten by diag \((S) \star A\) *diag \((S)\) and \(B\) by diag \((S) * B\).
2. IfFACT = N 'or E', the Cholesky decom position is used to
factor the \(m\) atrix A (afterequilibration ifFACT \(=\mathrm{E}\) )
as
\(A=U * * H * U\), if \(U P L O=U\) ', or
\(\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\), if \(\mathrm{UPLO}=\mathrm{L}\) ',
\(w\) here \(U\) is an upper triangularband \(m\) atrix, and \(L\) is a low er triangularband \(m\) atrix.
3. If the leading i-by-iprincipal \(m\) inor is not positive definite,
then the routine retums w ith \(\mathbb{N} F O=\) i. O therw ise, the factored
form ofA is used to estim ate the condition num ber of the \(m\) atrix
A. If the reciprocal of the condition num ber is less than \(m\) achine precision, \(\mathbb{N F F O}=\mathrm{N}+1\) is retumed as a w aming, but the routine
still goes on to solve for X and com pute errorbounds as described below .
4.The system of equations is solved for X using the factored form of A.
5. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw and error estim ates
for主.
6. Ifequilibration w as used, the \(m\) atrix \(X\) is prem ultiplied by diag (S) so that it solves the original system before equilibration.

\section*{ARGUMENTS}

\section*{FACT (input)}

Specifies w hether or not the factored form of the \(m\) atrix \(A\) is supplied on entry, and ifnot, whether them atrix A should be equilibrated before it is factored. = \(\mathrm{F}^{\prime}\) : On entry, AF contains the factored form ofA. IfEQUED \(=Y\) ', the m atrix A has been equilibrated \(w\) ith scaling factors given by \(S\).
A and A F w illnotbe m odified. = N ': Them atrix A w illbe copied to AF and factored.
\(=\mathrm{E}\) ': The matrix A w ill be equilibrated if necessary, then copied to AF and factored.

UPLO (input)
\(=\mathrm{U}:\) Upper triangle of \(A\) is stored;
\(=\mathbb{L}\) ': Low ertriangle ofA is stored.

N (input) The num ber of linearequations, ie., the order of them atrix \(A . N>=0\).

ND IA G (input)
The num ber of superdiagonals of the \(m\) atrix \(A\) if \(\mathrm{UPLO}=\mathrm{U}\) ', orthe num ber of subdiagonals ifU PLO
\(=L^{\prime} . \mathrm{ND} \mathbb{I} \mathrm{G}>=0\) 。

NRHS (input)
The num ber of right-hand sides, i.e., the num ber of colum ns of the m atrices B and X . NRH S \(>=0\).

A (input/output)
O n entry, the upper or low er triangle of the H er\(m\) itian band \(m\) atrix A, stored in the firstND IA G +1 row s of the anay, exceptifFACT = \(\mathrm{F}^{\prime}\) and EQUED \(=Y\) ', then A m ustcontain the equilibrated \(m\) atrix diag \((S) \star A\) *diag \((S)\). The \(j\) th colum n of \(A\) is stored in the \(j\) th colum \(n\) of the amay \(A\) as follow \(s\) : if \(\mathrm{UPLO}=\mathrm{U} ', \mathrm{~A} \mathbb{N D I A G + 1 + i - j})=A(i, j\) for \(\max (1, j \mathrm{j} D \mathrm{IA} \mathrm{G})<=\dot{i}=\dot{j}\) if \(\mathrm{UPLO}=\mathbb{L}\) ', A ( \(1+i-j\) ) \(=A(i, j)\) for \(j=i<=m\) in \((\mathbb{N}, j+N D\) IAG \()\). See below for furtherdetails.

On exit, ifFACT = E' and EQUED = \(Y^{\prime}\), A is overw rilten by diag \((S) \star A\) *diag \((S)\).

LD A (input)
The leading dim ension of the array A. LD A >= N D IA G +1.

AF (input/output)

If \(F A C T=F '\), then \(A F\) is an inputargum ent and on entry contains the triangular factorU orL from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(L * L * * H\) of the band \(m\) atrix \(A\), in the sam e storage form atas A (see A). IfEQUED = \(Y\) ', then AF is the factored form of the equilibrated \(m\) atrix A.

If FA C T = N ', then AF is an output argum ent and on exit retums the triangular factorU orL from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) L*L** H .

If FACT = E', then AF is an output argum ent and on exit retums the triangular factorU orl from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(L * L * * H\) of the equilibrated \(m\) atrix \(A\) (see the description of \(A\) for the form of the equilibrated \(m\) atrix).

LDAF (input)
The leading dim ension of the array AF. LDAF >= ND IA G +1.

EQUED (input)
Specifies the form of equilibration thatw as done. \(=\mathrm{N}\) ': N o equilibration (alw ays true iffA C T = N \({ }^{2}\).
\(=Y^{\prime}:\) Equilibration \(w\) as done, i.e., A has been
replaced by diag (S) * A * diag (S). EQUED is an inputargum ent if \(F A C T=F\) '; otherw ise, it is an outputargum ent.
\(S\) (input/output)
The scale factors forA; notaccessed if EQUED = \(\mathrm{N}^{\prime} . \mathrm{S}\) is an inputargum entifFACT = F '; otherw ise, S is an outputargum ent. IfFACT = \(\mathrm{F}^{\prime}\) and EQUED = Y', each elem entofs m ustbe positive.

B (input/output)
On entry, the \(\mathrm{N}-\mathrm{by}-\mathrm{NRH} \mathrm{S}\) righthand side m atrix B . On exit, if EQUED \(=N\) ', \(B\) is notm odified; if EQUED = \(Y\) ', \(B\) is overw rilten by diag ( S ) * B .

LD B (input)
The leading dim ension of the anay B . LD B >= \(\max (1, N)\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=\mathrm{N}+1\), the N -by-NRH S solution
\(m\) atrix \(X\) to the original system ofequations.
\(N\) ote that ifEQUED \(=Y\) ', A and \(B\) are m odified on exit, and the solution to the equilibrated system is inv (diag \((S)) * X\).

\section*{LD X (input)}

The leading dim ension of the array X . LD X >= \(\max (1, N)\).

\section*{RCOND (output)}

The estim ate of the reciprocal condition num berof the \(m\) atrix \(A\) afterequilibration (ifdone). If RCOND is less than them achine precision (in particular, ifRCOND \(=0\) ), the \(m\) atrix is singular to w orking precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO \(>0\).

\section*{FERR (output)}

The estim ated forw ard emrorbound for each solution vector \(X()\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{O}), \operatorname{FERR}(\mathcal{O})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{D})\)-XTRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(j)\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector X (i) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

\section*{W ORK (w orkspace)}
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=i\), and \(i\) is
\(<=\mathrm{N}\) : the leading m inoroforderiof A is not positive definite, so the factorization could not be com pleted, and the solution has not been com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1\) : U is nonsingular, butRCOND is less than machine precision, \(m\) eaning that the \(m\) atrix is singular to w ork-
ing precision. Nevertheless, the solution and error bounds are com puted because there are a num ber of situations w here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, when \(N=6, N D \mathbb{I A G}=2\), and UPLO \(=U\) ':

Tw o-dim ensional storage of the \(H\) erm itian \(m\) atrix A:
```

al1 al2 al3
a22 a23 a24
a33 a34 a35
a44 a45 a46
a55 a56

```
(aijong (äi)) a66
\(B\) and storage of the upper triangle of \(A\) :
* * a13 a24 a35 a46
* a12 a23 a34 a45 a56
a11 a22 a33 a44 a55 a66
Sim ilarly, if UPLO = L'the form atofA is as follow s:
a11 a22 a33 a44 a55 a66
a21 a32 a43 a54 a65 *
a31 a42 a53 a64 * *

A rray elem entsm arked * are notused by the routine.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zpbtf2 -com pute the C holesky factorization of a com plex Herm itian positive definite band \(m\) atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPBTF2(UPLO,N,KD,AB,LDAB,INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX AB (LDAB,*)
INTEGERN,KD,LDAB,INFO
SU BROUT\mathbb{NE ZPBTF2_64(UPLO,N,KD,AB,LDAB,}\mathbb{N}FO)
CHARACTER * 1 UPLO
DOUBLE COM PLEXAB (LDAB,*)
INTEGER*8N,KD,LDAB,INFO

```

\section*{F95 INTERFACE}

SU BROUTINE PBTF2 (UPLO, \(\mathbb{N}], K D, A B,[L D A B],[\mathbb{N F O}])\)

CHARACTER (LEN=1)::UPLO
COMPLEX (8), D IM ENSION (:,:) ::AB
\(\mathbb{N} T E G E R:: N, K D, L D A B, \mathbb{N} F O\)

SU BROUTINE PBTF2_64 (UPLO, N ],KD ,AB, [LDAB], [NFO ])

CHARACTER (LEN=1) ::UPLO
COM PLEX (8), D IM ENSION (:,:) ::AB
\(\mathbb{N} T E G E R(8):: N, K D, L D A B, \mathbb{N} F O\)
void zpbtf2 (charuple, intn, int kd, doublecom plex *ab, int ldab, int *info);
void zpbtf2_64 (charuplo, long n, long kd, doublecom plex *ab, long ldab, long *info);

\section*{PURPOSE}
zpbtf2 com putes the Cholesky factorization of a com plex H er\(m\) itian positive definite band \(m\) atrix \(A\).

The factorization has the form
\(A=U^{\prime} \star U\), if \(U P L O=U\) ', or
\(A=L * L \prime\) if \(\mathrm{U} P \mathrm{PLO}=\mathrm{L}\) ',
w here U is an uppertriangularm atrix, U 'is the conjugate transpose of \(U\), and \(L\) is low ertriangular.

This is the unblocked version of the algorithm, calling Level2 BLAS.

\section*{ARGUMENTS}

\section*{UPLO (input)}

Specifies w hether the upper or low er triangular
part of the \(H\) erm itian \(m\) atrix \(A\) is stored:
= U ': U pper triangular
= LL: Low er triangular

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

KD (input)
The num ber of super-diagonals of the m atrix A if \(\mathrm{UPLO}=\mathrm{U}\) ', or the num ber of sub-diagonals if UPLO
\(=\mathbb{L} . \mathrm{KD}>=0\) 。

A B (input/output)
O n entry, the upper or low er triangle of the H er\(m\) tiian band \(m\) atrix \(A\), stored in the firstKD +1 row s of the amay. The \(j\) th colum n of A is stored in the \(j\) th colum n of the array A B as follow s: if \(\mathrm{UPLO}=\mathrm{U}\) ', AB \((k d+1+i-j, j)=A(i, j)\) for \(\max (1, j\) \(\mathrm{kd})<=i<=\dot{j}\) ifUPLO \(=L ', A B(1+i-j, j)=A(i, j)\) for \(\dot{j}=i<=m\) in \((n, j+k d)\).

On exit, if \(\mathbb{N} F O=0\), the triangular factor \(U\) orL
from the Cholesky factorization \(\mathrm{A}=\mathrm{U}\) * U or \(\mathrm{A}=\) L * L ' of the band m atrix A , in the sam e storage form atas A.

LDAB (input)
The leading dim ension of the array AB. LD AB >= KD+1.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-k\), the \(k\)-th argum enthad an illegalvalue \(>0:\) if \(\mathbb{N} F O=k\), the leading \(m\) inoroforderk is notpositive definite, and the factorization could notbe com pleted.

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, when \(\mathrm{N}=6, \mathrm{KD}=2\), and \(\mathrm{U} P L O=\mathrm{U}\) ':

On entry: On exit:
* * a13 a24 a35 a46 * * u13 u24 u35
u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55 u66

Sim ilarly, if UPLO = L'the form atofA is as follow s:
On entry: On exit:

166
a21 a32 a43 a54 a65 * \(121 \quad 132143154165\)
*
a31 a42 a53 a64 * * 131142153164 *

A ray elem entsm arked * are not used by the routine.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zpbtrf-com pute the Cholesky factorization of a complex H erm itian positive definite band \(m\) atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPBTRF (UPLO,N,KD,A,LDA,INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*)
\mathbb{NTEGERN,KD,LDA,}\mathbb{NFO}
SU BROUT\mathbb{NE ZPBTRF_64(UPLO,N ,KD,A ,LDA , INFO)}
CHARACTER * 1 UPLO
D OUBLE COM PLEX A (LDA,*)
INTEGER*8N,KD,LDA,INFO

```

\section*{F95 INTERFACE}
```

SU BROUTINE PBTRF (UPLO, $\mathbb{N}], K D, A,[L D A],[\mathbb{N F O}])$
CHARACTER (LEN=1)::UPLO
COM PLEX (8),D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R:: N, K D, L D A, \mathbb{N} F O$
SU BROUTINE PBTRF_64 (UPLO, $\mathbb{N}], K D, A,[L D A],[\mathbb{N F O}])$
CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R(8):: N, K D, L D A, \mathbb{N} F O$

```
void zpbtrf(charuplo, intn, intkd, doublecom plex *a, int lda, int *info);
void zpbtrf_64 (charuplo, long n, long kd, doublecom plex *a, long lda, long *info);

\section*{PURPOSE}
zpbtrf com putes the C holesky factorization of a com plex H er\(m\) itian positive definite band \(m\) atrix A.

The factorization has the form
\[
\begin{aligned}
& A=U * * H * U, \text { if } U P L O=U ' \text {, or } \\
& A=L * L * * H, \text { if } U P L O=L^{\prime},
\end{aligned}
\]
\(w\) here \(U\) is an uppertriangularm atrix and \(L\) is low er triangular.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': Uppertriangle of \(A\) is stored;
\(=1 \mathrm{~L}\) ': Low er triangle of A is stored.

N (input) The order of them atrix \(A . N>=0\).

KD (input)
The num berof superdiagonals of the \(m\) atrix \(A\) if
\(\mathrm{UPLO}=\mathrm{U}\) ', or the num ber of subdiagonals if \(\mathrm{U} P L O\)
\(=L^{\prime} . K D>=0\) 。

A (input/output)
O \(n\) entry, the upper or low er triangle of the H er\(m\) itian band \(m\) atrix \(A\), stored in the firstKD +1 row s of the array. The \(j\) th colum n of A is stored in the \(j\) th colum \(n\) of the amay \(A\) as follow \(s\) : if \(\mathrm{UPLO}=U ', A(k d+1+i-j, j)=A(i, j)\) for \(m a x(1, j\) \(\mathrm{kd})<=i<=\dot{j}\) if \(\mathrm{UPLO}=L^{\prime}, A(1+i-j, j)=A(i, j)\) for \(j<=i<=m\) in \((n, j+k d)\).

On exit, if \(\mathbb{N} F O=0\), the triangular factor \(U\) orL from the Cholesky factorization \(A=U * * H * U\) orA \(=\) L*L**H of the band \(m\) atrix A, in the sam e storage form atas A.

The leading dim ension of the array A. LDA >= K D +1 .
\(\mathbb{N F O}\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the leading \(m\) inoroforder is notpositive definite, and the factorization could notbe com pleted.

\section*{FURTHER DETAILS}

The band storage scheme is illustrated by the follow ing exam ple, when \(N=6, K D=2\), and \(U P L O=U:\)
On entry: On exit:
```

    * a13 a24 a35 a46 * * u13 u24 u35
    u46
* a12 a23 a34 a45 a56 * u12 u23 u34 u45
u56
a11 a22 a33 a44 a55 a66 u11 u22 u33 u44 u55
u66

```
Sim ilarly, if UPLO = L 'the form atofA is as follow s:
On entry: Onexit:
    a11 a22 a33 a44 a55 a66 \(111 \quad 122 \quad 133144 \quad 155\)
166
    a21 a32 a43 a54 a65 * \(121 \quad 132143154165\)
*
    a31 a42 a53 a64 * * 131142153164 *
*

A rray elem entsm arked * are notused by the routine.
C ontributed by
PeterM ayes and G inseppe Radicati, \(\mathbb{B M}\) EC SEC , Rom e, M arch 23,1989

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zpbtrs - solve a system of linear equations A *X \(=B\) w ith a \(H\) erm tian positive definite band \(m\) atrix \(A\) using the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) com puted by CPBTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPBTRS (UPLO,N,KD,NRHS,A,LDA,B,LDB, INFO)}
CHARACTER * 1 UPLO
D OUBLE COM PLEX A (LDA,*),B (LDB,*)
\mathbb{NTEGER N,KD,NRHS,LDA,LDB,INFO}
SUBROUT\mathbb{NE ZPBTRS_64(UPLO,N,KD,NRHS,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
\mathbb{NTEGER*8 N,KD,NRHS,LDA,LDB,INFO}

```
F95 INTERFACE
    SU BROUTINE PBTRS (UPLO, \(\mathbb{N}], K D, \mathbb{N R H S}], A,[L D A], B,[L D B],[\mathbb{N F O}])\)
    CHARACTER (LEN=1)::UPLO
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A,B
    \(\mathbb{N} T E G E R:: N, K D, N R H S, L D A, L D B, \mathbb{N} F O\)
    SU BROUTINE PBTRS_64 (UPLO, \(\mathbb{N}], K D, \mathbb{N} R H S], A,[L D A], B,[L D B]\),
        [ \(\mathbb{N} F O\) ])
    CHARACTER (LEN=1)::UPLO
    COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A,B
    \(\mathbb{N}\) TEGER (8) ::N,KD,NRHS,LDA, LD B, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zpbtrs (charuplo, intn, intkd, int nrhs, doublecom plex *a, intlda, doublecom plex *b, int ldb, int *info);
void zpbtrs_64 (charuplo, long n, long kd, long nrhs, doublecom plex *a, long lda, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
zpbtrs solves a system of linear equations \(A * X=B\) with a H erm itian positive definite band \(m\) atrix A using the C holesky factorization \(A=U * * H * U\) orA \(=L \star L * * H\) com puted by CPBTRF .

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': Upper triangular factor stored in A ;
\(=L^{\prime}\) : Low er triangular factor stored in A.

N (input) The order of the matrix A. \(\mathrm{N}>=0\).

KD (input)
The num ber of superdiagonals of the \(m\) atrix \(A\) if U PLO \(=\mathrm{U}\) ', orthe num ber of subdiagonals if U PLO \(=\mathrm{L}^{\prime} . \mathrm{KD}>=0\) 。

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS \(>=0\).

A (input) The triangular factor \(U\) or \(L\) from the Cholesky factorization \(A=U * * H * U\) orA \(=L * L * * H\) of the band \(m\) atrix A, stored in the first KD +1 row \(s\) of the array. The jth colum n of j orL is stored in the \(j\) th colum n of the array A as follow s: if UPLO \(=U \prime, A(k d+1+i-j)=U(i, j)\) for \(m a x(1, j\)
\(\mathrm{kd})<=i<=\dot{j}\) ifUPLO \(=\mathrm{L}\); \(\mathrm{A}(1+i-j)=\mathrm{j}(i, j)\)
for \(\dot{j}=i<=m\) in \((n, j+k d)\).

LD A (input)
The leading dim ension of the aray A. LDA >= K D +1 .

B (input/output)
On entry, the righthand side m atrix B. On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the anay B . LD B >= \(\max (1, N)\).
\(\mathbb{I N F O}\) (output)
= 0 : successfinlexit
<0: if \(\mathbb{N F O}=-i\), the \(i\)-th argum ent had an illegal value

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zpocon -estim ate the reciprocal of the condition num ber (in the 1-norm ) of a com plex H erm tian positive definitem atrix using the C holesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) com puted by CPO TRF

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)
INTEGERN,LDA, INFO
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION W ORK2 (*)
SUBROUTINE ZPOCON_64 UPLO,N,A,LDA,ANORM,RCOND,WORK,W ORK 2,
\mathbb{NFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGER*8N,LDA,INFO}
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION W ORK2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE POCON (UPLO, \(\mathbb{N}], A,[L D A], A N O R M, R C O N D,[W\) ORK ], [W ORK2], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D IM ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: N, L D A, \mathbb{N} F O\)

REAL (8) ::ANORM,RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::W ORK2

SUBROUTINE POCON_64 (UPLO, \(\mathbb{N}], A,[L D A], A N O R M, R C O N D,[W O R K],[W O R K 2]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : : A
\(\mathbb{N} T E G E R(8):: N, L D A, \mathbb{N F O}\)
REAL (8) ::ANORM,RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zpocon (charuplo, intn, doublecom plex *a, int lda, double anorm, double *rcond, int *info);
void zpocon_64 (charuple, long n, doublecom plex *a, long lda, double anorm , double *roond, long *info);

\section*{PURPOSE}
zpocon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a com plex \(H\) erm titian positive definitem atrix using the Cholesky factorization \(A=U * * H * U\) or \(A=L \star L * * H\) com puted by CPO TRF .

A \(n\) estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND \(=1 /(\operatorname{ANORM} \star\) norm (inv (A )) ).

\section*{ARGUMENTS}

\section*{UPLO (input)}
\(=U\) ': U ppertriangle of A is stored;
\(=L^{\prime}\) ': Low er triangle ofA is stored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input) The triangular factor \(U\) or L from the Cholesky factorization \(A=U * * H * U\) orA \(=L * L * * H\), as com puted by CPOTRF.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

ANORM (input)
The 1-norm (or infinity-norm ) of the Herm titian \(m\) atrix A.

\section*{RCOND (output)}

The reciprocal of the condition number of the \(m\) atrix \(A\), com puted as RCOND \(=1\) ( \(A N O R M * A \mathbb{N} V N M\) ), where \(A \mathbb{N} V N M\) is an estim ate of the 1 -nom of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{N N}\) FO \(=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zpoequ - com pute row and colum n scalings intended to equilibrate a H erm itian positive definite \(m\) atrix \(A\) and reduce its condition num ber (w ith respect to the tw o-norm )

\section*{SYNOPSIS}
```

SUBROUTINE ZPOEQU N,A,LDA,SCALE,SCOND,AMAX,\mathbb{NFO)}
D OUBLE COM PLEX A (LDA,*)
NNTEGERN,LDA,}\mathbb{N}F
DOUBLE PRECISION SCOND,AMAX
DOUBLE PRECISION SCALE (*)
SUBROUT\mathbb{NE ZPOEQU_64(N,A,LDA,SCALE,SCOND,AM AX,INFO)}
DOUBLE COM PLEX A (LDA,*)
INTEGER*8N,LDA,INFO
DOUBLE PRECISION SCOND,AMAX
DOUBLE PRECISION SCALE (*)
F95 INTERFACE

```

```

    COM PLEX (8),D IM ENSION (:,:) ::A
    \mathbb{NTEGER ::N,LDA,}\mathbb{N}FO
    REAL (8) ::SCOND,AMAX
    REAL (8),D IM ENSION (:) ::SCALE
    SUBROUT\mathbb{NE POEQU_64 (N ],A,[LDA ],SCALE,SCOND,AM AX,[INFO ])}
    COM PLEX (8),D IM ENSION (:,:) ::A
    \mathbb{NTEGER (8)::N,LDA,\mathbb{NFO}}\mathbf{N}=\mp@code{L}
    ```

REAL (8) :: SCOND,AMAX
REAL (8),D \(\mathbb{I M}\) ENSION (:) ::SCALE

\section*{C INTERFACE}
\#include <sunperfh>
void zpoequ (intn, doublecom plex *a, int lda, double *scale, double *scond, double *am ax, int *info);
void zpoequ_64 (long n, doublecom plex *a, long lda, double *scale, double *scond, double *am ax, long *info);

\section*{PURPOSE}
zpoequ computes row and colum n scalings intended to equilibrate a Herm itian positive definite matrix \(A\) and reduce its condition num ber (w ith respect to the two-norm ). S contains the scale factors, \(S(i)=1 /\) sqrt (A \((i, i))\), chosen so that the scaled matrix B w ith elem ents \(\mathrm{B}(i, 1)=\) S (i)*A ( \(i, j\) ) \({ }^{\text {S }}\) ( \(\bar{j}\) ) has ones on the diagonal. This choice of \(S\) puts the condition num berofB \(w\) ithin a factor \(N\) of the sm allest possible condition num ber over allpossible diagonal scalings.

\section*{ARGUMENTS}

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
A (input) The N -by -N Herm tian positive definite m atrix whose scaling factors are to be com puted. Only the diagonalelem ents ofA are referenced.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

SCALE (output)
If \(\mathbb{N} F O=0, S C A L E\) contains the scale factors for A.

\section*{SCOND (output)}

If \(\mathbb{N} F O=0, S C A L E\) contains the ratio of the \(s m\) allest SCA LE (i) to the largestSCA LE (i). IfSCOND \(>=0.1\) and AMAX is neither too large nor too sm all, it is notw orth scaling by SCA LE .

A bsolute value of largestm atrix elem ent. If A M A X is very close to overflow orvery close to underflow , the \(m\) atrix should be scaled.
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue
> 0 : if \(\mathbb{N F O}=\) i, the \(i\)-th diagonal elem ent is nonpositive.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zporfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is \(H\) erm tian positive definite,

\section*{SYNOPSIS}
```

SUBROUTINE ZPORFS (UPLO,N,NRHS,A,LDA,AF,LDAF,B,LDB,X,LDX,
FERR,BERR,W ORK,W ORK2,INFO)

```
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA ,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
\(\mathbb{N}\) TEGER N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
D OUBLE PRECISION FERR (*), BERR (*), W ORK 2 (*)
SU BROUTINE ZPORFS_64 (UPLO,N,NRHS,A,LDA,AF,LDAF,B,LDB,X,LDX,
    FERR,BERR,W ORK,WORK2, \(\mathbb{N} F O\) )
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
\(\mathbb{N} T E G E R * 8 N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)
D OUBLE PRECISION FERR (*), BERR (*), W ORK 2 (*)

\section*{F95 INTERFACE}

SU BROUTINE PORFS (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F], B,[L D B]\), X, [LDX],FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : : : A, AF, B, X
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D A F, L D B, L D X, \mathbb{N} F O\)

SU BROUTINE PORFS_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], A F,[L D A F], B,[L D B]\), \(\mathrm{X},[\mathrm{LD} \mathrm{X}], \mathrm{FERR}, \mathrm{BERR},[\mathbb{W} O R K],[\mathbb{W} O R K 2],[\mathbb{N} F O])\)

CHARACTER (LEN=1) :: UPLO
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M} E N S I O N(:,:):: A, A F, B, X\)
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{I}\) ENSION (:) :: FERR,BERR,W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zporfs (charuplo, intn, int nhs, doublecom plex *a, int lda, doublecom plex *af, int ldaf, doublecom plex *b, int ldb, doublecom plex *x, int ldx, double * ferr, double *berr, int *info);
void zporfs_64 (charuplo, long n, long nrhs, doublecom plex *a, long lda, doublecom plex *af, long ldaf, doublecom plex *b, long lalb, doublecom plex *x, long ldx, double * ferr, double *berr, long *info);

\section*{PURPOSE}
zporfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is H erm tian positive definite, and provides errorbounds and backw ard erroresti\(m\) ates for the solution.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': U pper triangle of A is stored;
= \(\mathbb{L}\) ': Low ertriangle ofA is stored.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of right hand sides, i.e., the num ber
of colum ns of the m atrices B and X . NRHS \(>=0\).

A (input) The H erm Itian \(m\) atrix A . IfUPLO \(=\mathrm{U}\) ', the leading N -by- N uppertriangularpartof A contains the upper triangularpart of the \(m\) atrix \(A\), and the strictly low ertriangularpart ofA is not referenced. IfUPLO = 'L', the leading N boy N lower
triangularpart ofA contains the low er triangular part of the \(m\) atrix A, and the strictly upper triangular part ofA is not referenced.

LD A (input)
The leading dim ension of the anay A. LDA >= \(\max (1, N)\).

AF (input)
The triangular factor \(U\) or \(L\) from the Cholesky factorization \(A=U * * H * U\) orA \(=L{ }^{*} L^{* *}{ }^{*}\), as com puted by CPOTRF.
LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, \mathbb{N})\).
\(B\) (input) The righthand side \(m\) atrix \(B\).
LD B (input)
The leading dim ension of the aray \(B\). LD B \(>=\) \(\max (1, \mathbb{N})\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by CPO TRS. On exit, the im proved solution \(m\) atrix \(X\).

\section*{LD X (input)}

The leading dim ension of the array X . LDX >= \(\max (1, \mathbb{N})\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X()\) ) the \(j\) th column of the solution \(m\) atrix \(X\) ). If XTRUE is the true solution comesponding to \(X(\mathcal{O}), \operatorname{FERR}(\mathcal{O})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{H})\)-XTRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(j)\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector \(X(\mathcal{)}\) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )

W ORK2 (w orkspace)
dim ension \(\mathbb{N}\) )

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the \(i\)-th argum ent had an illegal value

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zposv -com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUTINE ZPOSV (UPLO,N,NRHS,A,LDA,B,LDB,INFO)
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGERN,NRHS,LDA,LDB,INFO
SUBROUT\mathbb{NE ZPOSV_64(UPLO,N,NRHS,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGER*8N,NRHS,LDA,LDB, NNFO

```
F95 INTERFACE
    SU BROUTINEPOSV (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N F O}])\)
    CHARACTER (LEN=1)::UPLO
    COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : ) : : A, B
    \(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, \mathbb{N F O}\)
    SU BROUTINE POSV_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO
    COMPLEX (8),D \(\mathbb{D}\) ENSION (:,:) ::A,B
    \(\mathbb{N}\) TEGER (8) ::N,NRHS,LDA,LDB, \(\mathbb{N} F O\)
C INTERFACE
    \#include <sunperfh>
void zposv (charuple, intn, intnrhs, doublecom plex *a, int lda, doublecom plex *b, int ldb, int *info);
void zposv_64 (charuplo, long n, long nrhs, doublecom plex
*a, long lda, doublecom plex *b, long ldb, long
*info);

\section*{PURPOSE}
zposv com putes the solution to a com plex system of linear equations
\(A * X=B\), where \(A\) is an \(N\) boy \(-N\) H erm itian positive definte \(m\) atrix and \(X\) and \(B\) are \(N\)-by-N RH S \(m\) atrices.
The Cholesky decom position is used to factorA as
\(A=U * * H * U\), if \(U P L O=U\) ', or
\(A=L * L * * H\), if UPLO \(=L^{\prime}\) ',
where \(U\) is an upper triangularm atrix and \(L\) is a low ertriangular \(m\) atrix. The factored form ofA is then used to solve the system of equations \(A * X=B\).

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': U pper triangle of A is stored;
\(=\mathbb{L}\) ': Low ertriangle ofA is stored.

N (input) The num ber of linearequations, ie., the order of the matrix A. \(N>=0\).

NRHS (input)
The num ber of right hand sides, i.e., the num ber of colum ns of the m atrix B. NRHS \(>=0\).

A (input/output)
O n entry, the \(H\) erm itian m atrix A. If UPLO = \(U^{\prime}\), the leading N -oy N uppertriangularpart of \(A\) contains the upper triangularpart of the \(m\) atrix \(A\), and the strictly low er triangularpart of A is not referenced. If UPLO \(=\mathrm{L}\) ', the leading N -by N low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of A is notreferenced.

On exit, if \(\mathbb{N F O}=0\), the factor \(U\) orL from the Cholesky factorization \(A=U * * H * U\) orA \(=L * L * * H\).

LD A (input)
The leading dim ension of the anay A. LD A >= \(\max (1, N)\).

B (input/output)
On entry, the N -by-NRHS righthand side matrix B.
On exi, if \(\mathbb{N} F O=0\), the \(N\) by \(N\) RH S solution
matrix \(X\).

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO = -i, the i-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N F O}=\mathrm{i}\), the leading m inoroforderiof
A is notpositive definite, so the factorization could not.be com pleted, and the solution has not been com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zposvx - use the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) to com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPOSVX (FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,EQUED,}
S,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK2,INFO)
CHARACTER * 1FACT,UPLO,EQUED
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
\mathbb{N TEGER N,NRHS,LDA,LDAF,LDB,LDX, INFO}
DOUBLE PRECISION RCOND
DOUBLE PRECISION S (*),FERR (*),BERR (*),W ORK 2 (*)
SUBROUT\mathbb{NE ZPOSVX_64\&ACT,UPLO,N,NRHS,A,LDA,AF,LDAF,EQUED,}
S,B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,W ORK2, INFO)
CHARACTER * 1 FACT,UPLO,EQUED
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
INTEGER*8 N,NRHS,LDA,LDAF,LD B,LDX, INFO
DOUBLE PRECISION RCOND
DOUBLE PRECISION S (*),FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE POSVX \(\mathbb{E A C T}, \mathrm{UPLO}, \mathbb{N}], \mathbb{N R H S}], A,[L D A], A F,[L D A F]\), EQUED, \(\mathrm{S}, \mathrm{B},[\mathrm{LD} B], \mathrm{X},[\mathrm{LDX}], R C O N D, F E R R, B E R R,[W\) ORK], [W ORK 2], [ \(\mathbb{N F F O}\) ])

CHARACTER (LEN=1)::FACT,UPLO,EQUED

COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A, AF, B, X
\(\mathbb{N}\) TEGER :: \(N\),NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
REAL (8) :: RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::S,FERR,BERR,W ORK 2
SUBROUTINE POSVX_64 \(\mathbb{F A C T}, \mathrm{UPLO}, \mathbb{N}], \mathbb{N R H S}], A,[L D A], A F,[L D A F]\), EQUED, \(S, B,[L D B], X,[L D X], R C O N D, F E R R, B E R R,[W\) ORK], [W ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::FACT,UPLO,EQUED
COM PLEX (8), D IM ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A, AF,B,X
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
REAL (8) ::RCOND
REAL (8),D \(\mathbb{I}\) ENSION (:) ::S,FERR,BERR,W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zposvx (char fact, charuplo, int n, int nins, doublecom plex *a, int lda, doublecom plex *af, int ldaf, charequed, double *s, doublecom plex *b, int ldb, doublecom plex *x, int ldx, double *rcond, double * ferr, double *berr, int *info);
void zposvx_64 (char fact, char uplo, long n, long nrhs, doublecom plex *a, long lda, doublecom plex *af, long ldaf, char equed, double *s, doublecom plex *b, long ldb, doublecom plex *x, long ldx, double *rcond, double *ferr, double *berr, long *info);

\section*{PURPOSE}
zposvx uses the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) L*L**H to com pute the solution to a com plex system of linear equations
\(A\) * \(X=B\), where \(A\) is an \(N\)-by-N H erm itian positive definte m atrix and X and B are N -by-N RH S m atrioes.

Errorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT = E', real scaling factors are com puted to equilibrate
the system :
\(\operatorname{diag}(S) * A * \operatorname{diag}(S) * \operatorname{inv}(\operatorname{diag}(S)) * X=\operatorname{diag}(S) * B\)
\(W\) hether or not the system \(w\) illbe equilibrated depends on
the
scaling of the m atrix A , but ifequilibration is used, A is
overw rilten by diag \((\mathrm{S}) \star A\) *diag \((\mathrm{S})\) and B by diag \((\mathrm{S}) \star \mathrm{B}\).
2. IfFACT = N 'or E', the Cholesky decom position is used to
factorthem atrix A (afterequilibration ifFACT = E )
as
\[
\begin{aligned}
& A=U * * H * U, \text { if } U P L O=U ', \text { or } \\
& A=L * L * * H, \text { if } U P L O=L '
\end{aligned}
\]
\(w\) here \(U\) is an uppertriangularm atrix and \(L\) is a low er triangular
\(m\) atrix.
3. If the leading iboy-iprincipal \(m\) inor is not positive definite, then the routine retums w ith \(\mathbb{N F O}=\) i. O therw ise, the factored
form ofA is used to estim ate the condition num ber of the \(m\) atrix
A. If the reciprocal of the condition num ber is less than \(m\) achine precision, \(\mathbb{N} F O=\mathrm{N}+1\) is retumed as a w aming, but the routine still goes on to solve for \(X\) and com pute emorbounds as described below .
4. The system ofequations is solved forX using the factored form
```

ofA.

```
5. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates for it.
6. If equilibration \(w\) as used, the \(m\) atrix \(X\) is prem ultiplied by diag (S) so that it solves the original system before equilibration.

\section*{ARGUMENTS}

FACT (input)
Specifies w hether or not the factored form of the \(m\) atrix \(A\) is supplied on entry, and ifnot, w hether
them atrix A should be equilibrated before it is factored. = F : O n entry, AF contains the factored form of \(A\). IfEQUED \(=Y\) ', the \(m\) atrix \(A\) has been equilibrated \(w\) ith scaling factors given by \(S\). A and AF w illnotbe m odified. = N ': Them atrix A w illbe copied to A F and factored.
\(=\mathrm{E}\) ': The matrix A w ill be equilibrated if necessary, then copied to AF and factored.

UPLO (input)
\(=U\) ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.
N (input) The num ber of linear equations, i.e., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrices B and X. NRHS >=0.
A (input/output)
On entry, the \(H\) erm itian \(m\) atrix A, exceptifFA C T = \(\mathrm{F}^{\prime}\) and EQUED = Y ', then A m ust contain the equilibrated \(m\) atrix diag \((S) * A\) *diag \((S)\). If UPLO \(=\) U ', the leading N -by-N uppertriangular part of A contains the upper triangularpart of the \(m\) atrix
A, and the strictly low ertriangularpartofA is not referenced. If U PLO \(=\mathrm{L}\) ', the leading N -by -N low er triangularpartofA contains the low ertriangularpart of the matrix A, and the strictly upper triangular partofA is not referenced. A is notm odified ifFACT = F or \(\mathrm{N}^{\prime}\), or if \(\mathrm{FACT}=\) E'and EQUED = N 'on exit.

On exit, ifFACT = E' and EQUED = \(\mathrm{Y}^{\prime}, \mathrm{A}\) is overw ritten by diag \((S) \star A\) *diag \((S)\).

LD A (input)
The leading dim ension of the anay A. LDA >= \(\max (1, \mathbb{N})\).

AF (output)
IfFACT = \(\mathrm{F}^{\prime}\), then AF is an input argum entand on entry contains the triangular factor \(U\) orL from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{H}\), in the sam e storage form at as A. IfEQ U ED ne. \(N\) ', then \(A F\) is the factored form of the equilibrated \(m\) atrix diag \((S) \star A * \operatorname{diag}(S)\).

IfFACT = N ', then AF is an output argum ent and on exit retums the triangular factor \(U\) or \(L\) from
the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) of the originalm atrix A .

IfFACT = E', then AF is an output argum ent and on exitretums the triangular factor \(U\) orL from the Cholesky factorization \(A=U * * H * U\) or \(A=\) \(L * L * * H\) of the equilibrated \(m\) atrix \(A\) (see the description ofA for the form of the equilibrated m atrix).

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, N)\).

EQUED (input)
Specifies the form of equilibration thatw as done.
\(=\mathrm{N}\) ': N o equilibration (alw ays true ifFACT = N 7 。
\(=Y\) ': Equilibration w as done, i.e., A has been
replaced by diag \((\mathrm{S})\) * \(A\) * diag \((\mathrm{S})\). EQUED is an inputargum entiffACT = \(\mathrm{F}^{\prime}\); otherw ise, it is an output argum ent.

S (input/output)
The scale factors forA; not accessed if EQUED =
\(\mathrm{N}^{\prime} . \mathrm{S}\) is an inputargum entifFACT=F'; other-
W ise, S is an outputargum ent. IfFACT \(=\mathrm{F}^{\prime}\) and
\(E Q U E D=Y\) ', each elem entof m ust.be positive.

B (input/output)
On entry, the N boy-NRHS righthand side matrix B.
On exit, if EQUED = \(N^{\prime}\) ', B is notm odified; if
EQUED \(=Y ', B\) is overw ritten by diag \((S) * B\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

X (output)
If \(\mathbb{N}\) FO \(=0\) or \(\mathbb{N} F O=N+1\), the N -by -N R H S solution
\(m\) atrix \(X\) to the original system of equations.
\(N\) ote that ifEQUED \(=Y^{\prime}, A\) and \(B\) are \(m\) odified on
exit, and the solution to the equilibrated system
is inv \((\) diag \((S)) \star X\).

LD X (input)
The leading dim ension of the array X . LD X >= \(\max (1, N)\).

The estim ate of the reciprocal condition num ber of the matrix A after equilibration (if done). If RCOND is less than the \(m\) achine precision (in particular, ifRCOND \(=0\) ), the \(m\) atrix is singular to w orking precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO \(>0\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X()\) ) the \(j\) th column of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{H})\), FERR ( \()\) ) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{D})-X\) TRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(j)\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector \(X(\mathcal{)}\) ) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthad an ille-
galvalue
\(>0:\) if \(\mathbb{N F O}=i\), and \(i\) is
\(<=N\) : the leading \(m\) inor oforderiof \(A\) is not positive definite, so the factorization could not be com pleted, and the solution has not been com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1\) : U is nonsingular, butRCOND is less than machine precision, m eaning that the \(m\) atrix is singularto w orking precision. Nevertheless, the solution and error bounds are com puted because there are a num berof situationsw here the com puted solution can bem ore accurate than the value ofRC O N D w ould suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zpotf2 -com pute the Cholesky factorization of a com plex Herm itian positive definite m atrix A

\section*{SYNOPSIS}
```

    SUBROUT\mathbb{NE ZPOTF2(UPLO,N,A,LDA, INFO)}
    CHARACTER * 1 UPLO
    DOUBLE COM PLEXA (LDA,*)
    INTEGERN,LDA,}\mathbb{N}F
    SUBROUT\mathbb{NE ZPOTF2_64(UPLO,N,A,LDA, INFO)}
    CHARACTER * 1 UPLO
    DOUBLE COM PLEXA (LDA,*)
    INTEGER*8N,LDA,INFO
    F95 INTERFACE
SUBROUT\mathbb{NE POTF2 (UPLO, NN,A,[LDA ], [NFO])}
CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D IM ENSION (:,:) ::A
\mathbb{NTEGER ::N,LDA,}\mathbb{NFO}
SUBROUT\mathbb{NE POTF2_64 (UPLO, N ],A, [LDA ], [NFO ])}
CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D IM ENSION (:,:) ::A
\mathbb{NTEGER (8)::N,LDA,}\mathbb{NFO}

```
C INTERFACE
    \#include <sunperfh>
void zpotf2 (charuplo, intn, doublecom plex *a, int lda, int *info);
void zpotf2_64 (char uple, long n, doublecom plex *a, long lda, long *info);

\section*{PURPOSE}
zpotf2 com putes the Cholesky factorization of a com plex H er\(m\) itian posilive definite m atrix A .

The factorization has the form
\(A=U^{\prime} \star U\), if \(U P L O=U '\) 'or
\(A=L * L \prime\), ifUPLO = L',
\(w\) here \(U\) is an uppertriangularm atrix and \(L\) is low er triangular.

This is the unblocked version of the algorithm, calling Level2 BLAS.

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the upper or low er triangular
part of the \(H\) erm itian \(m\) atrix \(A\) is stored. \(=U\) ':
U pper triangular
= IL ': Low ertriangular

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
On entry, the \(H\) erm itian matrix A. If \(\mathrm{UPLO}=\mathrm{U}\) ', the leading \(n\) by \(n\) upper triangularpart of A contains the upper triangular part of the \(m\) atrix \(A\), and the strictly low er triangularpart of A is not referenced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading n by n low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of \(A\) is not referenced.

On exit, if \(\mathbb{N F O}=0\), the factor \(U\) orL from the
Cholesky factorization \(A=U\) * \(U\) or \(A=L * L\) '.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-\mathrm{k}\), the k -th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N} F O=k\), the leading \(m\) inor oforder \(k\) is notpositive definite, and the factorization could notbe com pleted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zpotrf-com pute the C holesky factorization of a complex Herm itian positive definite m atrix A

\section*{SYNOPSIS}
```

SUBROUTINE ZPOTRF(UPLO,N,A,LDA, INFO)
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*)
\mathbb{NTEGERN,LDA,INFO}
SUBROUT\mathbb{NE ZPOTRF_64(UPLO,N,A,LDA, INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEXA (LDA,*)
INTEGER*8N,LDA,INFO
F95 INTERFACE
SUBROUT\mathbb{NE POTRF (UPLO, NN,A, [LDA ], [NFO])}
CHARACTER (LEN=1)::UPLO
COM PLEX (8),D IM ENSION (:,:) ::A
\mathbb{NTEGER ::N,LDA,}\mathbb{NFO}
SUBROUT\mathbb{NE POTRF_64 (UPLO, N ],A, [LDA ],[NFO ])}
CHARACTER (LEN=1)::UPLO
COM PLEX (8),D IM ENSION (:,:) ::A
\mathbb{NTEGER (8)::N,LDA,}\mathbb{NFO}

```
C INTERFACE
    \#include <sunperfh>
void zpotrf(charuplo, intn, doublecom plex *a, int lda, int *info);
void zpotrf_64 (charuplo, long n, doublecom plex *a, long lda, long *info);

\section*{PURPOSE}
zpotrf com putes the Cholesky factorization of a com plex H er\(m\) itian posilive definite m atrix A .

The factorization has the form
\(A=U * * H * U\), if \(U P L O=U\) ', or
\(A=L * L^{* *} H\), if \(\mathrm{UPLO}=\mathrm{L}\) ',
\(w\) here \(U\) is an upper triangularm atrix and \(L\) is low er triangular.

This is the block version of the algorithm, calling Level 3 BLAS .

\section*{ARGUMENTS}

UPLO (input)
\(=U\) ': U pper triangle of \(A\) is stored;
\(=\mathbb{L}\) ': Low ertriangle of \(A\) is stored.

N (input) The order of them atrix A. N \(>=0\).

A (input/output)
O n entry, the H erm itian m atrix A. If UPLO = U', the leading N -by -N uppertriangularpartof A contains the upper triangularpart of the \(m\) atrix \(A\), and the strictly low er triangularpart of \(A\) is not referenced. If UPLO = 'L', the leading N -by N low er triangularpant of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of A is notreferenced.

On exit, if \(\mathbb{N F O}=0\), the factor U orL from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\).

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).
= 0: successfulexit
< 0 : if \(\mathbb{N}\) FO \(=-\)-i, the i-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the leading \(m\) inoroforder \(i\) is not positive definite, and the factorization could notbe com pleted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zpotri-com pute the inverse of a com plex H em itian positive definite \(m\) atrix \(A\) using the Cholesky factorization \(A=\) \(\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) com puted by CPO TRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPOTRI(UPLO,N,A,LDA, INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*)
\mathbb{NTEGER N,LDA,}\mathbb{N}FO
SUBROUT\mathbb{NE ZPOTRI_64(UPLO,N,A,LDA, INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*)
INTEGER*8N,LDA,INFO
F95 INTERFACE
SU BROUT\mathbb{NE POTRI(UPLO, N ],A , [LD A ], [NNFO ])}
CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D IM ENSION (:,:) ::A
\mathbb{NTEGER ::N,LDA,}\mathbb{NFO}
SU BROUT\mathbb{NE POTRI_64 (UPLO, N ],A, [LDA ], [NNO ])}
CHARACTER (LEN=1)::UPLO
COM PLEX (8),D IM ENSION (:,:) ::A
\mathbb{NTEGER (8)::N,LDA,\mathbb{NFO}}\mathbf{~}=\mp@code{L}

```
void zpotri(charuple, intn, doublecom plex *a, int lda, int *info);
void zpotri_ 64 (char uplo, long n, doublecom plex *a, long lda, long *info);

\section*{PURPOSE}
zpotricom putes the inverse of a com plex H erm itian positive definite \(m\) atrix \(A\) using the Cholesky factorization \(A=\) \(\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) com puted by CPO TRF .

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) ': Uppertriangle of \(A\) is stored;
\(=\mathbb{L}\) ': Low er triangle ofA is stored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O \(n\) entry, the triangular factor \(U\) or \(L\) from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\), as com puted by CPO TRF. On exit, the upper or low er triangle of the ( H erm itian) inverse of \(A\), overw riling the input factorU orL.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the i-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the \((i, i)\) elem entof the factor
U orL is zero, and the inverse could not.be com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zpotrs - solve a system of linear equations A *X = B w ith a Herm itian positive definite \(m\) atrix \(A\) using the Cholesky factorization \(A=U * * H * U\) orA \(=L * L * * H\) com puted by CPO TRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPOTRS (UPLO,N,NRHS,A,LDA,B,LDB,INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGERN,NRHS,LDA,LDB,NNFO

```
SU BROUTINE ZPOTRS_64 (UPLO,N,NRHS,A,LDA,B,LDB, INFO)
CHARACTER * 1 UPLO
D OUBLE COM PLEXA (LDA,*), B (LDB,*)
\(\mathbb{N}\) TEGER*8N,NRHS,LDA,LDB, \(\mathbb{N} F O\)
F95 INTERFACE
    SU BROUTINE POTRS (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N F O}])\)
    CHARACTER (LEN=1) ::UPLO
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A,B
    \(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, \mathbb{N} F O\)
    SU BROUTINE POTRS_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], B,[L D B],[\mathbb{N F O}])\)
    CHARACTER (LEN=1) ::UPLO
    COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : ) : : A, B
    \(\mathbb{N} \operatorname{TEGER}(8):: N, N R H S, L D A, L D B, \mathbb{N} F O\)
\#include <sunperfh>
void zpotrs (char uple, intn, int nrhs, doublecom plex *a, int lda, doublecom plex *b, int ldb, int *info);
void zpotrs_64 (charuplo, long n, long nrhs, doublecom plex
*a, long lda, doublecom plex *b, long ldb, long
*info);

\section*{PURPOSE}
zpotes solves a system of linear equations \(A * X=B\) with a H erm itian positive definite m atrix A using the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) com puted by CPOTRF.

\section*{ARGUMENTS}

UPLO (input)
\(=U\) ': U pper triangle ofA is stored;
\(=\mathrm{L}\) ': Low er triangle ofA is stored.

N (input) The order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the m atrix B. NRH S \(>=0\).

A (input) The triangular factorU or L from the Cholesky
factorization \(A=U * * H * U\) orA \(=L * L * * H\), as com puted by CPOTRF.

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, N)\).

B (input/output)
O \(n\) entry, the righthand side \(m\) atrix \(B\). On exit, the solution \(m\) atrix X .

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zppcon -estim ate the reciprocal of the condition num ber (in the 1 -norm ) of a com plex H erm itian positive definite packed \(m\) atrix using the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) L*L**H com puted by CPPTRF

\section*{SYNOPSIS}

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*), W ORK (*)
\(\mathbb{N}\) TEGER \(N, \mathbb{I N F O}\)
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION W ORK 2 (*)
SU BROUTINE ZPPCON_64 (UPLO,N,A,ANORM,RCOND,WORK,WORK2, \(\mathbb{N} F O\) )
CHARACTER * 1 UPLO
DOUBLE COM PLEXA (*), W ORK (*)
\(\mathbb{N}\) TEGER*8 N, \(\mathbb{N}\) FO
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION W ORK 2 (*)

\section*{F95 INTERFACE}

SU BROUTINE PPCON (UPLO, N,A,ANORM,RCOND, [W ORK], [W ORK2], [ \(\mathbb{N} F O]\) )
CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D IM ENSION (:) ::A,W ORK
\(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
REAL (8) ::ANORM ,RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ORK 2

CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D IM ENSION (:) ::A,W ORK
\(\mathbb{N}\) TEGER (8) ::N, \(\mathbb{N} F O\)
REAL (8) ::ANORM,RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void zppcon (char uple, int n, doublecom plex *a, double anorm , double *rcond, int *info);
void zppcon_64 (char uplo, long n, doublecom plex *a, double anorm , double *rcond, long *info);

\section*{PURPOSE}
zppcon estim ates the reciprocal of the condition num ber (in the 1 -norm ) of a com plex H erm itian positive definite packed \(m\) atrix using the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) L*L**H com puted by CPPTRF .

A \(n\) estim ate is obtained fornorm (inv (A ) ), and the reciprocal of the condition num ber is com puted as RCOND = 1 / ANORM * norm (inv (A))).

\section*{ARGUMENTS}

UPLO (input)
= U : U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
A (input) The triangular factorU or \(L\) from the Cholesky
factorization \(A=U * * H * U\) orA \(=L * L * * H\), packed colum nw ise in a linearanay. The jth colum \(n\) of \(U\) or \(L\) is stored in the array A as follow s: if UPLO = U', A \((i+(j-1) * j 2)=U(i, j)\) for \(1<=i<=j\) if \(U P L O=L ', A(i+(j-1) *(2 n-j / 2)=L(i, 7)\) for j=i<=n.

ANORM (input)
The 1-norm (or infinity-norm) of the Herm tian matrix A.

\section*{RCOND (output)}

The reciprocal of the condition num ber of the
\(m\) atrix \(A\), com puted as RCOND = 1/(ANORM *A \(\mathbb{N} V N M)\),
where \(A \mathbb{N} V N M\) is an estim ate of the 1 -nom of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an ille-
galvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zppequ - com pute row and colum n scalings intended to equilibrate a Herm itian positive definite \(m\) atrix A in packed storage and reduce its condition num ber (w ith respect to the tw o-norm )

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPPEQU (UPLO,N,A,SCALE,SCOND,AMAX,INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*)
INTEGERN,\mathbb{NFO}
DOUBLE PRECISION SCOND,AMAX
DOUBLE PRECISION SCALE (*)
SUBROUT\mathbb{NE ZPPEQU_64(UPLO,N,A,SCALE,SCOND,AMAX,INFO )}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*)
INTEGER*8 N, INFO
DOUBLE PRECISION SCOND,AMAX
DOUBLE PRECISION SCALE (*)
F95 INTERFACE
SUBROUT\mathbb{NE PPEQU (UPLO, N ],A,SCALE,SCOND,AMAX,[NFO])}
CHARACTER (LEN=1) ::UPLO
COMPLEX (8),DIM ENSION (:) ::A
\mathbb{NTEGER ::N,\mathbb{NFO}}\mathbf{N}=\mp@code{N}
REAL (8) ::SCOND,AM AX
REAL (8),D IM ENSION (:) ::SCALE

```

CHARACTER (LEN=1) :: UPLO
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::A
\(\mathbb{N}\) TEGER (8) :: N , \(\mathbb{N} F O\)
REAL (8) :: SCOND,AMAX
REAL (8), D \(\mathbb{M}\) ENSION (:) ::SCALE

\section*{C INTERFACE}
\#include <sunperfh>
void zppequ (char uple, int n, doublecom plex *a, double
*scale, double *scond, double *am ax, int *info);
void zppequ_64 (charuplo, long n, doublecom plex *a, double
*scale, double *scond, double *am ax, long *info);

\section*{PURPOSE}
zppequ com putes row and colum n scalings intended to equilibrate a Herm itian positive definite \(m\) atrix A in packed storage and reduce its condition num ber (w ith respect to the two-norm ). S contains the scale factors, \(S(i)=1 /\) sqit \((A\) ( \(i, i))\), chosen so that the scaled \(m\) atrix \(B\) w ith elem ents \(B(i, j)=S(i) * A(i, j) * S(i)\) has ones on the diagonal. This choioe ofS puts the condition num ber of B w ithin a factor N of the sm allest possible condition num ber over all possible diagonal scalings.

\section*{ARGUMENTS}

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input) The upper or low er triangle of the H erm itian \(m\) atrix A, packed colum nw ise in a linear array. The \(j\) th column of A is stored in the anay A as follow s: if UPLO = U',A ( \(i+(j-1) * j 2)=A(i, 1)\) for \(1<=i<=j\) ifUPLO \(=L\) ', A ( \(\left.i+(j-1)^{*}(2 n-j) / 2\right)\) \(=A(i, 7)\) for \(j=i<=n\).

SCALE (output)
If \(\mathbb{N} F O=0, S C A L E\) contains the scale factors for A.

\section*{SCOND (output)}

If \(\mathbb{N} F O=0, S C A\) LE contains the ratio of the sm allest SCA LE (i) to the largestSCA LE (i). If SC OND \(>=0.1\) and \(A M A X\) is neither too large nor too sm all, it is notw orth scaling by SC A LE .

AMAX (output)
A bsolute value of largestm atrix elem ent. If A M A X is very close to overflow orvery close to underflow , the m atrix should be scaled.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue
> 0 : if \(\mathbb{N} F O=i\), the \(i\) th diagonal elem ent is nonpositive.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zpprfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is H em itian positive definite and packed, and provides emrorbounds and backw ard error estim ates for the solution

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPPRFS (UPLO,N,NRHS,A,AF,B,LDB,X,LDX,FERR,BERR,}
W ORK,W ORK2, INFO)
CHARACTER * 1 UPLO
DOUBLE COM PLEXA (*),AF (*),B (LD B ,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)
SUBROUT\mathbb{NE ZPPRFS_64 (UPLO,N,NRHS,A,AF,B,LDB,X,LDX,FERR,}
BERR,W ORK,W ORK2,INFO)
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*),AF (*),B (LD B,*),X (LDX,*),W ORK (*)
INTEGER*8N,NRHS,LDB,LDX,INFO
D OUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE PPRFS (UPLO,N, NRHS],A,AF,B,[LDB],X, [LDX],FERR, BERR, [W ORK], [W ORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::A,AF,W ORK
COM PLEX (8), D IM ENSION (:,:) ::B,X
\(\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

SU BROUTINE PPRFS_64 (UPLO,N, \(\mathbb{N} R H S], A, A F, B,[L D B], X,[L D X], F E R R\), BERR, [W ORK], [W ORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::A,AF,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : : : B, X
\(\mathbb{N}\) TEGER ( 8 ) ::N,NRHS,LDB,LDX, \(\mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include < sunperfh>
void zppnfs (charuplo, intn, int nrhs, doublecom plex *a, doublecom plex *af, doublecom plex *b, int ldb, doublecom plex *x, int ldx, double *ferr, double *ber, int*info);
void zpprfs_64 (charuplo, long n, long nrhs, doublecom plex *a, doublecom plex *af, doublecom plex *b, long ldb, doublecom plex *x, long ldx, double *ferr, double *berr, long *info);

\section*{PURPOSE}
zpprfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is \(H\) erm itian positive definite and packed, and provides errorbounds and backw ard error estim ates for the solution.

\section*{ARGUMENTS}

UPLO (input)
\(=U\) : U pper triangle ofA is stored;
= L' ': Low ertriangle ofA is stored.

N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).
NRHS (input)
The num ber of right hand sides, i.e., the num ber
of collm ns of the m atrices B and X. NRHS >=0.
A (input) The upper or low er triangle of the H erm tian
\(m\) atrix A, packed colum nw ise in a linear array.
The \(j\) th column of A is stored in the array A as
follows: if UPLO = U', A \((i+(j-1) * j 2)=A(i, 7)\)
for \(1<=i<=j\) ifUPLO \(=\mathrm{L}\) ', A \((i+(j-1) *(2 n-j) / 2)\)
\(=A(i, j)\) for \(j=i<=n\).

\section*{AF (input)}

The triangular factor \(U\) or \(L\) from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) orA \(=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\), as com puted by SPPTRF \(/ C P P T R F\), packed collm nw ise in a linear anay in the sam e form atas A (see A).
\(B\) (input) The righthand side m atrix \(B\).

LD B (input)
The leading dim ension of the array \(B\). LD B >= max (1,N).
\(X\) (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by CPPTRS. On exit, the im proved solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the array X . LD X >= \(\max (1, \mathbb{N})\).

FERR (output)
The estim ated forw ard enrorbound for each solution vector \(X()\) ) the \(j\) th colum \(n\) of the solution matrix X). If XTRUE is the true solution comesponding to \(X(\mathcal{O}), \operatorname{FERR}(\mathcal{O})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\nu)\)-XTRU \(E\) ) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{)}\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vectorX (i) (ie., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension ( N )
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zppsv - com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPPSV (UPLO,N,NRHS,A,B,LDB,INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*),B (LDB,*)
INTEGERN,NRHS,LDB,INFO
SU BROUTINE ZPPSV_64(UPLO,N,NRHS,A,B,LDB,INFO)
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*),B (LDB,*)
INTEGER*8N,NRHS,LDB,INFO

```

\section*{F95 INTERFACE}
```

SU BROUTINE PPSV (UPLO,N, NRHS],A,B, [LDB], [ $\mathbb{N} F O$ ])
CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D $\mathbb{M}$ ENSION (:) ::A
COM PLEX (8),D IM ENSION (:,:) :: B
$\mathbb{N}$ TEGER :: $\mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} \mathrm{B}, \mathbb{N}$ FO
SU BROUTINE PPSV_64 (UPLO,N, $\mathbb{N} R H S], A, B,[L D B],[\mathbb{N F O}])$
CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D $\mathbb{M}$ ENSION (:) ::A
COM PLEX (8), D $\mathbb{M}$ ENSION (: : : : : B
$\mathbb{N}$ TEGER (8) :: N ,NRHS,LDB, $\mathbb{N} F O$

```

\section*{C INTERFACE}
\#include <sunperfh>
void zppsv (charuple, intn, int nrhs, doublecom plex *a, doublecom plex *b, int ldb, int *info);
void zppsv_64 (charuplo, long n, long nrhs, doublecom plex
*a, doublecom plex *b, long ldo, long *info);

\section*{PURPOSE}
zppsv com putes the solution to a com plex system of linear equations
\(A * X=B\), where \(A\) is an \(N\)-by \(-N\) H erm itian positive defintiem atrix stored in packed form at and X and B are N -by-N RH S \(m\) atrices.

The Cholesky decom position is used to factorA as
\[
\begin{aligned}
& A=U * * H * U, \text { if } U P L O=U ' \text { or } \\
& A=L * L * * H, \text { if } U P L O=L ',
\end{aligned}
\]
where \(U\) is an uppertriangularm atrix and \(L\) is a low er triangular \(m\) atrix. The factored form of \(A\) is then used to solve the system ofequations \(A *=B\).

\section*{ARGUMENTS}

UPLO (input)
= U ': U pper triangle ofA is stored;
= L': Low er triangle ofA is stored.
N (input) The num ber of linearequations, i.e., the order of them atrix \(A . N>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colm ns of them atrix B. NRHS \(>=0\).

A (input/output)
On entry, the upper or low ertriangle of the Her\(m\) tian \(m\) atrix A, packed colum nw ise in a linear array. The jth column of A is stored in the array A as follows: if UPLO \(=U U^{\prime}, A(i+(j\)
 \((j-1) *(2 n-j / 2)=A(i, 7)\) for \(\dot{j}=i<=n\). See below for further details.

On exit, if \(\mathbb{N} F O=0\), the factor \(U\) orL from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\), in the sam e storage form at as A.

B (input/output)
On entry, the N -by -NRH S righthand side m atrix B. On exit, if \(\mathbb{N F O}=0\), the N boy \(-\mathrm{NRH} S\) solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the i-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N}\) FO \(=i\), the leading \(m\) inoroforder iof \(A\) is notposilive definite, so the factorization could notbe com pleted, and the solution has not been com puted.

\section*{FURTHER DETAILS}

The packed storage schem e is illustrated by the follow ing
exam ple w hen \(N=4, \mathrm{UPLO}=\mathrm{U}\) ':

Tw o-dim ensionalstorage of the H erm itian m atrix A :
```

a11 a12 a13 a14
a22 a23 a24
a33 a34 (aij= congj (aï))
a44

```

Packed storage of the upper triangle ofA:
\(A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]\)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zppsvx -use the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) to com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPPSVX (FACT,UPLO,N,NRHS,A,AF,EQUED,S,B,LDB,}
X,LDX,RCOND,FERR,BERR,WORK,W ORK2,INFO)
CHARACTER * 1FACT,UPLO,EQUED
DOUBLE COM PLEX A (*),AF (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
DOUBLE PRECISION RCOND
DOUBLE PRECISION S (*),FERR (*),BERR (*),W ORK 2 (*)
SUBROUT\mathbb{NE ZPPSVX_64(FACT,UPLO,N,NRHS,A,AF,EQUED,S,B,}
LDB,X,LDX,RCOND,FERR,BERR,WORK,W ORK 2, INFO)
CHARACTER * 1 FACT,UPLO,EQUED
DOUBLE COM PLEXA (*),AF (*),B (LDB,*),X (LDX ,*),W ORK (*)
\mathbb{NTEGER*8N,NRHS,LDB,LDX,}\mathbb{N}FO
DOUBLE PRECISION RCOND
D OUBLE PRECISION S (*),FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE PPSVX (EACT,UPLO, \(\mathbb{N}], \mathbb{N} R H S], A, A F, E Q U E D, S, B\), [LDB], \(\mathrm{X},[\mathrm{LD} \mathrm{X}], R C O N D, F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N F O}])\)

CHARACTER (LEN=1) ::FACT,UPLO,EQUED
COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::A,AF,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::B,X
\(\mathbb{N}\) TEGER : \(: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} \mathrm{B}, \mathrm{LD} \mathrm{X}, \mathbb{N} \mathrm{FO}\)
REAL (8) :: RCOND
REAL (8), D \(\mathbb{M} E N S I O N(:):: S, F E R R, B E R R, W\) ORK 2

SUBROUTINE PPSVX_64 (FACT, UPLO, \(\mathbb{N}], \mathbb{N} R H S], A, A F, E Q U E D, S, B\), \([\) [LD \(], \mathrm{X},[\mathrm{LD} \mathrm{X}], R C O N D, F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::FACT, UPLO, EQUED
COMPLEX (8), D \(\mathbb{I M} E N S I O N(:):: A, A F, W O R K\)
COM PLEX (8), D \(\mathbb{M}\) ENSION (: \(:\) : : : B , X
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDB,LDX, \(\mathbb{N}\) FO
REAL (8) :: RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::S,FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void zppsvx (char fact, charuplo, int n, int nins, doublecom plex *a, doublecom plex *af, char equed, double *s, doublecom plex *b, int ldb, doublecom plex *x, int ldx, double *roond, double *ferr, double *berr, int *info);
void zppsvx_64 (char fact, char uplo, long n, long nrhs, doublecom plex *a, doublecom plex *af, charequed, double *s, doublecom plex *b, long ldb, doublecom plex *x, long ldx, double *roond, double *ferr, double *berr, long *info);

\section*{PURPOSE}
zppsvx uses the C holesky factorization \(A=U * * H * U\) or \(A=\) L*L**H to com pute the solution to a com plex system of linear equations
\(A * X=B\), where \(A\) is an \(N\) boy \(N\) H erm itian positive defintem atrix stored in packed form atand \(X\) and \(B\) are \(N\) boy-N RHS m atrices.

Emorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT = E', real scaling factors are computed to equilibrate
the system :
\(\operatorname{diag}(S) * A * \operatorname{diag}(S) * \operatorname{inv}(\operatorname{diag}(S)) * X=\operatorname{diag}(S) * B\)
W hether or not the system w illbe equilibrated depends on the
scaling of the m atrix A , but ifequilibration is used, A
overw rilten by diag \((\mathrm{S}) \star A\) *diag \((\mathrm{S})\) and B by diag \((\mathrm{S}) \star\) B .
2. IfFACT = N 'or E', the Cholesky decom position is used to
factor them atrix A (afterequilibration ifFACT =E)
as
\(A=U{ }^{\star} U\), if \(U P L O=U\) ', or
\(A=L * L '\) if \(U P L O=L \prime\) ',
\(w\) here \(U\) is an upper triangularm atrix, \(L\) is a low er triangular
m atrix, and 'indicates conjugate transpose.
3. If the leading iboy-iprincipal \(m\) inor is not positive definite,
then the routine retums \(w\) ith \(\mathbb{N F O}=\) i. O therw ise, the factored
form ofA is used to estim ate the condition num ber of the \(m\) atrix
A. If the reciprocal of the condition num ber is less than \(m\) achine
precision, \(\mathbb{N} F O=N+1\) is retumed as a w aming, but the routine
stillgoes on to solve forX and com pute emorbounds as described below .
4. The system ofequations is solved forX using the factored form
of A.
5. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for it.
6. If equilibration \(w\) as used, the \(m\) atrix \(X\) is prem ultiplied by
diag (S) so that it solves the original system before equilibration.

\section*{ARGUMENTS}

FACT (input)
Specifies w hether ornotthe factored form of the \(m\) atrix \(A\) is supplied on entry, and if not, whether them atrix A should be equilibrated before it is factored. = F': On entry, AF contains the fac-
tored form ofA. IfEQUED \(=Y\) ', them atrix \(A\) has been equilibrated \(w\) ith scaling factors given by \(S\). A and A F w illnotbe m odified. = N ': The m atrix A w illibe copied to A F and factored.
\(=\mathrm{E}\) : The matrix A w ill be equilibrated if necessary, then copied to AF and factored.

UPLO (input)
= U : U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The num ber of linear equations, i.e., the order of the matrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrices B and X. NRHS >=0.
A (input/output)
O \(n\) entry, the upper or low er triangle of the Her \(m\) itian matrix A, packed colum nw ise in a linear array, except ifFACT \(=F^{\prime}\) and \(E Q U E D=Y '\), then A \(m\) ust contain the equilibrated \(m\) atrix diag ( S ) \(\mathrm{A}^{\mathrm{A}}\) *diag ( S ). The jth column ofA is stored in the array A as follow s: if UPLO = U', A (i+ \((j-1) * \dot{j} 2)=A(i, 7)\) for \(1<=i<=j\) if \(U P L O=L^{\prime}\) ', A \((i+(j-1) *(2 n-j) / 2)=A(i, j)\) for \(j=i<=n\). See below for further details. A is not \(m\) odified if FACT = F' or \(\mathrm{N}^{\prime}\), orifFACT = E'andEQUED = N 'on exit.

On exit, ifFACT = E' and EQUED = \(\mathrm{Y}^{\prime}\), A is overw rilten by diag \((S) * A * d i a g(S)\).

AF (input/output)
If \(F A C T=F '\), then \(A F\) is an inputargum ent and on entry contains the triangular factorU orL from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{H}\), in the sam e storage form at as A. IfEQ U ED ne. \(N\) ', then \(A F\) is the factored form of the equilibrated \(m\) atrix \(A\).

IfFACT = N ', then AF is an output argum ent and on exit retums the triangular factor \(U\) or \(L\) from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) of the originalm atrix A.

If \(F A C T=E\) ', then \(A F\) is an output argum ent and on exit retums the triangular factor \(U\) or \(L\) from the Cholesky factorization \(\mathrm{A}=\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\) \(L * L * * H\) of the equilibrated \(m\) atrix \(A\) (see the
description of \(A\) for the form of the equilibrated \(m\) atrix).

EQUED (input)
Specifies the form of equilibration thatw as done.
= N ': N o equilibration (alw ays true ifFA C T = N \({ }^{1}\).
= Y': Equilibration w as done, i.e., A has been replaced by diag (S) * A * diag (S). EQUED is an inputargum ent if \(\mathrm{FACT}=\mathrm{F}\) '; otherw ise, it is an outputargum ent.

S (input/output)
The scale factors forA; not accessed if \(\mathrm{EQUED}=\) \(\mathrm{N}^{\prime} . \mathrm{S}\) is an inputargum entifFACT = F '; otherw ise, S is an outputargum ent. IfFACT \(=\mathrm{F}^{\prime}\) and EQUED = \(Y\) ', each elem entofs m ustbe positive.

B (input/output)
On entry, the N -by-NRH S righthand sidem atrix B. On exit, if EQUED \(=N\) ', \(B\) is notm odified; if EQUED = Y',B is overw rilten by diag \((S)\) * \(B\).

LD B (input)
The leading dim ension of the anay B . LD B >= max (1,N).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=N+1\), the N -by-NRHS solution
\(m\) atrix \(X\) to the original system ofequations.
\(N\) ote that if EQ UED \(=Y\) ', \(A\) and \(B\) are \(m\) odified on exit, and the solution to the equilibrated system is inv (diag (S))*X .

LD X (input)
The leading dim ension of the array X . LD X >= \(\max (1, N)\).

RCOND (output)
The estim ate of the reciprocal condition num berof the matrix A after equilibration (if done). If
RCOND is less than the \(m\) achine precision (in particular, ifRCOND \(=0\) ), the \(m\) atrix is singular to w orking precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO \(>0\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X(\mathcal{)}\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution
conesponding to \(X(\mathcal{O}), \operatorname{FERR}(\mathcal{O})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{D})\)-XTRUE) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true enror.

BERR (output)
The com ponentw ise relative backw ard error of each solution vectorX ( \(\mathcal{j}\) ) (i.e., the sm allest relative change in any elem entofA orB thatm akes \(X(\mathcal{j})\) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N F O}=\mathrm{i}\), and i is
\(<=N\) : the leading \(m\) inoroforderiof \(A\) is not positive definite, so the factorization could not be com pleted, and the solution has not been com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1\) : U is nonsingular, but RCOND is less than machine precision, m eaning that the \(m\) atrix is singularto \(w\) orking precision. Nevertheless, the solution and error bounds are com puted because there are a num ber of situationsw here the com puted solution can bem ore accurate than the value ofRC OND w ould suggest.

\section*{FURTHER DETAILS}

The packed storage schem e is illustrated by the follow ing exam ple when \(\mathrm{N}=4\), UPLO = U':

Tw o-dim ensional storage of the \(H\) erm itian m atrix A :
```

a11 a12 a13 a14
a22 a23 a24
a33 a34 (aij= conjg (aji))
a44

```

Packed storage of the upper triangle ofA :

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zpptrf-com pute the C holesky factorization of a com plex H erm itian positive definite m atrix A stored in packed form at

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPPTRF (UPLO,N,A , \mathbb{NFO )}}\mathbf{N}=(
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*)
INTEGERN,\mathbb{NFO}
SU BROUT\mathbb{NE ZPPTRF_64 (UPLO,N,A,\mathbb{NFO)}}\mathbf{~}\mathrm{ ( }
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*)
\mathbb{NTEGER*8N,INFO}
F95 INTERFACE

```

```

CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D IM ENSION (:) ::A
INTEGER::N,\mathbb{NFO}
SUBROUTINE PPTRF_64 (UPLO,N,A,[\mathbb{NFO ])}
CHARACTER (LEN=1)::UPLO
COM PLEX (8),D IM ENSION (:) ::A
\mathbb{NTEGER (8)::N,\mathbb{NFO}}\mathbf{N}=\mp@code{N}

```
void zpptrf_64 (charuplo, long n, doublecom plex *a, long *info);

\section*{PURPOSE}
zpptrf com putes the Cholesky factorization of a com plex H er\(m\) itian positive definite \(m\) atrix A stored in packed form at.

The factorization has the form
\(A=U * * H * U\), if \(U P L O=U '\), or
\(A=L * L \star * H\), if \(U P L O=L '\),
\(w\) here \(U\) is an upper triangular \(m\) atrix and \(L\) is lower triangular.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) : U pper triangle ofA is stored;
\(=\mathbb{L}\) ': Low er triangle of A is stored.

N (input) The order of them atrix A. \(\mathrm{N}>=0\).

A (input/output)
O n entry, the upper or low er triangle of the H er\(m\) tian \(m\) atrix A, packed colum nw ise in a linear array. The \(j\) th colum n of \(A\) is stored in the amay A as follows: if UPLO = U', A (i+ (j 1) \({ }^{j} 2\) ) \(=A(i, j)\) for \(1<=i<=j\) ifUPLO \(=\mathbb{L}^{\prime}, A(i+\) \((j-1)^{\star}(2 n-7 / 2)=A(i, 7)\) for \(\dot{j}=i<=n\). See below for furtherdetails.

On exit, if \(\mathbb{N F O}=0\), the triangular factor \(U\) orL from the Cholesky factorization \(A=U * * H * U\) or \(A=\) L *L **H, in the sam e storage form atas A.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the i-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the leading \(m\) inor of order is notpositive definite, and the factorization could notbe com pleted.

\section*{FURTHER DETAILS}

The packed storage scheme is illustrated by the follow ing
exam ple when \(\mathrm{N}=4, \mathrm{UPLO}=\mathrm{U}\) ':

Tw o-dim ensional storage of the \(H\) erm itian \(m\) atrix A :
```

a11 a12 a13 a14
a22 a23 a24
a33 a34 (aij= con`g (aji))
a44

```

Packed storage of the upper triangle ofA :
\[
A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]
\]

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zpptri-com pute the inverse of a com plex H erm itian positive definite \(m\) atrix \(A\) using the Cholesky factorization \(A=\) \(\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) com puted by CPPTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPPTRI(UPLO,N,A, INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*)
\mathbb{NTEGER N,\mathbb{NFO}}\mathbf{~}=0
SU BROUT\mathbb{NE ZPPTRI_64 (UPLO,N,A,NNFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*)
INTEGER*8 N,\mathbb{NFO}

```
F95 INTERFACE
    SU BROUTINE PPTRI(UPLO , N, A, [ \(\mathbb{N F O}\) ])
    CHARACTER (LEN=1) ::UPLO
    COM PLEX (8),D \(\mathbb{I}\) ENSION (:) ::A
    \(\mathbb{N} T E G E R:: N, \mathbb{N F O}\)
    SU BROUTINE PPTRI_64 (UPLO,N,A, [NFO ])
    CHARACTER (LEN=1)::UPLO
    COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::A
    \(\mathbb{N}\) TEGER ( 8 ) :: N, \(\mathbb{N} F O\)
\#include <sunperfh>
void zpptri(charuple, intn, doublecom plex *a, int *info);
void zpptri_64 (charuplo, long n, doublecom plex *a, long *info);

\section*{PURPOSE}
zpptricom putes the inverse of a com plex H erm itian posilive definite \(m\) atrix \(A\) using the Cholesky factorization \(A=\) \(\mathrm{U} * * \mathrm{H} * \mathrm{U}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{~L} * * \mathrm{H}\) com puted by CPPTRF.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) : U pper triangular factor is stored in A;
\(=\mathbb{L}\) ': Low er triangular factor is stored in A.

N (input) The order of them atrix \(A . N>=0\).

A (input/output)
O n entry, the triangular factor \(U\) or \(L\) from the Cholesky factorization \(A=U * * H * U\) or \(A=L * L * * H\), packed colum nw ise as a linear anray. The jth colum \(n\) of \(U\) orL is stored in the array A as fol lows: ifUPLO \(=U\) ', A \((i+(j 1) \star j 2)=U(i, j)\)
for \(1<=i<=j\) if UPLO \(=L \prime A(i+(j-1) *(2 n-j) / 2)\)
\(=L(i, j)\) for \(j=i<=n\).

On exit, the upper or low er triangle of the (Herm itian) inverse of \(A\), overw riting the input factor
U orL.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the ( \((1, i)\) elem entof the factor
U orL is zero, and the inverse could not.be com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zpptrs -solve a system of linear equations \(A * X=B\) w th a Herm itian positive definite \(m\) atrix \(A\) in packed storage using the Cholesky factorization \(A=U * * H * U\) orA \(=L * L * * H\) com puted by CPPTRF

\section*{SYNOPSIS}
```

SUBROUTINE ZPPTRS(UPLO,N,NRHS,A,B,LDB,\mathbb{NFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*),B (LDB ,*)
\mathbb{NTEGER N,NRHS,LDB,INFO}
SUBROUT\mathbb{NE ZPPTRS_64(UPLO,N,NRHS,A,B,LDB,NNFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*),B (LDB,*)
INTEGER*8N,NRHS,LDB,INFO

```

\section*{F95 INTERFACE}
```

SU BROUTINE PPTRS (UPLO,N, NRHS],A,B,[LDB],[NFO])
CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D $\mathbb{I M}$ ENSION (:) ::A
COM PLEX (8),D IM ENSION (:,:) ::B
$\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O$
SU BROUTINE PPTRS_64 (UPLO ,N, $\mathbb{N} R \mathrm{R}$ S], A, B, [LDB], [ $\mathbb{N} F \mathrm{FO}$ ])
CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D $\mathbb{I M}$ ENSION (:) ::A
COM PLEX (8),D IM ENSION (: : : : : B

```
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDB, \(\mathbb{N}\) FO

\section*{C INTERFACE}
\#include <sunperfh>
void zpptrs (charuplo, intn, int nrhs, doublecom plex *a, doublecom plex *b, int ldlo, int *info);
void zpptrs_64 (charuplo, long n, long nrhs, doublecom plex *a, doublecom plex *b, long ldlo, long *info);

\section*{PURPOSE}
zpptrs solves a system of linear equations \(A * X=B\) with \(a\) H erm itian positive definite \(m\) atrix \(A\) in packed storage using the Cholesky factorization \(A=U * * H * U\) orA \(=L \star L * * H\) com puted by CPPTRF.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}\) : U pper triangle ofA is stored;
\(=\mathbb{L}\) ': Low er triangle of \(A\) is stored.

N (input) The order of the matrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRH S \(>=0\).

A (input) The triangular factorU or L from the Cholesky
factorization \(A=U * * H * U\) or \(A=L * L * * H\), packed
colum nw ise in a linearanray. The jth colum n of
U or L is stored in the aray A as follow s: if \(\mathrm{UPLO}=\mathrm{U}\) ', \(\mathrm{A}(i+(j-1) \star j 2)=\mathrm{U}(i, j)\) for \(1<=i<=j\) if UPLO \(=L \prime\) ' \(A(i+(j-1) *(2 n-j) / 2)=L(i, 7)\) for \(j=i<=n\) 。

B (input/output)
O \(n\) entry, the righthand side m atrix B. On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
< 0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zptcon - com pute the reciprocal of the condition num ber (in the 1 -norm ) of a com plex H erm titian positive definite tridiagonalm atrix using the factorization \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\) or \(\mathrm{A}=\) \(\mathrm{U} * * \mathrm{H} * \mathrm{D} * \mathrm{U}\) com puted by CPTTRF

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZPTCON N,D IAG,OFFD,ANORM ,RCOND,W ORK,NNFO)}
DOUBLE COM PLEX OFFD (*)
\mathbb{NTEGER N,}\mathbb{N}FO
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION DIAG(*),W ORK(*)
SU BROUTINE ZPTCON_64 N,D IAG,OFFD,ANORM,RCOND,W ORK,\mathbb{NFO )}
DOUBLE COM PLEX OFFD (*)
INTEGER*8N,\mathbb{NFO}
DOUBLE PRECISION ANORM,RCOND
DOUBLE PRECISION DIAG (*),WORK(*)

```

\section*{F95 INTERFACE}

SUBROUTINEPTCON ( \(\mathbb{N}], D \mathbb{A} G, O F F D, A N O R M, R C O N D,[\mathbb{N} O R K],[\mathbb{N} F O])\)

COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::OFFD
\(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
REAL (8) ::ANORM,RCOND
REAL (8),D IM ENSION (:) ::D IA G,W ORK
SU BROUTINEPTCON_64 ( \(\mathbb{N}], D \mathbb{I} G, O F F D, A N O R M, R C O N D,[W O R K],[\mathbb{N} F O])\)

\section*{C INTERFACE}
\#include <sunperfh>
void zptoon (intn, double *diag, doublecom plex *offd, double anorm, double *rcond, int *info);
void zptoon_64 (long n, double *diag, doublecom plex *offd, double anorm , double *rcond, long *info);

\section*{PURPOSE}
zptcon com putes the reciprocal of the condition num ber (in the 1 -norm ) of a com plex \(H\) erm tian positive definite tridiagonalm atrix using the factorization \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\) or \(\mathrm{A}=\) \(\mathrm{U} * * \mathrm{H} * \mathrm{D} * \mathrm{U}\) com puted by CPTTRF.

Norm (inv (A )) is com puted by a direct method, and the reciprocal of the condition num ber is com puted as
```

RCOND =1 / (ANORM * nom (inv (A))).

```

\section*{ARGUMENTS}

N (input) The order of them atrix A. \(\mathrm{N}>=0\).

D IA G (input)
Then diagonalelem ents of the diagonal \(m\) atrix
D IA G from the factorization of \(A\), as com puted by
CPTTRF.

OFFD (input)
The ( \(n-1\) ) off-diagonalelem ents of the unit bidiagonal factorU orL from the factorization ofA, as com puted by CPTTRF .

ANORM (input)
The 1 -norm of the originalm atrix A.

RCOND (output)
The reciprocal of the condition num ber of the \(m\) atrix \(A\), com puted as RCOND \(=1 /(A N O R M * A \mathbb{N} V N M)\), where \(A \mathbb{I N V N M}\) is the 1 -norm of inv ( \(A\) ) com puted in this routine.

W ORK (w orkspace)
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

Them ethod used is described in N icholas J . H igham, "E fficient A lgorithm s for C om puting the C ondition N um berof a TridiagonalM atrix", SIA M J.Sci.Stat. C om put., V ol. 7, No. 1, January 1986.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zpteqr-com pute alleigenvalues and, optionally, eigenvectors of a sym \(m\) etric positive definite tridiagonalm atrix by first factoring the \(m\) atrix using SPTTRF and then calling CBD SQ R to com pute the singularvalues of the bidiagonal factor

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPTEQR(COMPZ,N,D,E,Z,LDZ,W ORK,\mathbb{NFO)}}\mathbf{N}=(
CHARACTER * 1 COM PZ
DOUBLE COM PLEX Z (LD Z,*)
\mathbb{NTEGER N,LD Z,}\mathbb{N}FO
DOUBLE PRECISION D (*),E (*),W ORK (*)
SUBROUT\mathbb{NE ZPTEQR_64(COMPZ,N,D,E,Z,LD Z,W ORK,INFO)}
CHARACTER * 1 COMPZ
DOUBLE COM PLEX Z (LD Z,*)
\mathbb{NTEGER*8N,LD Z,NNFO}
DOUBLE PRECISION D (*),E (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE PTEQR COMPZ, \(\mathbb{N}], D, E, Z,[L D Z],[W O R K],[\mathbb{N} F O])\)
CHARACTER (LEN=1) ::COM PZ
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) :: Z
\(\mathbb{N} T E G E R:: N, L D Z, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK
SU BROUTINE PTEQR_64 (COMPZ, \(\mathbb{N}], D, E, Z,[L D Z],[W O R K],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::COMPZ
COM PLEX (8), D IM ENSION (:,:) :: Z
\(\mathbb{N}\) TEGER (8) :: N, LD Z, \(\mathbb{N}\) FO
REAL (8),D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zpteqr(charcom pz, intn, double *d, double *e, doublecom plex *z, int ldz, int*info);
void zpteqr_64 (charcom pz, long n, double *d, double *e, doublecom plex *z, long ldz, long *info);

\section*{PURPOSE}
zpteqr com putes alleigenvalues and, optionally, eigenvectors of a sym \(m\) etric posilive definite tridiagonalm atrix by first factoring the \(m\) atrix using SPTTRF and then calling CBD SQR to com pute the singularvalues of the bidiagonal factor.

This routine com putes the eigenvalues of the positive definte tridiagonal m atrix to high relative accuracy. This m eans that if the eigenvalues range overm any orders ofm agnitude in size, then the sm alleigenvalues and comesponding eigenvectors \(w\) illbe com puted m ore accurately than, for exam ple, w ith the standard \(Q R\) m ethod.

The eigenvectors of a fullorband positive definite Herm itian \(m\) atrix can also be found ifCHETRD, CHPTRD, orCHBTRD has been used to reduce this \(m\) atrix to tridiagonal form . (The reduction to tridiagonal form, how ever, \(m\) ay preclude the possibility of obtaining high relative accuracy in the sm all eigenvalues of the originalm atrix, if these eigenvalues range overm any orders ofm agnitude.)

\section*{ARGUMENTS}

COM PZ (input)
= N ': C om pute eigenvalues only .
\(=\mathrm{V}\) : C om pute eigenvectors of originalH erm itian \(m\) atrix also. A may \(Z\) contains the unitary \(m\) atrix used to reduce the originalm atrix to tridiagonal form . = I': C om pute eigenvectors of tridiagonal \(m\) atrix also.

N (input) The order of the m atrix. \(\mathrm{N}>=0\).
D (input/output)
O n entry, the \(n\) diagonalelem ents of the tridiagonalm atrix. On norm alexit, D contains the eigenvalues, in descending order.

E (input/output)
O \(n\) entry, the ( \(n-1\) ) subdiagonal elem ents of the tridiagonal matrix. On exit, E hasbeen destroyed.

Z (input) \(O n\) entry, if \(C O M P Z=V\) ', the unitary \(m\) atrix used in the reduction to tridiagonal form. On exit, if \(C O M P Z=V\) ', the orthonorm aleigenvectors of the original Herm tian matrix; if COM PZ = I', the orthonorm al eigenvectors of the tridiagonal \(m\) atrix. If \(\mathbb{N} F O>0\) on exit, \(Z\) contains the eigenvectors associated with only the stored eigenvalues. If COMPZ \(=N\) ', then \(Z\) is not referenced.

LD Z (input)
The leading dim ension of the amray Z . LD \(\mathrm{Z}>=1\), and ifCOM PZ \(=V\) 'or \(I\) ', LD \(Z>=\max (1, N)\).

W ORK (w orkspace)
dim ension ( \(4 * \mathrm{~N}\) )
\(\mathbb{N} F O\) (output)
= 0: successfulexit.
< \(0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue.
> 0 : if \(\mathbb{N} F O=i\), and \(i\) is: <= \(N\) the Cholesky factorization of the \(m\) atrix could notbe perform ed because the \(i\)-th principalm inorw as not positive definite. > N the SVD algorithm failed to converge; if \(\mathbb{N} F O=N+\) i, ioff-diagonal elem ents of the bidiagonal factor did not converge to zero.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zptrfs -im prove the com puted solution to a system of linear equations when the coefficientm atrix is \(H\) erm itian positive definite and tridiagonal, and provides error bounds and backw ard errorestim ates for the solution

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZPTRFS (UPLO,N,NRHS,D IAG,OFFD,D IA GF,OFFDF,B,LDB,X,}
LDX,FERR,BERR,W ORK,W ORK2, NNFO)
CHARACTER * 1 UPLO
DOUBLE COM PLEX OFFD (*),OFFDF (*), B (LDB,*), X (LDX,*),
W ORK (*)
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
DOUBLE PRECISION DIAG (*), DIAGF (*), FERR (*), BERR (*),
W ORK2(*)
SUBROUT\mathbb{NE ZPTRFS_64 (UPLO,N,NRHS,D IAG,OFFD,D IAGF,OFFDF,B,LDB,}
X,LDX,FERR,BERR,W ORK,W ORK 2, INFO)
CHARACTER * 1 UPLO
DOUBLE COM PLEX OFFD (*), OFFDF (*), B (LDB,*), X (LDX,*),
W ORK (*)
INTEGER*8N,NRHS,LDB,LDX,\mathbb{NFO}
DOUBLE PRECISION DIAG (*), DIAGF (*), FERR (*), BERR (*),
W ORK2(*)

```

\section*{F95 INTERFACE}

SU BROUTINEPTRFS (UPLO, \(\mathbb{N}], \mathbb{N} R H S], D \mathbb{I} G, O F F D, D \mathbb{A} G F, O F F D F, B,[L D B]\), X, [LDX],FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1)::UPLO

COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::OFFD,OFFDF,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: \(:\) : : : B , X
\(\mathbb{N}\) TEGER :: N, NRHS,LDB,LDX, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D IA G, D IAGF,FERR,BERR,WORK2

SU BROUT \(\mathbb{N} E\) PTRFS_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], D \mathbb{I A} G, O F F D, D I A G F, O F F D F, B\), [LDB],X, [LDX ], FERR, BERR, [W ORK], [W ORK2], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1) :: UPLO
COM PLEX (8), D \(\mathbb{M} E N S I O N(:):\) OFFD , OFFDF,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : B , X
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LD B, LD X , \(\mathbb{N}\) FO
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D IA G, D IA GF,FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void zptrfs (char uplo, intn, intnrhs, double *diag, doublecom plex *offd, double *diagf, doublecom plex *offdf, doublecom plex *b, int ldb, doublecom plex \({ }^{*} x\), int \(l d x\), double *ferr, double *berr, int *info);
void zptrfs_64 (charuplo, long n, long nms, double *diag, doublecom plex *offf, double *diagf, doublecom plex *offdf, doublecom plex *b, long ldb, doublecom plex *x, long ldx, double *ferr, double *berr, long *info);

\section*{PURPOSE}
zptrfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is \(H\) erm itian posilive definite and tridiagonal, and provides error bounds and backw ard errorestim ates for the solution.

\section*{ARGUMENTS}
```

UPLO (input)
Specifies w hether the superdiagonal or the subdiagonal of the tridiagonalm atrix A is stored and the form of the factorization:
$=\mathrm{U}$ : OFFD is the superdiagonalofA, and $\mathrm{A}=$ $\mathrm{U} * * \mathrm{H} * \mathrm{D} \mathbb{I}$ G * U ;
= L': OFFD is the subdiagonal of A , and $\mathrm{A}=$ $\mathrm{L} * \mathrm{D} \operatorname{IA} \mathrm{G} * \mathrm{~L} * * \mathrm{H}$. (The two form sare equivalentifA is real.)

```

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of right hand sides, i.e., the num ber
of colum ns of them atrix B. NRHS \(>=0\).
D IA G (input)
The n realdiagonalelem ents of the tridiagonal \(m\) atrix A.

OFFD (input)
The ( \(n-1\) ) off-diagonalelem ents of the tridiagonal matrix A (see UPLO).
D IA GF (input)
The \(n\) diagonalelem ents of the diagonal \(m\) atrix
D IA G from the factorization com puted by CPTTRF .
OFFDF (input)
The ( \(n-1\) ) off-diagonalelem ents of the unit bidiagonal factor \(U\) orL from the factorization com puted by CPTTRF (see UPLO).
\(B\) (input) The righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the aray B . LD B >= \(\max (1, N)\).

X (input/output)
On entry, the solution \(m\) atrix \(X\), as com puted by CPTTRS. On exit, the im proved solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the array X . LD X >= \(\max (1, N)\).

\section*{FERR (output)}

The forw ard errorbound foreach solution vector X (i) (the \(j\) th colum \(n\) of the solution \(m\) atrix X ). IfXTRUE is the true solution corresponding to \(X(\mathcal{j})\), FERR \((\mathcal{j})\) is an estim ated upperbound for the \(m\) agnitude of the largestelem ent in (X ( \()\)-X TRU E) divided by the \(m\) agnitude of the largestelem ent in \(\mathrm{X}(\mathrm{J})\).

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vectorX ( \(j\) ) (ie., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace) dim ension \((\mathbb{N})\)

W ORK 2 (w orkspace) dim ension \(\mathbb{N}\) )
INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zptsv -com pute the solution to a com plex system of linear equations \(A * X=B\), where \(A\) is an \(N\) by \(-N\) Herm itian positive definite tridiagonalm atrix, and X and B are N -by-NRHS m atrices.

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPTSV N,NRHS,D IAG,SUB,B,LDB,INFO)}
DOUBLE COM PLEX SUB (*),B (LDB,*)
INTEGERN,NRHS,LDB,NNFO
DOUBLE PRECISION DIAG (*)
SUBROUT\mathbb{NE ZPTSV_64 N,NRHS,D IAG,SUB,B,LDB,INFO)}
DOUBLE COM PLEX SUB(*),B (LDB,*)
\mathbb{NTEGER*8N,NRHS,LDB,NNFO}
DOUBLE PRECISION DIAG (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE PTSV ( $\mathbb{N}], \mathbb{N} R H S], D \mathbb{A} G, S U B, B,[L D B],[\mathbb{N} F O])$
COM PLEX (8),D IM ENSION (:) ::SUB
COM PLEX (8),D IM ENSION (:,:) :: B
$\mathbb{N} T E G E R:: N, N R H S, L D B, \mathbb{N} F O$
REAL (8),D $\mathbb{M}$ ENSION (:) ::D IA G
SU BROUTINEPTSV_64 ( $\mathbb{N}], \mathbb{N} R H S], D \mathbb{A} G, S U B, B,[L D B],[\mathbb{N} F O])$
COMPLEX (8),D $\mathbb{I M}$ ENSION (:) ::SUB
COM PLEX (8),D IM ENSION (:,:) :: B
$\mathbb{N}$ TEGER (8) :: N,NRHS,LDB, $\mathbb{N} F O$

```

\section*{C INTERFACE}
\#include <sunperfh>
void zptsv (intn, int nrhs, double *diag, doublecom plex *sub, doublecom plex *b, int ldb, int *info);
void zptsv_64 (long n, long nrhs, double *diag, doublecom plex *sub, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
zptsv com putes the solution to a com plex system of linear equations \(A * X \quad=B\), where \(A\) is an \(N\)-by \(-\mathbb{N}\) Hem itian positive definite tridiagonalm atrix, and X and B are N -by-NRHS m atrices.

A is factored as \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\), and the factored form of A is then used to solve the system of equations.

\section*{ARGUMENTS}

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the m atrix B. NRHS >=0.

D IA G (input/output)
O n entry, the n diagonalelem ents of the tridiagonalm atrix A. On exit, the \(n\) diagonalelem ents of the diagonalm atrix D IA G from the factorization \(A\) \(=\mathrm{L} * \mathrm{D} \mathbb{I A}\) * \(\mathrm{L}^{* *}{ }^{\mathrm{H}}\).

SU B (input/output)
O \(n\) entry, the ( \(n-1\) ) subdiagonal elem ents of the tridiagonalm atrix A. On exit, the ( \(n-1\) ) subdiagonalelem ents of the unitbidiagonal factorL from the L*D IA G *L**H factorization ofA. SUB can also be regarded as the superdiagonal of the unitbidiagonal factor \(U\) from the \(U * * H * D \mathbb{I A} G * U\) factorization of A.

B (input/output)
On entry, the \(\mathrm{N}-\mathrm{by}-\mathrm{N}\) RH S righthand side \(m\) atrix B .
On exit, if \(\mathbb{N F O}=0\), the \(N\)-by-NRH S solution
m atrix X .

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

IN FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N F O}=i\), the leading \(m\) inoroforder \(i\) is not positive definite, and the solution has not been com puted. The factorization has not been com pleted unless \(i=N\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zptsvx - use the factorization \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{H}\) to com pute the solution to a com plex system of linear equations \(A * X=B\), w here A is an N -by N H erm itian positive definite tridiagonal \(m\) atrix and \(X\) and \(B\) are \(N\)-by \(-N\) R H S \(m\) atrices

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPTSVX (FACT,N,NRHS,D IAG,SUB,DIAGF,SUBF,B,LDB,X,}
LDX,RCOND,FERR,BERR,W ORK,W ORK 2, INFO)
CHARACTER * 1 FACT
DOUBLE COM PLEX SUB (*),SUBF (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
DOUBLE PRECISION RCOND
DOUBLE PRECISION DIAG (*), DIAGF (*), FERR (*), BERR (*),
W ORK2(*)
SUBROUT\mathbb{NE ZPTSVX_64\&ACT,N,NRHS,DIAG,SUB,D IAGF,SUBF,B,LDB,}
X,LDX,RCOND,FERR,BERR,W ORK,WORK2, NNFO)
CHARACTER * 1FACT
DOUBLE COM PLEX SUB (*),SUBF (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGER*8N,NRHS,LDB,LDX,\mathbb{NFO}
DOUBLE PRECISION RCOND
DOUBLE PRECISION DIAG (*), DIAGF (*), FERR (*), BERR (*),
W ORK2(*)

```

\section*{F95 INTERFACE}

SU BROUTINE PTSVX \(\mathbb{F A C T}, \mathbb{N}], \mathbb{N} R H S], D \mathbb{I A G}, S U B, D \mathbb{A} G F, S U B F, B,[L D B]\), X, [LDX ],RCOND ,FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N F O}]\) )

COM PLEX (8), D \(\mathbb{M}\) ENSION (:) :: SUB , SUBF,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : B , X
\(\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N} F O\)
REAL (8) ::RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D IA G, D IA GF,FERR,BERR,W ORK 2

SUBROUTINE PTSVX_64 (FACT, \(\mathbb{N}], \mathbb{N R H S}], D I A G, S U B, D I A G F, S U B F, B\), \([\) [LB \(], \mathrm{X},[\llbracket D X], R C O N D, F E R R, B E R R,[\mathbb{O} O K],[W O R K 2],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::FACT
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) :: SUB , SUBF, W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: \(:\) : : : B , X
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LD B , LD X , \(\mathbb{N}\) FO
REAL (8) :: RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) :: D \(\mathbb{A} G, D \mathbb{I A G F}, F E R R, B E R R, W\) ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void zptsvx (char fact, intn, intnrhs, double *diag, doublecom plex *sub, double *diagf, doublecom plex *subf, doublecom plex *b, int ldb, doublecom plex
*x, int ldx, double *roond, double *ferr, double
*berr, int*info);
void zptsvx_64 (char fact, long n, long nrhs, double *diag, doublecom plex *sub, double *diagf, doublecom plex *subf, doublecom plex *b, long ldb, doublecom plex
*x, long ldx, double *roond, double * ferr, double
*berr, long *info);

\section*{PURPOSE}
zptsvx uses the factorization \(A=L * D * L * * H\) to com pute the solution to a com plex system of linear equations \(A * X=B\), where A is an N boy N H erm itian positive definite tridiagonal \(m\) atrix and \(X\) and \(B\) are \(N\) boy- N R H S \(m\) atrioes.

Emorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT \(=N\) ', them atrix \(A\) is factored as \(A=L * D * L * * H\), where L
is a unit low erbidiagonalm atrix and \(D\) is diagonal. The factorization can also be regarded as having the form
\(A=U * * H * D * U\).
2. If the leading iboy-iprincipal \(m\) inor is not positive
definite,
then the routine retums w ith \(\mathbb{N} F O=\) i. O therw ise, the factored
form ofA is used to estim ate the condition num ber of the \(m\) atrix
A. If the reciprocal of the condition num ber is less than \(m\) achine
precision, \(\mathbb{N N}\) FO = N +1 is retumed as a w aming, but the routine
still goes on to solve for X and com pute errorbounds as described below .
3.The system ofequations is solved for \(X\) using the factored form of A.
4. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for it.

\section*{ARGUMENTS}

FACT (input)
Specifies w hether ornot the factored form of the \(m\) atrix \(A\) is supplied on entry. \(=F\) : On entry, \(D I A G F\) and SUBF contain the factored form of \(A\). D IA G, SUB, D IA GF, and SUBF w ill notbe m odified.
\(=N\) ': Them atrixA w illbe copied to D IA GF and SUBF and factored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, ie., the num ber
of collm ns of the m atrices B and X. NRHS >=0.
D IA G (input)
The \(n\) diagonalelem ents of the tridiagonal \(m\) atrix
A.

SU B (input)
The ( \(n-1\) ) subdiagonalelem ents of the tridiagonal \(m\) atrix A.

DIAGF (input/output)
If \(F A C T=F '\), then \(D I A G F\) is an input argum ent and
on entry contains the \(n\) diagonalelem ents of the diagonalm atrix D IA G from the L*D IA G *L**H factorization ofA. IfFACT = N', then D IA GF is an out putargum entand on exitcontains the n diagonal elem ents of the diagonal \(m\) atrix \(D\) IA G from the \(\mathrm{L} * \mathrm{D} \mathbb{I}\) G *L**H factorization of .

SUBF (input/output)
IfFACT = F ', then SUBF is an inputargum ent and on entry contains the ( \(n-1\) ) subdiagonalelem ents of the unit bidiagonal factor \(L\) from the \(\mathrm{L} * \mathrm{D}\) IA \(G * \mathrm{~L}^{* *} \mathrm{H}\) factorization of \(\mathrm{A} . \mathrm{IfFACT}=\mathrm{N}\) ', then SUBF is an outputargum entand on exit contains the ( \(n-1\) ) subdiagonalelem ents of the unit bidiagonal factor \(L\) from the \(L * D I A G * L * * H\) factorization of A.
\(B\) (input) On entry, the N -by-N RH S righthand side m atrix B.
U nchanged on exit.
LD B (input)
The leading din ension of the array \(B\). LD B >= \(\max (1, N)\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=\mathrm{N}+1\), the \(\mathrm{N}-\) by -NRH S solution
\(m\) atrix \(X\).

LD X (input)
The leading dim ension of the amay X . LD X >= \(\max (1, N)\).

RCOND (output)
The reciprocalcondition num berof the \(m\) atrix \(A\). If RCOND is less than them achine precision (in particular, ifRCOND \(=0\) ), the \(m\) atrix is singular to w orking precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO \(>0\).

\section*{FERR (output)}

The forw ard errorbound foreach solution vector \(X\) (i) (the jth colum \(n\) of the solution \(m\) atrix \(X\) ). IfXTRUE is the true solution corresponding to \(X(\mathcal{J}), \operatorname{FERR}(\mathcal{J})\) is an estim ated upperbound for the \(m\) agnitude of the largestelem ent in (X ( \()\)-X TRU E) divided by the \(m\) agnitude of the largestelem ent in X ( \()\).

BERR (output)
The com ponentw ise relative backw ard emorof each
solution vectorX (i) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{j}\) ) an exactsolution).

W ORK (w orkspace)
dim ension (N)
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue
\(>0\) : if \(\mathbb{N} F O=i\), and \(i\) is
\(<=N\) : the leading \(m\) inor oforderiof \(A\) is not positive definite, so the factorization could not be com pleted, and the solution has not been com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1\) : U is nonsingular, butRCOND is less than machine precision, m eaning that the \(m\) atrix is singularto \(w\) orking precision. Nevertheless, the solution and error bounds are com puted because there are a num ber of situationsw here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zpttrf-com pute the L *D *L 'factorization of a com plex Herm tian positive definite tridiagonalm atrix A

\section*{SYNOPSIS}
```

SU BROUTINE ZPTTRF N,DIAG,OFFD,INFO)
DOUBLE COM PLEX OFFD (*)
INTEGERN,\mathbb{NFO}
DOU BLE PRECISION DIAG (*)

```

```

DOUBLE COM PLEX OFFD (*)
\mathbb{NTEGER*8N,INFO}
DOUBLE PRECISION DIAG (*)

```
F95 INTERFACE
    SU BROUTINE PTTRF ( \(\mathbb{N}], D \mathbb{I A G}, O F F D,[\mathbb{N} F O]\) )
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::OFFD
    \(\mathbb{N} T E G E R:: N, \mathbb{N} F O\)
    REAL (8), D \(\mathbb{M}\) ENSION (:) ::D IA G
    SU BROUTINE PTTRF_64 ( \(\mathbb{N}\) ],D IA G , OFFD , [ \(\mathbb{N} F O]\) )
    COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::OFFD
    \(\mathbb{N} T E G E R(8):: N, \mathbb{N F O}\)
    REAL (8), D \(\mathbb{M}\) ENSIO N (:) ::D IA G
C INTERFACE
    \#include < sunperfh>
void zpttrf(intn, double *diag, doublecom plex *offd, int *info);
void zpttrf_64 (long n, double *diag, doublecom plex *offd, long *info);

\section*{PURPOSE}
zpttrf com putes the \(L\) *D *L 'factorization of a com plex Herm itian posilive definite tridiagonalm atrix A. The factorization \(m\) ay also be regarded as having the form \(A=U\) *D *U .

\section*{ARGUMENTS}

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

D IA G (input/output)
O n entry, the \(n\) diagonalelem ents of the tridiagonalm atrix A. On exit, the \(n\) diagonalelem ents of the diagonalm atrix D IA G from the \(\mathrm{L} * \mathrm{D}\) IA G *L' factorization of A.

OFFD (input/output)
O \(n\) entry, the \((n-1)\) subdiagonal elem ents of the tridiagonalm atrix A. On exit, the \((n-1)\) subdiagonalelem ents of the unit.bidiagonal factorL from the \(L * D I A G * L\) ' factorization ofA. OFFD can also be regarded as the superdiagonal of the unitbidiagonal factor \(U\) from the \(U\) *D IA G *U factorization of A.

IN FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N N F}=-\mathrm{k}\), the k -th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=k\), the leading \(m\) inor of orderk is notpositive definite; if \(k<N\), the factorization could notbe com pleted, while if \(k=N\), the factorization w as com pleted, butD \(\mathbb{A} G \mathbb{N})=0\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zpters-solve a tridiagonalsystem of the form \(A * X=B\) using the factorization \(A=U{ }^{*} D * U\) orA \(=L * D * L\) 'com puted by CPTTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPTTRS (UPLO,N,NRHS,D IA G,OFFD,B,LDB, NFO )}
CHARACTER * 1 UPLO
DOUBLE COM PLEX OFFD (*),B (LDB,*)
\mathbb{NTEGER N,NRHS,LDB,INFO}
DOUBLE PRECISION DIAG (*)
SU BROUT\mathbb{NE ZPTTRS_64(UPLO,N ,NRHS,D IA G ,OFFD,B,LDB,INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX OFFD (*),B (LD B ,*)
INTEGER*8N,NRHS,LDB,\mathbb{NFO}
DOUBLE PRECISION DIAG (*)

```

\section*{F95 INTERFACE}

SU BROUTINE PTTRS (UPLO, \(\mathbb{N}], \mathbb{N} R H S], D \mathbb{A} G, O F F D, B,[L D B],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::OFFD
COM PLEX (8),D IM ENSION (:,:) :: B
\(\mathbb{N}\) TEGER ::N,NRHS,LDB, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D IA G

SU BROUTINE PTTRS_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], D \mathbb{A} G, O F F D, B,[L D B],[\mathbb{N} F O])\)

CHARACTER (LEN=1)::UPLO

COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::OFFD
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::B
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDB, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D IA G

\section*{C INTERFACE}
\#include <sunperfh>
void zpttrs (charuplo, intn, intnms, double *diag, doublecom plex *offd, doublecom plex *b, int ldb, int *info);
void zpttrs_64 (charuple, long n, long nrhs, double *diag, doublecom plex *offd, doublecom plex *b, long lalo, long *info);

\section*{PURPOSE}
zpttrs solves a tridiagonal system of the form
\(A * X=B\) using the factorization \(A=U * D * U\) or \(A=\) L *D *L 'com puted by CPTTRF. D is a diagonalm atrix specified in the vectorD, U (orL) is a unitbidiagonalm atrix whose superdiagonal (subdiagonal) is specified in the vector \(E\), and X and B are N by NRH S m atriges.

\section*{ARGUMENTS}

\section*{UPLO (input)}

Specifies the form of the factorization and
w hether the vector OFFD is the superdiagonal of the upperbidiagonal factorU or the subdiagonal of the low er bidiagonal factorL. \(=U\) ': \(A=\) U *D IA G *U , OFFD is the superdiagonal of \(U\) \(=\mathbb{L}\) ': A \(=\mathrm{L} * \mathrm{D}\) IA \(G *\) ' ', OFFD is the subdiagonal of L

N (input) The order of the tridiagonalm atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRH S >=0 .

D IA G (input)
The \(n\) diagonalelem ents of the diagonal \(m\) atrix
\(D I A G\) from the factorization \(A=U\) *D \(I A G * U\) orA \(=\)
L *D IA G *L '.

OFFD (input/output)
If \(U P L O=U\) ', the ( \(n-1\) ) superdiagonalelem ents of
the unit bidiagonal factor \(U\) from the factorization \(A=U\) *D IA G *U. IfUPLO \(=L^{\prime}\) ', the ( \(n-1\) ) subdiagonal elem ents of the unitbidiagonal factor \(L\) from the factorization \(A=L * D \operatorname{IA} G * L\) '.

B (input/output)
On entry, the righthand side vectors \(B\) for the system of linearequations. On exit, the solution vectors, \(X\).

LD B (input)
The leading dim ension of the array \(B\). LD B \(>=\) max ( \(1, N\) ).
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0\) : if \(\mathbb{N} F O=-\mathrm{k}\), the k -th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zptts2 - solve a tridiagonalsystem of the form A * X = B using the factorization \(A=U{ }^{*} D * U\) orA \(=L * D * L\) 'com puted by CPTTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZPTTS2 (UUPLO,N,NRHS,D ,E,B,LDB)}
DOUBLE COM PLEXE (*),B (LDB,*)
INTEGER IUPLO,N,NRHS,LDB
DOUBLE PRECISION D (*)

```

```

DOUBLE COM PLEXE (*),B (LDB,*)
INTEGER*8 \mathbb{UPLO,N,NRHS,LDB}
DOUBLE PRECISION D (*)

```
F95 INTERFACE
    SU BROUTINE ZPTTS2 (UPLO ,N,NRHS,D ,E,B,LDB)
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::E
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::B
    \(\mathbb{N} T E G E R:: \mathbb{U} P L O, N, N R H S, L D B\)
    REAL (8),D \(\mathbb{M}\) ENSION (:) ::D
    SU BROUTINE ZPTTS2_64 ( \(\mathbb{U}\) PLO ,N,NRHS,D ,E,B,LDB)
    COMPLEX (8), D IM ENSION (:) :: E
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::B
    \(\mathbb{N}\) TEGER ( 8 ): : \(\mathbb{U}\) PLO , N,NRHS,LDB
    REAL (8),D \(\mathbb{M}\) ENSION (:) ::D

\section*{C INTERFACE}
\#include < sunperfh>
void zptts2 (int iuplo, intn, int nhs, double *d, doublecom plex *e, doublecom plex *b, int ldb);
void zptts2_64 (long iuplo, long n, long nihs, double *d, doublecom plex *e, doublecom plex *b, long ldb);

\section*{PURPOSE}
zptts2 solves a tridiagonal system of the form
\(A * X=B\) using the factorization \(A=U{ }^{*} D * U\) or \(A=\) L * D *L 'com puted by CPTTRF. D is a diagonalm atrix specified in the vectorD, U (orL) is a unitbidiagonalm atrix whose superdiagonal (subdiagonal) is specified in the vectore, and \(X\) and \(B\) are N by NRH S m atrioes.

\section*{ARGUMENTS}

IUPLO (input)
Specifies the form of the factorization and whether the vectorE is the superdiagonal of the upperbidiagonal factor \(U\) or the subdiagonal of the low erbidiagonal factor \(L .=1: A=U * D * U\), \(E\) is the superdiagonalof \(U\)
\(=0: A=L * D * L\) ' E is the subdiagonalof L
N (input) The order of the tridiagonalm atrix \(\mathrm{A} . \mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRHS \(>=0\).
\(D\) (input) The \(n\) diagonalelem ents of the diagonal \(m\) atrix \(D\)
from the factorization \(A=U{ }^{*} D * U\) orA \(=L * D * L '\).
\(E\) (input) If \(\mathbb{U P L O}=1\), the \((n-1)\) superdiagonalelem ents of the unit bidiagonal factor \(U\) from the factorization \(A=U\) *D *U. If \(\mathbb{I} P L O=0\), the ( \(n-1\) ) subdiagonalelem ents of the unitbidiagonal factorL from the factorization \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L}\).

B (input/output)
On entry, the righthand side vectors \(B\) for the
system of linearequations. On exit, the solution
vectors, X .
LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, N)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zrot-apply a plane rotation, w here the cos (C) is realand the sin (S) is com plex, and the vectors \(X\) and \(Y\) are com plex

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZROT N,X,\mathbb{NCX,Y, INCY,C,S)}}\mathbf{N}=()
DOUBLE COM PLEX S
DOUBLE COM PLEX X (*),Y (*)
\mathbb{NTEGERN,}\mathbb{NCX,\mathbb{NCY}}\mathbf{}\mathrm{ CN}
DOUBLE PRECISION C
SUBROUT\mathbb{NE ZROT_64 N,X,\mathbb{NCX,Y,INCY,C,S)}}\mathbf{N}\mathrm{ ( }
DOUBLE COM PLEX S
DOUBLE COM PLEX X (*),Y (*)
INTEGER*8N,\mathbb{NCX,INCY}
DOUBLE PRECISION C

```

\section*{F95 INTERFACE}
```

SU BROUTINE ROT ( $\mathbb{N}$ ], X, $[\mathbb{N C X}], Y,[\mathbb{N} C Y], C, S)$
COM PLEX (8) ::S
COMPLEX (8),D IM ENSION (:) ::X,Y
$\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y$
REAL (8) ::C
SU BROUTINEROT_64 ( $\mathbb{N}$ ], $\mathrm{X},[\mathbb{N} C X], Y,[\mathbb{N} C Y], C, S)$
COM PLEX (8) ::S
COM PLEX (8),D $\mathbb{M}$ ENSION (:) ::X,Y
$\mathbb{N} \operatorname{TEGER}(8):: N, \mathbb{N} C X, \mathbb{N} C Y$

```

REAL (8) ::C

\section*{C INTERFACE}
\#include < sunperfh>
void zrot(intn, doublecom plex *x, int incx, doublecom plex
*y, int incy, double c, doublecom plex *s);
void zrot 64 (long n, doublecom plex *x, long incx, doublecom plex *y, long incy, double c, doublecom plex *s);

\section*{PURPOSE}
zrot applies a plane rotation, w here the cos (C) is real and the \(\sin (S)\) is com plex, and the vectors \(X\) and \(Y\) are com plex.

\section*{ARGUMENTS}

N (input)
The num berof elem ents in the vectors X and Y .

X (input/output)
On input, the vector \(X\). On output, \(X\) is overw ritten \(w\) ith \(C * X+S * Y\).
\(\mathbb{N C X}\) (input)
The increm ent betw een successive values of \(Y\).
\(\mathbb{N} C X<>0\).

Y (input/output)
On input, the vector \(Y\). On output, \(Y\) is overw ritten with CONJG (S)*X + C*Y.
\(\mathbb{N C Y}\) (input)
The increm ent betw een successive values of \(Y\).
\(\mathbb{N} C Y\) <> 0 .

C (input)
\(S\) (input)
C and S define a rotation
[ C S ]
[-oonjg (S) C ]
where \(\mathrm{C} * \mathrm{C}+\mathrm{S}{ }^{*} \mathrm{CON} \operatorname{JG}(\mathrm{S})=1.0\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

zrotg -C onstructa G iven S plane rotation

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZROTG (A,B,C,S)}
DOUBLE COM PLEX A,B,S
DOUBLE PRECISION C
SUBROUTINE ZROTG_64(A,B,C,S)
DOUBLE COM PLEX A,B,S
DOUBLE PRECISION C

```
F95 INTERFACE
    SUBROUTINEROTG (A, B , C, S)
    COM PLEX (8) ::A, B, S
    REAL (8) ::C
    SU BROUTINEROTG_64(A,B,C,S)
    COM PLEX (8) ::A,B,S
    REAL (8) ::C
C INTERFACE
    \#include <sunperfh>
    void zrotg (doublecom plex *a, doublecom plex *b, double *c,
        doublecom plex *s);
    void zrotg_64 (doublecom plex *a, doublecom plex *b, double *c,

\section*{PURPOSE}
zrotg C onstructa \(G\) iven \(S\) plane rotation that \(w\) ill annihilate an elem entof a vector.

\section*{ARGUMENTS}

A (input/output)
O n entry, A contains the entry in the firstvector that comesponds to the elem ent to be annihilated in the second vector. On exit, contains the nonzero elem ent of the rotated vector.
B (input)
On entry, B contains the entry to be annihilated in the second vector. U nchanged on exit.

C (output)
On exit, \(C\) and \(S\) are the elem ents of the rotation \(m\) atrix thatw illibe applied to annihilate \(B\).

S (output)
On exit, C and S are the elem ents of the rotation \(m\) atrix thatw ill.be applied to annihilate \(B\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

zscal-C om pute y := alpha * y

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZSCALN,ALPHA,Y,INCY)}
D OUBLE COM PLEX ALPHA
DOUBLE COM PLEX Y (*)
INTEGERN,\mathbb{NCY}
SUBROUT\mathbb{NE ZSCAL_64 N,ALPHA,Y,\mathbb{NCY)}}\mathbf{N}\mathrm{ (S)}
DOUBLE COM PLEX ALPHA
DOUBLE COM PLEX Y (*)
INTEGER*8N,\mathbb{NCY}
F95 INTERFACE

```

```

    COM PLEX (8) ::A LPHA
    COM PLEX (8),D IM ENSION (:) ::Y
    \mathbb{NTEGER ::N,\mathbb{NCY}}\mathbf{N}=\mp@code{N}
    SU BROUTINE SCAL_64 (N ],ALPHA,Y,[\mathbb{N CY ])}
    COM PLEX (8) ::A LPHA
    COM PLEX (8),D IM ENSION (:) ::Y
    INTEGER (8)::N,\mathbb{NCY}
    C INTERFACE
\#include <sunperfh>

```
void zscal(intn, doublecom plex *alpha, doublecom plex *y, intincy);
void zscal 64 (long \(n\), doublecom plex *alpha, doublecom plex *y, long incy);

\section*{PURPOSE}
zscalC om pute \(y:=\) alpha * \(y\) w here alpha is a scalar and \(y\) is an \(n\)-vector.

\section*{ARGUMENTS}

N (input)
O n entry, N specifies the num ber of elem ents in the vector. N m ustbe at least one for the subroutine to have any visible effect. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

Y (input/output)
( \(1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C Y)\) ). On entry, the increm ented array \(Y\) m ustcontain the vectory. On exit, \(Y\) is overw ritten by the updated vectory.
\(\mathbb{N} C Y\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of Y. \(\mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- PURPOSE
- ARGUMENTS

\section*{NAME}
zsctr-Scatters elem ents from \(x\) into \(y\).

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZSCTR NZ,X,INDX,Y)}
DOUBLE COM PLEX X (*),Y (*)
INTEGERNZ
INTEGER \mathbb{NDX (*)}
SUBROUT\mathbb{NE ZSCTR_64 NZ,X,INDX,Y)}
DOUBLE COM PLEX X (*),Y (*)
INTEGER*8NZ
INTEGER*8 INDX (*)
F95 INTERFACE
SUBROUTINE SCTR(NZ],X,NNDX,Y)
COM PLEX (8),D IM ENSION (:) ::X,Y
\mathbb{NTEGER ::NZ}
\mathbb{NTEGER,D IM ENSION (:) :: \mathbb{NDX}}\mathbf{N}=\mp@code{N}
SUBROUT\mathbb{NE SCTR_64(NZ I,X,INDX,Y)}
COM PLEX (8),D IM ENSION (:) ::X,Y
INTEGER (8)::NZ
\mathbb{NTEGER (8),D IM ENSION (:) :: \mathbb{NDX}}\mathbf{N}=\mp@code{l}

```

\section*{PURPOSE}
in fullstorage form .
do \(i=1, n\)
\(y(\) indx (i) \()=x(i)\)
enddo

\section*{ARGUMENTS}

N Z (input) - \(\mathbb{N}\) TEGER
N um ber of elem ents in the com pressed form .
U nchanged on exit.

X (input)
V ector containing the values to be scattered from com pressed form into fill storage form. U nchanged on exit.
\(\mathbb{N} D X\) (input) \(-\mathbb{N} T E G E R\)
\(V\) ector containing the indiges of the com pressed form. It is assum ed that the elem ents in \(\mathbb{N} D \mathrm{X}\) are distinctand greater than zero. U nchanged on exit.

Y (output)
V ectorw hose elem ents specified by indx have been set to the corresponding entries ofx. Only the elem ents corresponding to the indices in indx have been m odified.

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
zskym m -Skyline form atm atrix-m atrix m ultiply

\section*{SYNOPSIS}

SUBROUTINE ZSKYMM (TRANSA, M, N, K,ALPHA,DESCRA, * VAL,PNTR, B,LDB,BETA, C,LDC,WORK,LWORK)
\(\mathbb{N} T E G E R\) TRANSA, M,N,K,DESCRA (5),
* LDB,LDC,LWORK
\(\mathbb{N} T E G E R \quad\) PNTR ( \({ }^{( }\)),
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (NNZ),B (LDB,*), C (LDC,*),W ORK (LW ORK)
SUBROUTINE ZSKYMM_64(TRANSA, M,N,K,ALPHA,DESCRA, * VAL,PNTR, B,LDB,BETA, C,LDC,W ORK,LW ORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, \(\mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{DESCRA}\) (5),
* LDB,LDC,LW ORK
\(\mathbb{N} T E G E R * 8\) PNTR (*) ,
D OUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEXVAL (NNZ), B (LDB,*), C (LDC \(\left.\boldsymbol{c}^{\star}\right), W\) ORK (LW ORK)
where \(N \mathrm{NZ}=\operatorname{PN} \operatorname{TR}(\mathbb{K}+1)\)-PN TR (1) (upper triangular)
NN Z = PNTR (M+1)PNTR (1) (low er triangular)
PN TR 0 size \(=(\mathbb{K}+1)\) (uppertriangular)
PNTR () size \(=(\mathrm{M}+1\) ) (low ertriangular)

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE SKYMM (TRANSA,M, N ],K,ALPHA,DESCRA,VAL,}

* PNTR, B, [LDB],BETA,C, [LDC], [W ORK], [LW ORK])
INTEGER TRANSA,M,K
\mathbb{NTEGER,D IM ENSION (:) :: DESCRA, PNTR}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D IM ENSION(:) :: VAL

```

DOUBLE COM PLEX,D \(\mathbb{I}\) ENSION (: : : :: B, C

SU BROUTINE SKYMM_64(TRANSA, M, \(\mathbb{N}], K, A L P H A, D E S C R A, V A L\),
* PNTR, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
\(\mathbb{N} T E G E R * 8\) TRANSA, M, K
\(\mathbb{N} T E G E R * 8, D \mathbb{M}\) ENSION (:) :: DESCRA, PNTR
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D \(\mathbb{M}\) ENSION (:) :: VAL
DOUBLE COM PLEX,D \(\mathbb{I}\) ENSION (:, :) :: B,C

\section*{DESCRIPTION}
\[
C \text { <-alpha op (A) B + beta C }
\]
where A LPHA and BETA are scalar, \(C\) and \(B\) are dense \(m\) atrices, \(A\) is a \(m\) atrix represented in skyline form at and op(A) is one of \(\mathrm{op}(\mathrm{A})=\mathrm{A}\) or \(\mathrm{op}(\mathrm{A})=A^{\prime}\) or \(\mathrm{op}(\mathrm{A})=\operatorname{conj}\left(\mathrm{A}^{\prime}\right)\).
( 'indicates m atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
0 : operate w ith m atrix
1 : operate w th transpose m atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix. 2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um berof row \(s\) in \(m\) atrix A

N \(\quad N\) um berof colum ns in matrix C
K \(\quad\) Num berof colum ns in matrix A

A LPH A Scalar param eter
DESCRA 0 D escriptor argum ent. Five elem ent integer amay
DESCRA (1) m atrix structure
0 : general (NOT SUPPORTED)
1 : symmetric ( \(A=A\) )
2 : Herm Aian ( \(\mathrm{A}=\mathrm{CONJG}\) (A))
3 :Triangular
4 : Skew (A nti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJ}\) ( A ) )
D ESCRA (2) upper/low er triangular indicator
1 : low er

2 :upper
DESCRA (3) main diagonal type
0 :non-unit
1 : unit
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED)
0 :C C ++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT \(\mathbb{I}\) PLEM ENTED)
0 : unknown
1 : no repeated indices

VAL () array contain the nonzeros of in in skyline profile form . Row -oriented ifD ESCRA (2) = 1 (low er triangular), Colum \(n\) oriented ifD ESCRA (2) \(=2\) (upper triangular).

PN TR ( integer anray of length M +1 (low er triangular) or \(\mathrm{K}+1\) (uppertriangular) such thatPN TR (I) PN TR (1)+1 points to the location in VAL of the firstelem ent of the skyline profile in row (colum n) I.

B 0 rectangular anray w ith first dim ension LD B .
LD B leading dim ension ofB
BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC
W ORK () scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK anay. LW ORK is not referenced in the currentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at:
http://m ath nistgov/n cso/Staff/k Rem ington/fspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S)
Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse .ps

\section*{NOTES/BUGS}

The SK Y data structure is not supported for a generalm atrix structure (DESCRA (1)=0).

A lso not supported:
1. low ertriangularm atrix \(A\) of size \(m\) by \(n\) where \(m>n\)
2. uppertriangularm atrix A of size m by n wherem < n

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
zskysm -Skyline form at triangular solve

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZSKYSM(TRANSA,M,N,UNITD,DV,ALPHA,DESCRA,}

* VAL,PNTR,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,M,N,UNITD,DESCRA (5),}
* LDB,LDC,LW ORK
INTEGER PNTR(*),
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX DV M ),VAL NNZ),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
SUBROUTINE ZSKYSM_64(TRANSA, M,N,UNITD,DV,ALPHA,DESCRA,
* VAL,PNTR,
* B,LDB,BETA, C,LDC,W ORK,LW ORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, M,N,UNITD,DESCRA (5),
* LDB,LDC,LWORK
\(\mathbb{N} T E G E R * 8 \operatorname{PNTR}(\star)\),
D OUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX DV M),VAL NNZ), B (LDB,*), C (LDC ,*), W ORK (LW ORK)
where \(\mathrm{NN} Z=\operatorname{PN} \operatorname{TR}(\mathrm{M}+1)\)-PN TR (1) (uppertriangular)
    \(\mathrm{NN} Z=\operatorname{PNTR}(\mathbb{K}+1)\)-PNTR (1) (low ertriangular)
    PNTR () size \(=(M+1\) ) (uppertriangular)
    PNTR 0 ) size \(=(\mathbb{K}+1)\) (low ertriangular)

\section*{F95 INTERFACE}

SUBROUTINE SKYSM (TRANSA, M, \(\mathbb{N}\) ],UNITD,DV,ALPHA,DESCRA,VAL, * PNTR, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
\(\mathbb{N} T E G E R\) TRANSA,M,UNITD
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:)::\) DESCRA, PNTR

DOUBLE COMPLEX ALPHA,BETA
D OUBLE COM PLEX ,D \(\mathbb{I M} E N S I O N\) (:) :: VAL, DV
DOUBLE COM PLEX ,D \(\mathbb{M}\) ENSION (: : : : : B , C

SU BROUTINE SKYSM _64 (TRANSA , M, \(\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A\),
* VAL,PNTR, B, [LDB],BETA, C, [LDC], [W ORK], [LW ORK])
\(\mathbb{N} T E G E R * 8\) TRANSA, M, UNITD
\(\mathbb{N} T E G E R * 8, D \mathbb{M}\) ENSION (:) :: DESCRA, PNTR
DOUBLECOMPLEX ALPHA,BETA
DOUBLE COM PLEX,D \(\mathbb{I M} E N S I O N(:):: V A L, D V\)
DOUBLE COM PLEX ,D \(\mathbb{M}\) ENSION (: : : : : B , C

\section*{DESCRIPTION}
\[
C<-A L P H A \quad \text { op (A) B + BETA } C \quad C<-A L P H A D \text { op (A) B + BETA C }
\] \(C<-A L P H A \operatorname{Op}(A) D B+B E T A C\)
where A LPHA andBETA are scalar, \(C\) and \(B\) are \(m\) by \(n\) dense \(m\) atrices, \(D\) is a diagonalscaling \(m\) atrix, \(A\) is a unit, ornon-unit, upper or low er triangularm atrix represented in skyline form at and \(o p(A)\) is one of
```

op (A ) = inv (A ) or op (A ) = inv (A ) or op (A ) = inv (oonjg (A')).

``` (inv denotesm atrix inverse, 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRAN SA Indicates how to operate \(w\) th the sparse \(m\) atrix 0 : operate \(w\) ith \(m\) atrix
1 : operate w th transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M \(\quad N\) um ber of row \(s\) in matrix A
\(N \quad N\) um berof colum \(n s\) in \(m\) atrix \(C\)

UN ITD Type of scaling:
1 : Identily \(m\) atrix (argum entD V [] is ignored)
2 : Scale on left (row scaling)
3 : Scale on right (colum \(n\) scaling)
4 :A utom atic row orcolum n scaling (see section N O TES for further details)

DV () A rray of length M containing the diagonalentries of the scaling diagonalm atrix D .

ALPHA Scalarparam eter
```

D ESCRA () D escriptor argum ent. Fi̇ve elem entinteger anay
DESCRA (1) m atrix structure
0 :general
1 : symm etric (A=A)
2:H erm itian (A = CON JG (A ))
3:Triangular
4 : Skew (A nti)-Symm etric (A=-A )
5 :D iagonal
6:Skew Herm itian (A= CON JG (A ) )
N ote: For the routine, D ESCRA (1)=3 is only supported.
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 :non-unit
1 :unit
DESCRA (4) A nay base $\mathbb{N} O T \mathbb{M}$ PLEM ENTED)
$0: C / C++$ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? (NOT $\mathbb{M}$ PLEM ENTED)
0 : unknown
1 : no repeated indices

```

VAL () array contain the nonzeros ofA in skyline profile form .
Row -oriented ifD ESCRA (2) = 1 (low er triangular), colum n oriented ifD ESCRA (2) \(=2\) (upper triangular) .

PN TR () integer array of length \(M+1\) (low ertriangular) or
\(\mathrm{K}+1\) (upper triangular) such thatPN TR (I)-PN TR (1)+1 points to the location in VAL of the firstelem ent of the skyline profile in row (colum n) I.

B 0 rectangular anay w ith first dim ension LD B .

LD B leading dim ension ofB

BETA Scalarparam eter

C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC

W ORK () scratch amay of length LW ORK.
On exit, ifLW ORK \(=-1, W\) ORK (1) retums the optim um LW ORK.

LW ORK length ofW ORK array. LW ORK should be at leastM .

Forgood perform ance, LW O RK should generally be larger.

For optim um perform ance on \(m\) ultiple processors, LW ORK \(>=M\) *N _CPU S where N_CPUS is the m axim um num berof processors available to the program .

If LW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser'S G uide available at:
http://m ath nist.gov/m cso/Staff/K Rem ington/Espblas/
"D ocum ent forthe B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlio .org/utk/papers/sparse _ps

\section*{NOTES /BUGS}
1.A lso notsupported:
a. low er triangularm atrix A ofsizem by n wherem \(>n\)
b. upper triangularm atrix \(A\) of size \(m\) by \(n\) where \(m<n\)
2. N o test for singularity ornear-singularity is included in this routine. Such tests m ust.be perform ed before calling this routine.
3. If U N ITD \(=4\), the routine scales the row s of \(A\) if \(D E S C R A(2)=1\) and the colum ns ofA if \(D E S C R A(2)=2\) such that their 2 -norm s are one. The scaling \(m\) ay im prove the accuracy of the com puted solution. C orresponding entries of V A L are changed only in this particular case. O n retum D V m atrix stored as a vector contains the diagonalm atrix by w hich the row \(s\) (colum ns) have been scaled. U N ITD = 2 if \(D E S C R A(2)=1\) and UN ITD \(=3\) if \(D E S C R A(2)=2\) should be used for the next calls to the routine \(w\) ith overw rilten \(V A L\) and \(D V\).

WORK \((1)=0\) on retum if the scaling has been com pleted successfully, otherw ise WORK (1) = -iw here i is the row (colum n) num berw hich 2 -norm is exactly zero.
4. If \(D E S C R A(3)=1\) and \(U \mathrm{~N}\) ITD \(<4\), the unit diagonalelem ents
\(m\) ightorm ightnotbe referenced in the SK Y representation of a sparse m atrix. They are not used anyw ay in these cases. ButifUN ITD = 4, the unit diagonalelem ents M U ST be referenced in the SK Y representation.
5.The routine can be applied for solving triangular system \(s\) w hen the upper or low er triangle of the general sparse \(m\) atrix \(A\) is used. H ow ever DESCRA (1) m ustbe equal to 3 in this case.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zspcon -estim ate the reciprocal of the condition num ber (in the 1-norm ) of a com plex sym \(m\) etric packed \(m\) atrix \(A\) using the factorization \(A=U * D * U * * T\) or \(A=L * D * L * * T\) com puted by ZSPTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZSPCON(UPLO,N,AP,\mathbb{PIVOT,ANORM,RCOND,W ORK,INFO)}}\mathbf{N}\mathrm{ (NO}
CHARACTER * 1 UPLO
DOUBLE COM PLEX AP (*),W ORK (*)
INTEGERN,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION ANORM ,RCOND

```

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX AP (*),W ORK (*)
INTEGER*8N,\mathbb{NFO}
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION ANORM,RCOND

```

\section*{F95 INTERFACE}
```

SU BROUTINE SPCON (UPLO,N,AP, $\mathbb{P} \mathbb{I V O T}, A N O R M, R C O N D,[\mathbb{O R K}],[\mathbb{N} F O])$
CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D $\mathbb{M}$ ENSION (:) ::AP,W ORK
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
$\mathbb{N}$ TEGER,D $\mathbb{I}$ ENSION (:) :: $\mathbb{P} \mathbb{I V}$ OT
REAL (8) ::ANORM,RCOND

```

CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::AP,W ORK
\(\mathbb{N}\) TEGER (8) ::N, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V} O T\)
REAL (8) ::ANORM,RCOND

\section*{C INTERFACE}
\#include <sunperfh>
void zspcon (char uplo, int n, doublecom plex *ap, int
*ịívot, double anorm , double *rcond, int *info);
void zspcon_64 (charuplo, long n, doublecom plex *ap, long
* ịpívot, double anorm , double *rcond, long *info);

\section*{PURPOSE}
zspcon estim ates the reciprocal of the condition num ber (in the 1-norm ) of a com plex sym \(m\) etric packed \(m\) atrix A using the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * T\) com puted by ZSPTRF.

A n estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND = 1 / (ANORM * norm (inv (A))).

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) ': U ppertriangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\);
= L ': Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D}\) * \(\mathrm{L} * * \mathrm{~T}\).

N (input) The order of the matrix A. \(\mathrm{N}>=0\).

AP (input)
D ouble com plex array, dim ension \(\mathbb{N}^{*} \mathbb{N}+1\) )/2) The block diagonal \(m\) atrix \(D\) and the \(m\) ultipliers used to obtain the factorU orL as com puted by ZSPTRF, stored as a packed triangularm atrix.

PIVOT (input)
Integer array, dim ension \(\mathbb{N}\) ) D etails of the interchanges and the block structure ofD as determ ined by ZSPTRF.

\section*{ANORM (input)}

The 1-norm of the originalm atrix A.

\section*{RCOND (output)}

The reciprocal of the condition number of the \(m\) atrix \(A\), com puted as RCOND \(=1 /(A N O R M * A \mathbb{N} V N M)\), where \(A \mathbb{N} V N M\) is an estim ate of the 1 -nom of inv (A) com puted in this routine.

W ORK (w orkspace)
D ouble com plex array, dim ension (2*N )
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N N F O}=-i\), the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zsprfs - im prove the com puted solution to a system of linear equations \(w\) hen the coefficientm atrix is sym \(m\) etric indefinite and packed, and provides errorbounds and backw ard error estim ates for the solution

\section*{SYNOPSIS}
```

SUBROUTINE ZSPRFS (UPLO,N,NRHS,A,AF,\mathbb{PIVOT,B,LDB,X,LDX,FERR,}
BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEXA (*),AF (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGER N,NRHS,LDB,LDX,INFO
\mathbb{NTEGER IPIVOT (*)}
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)
SU BROUT\mathbb{NE ZSPRFS_64 (UPLO,N,NRHS,A,AF,\mathbb{PIVOT,B,LDB,X,LDX,}}\mathbf{N},\textrm{N},\textrm{N}
FERR,BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (*),AF (*),B (LD B,*),X (LDX ,*),W ORK (*)
INTEGER*8N,NRHS,LDB,LDX,INFO
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE SPRFS (UPLO, N, \(\mathbb{N} R H S], A, A F, \mathbb{P} \mathbb{I V O T}, B,[L D B], X,[L D X]\), FERR, BERR, [W ORK], [W ORK 2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::A,AF,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : ) :: B, X
\(\mathbb{N}\) TEGER : \(: \mathrm{N}, \mathrm{NRH} \mathrm{S}, \mathrm{LD} \mathrm{B}, \mathrm{LD} \mathrm{X}, \mathbb{N} F \mathrm{O}\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL (8), D \(\mathbb{I}\) ENSION (:) :: FERR,BERR,W ORK2

SU BROUTINE SPRFS_64 (UPLO,N, \(\mathbb{N} R H S], A, A F, \mathbb{P} \mathbb{I} O T, B,[L D B], X,[L D X]\), FERR, BERR, [W ORK], [W ORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) :: UPLO
COM PLEX (8), D \(\mathbb{I M} E N S I O N\) (:) ::A,AF,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : B , X
\(\mathbb{N}\) TEGER (8) ::N , NRHS,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8), D \(\mathbb{I M} E N S I O N(:):: F E R R, B E R R, W\) ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void zsprfs (charuplo, intn, int nhs, doublecom plex *a, doublecom plex *af, int*ipivot, doublecom plex *b, int ldb, doublecom plex *x, int ldx, double *ferr, double *berr, int *info);
void zsprfs_64 (charuplo, long n, long nrhs, doublecom plex *a, doublecom plex *af, long *ipivot, doublecom plex *b, long ldb, doublecom plex *x, long ldx, double * ferr, double *berr, long *info);

\section*{PURPOSE}
zspris im proves the com puted solution to a system of linear equations \(w\) hen the coefficientm atrix is sym \(m\) etric indefintie and packed, and provides errorbounds and backw ard error estim ates for the solution.

\section*{ARGUMENTS}

UPLO (input)
\(=U\) ': U pper triangle of A is stored;
= LL': Low ertriangle ofA is stored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber
of colum ns of the m atrioes B and X. NRHS >=0.

A (input) D ouble com plex amay, dim ension \(\mathbb{N} *(\mathbb{N}+1) / 2)\) The upper or low er triangle of the sym \(m\) etric \(m\) atrix \(A\),
packed colum nw ise in a linear array. The jth colum \(n\) of \(A\) is stored in the array \(A\) as follow \(s\) : if UPLO \(=U\) ', A \((i+(j 1) * j 2)=A(i, j)\) for \(1<=i<=j\) if \(\mathrm{UPLO}=\mathrm{L}^{\prime}, \mathrm{A}(i+(j 1) *(2 * n-j) / 2)=\) A \((i, j)\) for \(j=i<=n\).

AF (input)
D ouble com plex anray, dim ension \((\mathbb{N} * \mathbb{N}+1) / 2)\) The factored form of them atrix A. AF contains the block diagonalm atrix D and the m ultipliers used to obtain the factorU orL from the factorization \(A=U * D * U * * T\) or \(A=L * D * L * * T\) as com puted by C SP TRF, stored as a packed triangularm atrix.
IPIVOT (input)
Integer array, dim ension \((\mathbb{N})\) D etails of the interchanges and the block structure ofD as determ ined by CSPTRF.

B (input) D ouble com plex array, dim ension (LD B,NRHS) The righthand side m atrix \(B\).

LD B (input)
The leading dim ension of the array B . LD B \(>=\) \(\max (1, N)\).

X (input/output)
D ouble com plex array, dimension (LDX,NRHS) On entry, the solution \(m\) atrix \(X\), as com puted by CSPTRS. On exit, the im proved solution m atrix X .

\section*{LD X (input)}

The leading dim ension of the array X . LD X >= \(\max (1, N)\).

FERR (output)
D ouble precision array, dimension \(\mathbb{N} R H S\) ) The estim ated forw ard error bound foreach solution vectorX ( 7 ) (the \(j\) th column of the solution \(m\) atrix \(X)\). If \(X T R U E\) is the true solution corresponding to \(X(\mathcal{J}), \operatorname{FERR}\) ( \(\mathcal{I}\) ) is an estim ated upper bound for the \(m\) agnitude of the largestele\(m\) entin (X ( 1 ) -X TRUE) divided by the magninude of the largest elem entin \(X\) ( 1 ). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

BERR (output)
D ouble precision array, dim ension \((\mathbb{N}\) RHS) The com ponentw ise relative backw ard emor of each solution vector X (i) (ie., the sm allest relative
change in any elem entofA orB thatm akes X ( 7 ) an exactsolution).

W ORK (w orkspace)
D ouble precision anray, dim ension ( \(2 \star \mathrm{~N}\) )

W ORK2 (w orkspace)
Integer array, dim ension \((\mathbb{N})\)
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zspsv - com pute the solution to a com plex system of linear equations A * X = B,

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX AP (*),B (LDB **)
\mathbb{N TEGER N,NRHS,LDB, NNFO}
INTEGER \mathbb{PIVOT (*)}
SU BROUTINE ZSPSV_64(UPLO,N,NRHS,AP, \mathbb{P IVOT,B,LDB, NN FO )}
CHARACTER * 1 UPLO
DOUBLE COMPLEX AP (*),B(LDB,*)
INTEGER*8 N,NRHS,LDB, INFO
\mathbb{NTEGER*8 \mathbb{P IVOT (*)}}\mathbf{(})=

```

\section*{F95 INTERFACE}

SU BROUTINE SPSV (UPLO,N, \(\mathbb{N} R H S], A P, \mathbb{P} \mathbb{I V} O T, B,[L D B],[\mathbb{N} F O])\)
CHARACTER (LEN=1)::UPLO
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::AP
COM PLEX (8),D IM ENSION (:,:) ::B
\(\mathbb{N}\) TEGER :: N, NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
SU BROUTINE SPSV_64 (UPLO,N, NRHS],AP, \(\mathbb{P} \mathbb{I} \operatorname{OT}, \mathrm{B},[\operatorname{LD} B],[\mathbb{N} F O])\)
CHARACTER (LEN=1)::UPLO

COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::AP
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::B
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zspsv (charuplo, intn, int nrhs, doublecom plex *ap, int *ipivot, doublecom plex *b, int ldl, int *info);
void zspsv_64 (charuplo, long n, long nrhs, doublecom plex *ap, long *ipivot, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
zspsv com putes the solution to a com plex system of linear equations
\(A * X=B\), where \(A\) is an \(N\) boy \(-N\) symm etric \(m\) atrix stored in packed form at and X and B are N -by-N R H S m atrices.

The diagonalpivoting \(m\) ethod is used to factorA as
\(A=U * D * U * * T\), if \(U P L O=U\) ', or
\(A=L * D * L * * T\), if \(U P L O=L '\),
where U (orL) is a productofperm utation and unit upper
(low er) triangularm atrioes, D is sym \(m\) etric and block diagonalw th 1 boy-1 and 2 -by -2 diagonal blocks. The factored form of \(A\) is then used to solve the system of equations \(A\) * \(X=B\).

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}:\) Uppertriangle of \(A\) is stored;
= \(\mathrm{L}^{\prime}\) : Low er triangle of A is stored.

N (input) The num ber of linearequations, i.e., the order of them atrix A. N >= 0 .

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the m atrix B. NRH S >=0.

A P (input/output)
D ouble com plex aray, dim ension \(\left.\mathbb{N}^{*}(\mathbb{N}+1) / 2\right)\) On
entry, the upper or low ertriangle of the sym \(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linear array. The jth column of \(A\) is stored in the array AP as follow s: if UPLO \(=U^{\prime}, A P(i+(j\) \(\left.1)^{\star} j 2\right)=A\left(i, 7\right.\) for \(1<=i<=j\) ifUPLO \(=L^{\prime}, A P(i\) \(+(j-1)^{\star}(2 n-j / 2)=A(i, j)\) for \(j=i<=n\). See below for furtherdetails.

On exit, the block diagonalm atrix D and the mul tipliers used to obtain the factor \(U\) orL from the factorization \(A=U * D * U * * T\) orA \(=L * D * L * * T\) as com puted by CSPTRF, stored as a packed triangular \(m\) atrix in the sam e storage form atasA.

IP IV O T (output)
Integer aray, dim ension \((\mathbb{N})\) D etails of the interchanges and the block structure of \(D\), as determ ined by \(C\) SPTRF. If \(\mathbb{P} \mathbb{I V O T}(k)>0\), then row \(S\) and colum nsk and \(\mathbb{P} \mathbb{I}\) OT \((k)\) w ere interchanged, and \(D(k, k)\) is a 1 -by -1 diagonalblock. If \(\mathrm{UPLO}=\mathrm{U}^{\prime}\) and \(\mathbb{P} \mathbb{I V O T}(k)=\mathbb{P} \mathbb{I V O T}(k-1)<0\), then row \(s\) and colum nsk-1 and - \(\mathbb{P} \mathbb{I V O T}(k)\) were interchanged and D ( \(k-1: k, k-1 k)\) is a 2 -by-2 diagonalblock. If \(\mathrm{UPLO}=\mathbb{L}\) 'and \(\mathbb{P} \mathbb{I V O T}(\mathrm{k})=\mathbb{P} \mathbb{I V O T}(\mathrm{k}+1)<0\), then row \(s\) and colum ns \(k+1\) and \(-\mathbb{P} \mathbb{I V} \circ T(k)\) were inter changed and \(D(k: k+1, k: k+1)\) is a \(2-b y-2\) diagonal block.

B (input/output)
D ouble complex array, dimension (LDB,NRHS) On entry, the N -by -N RH S righthand side m atrix B. On exit, if \(\mathbb{N ~ F O ~}=0\), the N -by-N R H S solution m atrix X .

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue
\(>0\) : if \(\mathbb{N} F O=i, D(i, i)\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, so the solution could notbe com puted.

\section*{FURTHER DETAILS}
exam ple w hen \(N=4, \mathrm{UPLO}=\mathrm{U}\) ':

Tw o-dim ensional storage of the sym \(m\) etric \(m\) atrix A :
```

a11 a12 a13 a14
a22 a23 a24
a33 a34 (aij= aji)
a44

```

Packed storage of the upper triangle ofA :
```

AP = [a11,a12,a22,a13,a23,a33,a14,a24,a34,a44 ]

```

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
```

zspsvx - use the diagonal pivoting factorization A =
U *D *U **T or A = L*D *L**T to com pute the solution to a com -
plex system of linearequationsA * X = B,where A is an N-
by-N symm etric m atrix stored in packed form at and X and B
are N -by-N RH S m atrioes

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZSPSVX FACT,UPLO,N,NRHS,A,AF,\mathbb{PIVOT,B,LDB,X,LDX,}}\mathbf{N},\mp@code{N},
RCOND,FERR,BERR,W ORK,W ORK2, INFO)
CHARACTER * 1FACT,UPLO
DOUBLE COM PLEX A (*),AF (*),B (LD B ,*),X (LDX ,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,INFO
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION RCOND
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)

```

```

    LDX,RCOND,FERR,BERR,WORK,W ORK 2,INFO)
    CHARACTER * 1FACT,UPLO
DOUBLE COM PLEXA (*),AF (*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGER*8N,NRHS,LDB,LDX,NNFO
\mathbb{NTEGER*8 \mathbb{PIVOT (*)}}\mathbf{(})
DOUBLE PRECISION RCOND
D OUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE SPSVX (FACT,UPLO,N, NRHS],A,AF, \(\mathbb{P} I V O T, B,[L D B], X\), [LDX],RCOND,FERR,BERR, [W ORK], [W ORK2], [ \(\mathbb{N F O}])\)

CHARACTER (LEN=1) ::FACT,UPLO
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::A, AF,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:): : B, X
\(\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N F O}\)
\(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{V} O T\)
REAL (8) ::RCOND
REAL (8), D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

SUBROUTINE SPSVX_64 (FACT, UPLO,N, NRHS],A,AF, IPIVOT,B,[LDB],X, \([\) [LD \(], R C O N D, F E R R, B E R R,[W O R K],[W O R K 2],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::FACT, UPLO
COMPLEX (8), D \(\mathbb{M} E N S I O N(:):: A, A F, W\) ORK
COM PLEX (8), D IM ENSION (: ::) ::B,X
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDB,LDX, \(\mathbb{N}\) FO
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION(:):: \(\mathbb{P} \mathbb{I V O T}\)
REAL (8) ::RCOND
REAL (8), D \(\mathbb{M} E N S I O N(:):: F E R R, B E R R, W\) ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void zspsvx (char fact, charuplo, int n, int nrhs, doublecom plex *a, doublecom plex *af, int *ípivot, doublecom plex *b, intldb, doublecom plex *x, int \(l d x\), double *roond, double *ferr, double *berr, int*info);
void zspsvx_64 (char fact, char uplo, long n, long nrhs, doublecom plex *a, doublecom plex *af, long *ipívot, doublecom plex *b, long ldl, doublecom plex *x, long \(l d x\), double *roond, double *ferr, double *berr, long *info);

\section*{PURPOSE}
zspsvx uses the diagonalpivoting factorization \(A=U * D * U * * T\) or \(A=L * D * L * * T\) to com pute the solution to a com plex system of linear equations \(A * X=B\), where \(A\) is an \(N\) boy -N sym \(m\) etric \(m\) atrix stored in packed form atand \(X\) and \(B\) are \(N\)-byN R H S m atrioes.

Emorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT \(=N\) ', the diagonalpivoting \(m\) ethod is used to
factorA as
\(A=U * D * U * * T\), if \(U P L O=U\) ', or
\(A=L * D * L * * T\), if \(\mathrm{L} P \mathrm{LO}=\mathrm{L}\) ',
where \(U\) (orL) is a productofperm utation and unitupper (low er)
triangularm atrioes and \(D\) is sym \(m\) etric and block diagonal w ith

1 -by-1 and 2 -by-2 diagonalblocks.
2. If som eD \((i, i)=0\), so thatD is exactly singular, then the routine
retums \(w\) ith \(\mathbb{N N F O}=\) i. O therw ise, the factored form of \(A\) is used
to estim ate the condition num ber of the \(m\) atrix \(A\). If the reciprocal of the condition num ber is less than m achine precision,
\(\mathbb{N} \mathrm{FO}=\mathrm{N}+1\) is retumed as a w aming, but the routine stillgoes on
to solve for \(X\) and compute error bounds as described below.
3.The system ofequations is solved forX using the factored form of A.
4. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for 五.

\section*{ARGUMENTS}

\section*{FACT (input)}

Specifies w hether ornot the factored form of \(A\) has been supplied on entry \(.=F ':\) On entry, AF and \(\mathbb{P} \mathbb{I V O T}\) contain the factored form of A. A, AF and \(\mathbb{P} \mathbb{I V O T}\) w ill not be modified. = N : : The \(m\) atrix A w ill.be copied to A F and factored.
```

UPLO (input)

```
\(=\mathrm{U}\) ': Upper triangle ofA is stored;
\(=\mathrm{L}\) ': Low er triangle of A is stored.

N (input) The num ber of linearequations, i.e., the order of them atrix A. N >=0.

NRHS (input)

The num ber of righthand sides, ie., the num ber of colum ns of the \(m\) atrices \(B\) and \(X . N R H S>=0\).

A (input) D ouble com plex array, dim ension \(\mathbb{N}^{*}(\mathbb{N}+1) / 2\) ) The upper or low er triangle of the sym \(m\) etric \(m\) atrix \(A\), packed colum nw ise in a linear array. The jth colum \(n\) of \(A\) is stored in the array \(A\) as follow \(s\) : ifUPLO \(=U^{\prime}, A(i+(j-1) * j 2)=A(i, j)\) for \(1<=i<=j\) if UPLO \(=L^{\prime}, A\left(i+(j-1)^{*}(2 * n-j) / 2\right)=\) A \((i, y)\) for \(j=i<=n\). See below for further details.

AF (input/output)
D ouble com plex array, dim ension \((\mathbb{N} *(\mathbb{N}+1) / 2\) ) If FACT \(=F^{\prime}\), then \(A F\) is an inputargum entand on entry contains the block diagonalm atrix D and the m ultipliers used to obtain the factor U or L from the factorization \(A=U * D * U * * T\) or \(A=L * D * L * * T\) as com puted by CSPTRF, stored as a packed triangular \(m\) atrix in the sam e storage form atas \(A\).

IfFACT = N ', then AF is an output argum ent and on exit contains the block diagonalm atrix \(D\) and the m ultipliers used to obtain the factor \(U\) or \(L\) from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) as com puted by CSPTRF, stored as a packed triangularm atrix in the sam e storage form at as A.

PIVOT (inputoroutput)
Integer array, dim ension (N) IfFACT = \(\mathrm{F}^{\prime}\), then
\(\mathbb{P I V O T}\) is an inputargum entand on entry contains details of the interchanges and the block structure ofD, as determ ined by CSPTRF. If \(\mathbb{P} \mathbb{I V} O T(k)\) \(>0\), then row sand columnsk and \(\mathbb{P} \mathbb{I} O T(k)\) were interchanged and \(D(k, k)\) is a 1 -by- 1 diagonal
block. If UPLO \(=\mathrm{U}\) 'and \(\mathbb{P I V O T}(k)=\mathbb{P} \mathbb{V} O T(k-1)\) \(<0\), then row s and colum ns k-1 and - \(\mathbb{P} \mathbb{I V}\) O ( \(k\) ) w ere interchanged and \(D(k-1 k, k-1 k)\) is a \(2-b y-2\) diagonal block. If UPLO = L' and \(\mathbb{P} \mathbb{I V O T}(k)=\) \(\mathbb{P I V O T}(k+1)<0\), then row \(s\) and colum ns \(k+1\) and \(-\mathbb{P} \mathbb{I V O T}(k)\) were interchanged and \(D(k \cdot k+1, k \cdot k+1)\) is a 2-by-2 diagonalblock.

IfFACT = \(N\) ', then \(\mathbb{P I V O T}\) is an output argum ent and on exit contains details of the interchanges and the block structure of D, as determ ined by CSPTRF.

B (input) D ouble com plex aray, dim ension (LD B NRHS) The N by -N RH \(S\) righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

X (output)
D ouble com plex array, dim ension (LD X ,NRHS) If \(\mathbb{N}\) FO
\(=0\) or \(\mathbb{N} F O=N+1\), the N boy \(-\mathrm{NRH} S\) solution \(m\) atrix
X .

LD X (input)
The leading dim ension of the aray X. LD X >= \(\max (1, N)\).
RCOND (output)
The estim ate of the reciprocal condition num ber of
the matrix A. IfRCOND is less than the m achine precision (in particular, if RCOND \(=0\) ), the \(m\) atrix is singular to \(w\) orking precision. This condition is indicated by a retum code of \(\mathbb{N N F O ~ > ~}\) 0 .

FERR (output)
D ouble complex array, dimension \((\mathbb{N} H S\) ) The estim ated forw ard error bound foreach solution vectorX ( 7 ) (the \(j\) th colum \(n\) of the solution \(m\) atrix \(X)\). If XTRUE is the true solution corresponding to \(X(\mathcal{I}), \operatorname{FERR}(\mathcal{I})\) is an estim ated upper bound for the \(m\) agnitude of the largestele\(m\) entin ( \(X(\mathcal{)})-X\) TRU \(E\) ) divided by the \(m\) agnitude of the largest elem entin \(X\) ( 1 ). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slightoverestim ate of the true error.

\section*{BERR (output)}

D ouble com plex array, dim ension \((\mathbb{N}\) RHS) The componentw ise relative backw ard emor of each solution vector X (i) (ie., the sm allest relative change in any elem entof \(A\) orB thatm akes \(X\) ( \()\) an exactsolution).

W ORK (w orkspace)
D ouble com plex array, dim ension (2*N )

W ORK2 (w orkspace)
Integer amay, dim ension \((\mathbb{N})\)

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0\) : if \(\mathbb{N F O}=\mathrm{i}\), and i is
\(<=\mathrm{N}: \mathrm{D}(i, i)\) is exactly zero. The factorization has been com pleted but the factorD is exactly singular, so the solution and error bounds could not be com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1\) : D is nonsingular, butRCOND is less than \(m\) achine precision, \(m\) eaning that the \(m\) atrix is singular to w orking precision. \(N\) evertheless, the solution and error bounds are com puted because there are a num ber of situationsw here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{FURTHER DETAILS}

The packed storage schem e is illustrated by the follow ing exam plew hen \(N=4, U P L O=U\) ':

Tw o-dim ensional storage of the sym m etric m atrix A :
```

a11 a12 a13 a14
a22 a23 a24
a33 a34 (aij= aj̈)
a44

```

Packed storage of the upper triangle ofA :
\(A=[a 11, a 12, a 22, a 13, a 23, a 33, a 14, a 24, a 34, a 44]\)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zsptrf-com pute the factorization of a com plex sym m etric \(m\) atrix A stored in packed form at using the B unch-K aufn an diagonalpivoting \(m\) ethod

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZSPTRF(UPLO,N,AP,\mathbb{PIVOT,INFO)}}\mathbf{N}\mathrm{ (N,}
CHARACTER * 1 UPLO
DOUBLE COM PLEX AP (*)
INTEGER N,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}
SUBROUT\mathbb{NE ZSPTRF_64(UPLO,N,AP,\mathbb{PIVOT,INFO)}}\mathbf{N},\mp@code{N}
CHARACTER * 1 UPLO
DOUBLE COM PLEX AP (*)
INTEGER*8N,\mathbb{NFO}
\mathbb{NTEGER *8 \mathbb{PIVOT (*)}}\mathbf{(})

```

\section*{F95 INTERFACE}
```

SU BROUTINE SPTRF (UPLO ,N,AP, $\mathbb{P} \mathbb{I V O T},[\mathbb{N} F O]$ )
CHARACTER (LEN=1) ::UPLO
COMPLEX (8), D IM ENSION (:) ::AP
$\mathbb{N} T E G E R:: N, \mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{V} O T$
SU BROUTINE SPTRF_64 (UPLO,N,AP, $\mathbb{P} \mathbb{I} \operatorname{OT},[\mathbb{N} F O$ ])
CHARACTER (LEN=1)::UPLO

```

\section*{C INTERFACE}
\#include <sunperfh>
void zsptrf(char uple, int n, doublecomplex *ap, int *ípivot, int*info);
void zsptrf_64 (charuplo, long n, doublecom plex *ap, long *ịivot, long *info);

\section*{PURPOSE}
zsptrf com putes the factorization of a com plex symm etric \(m\) atrix A stored in packed form atusing the Bunch-K aufm an diagonalpivoting m ethod:
\[
A=U * D * U * * T \text { or } A=L * D * L * * T
\]
where U (orL) is a product of perm utation and unit upper (low er) triangular matrioes, and \(D\) is sym \(m\) etric and block diagonalw ith 1 boy-1 and 2 -by-2 diagonalblocks.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}:\) Uppertriangle of is istored;
\(=\mathrm{L}\) ': Low er triangle ofA is stored.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A P (input/output)
D ouble com plex array, dim ension \((\mathbb{N} *(\mathbb{N}+1) / 2)\) On entry, the upper or low er triangle of the sym \(m\) etric \(m\) atrix A, packed colum nw ise in a linear anray. The \(j\) th column of \(A\) is stored in the array AP as follow s: ifUPLO \(=U\) ', AP (i \(+(j\) \(\left.1)^{\star} j 2\right)=A(i, j\) for \(1<=i<=j\) ifUPLO \(=L\) ', AP \((i\) \(+(j-1)^{*}(2 n-j / 2)=A(i, j)\) for \(j=i<=n\).

On exit, the block diagonalm atrix D and the multipliers used to obtain the factorU orL, stored as a packed triangularm atrix overw rilling A (see below for further details).

\section*{IPIVOT (output)}

Integer amay, dim ension \(\mathbb{N}\) ) D etails of the interchanges and the block structure of D. If \(\mathbb{P} \mathbb{I V O T}(\mathrm{k})>0\), then rows and colum ns k and IP IV O T \((k)\) w ere interchanged and \(D(k, k)\) is a 1 -by-1 diagonalblock. IfUPLO \(=\mathrm{U}^{\prime}\) and \(\mathbb{P} \mathbb{I V O T}(k)=\) \(\mathbb{P} \mathbb{I V O T}(k-1)<0\), then row s and colum nsk-1 and - \(\mathbb{P}\) IV O T \((k)\) w ere interchanged and \(D(k-1 * k, k-1 k)\) is a 2 -by-2 diagonal block. If UPLO = L'and \(\mathbb{P} \mathbb{I V} \circ T(k)=\mathbb{P} \mathbb{I} \circ T(k+1)<0\), then row s and colum \(n s\) \(\mathrm{k}+1\) and \(-\mathbb{P} \mathbb{I V O T}(\mathrm{k})\) were interchanged and D \((k: k+1, k: k+1)\) is a 2 -by -2 diagonalblock.

\section*{\(\mathbb{N} F O\) (output)}
= 0: successfiulexit
\(<0:\) if \(\mathbb{N F O}=-i\), the \(i\)-th argum ent had an illegal value
\(>0:\) if \(\mathbb{N F O}=i, D(i, i)\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix D is exactly singular, and division by zero w illoccur if it is used to solve a system of equations.

\section*{FURTHER DETAILS}

5-96 - B ased on m odifications by J. Lew is, B oeing C om puter Services

C om pany

If \(\mathrm{U} P \mathrm{LO}=\mathrm{U}\) ', then \(A=U * D * U\) ', where
\(U=P(n) \star U(n)^{\star} \ldots{ }^{*} P(k) U(k)^{\star} \ldots\),
i.e., \(U\) is a productof term \(s P(k) * U(k)\), where \(k\) decreases from \(n\) to 1 in steps of 1 or 2 , and \(D\) is a block diagonal \(m\) atrix \(w\) ith 1 -by-1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I V} O T(k)\), and \(U(k)\) is a unituppertriangularm atrix, such that if the diagonal
block \(D(k)\) is of orders \((s=1\) or 2 ), then
```

    ( I v 0 ) k-s
    U (k)=(0 I 0 ) s
( 0 0 I ) n-k
k-s s n-k

```

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(1, k-\) \(1, k)\). If \(s=2\), the uppertriangle of \((k)\) overw rites \(A(k-\) \(1, k-1), A(k-1, k)\), and \(A(k, k)\), and \(v\) overw rites \(A(1 k-2, k-\) 1 k).
```

IfU PLO = L', then A = L *D *L',w here
L = P (l)*L (l)* ... *P (k)*L (k)* ...,

```
i.e., \(L\) is a productofterm \(S P(k) * L(k)\), where \(k\) increases from 1 to \(n\) in steps of 1 or2, and \(D\) is ablock diagonal \(m\) atrix \(w\) th 1 -by -1 and 2 -by- 2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I V O T}(k)\), and \(L(k)\) is a unit low ertriangularm atrix, such that if the diagonal block \(D(k)\) is of orders ( \(s=1\) or 2 ), then
```

    ( I 0 0 ) k-1
    L (k)=( 0 I 0 ) s
    ( 0 v I ) n-k-s+1
        k-1 s n-k-s+1
    ```

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites A \((k+1 n, k)\). If \(s=2\), the low er triangle ofD ( \(k\) ) overw rites A \((k, k), A(k+1, k)\), and \(A(k+1, k+1)\), and \(v\) overw rites \(A(k+2 n, k k+1)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zsptri-com pute the inverse of a com plex sym m etric indefinite \(m\) atrix \(A\) in packed storage using the factorization \(A=\) \(\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by CSPTRF

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX AP (*),W ORK (*)
INTEGER N,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}
SUBROUT\mathbb{NE ZSPTRI_64(UPLO,N,AP,\mathbb{PIVOT,W ORK,INFO)}}\mathbf{(})=
CHARACTER * 1 UPLO
DOUBLE COM PLEX AP (*),W ORK (*)
INTEGER*8N,\mathbb{NFO}
\mathbb{NTEGER *8 \mathbb{PIVOT (*)}}\mathbf{(})
F95 INTERFACE

```

```

    CHARACTER (LEN=1) ::UPLO
    COMPLEX (8),DIM ENSION (:) ::AP,W ORK
    \mathbb{NTEGER ::N,\mathbb{NFO}}0=0
    \mathbb{NTEGER,D IM ENSION (:) ::\mathbb{PIVOT}}\mathbf{T}=1
    ```

```

    CHARACTER (LEN=1) ::UPLO
    COMPLEX (8),D IM ENSION (:) ::AP,W ORK
    ```
\(\mathbb{N}\) TEGER (8) ::N, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V} O T\)

\section*{C INTERFACE}
\#include < sunperfh>
void zsptri(char uplo, int n, doublecom plex *ap, int *ịívot, int*info);
void zsptri_ 64 (charuplo, long n, doublecom plex *ap, long *ịìivot, long *info);

\section*{PURPOSE}
zsptri computes the inverse of a complex symm etric indefinite \(m\) atrix \(A\) in packed storage using the factorization \(A=U * D * U * * T\) or \(A=L * D * L * * T\) com puted by ZSPTRF.

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) ': U pper triangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\);
= L': Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\).

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
AP (input/output)
D ouble com plex array, dim ension \(\mathbb{N}^{*}(\mathbb{N}+1) / 2\) ) On entry, the block diagonalm atrix D and the \(m\) ultipliers used to obtain the factor \(U\) or L as computed by ZSPTRF, stored as a packed triangular \(m\) atrix.

On exit, if \(\mathbb{N F F O}=0\), the (sym metric) inverse of the originalm atrix, stored as a packed triangular \(m\) atrix. The \(j\) th \(c o l u m n\) of inv ( \(A\) ) is stored in the amay AP as follows: ifUPLO = U',AP (i+ (j 1) \({ }^{j}(2)=\operatorname{inv}(A)(i, \gamma)\) for \(1<=i<=j\) ifUPLO \(=L^{\prime}\), AP \((i+(j-1) *(2 n-j / 2)=\operatorname{inv}(A)(i\rangle\),\() for \dot{j}=i<=n\).

PIVOT (input)
Integer array, dim ension (N) D etails of the interchanges and the block structure ofD as determ ined by ZSPTRF.

W ORK (w orkspace)
D ouble com plex array, dim ension \(\mathbb{N}\) )
\(\mathbb{N}\) FO (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N N F O}=i, D(i, i)=0\); the \(m\) atrix is singular and its inverse could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zsptrs-solve a system of linearequations \(A * X=B\) w th a com plex sym \(m\) etric \(m\) atrix A stored in packed form atusing the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by ZSPTRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZSPTRS (UPLO,N,NRHS,AP,\mathbb{P}\mathbb{INOT,B,LDB,INFO)}}\mathbf{N}\mathrm{ (N,N}

```
CHARACTER * 1 UPLO
DOUBLE COM PLEX AP (*), B (LDB,*)
\(\mathbb{N}\) TEGERN,NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R \mathbb{P} \mathbb{V} O T\left({ }^{\star}\right)\)
SU BROUTINE ZSPTRS_64 (UPLO,N,NRHS,AP, \(\mathbb{P} \mathbb{I V O T}, \mathrm{B}, \mathrm{LD} B, \mathbb{N} F O\) )
CHARACTER * 1 UPLO
DOUBLE COM PLEXAP (*), B (LDB,*)
\(\mathbb{N}\) TEGER*8N,NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER*8 \(\mathbb{P} \mathbb{I V O T}\) ( \({ }^{*}\) )

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E \operatorname{SPTRS}(\mathbb{U}\) PLO , N, \(\mathbb{N} R H S], A P, \mathbb{P} \mathbb{I V} O T, B,[L D B],[\mathbb{N} F O])\)
CHARACTER (LEN=1)::UPLO
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::AP
COM PLEX (8), D IM ENSION (:,:) ::B
\(\mathbb{N}\) TEGER :: N, NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
SU BROUTINE SPTRS_64 (UPLO ,N, NRHS],AP, \(\mathbb{P} \mathbb{I V} O T, B,[L D B],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D \(\mathbb{I}\) ENSION (:) ::AP
COM PLEX (8), D IM ENSION (:,:) ::B
\(\mathbb{N}\) TEGER (8) :: N, NRHS,LDB, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V} O T\)

\section*{C INTERFACE}
\#include <sunperfh>
void zsptrs (charuplo, intn, intnris, doublecom plex *ap, int *ipivot, doublecom plex *b, int ldb, int *info);
void zsptrs_64 (charuplo, long n, long nrhs, doublecom plex *ap, long *ipívot, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
zsptres solves a system of linearequations \(A * X=B\) w th a com plex sym \(m\) etric \(m\) atrix A stored in packed form at using the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by ZSPTRF.

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) ': U ppertriangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\);
= L': Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\).
N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS >=0.

AP (input)
D ouble com plex array, dim ension \((\mathbb{N} * \mathbb{N}+1) / 2\) ) The block diagonal \(m\) atrix \(D\) and the \(m\) ultipliers used to obtain the factorU orL as com puted by ZSPTRF, stored as a packed triangularm atrix.
\(\mathbb{P I V O T}\) (input)
Integer anay, dim ension \(\mathbb{N}\) ) D etails of the interchanges and the block structure ofD as determ ined
by ZSPTRF.

B (input/output)
D ouble complex array, dimension (LDB,NRHS) On
entry, the right hand side \(m\) atrix \(B\). On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zstedc - com pute alleigenvalues and, optionally, eigenvectors of a sym \(m\) etric tridiagonalm atrix using the divide and conquerm ethod

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZSTEDC (COM PZ,N,D,E,Z,LD Z,W ORK,LW ORK,RW ORK,LRW ORK,}
IN ORK,LIN ORK,INFO)
CHARACTER * 1 COMPZ
DOUBLE COM PLEX Z (LD Z,*),W ORK (*)
\mathbb{NTEGER N,LDZ,LW ORK,LRW ORK,LIN ORK,INFO}
INTEGER IN ORK (*)
DOUBLE PRECISION D (*),E (*),RW ORK (*)
SU BROUTINE ZSTEDC_64 COMPZ,N,D,E,Z,LD Z,W ORK,LW ORK,RW ORK,
LRW ORK,INORK,LINORK,INFO)
CHARACTER * 1 COM PZ
DOUBLE COM PLEX Z (LD Z,*),W ORK (*)
\mathbb{NTEGER*8 N,LDZ,LW ORK,LRW ORK,LIN ORK,INFO}
INTEGER*8 IN ORK (*)
DOUBLE PRECISION D (*),E (*),RW ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE STEDC COM PZ, \(\mathbb{N}], D, E, Z,[L D Z],[W O R K],[L W O R K],[R W O R K]\), [LRW ORK], [IW ORK], [LINORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::COM PZ
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D IM ENSION (:,:) ::Z
\(\mathbb{N}\) TEGER :: N, LD Z,LW ORK,LRW ORK,LIW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} E N S I O N(:):: \mathbb{I W}\) ORK
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D , E, RW ORK

SU BROUTINE STED C_64 (COMPZ, \(\mathbb{N}], D, E, Z,[L D Z],[W O R K],[L W O R K]\), [RW ORK], [LRW ORK], [ \(\mathbb{W}\) ORK], [LIN ORK], [ \(\mathbb{N} F O])\)

CHARACTER (LEN=1) : COMPZ
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : Z
\(\mathbb{N}\) TEGER (8) :: N, LD Z,LW ORK, LRW ORK, LIW ORK, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{I W}\) ORK
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D , E, RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zstedc (char com pz, intn, double *d, double *e, doublecom plex * \(z\), int ld \(z\), int *info);
void zstedc_64 (charcom pz, long n, double *d, double *e, doublecom plex *z, long ldz, long *info);

\section*{PURPOSE}
zstedc com putes alleigenvalues and, optionally, eigenvectors of a sym \(m\) etric tridiagonalm atrix using the divide and conquerm ethod. The eigenvectors of a fullorband com plex Herm itian matrix can also be found ifCHETRD orCHPTRD or CHBTRD has been used to reduce this \(m\) atrix to tridiagonal form.

This code \(m\) akes very \(m\) ild assum ptions about floating point arithm etic. It \(w\) illw ork on \(m\) achines \(w\) ith a guard digitin add/subtract, oron those binary \(m\) achines \(w\) thout guard digits which subtract like the C ray \(\mathrm{X}-\mathrm{M}\) P , C ray Y M P , C ray \(\mathrm{C}-90\), or C ray-2. Itcould conceivably fail on hexadecim al or decim al \(m\) achines \(w\) thout guard digits, butw e know of none. See SLA ED 3 for details.

\section*{ARGUMENTS}

COMPZ (input)
\(=\mathrm{N}^{\prime}:\) C om pute eigenvalues only .
= I': C om pute eigenvectors of tridiagonalm atrix
also.
\(=\mathrm{V}:\) C om pute eigenvectors of original H erm itian
\(m\) atrix also. On entry, \(Z\) contains the unitary
\(m\) atrix used to reduce the originalm atrix to tridiagonal form .

N (input) The dim ension of the sym \(m\) etric tridiagonalm atrix. \(\mathrm{N}>=0\).

D (input/output)
O n entry, the diagonalelem ents of the tridiagonal m atrix. On exit, if \(\mathbb{N} F O=0\), the eigenvalues in ascending order.

E (input/output)
O n entry, the subdiagonalelem ents of the tridiagonalm atrix. On exit, E has been destroyed.

Z (input) \(O n\) entry, if COMPZ \(=V\) ', then \(Z\) contains the unitary m atrix used in the reduction to tridiagonalform. On exit, if \(\mathbb{N F O}=0\), then if \(C O M P Z=\) V', Z contains the orthonorm aleigenvectors of the original H erm itian m atrix, and if \(\mathrm{COMPZ}=\mathrm{I}\) ', Z contains the orthonorm al eigenvectors of the sym m etric tridiagonalm atrix. If COMPZ \(=\mathrm{N}^{\prime}\) ', then Z is not referenced.

LD Z (input)
The leading din ension of the array Z . LD \(\mathrm{Z}>=1\). If eigenvectors are desired, then LD Z \(>=\mathrm{max}(1, \mathrm{~N})\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, \mathrm{~W}\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array W ORK. IfCOM PZ \(=\mathrm{N}^{\prime}\) or 'I', orN <=1,LW ORK mustbe at least1. If
COM PZ \(=V\) 'and \(N>1\), LW ORK m ust.be at least \(N * N\).

IfLW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of
theW ORK anray, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LW ORK is issued by X ERBLA.

RW ORK (w orkspace)
dim ension (LRW ORK) On exit, if \(\mathbb{N} F O=0\), RW ORK (1) retums the optim allRW ORK .

LRW ORK (input)
The dim ension of the array RW ORK. IfCOMPZ \(=N^{\prime}\) or \(\mathrm{N}<=1\), LRW ORK mustbe at least1. IfCOMPZ \(=\)

V'and \(\mathrm{N}>1\),LRW ORK m ustbe atleast1 \(+3 * \mathrm{~N}+\) \(2 * N * \lg +3 * N * * 2\), where \(\lg (N)=s m\) allest integerk such that \(2 \star * \mathrm{k}>=\mathrm{N}\). IfCOM PZ \(=\) I'and \(\mathrm{N}>1\),LRW ORK mustbe at least \(1+4 * N+2 * N * * 2\).

If LRW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the RW ORK array, retums this value as the first entry of the RW ORK amay, and no emorm essage related to LRW ORK is issued by XERBLA.

IV ORK (w orkspace/output)
On exit, if \(\mathbb{N} F O=0, \mathbb{I W}\) ORK (1) retums the optim al LIW ORK.

LIW ORK (input)
The dim ension of the anay \(\mathbb{I W}\) ORK. IfCOMPZ \(=\mathrm{N}^{\prime}\) or \(\mathrm{N}<=1\), LIW ORK m ustbe at least1. IfCOMPZ \(=\) V 'orN > 1, LIW ORK mustbe at least \(6+6 * N+\) \(5 * N * \operatorname{N}\). IfCOMPZ = I'orN \(>1\), LIV ORK must be at least \(3+5 * N\).

IfLIW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the \(\mathbb{I W}\) ORK array, retums this value as the first entry of the \(\mathbb{I W}\) ORK aray, and no errorm essage related to \(L \mathbb{I W} O R K\) is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit.
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue.
\(>0\) : The algorithm failed to com pute an eigenvalue while w orking on the subm atrix lying in row S and colum ns \(\mathbb{N}\) FO \(/ \mathbb{N}+1\) ) through \(m o d(\mathbb{N}\) FO, \(\mathbb{N}+1)\).

\section*{FURTHER DETAILS}

B ased on contributions by JeffR utter, C om puter Science D ivision, U niversity of C alifomia atB erkeley, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zstegr-Com pute \(T\)-sigm a_i= L_iD_iL_i^T, such that L_i
D_iL_i^T is a relatively robustrepresentation

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZSTEGR (JOBZ,RANGE,N,D,E,VL,VU, U, IU,ABSTOL,M,W,}
Z,LDZ,ISUPPZ,W ORK,LW ORK,IN ORK,LIW ORK,\mathbb{NFO)}

```
```

CHARACTER * 1 JOBZ,RANGE
DOUBLE COM PLEX Z (LD Z,*)

```

```

INTEGER ISUPPZ (*), IN ORK (*)
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISIOND (*),E (*),W (*),W ORK (*)

```
SU BROUTINE ZSTEGR_64 (JOBZ,RANGE,N,D,E,VL,VU, IL, IU,ABSTOL,M,
    W, Z,LD Z, ISUPPZ,W ORK,LW ORK, IN ORK,LIN ORK, \(\mathbb{N} F O\) )
CHARACTER * 1 JOBZ,RANGE
DOUBLE COM PLEX Z (LD Z,*)
\(\mathbb{N}\) TEGER*8N, \(\mathbb{I}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W\) ORK,LIN ORK, \(\mathbb{N}\) FO
\(\mathbb{I N} T E G E R * 8 \operatorname{ISUPPZ}\) (*) \(^{*}\), \(\mathbb{I N}\) ORK ( \({ }^{*}\) )
DOUBLE PRECISION VL,VU,ABSTOL
DOUBLE PRECISIOND (*), E (*), W (*), WORK (*)

\section*{F95 INTERFACE}

SU BROUTINE STEGR (JOBZ,RANGE, \(\mathbb{N}], D, E, V L, V U, \mathbb{L}, \mathbb{U}, A B S T O L, M\), W , Z, [LD Z ], ISUPPZ, \(\mathbb{W}\) ORK ], [LW ORK ], [ \(\mathbb{W}\) ORK ], [LIN ORK ], [ \(\mathbb{N} F O]\) )
\(\mathbb{N} T E G E R:: N, \mathbb{H}, \mathbb{U}, M, L D Z, L W O R K, L \mathbb{N} O R K, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{I M}\) ENSION (:) :: ISUPPZ, \(\mathbb{I}\) ORK
REAL (8) ::VL,VU,ABSTOL
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D , E, W ,W ORK

SU BROUT \(\mathbb{N} E\) STEGR_64 (OBZ,RANGE, \(\mathbb{N}], D, E, V L, V U, \mathbb{L}, \mathbb{U}, A B S T O L\), \(M, W, Z,[L D Z], I S U P P Z,[W O R K],[L W O R K],[\mathbb{W} O R K],[L \mathbb{W} O R K],[\mathbb{N} F O])\)

CHARACTER ( \(4 E N=1\) ) : : J B Z , RANGE
COMPLEX (8), D \(\mathbb{I M} E N S \mathbb{O N}(:,:\) ) ::Z
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{\Pi}, \mathbb{U}, \mathrm{M}, \mathrm{LD} Z, L W\) ORK,LINORK, \(\mathbb{N} F O\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{I M}\) ENSION (:) :: ISUPPZ , \(\mathbb{W}\) ORK
REAL (8) :: VL,VU,ABSTOL
REAL (8), D \(\mathbb{M} E N S I O N(:):: D, E, W, W O R K\)

\section*{C INTERFACE}
\#include <sunperfh>
void zstegr(char jobz, char range, intn, double *d, double
*e, double vl, double vu, int il, intin, double abstol, int \({ }_{\mathrm{m}}\), double \({ }_{\mathrm{W}}\), doublecom plex \({ }^{\mathrm{z}} \mathrm{z}\), int ldz, int *isuppz, int *info);
void zstegr_64 (char j.bz, char range, long n, double *d, double *e, double vl, double vu, long il, long iu, double abstol, long \({ }^{\prime}\) m, double \({ }^{*}\), , doublecom plex *z, long ldz, long *isuppz, long *info);

\section*{PURPOSE}
zstegrb) C om pute the eigenvalues, lam bda_j of L_i D_i L_i^T to high relative accuracy by the dqds algorithm,
(c) If there is a cluster of close eigenvalues, "choose"
sigm a_i
close to the cluster, and go to step (a),
(d) G iven the approxim ate eigenvalue lam boda_jofL_i D _i L_i^T,
com pute the corresponding eigenvectorby form ing a rank-revealing tw isted factorization.
The desired accuracy of the output can be specified by the inputparam eterA BSTOL.

Form ore details, see "A new O ( \(n^{\wedge} 2\) ) algorithm for the sym \(m\) etric tridiagonal eigenvahue/eigenvector problem ", by Inder吕D hillon, C om puterScience D ìvision TechnicalR eport N o. U CB C SD -97-971, U C Berkeley, M ay 1997.

N ote 1 : Cumently CSTEGR is only setup to find ALL the \(n\) eigenvalues and eigenvectors of \(T\) in \(O\left(n^{\wedge} 2\right)\) tim e

N ote 2 : Currently the routine CSTE \(\mathbb{N}\) is called when an appropriate sigm a_i cannot be chosen in step (c) above. CSTE IN invokesm odified G ram -Schm idt when eigenvalues are close.
N ote 3 : C STEGR w orks only on \(m\) achines \(w\) hich follow ieee-754 floating-point standard in their handling of infinities and N aN s. N orm alexecution of CSTEGR m ay create N aN s and infinities and hence \(m\) ay abortdue to a floating pointexception in environm ents w hich do notconform to the ieee standard.

\section*{ARGUMENTS}

JO B Z (input)
\(=\mathrm{N}:\) : Com pute eigenvalues only;
\(=\mathrm{V}^{\prime}:\) C om pute eigenvalues and eigenvectors.
RANGE (input)
= A ': alleigenvalues w illbe found.
\(=\mathrm{V}\) : alleigenvalues in the half-open interval (VL, VU ] w ill be found. = I': the II th through \(\mathbb{I U}\) th eigenvalues w illbe found.

N (input) The order of the m atrix. \(\mathrm{N}>=0\).

D (input/output)
O n entry, the \(n\) diagonalelem ents of the tridiagonalm atrix T.On exit, D is overw rilten.

E (input/output)
O \(n\) entry, the \((n-1)\) subdiagonal elem ents of the tridiagonal \(m\) atrix \(T\) in elem ents 1 to \(N-1\) of \(E\);
\(E(\mathbb{N})\) need notbe set. On exit, \(E\) is overw rilten.

VL (input)
IfRANGE=V', the low er and upper bounds of the interval to be searched foreigenvalues. VL < V U . N ot referenced ifRANGE = A 'or I'.

VU (input)
IfRANGE=V', the low er and upper bounds of the interval to be searched foreigenvalues. \(\mathrm{VL}<\mathrm{VU}\). N ot referenced ifRANGE=A 'or 'I'.

II (input)
IfRA N G E = ' 1 ', the indiges (in ascending order) of the smallest and largest eigenvalues to be retumed. \(1<=\mathbb{L}<=\mathbb{U}<=N\), if \(N>0\); \(\mathbb{L}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N otreferenced ifRANGE \(=\) A'or V'.

IU (input)
IfRA N G E = I', the indices (in ascending order) of the sm allest and largest eigenvalues to be retumed. \(1<=\mathbb{L}<=\mathbb{U}<=\mathrm{N}\), if \(\mathrm{N}>0\); \(\mathbb{L}=1\) and \(\mathbb{U}=0\) if \(\mathrm{N}=0\). N ot referenced ifRANGE= A 'or V'.

ABSTOL (input)
The absolute enror tolerance for the eigenvalues/eigenvectors. \(\mathbb{F} J O B Z=V^{\prime}\) ', the eigenvalues and eigenvectors output have residual norms bounded by ABSTOL, and the dotproducts betw een different eigenvectors are bounded by ABSTOL. If ABSTOL is less than N *EPS*| \(\mid\), then \(\mathrm{N} * E P S *|T| w\) ill be used in its place, w here EPS is the \(m\) achine precision and \(F \mid\) is the 1 -norm of the tridiagonalm atrix. The eigenvalues are com puted to an accuracy ofEPS* \(\mid\) |imespective of A BSTOL. If high relative accuracy is im portant, setA BSTO L to DLAMCH (Safem inim um '). See Barlow and Dem m el "C om puting A ccurate Eigensystem s of Scaled D iagonally D om inantM atrices", LA PA CK W orking N ote \#7 for a discussion of \(w\) hich \(m\) atrioes define their eigenvalues to high relative accuracy.

M (output)
The total num ber ofeigenvalues found. \(0<=\mathrm{M}\) <= N. IfRANGE = \(A\) ', \(M=N\), and ifRANGE = \(\mathrm{I}^{\prime}, \mathrm{M}=\) \(\mathbb{U}-\mathbb{I}+1\).

W (output)
The firstM elem ents contain the selected eigenvalues in ascending order.
\(Z\) (input) If \(\mathcal{O B Z}=\mathrm{V}^{\prime}\), then if \(\mathbb{N F O}=0\), the first M colum ns of \(Z\) contain the orthonorm aleigenvectors of them atrix T corresponding to the selected eigenvalues, w ith the i-th colum n of Z holding the eigenvector associated with W (i). If \(\mathrm{JOBZ}=\mathrm{N}\) ', then \(Z\) is not referenced. N ote: the userm ust ensure that at leastm ax ( \(1, \mathrm{M}\) ) colum ns are supplied in the array \(Z\); ifRANGE = V', the exact value of \(M\) is not know \(n\) in advance and an upperbound \(m\) ust be used.

LD Z (input)
The leading \(d i m\) ension of the array \(Z . L D Z>=1\), and if \(\mathrm{JOBZ}=\mathrm{V}\) ', LD Z >= \(\mathrm{max}(1, \mathrm{~N})\).

\section*{ISU PPZ (output)}

The support of the eigenvectors in \(Z\), i.e., the indices indicating the nonzero elem ents in Z . The i-th eigenvector is nonzero only in elem ents ISU PPZ ( \(2 * i-1\) ) through ISU PPZ (2*i).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK ( 1 ) retums the optim al (and minim al) LW ORK .

LW ORK (input)
The dim ension of the array W ORK. LW ORK >= \(\max (1,18 * N)\)

IfLW O RK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK aray, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{I W}\) ORK (w orkspace/output)
On exit, if \(\mathbb{N}\) FO \(=0, \mathbb{I W}\) ORK (1) retums the optim al LIW ORK.

LIV ORK (input)
The dim ension of the array \(\mathbb{I W}\) ORK. L \(\mathbb{I W}\) ORK >= \(\max (1,10 * N)\)

IfLIW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the \(\mathbb{I W}\) ORK array, retums this value as the first entry of the \(\mathbb{I W}\) ORK aray, and no errorm essage related to LIW ORK is issued by XERBLA .
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the i-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=1\), intemalerror in \(S L A R R E\), if \(\mathbb{N} F O=2\), intemalemor in CLARRV.

\section*{FURTHER DETAILS}

B ased on contributions by
Inder \(\ddot{H}\) D hillon, \(\mathbb{B M}\) A \(\operatorname{lm}\) aden, U SA
O sniM arques, LBNLN ER SC , U SA
\(K\) en Stanley, C om puterScience D ivision, U niversity of C alifomia atB erkeley, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zstein - com pute the eigenvectors of a real sym \(m\) etric tridiagonal \(m\) atrix \(T\) comesponding to specified eigenvalues, using inverse iteration

\section*{SYNOPSIS}

```

    \mathbb{FA}|,\mathbb{NNOO}
    DOUBLE COM PLEX Z (LD Z,*)
\mathbb{NTEGER N,M,LDZ,INFO}

```

```

DOUBLE PRECISIOND (*),E (*),W (*),WORK (*)
SUBROUT\mathbb{NE ZSTEIN_64N,D,E,M,W, IBLOCK,ISPLIT,Z,LDZ,W ORK,}
\mathbb{N}
DOUBLE COM PLEX Z (LD Z,*)
\mathbb{NTEGER*8N,M,LDZ,INFO}

```

```

DOUBLE PRECISION D (*),E (*),W (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE STE \(\mathbb{N}\) ( \(\mathbb{N}], D, E, \mathbb{M}], W, \mathbb{B L O C K}, \operatorname{ISPLIT}, \mathrm{Z},[L D Z],[W\) ORK], [ \(\mathbb{I N}\) ORK], \(\mathbb{F A} \mathbb{I},[\mathbb{N} F O])\)

COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : : : Z
\(\mathbb{N} T E G E R:: N, M, L D Z, \mathbb{N F O}\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{B L O C K}, \operatorname{ISPLIT}, \mathbb{I}\) ORK, \(\mathbb{F} A \mathbb{I}\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::D ,E,W ,W ORK
SU BROUTINE STE \(\left.\mathbb{N} \_64(\mathbb{N}], D, E, \mathbb{M}\right], W, \mathbb{B L O C K}, \operatorname{ISPLIT}, \mathrm{Z},[\operatorname{LD} Z]\),
[W ORK], [IW ORK], \(\mathbb{F} A \mathbb{I},[\mathbb{N F O}])\)

COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::Z
\(\mathbb{N} T E G E R(8):: N, M, L D Z, \mathbb{N F O}\)
\(\mathbb{N}\) TEGER (8), D \(\mathbb{I M}\) ENSION (:) :: \(\mathbb{B L O C K}, \operatorname{ISPLIT}, \mathbb{I N} O R K, \mathbb{F A} \mathbb{I}\) REAL (8),D \(\mathbb{M}\) ENSION (:) ::D , E, W ,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zstein (intn, double *d, double \({ }^{*}\), intm, double \({ }^{*}\) w, int *iblock, int *isplit, doublecom plex *z, int ldz, int *ifail, int *info);
void zstein_64 long n, double *d, double *e, long m, double *w, long *iblock, long *isplit, doublecom plex *z, long ldz, long *ifail, long *info);

\section*{PURPOSE}
zstein com putes the eigenvectors of a realsym \(m\) etric tridiagonal \(m\) atrix \(T\) corresponding to specified eigenvahues, using inverse iteration.

Them axim um num ber of terations allow ed for each eigenvector is specified by an intemalparam eterM A X ITS (currently set to 5).

A though the eigenvectors are real, they are stored in a com plex array, which m ay be passed to CUNM TR orCUPM TR for back transform ation to the eigenvectors of a com plex H erm itian \(m\) atrix w hich \(w\) as reduced to tridiagonal form .

\section*{ARGUMENTS}

N (input) The order of the m atrix. \(\mathrm{N}>=0\).

D (input) The \(n\) diagonalelem ents of the tridiagonal \(m\) atrix T.

E (input) The ( \(\mathrm{n}-1\) ) subdiagonalelem ents of the tridiagonal \(m\) atrix \(T\), stored in elem ents 1 to \(N-1 ; E(\mathbb{N})\) need notbe set.

M (input) The num ber of eigenvectors to be found. \(0<=\mathrm{M}<=\) N .

W (input) The firstM elem ents of \(W\) contain the eigenvalues for which eigenvectors are to be com puted. The eigenvalues should be grouped by split-off block and ordered from sm allest to largestw ithin the block. (The output anay \(W\) from SSTEBZ w ith ORDER = B'is expected here.)

IBLOCK (input)
The subm atrix indices associated \(w\) ith the corresponding eigenvalues in W ; \(\mathbb{B L O C K}(i)=1\) if eigenvalueW (i) belongs to the first subm atrix from the top, \(=2\) ifW (i) belongs to the second subm atrix, etc. (The output array \(\mathbb{B L O C K}\) from SSTEBZ is expected here.)

ISPLIT (input)
The splilting points, atw hich \(T\) breaks up into subm atrices. The first subm atrix consists of row s/columns 1 to ISPLIT ( 1 ), the second of row s/colum ns ISPLIT ( 1 ) +1 through ISPLIT (2), etc. (The outputarray ISPLIT from SSTEBZ is expected here.)

\section*{Z (output)}

The com puted eigenvectors. The eigenvector associated w ith the eigenvalue W (i) is stored in the \(i-t h\) colum \(n\) of \(Z\). A ny vectorw hich fails to converge is set to its cument iterate afterM AXITS terations. The im aginary parts of the eigenvectors are set to zero.

\section*{LD Z (input)}

The leading dim ension of the aray \(Z\). LD \(Z \quad>=\) \(\max (1, N)\).

W ORK (w orkspace)
dim ension ( \(5 * \mathrm{~N}\) )
IN ORK (w orkspace)
dim ension (N)

IFA II (output)
On norm alexit, allelem ents of \(\mathbb{F} A \mathbb{I}\) are zero.
If one orm ore eigenvectors fail to converge after
M AX ITS iterations, then their indices are stored
in array \(\mathbb{F A} \mathbb{I}\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an ille-
galvalue
\(>0:\) if \(\mathbb{N F O}=i\), then \(i\) eigenvectors failed to converge in M AXITS terations. Their indioes are stored in array \(\mathbb{F A} \mathbb{I I}\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zsteqr-com pute alleigenvalues and, optionally, eigenvectors of a sym \(m\) etric tridiagonalm atrix using the im plicit \(Q L\) orQ \(R\) m ethod

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZSTEQR (COMPZ,N,D,E,Z,LDZ,W ORK,INFO)}
CHARACTER * 1 COMPZ
DOUBLE COM PLEX Z (LD Z,*)
\mathbb{NTEGER N,LD Z,INFO}
DOUBLE PRECISION D (*),E (*),W ORK (*)
SUBROUT\mathbb{NE ZSTEQR_64 COM PZ,N,D,E,Z,LD Z,W ORK,INFO)}
CHARACTER * 1 COMPZ
DOUBLE COM PLEX Z (LD Z,*)
INTEGER*8N,LD Z,INFO
DOUBLE PRECISION D (*),E (*),W ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE STEQR (COMPZ, \(\mathbb{N}], D, E, Z,[L D Z],[W O R K],[\mathbb{N} F O])\)

CHARACTER (LEN=1) ::COM PZ
COM PLEX (8),D IM ENSION (:,:) :: Z
\(\mathbb{N} T E G E R:: N, L D Z, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK
SU BROUTINE STEQR_64 (COMPZ, \(\mathbb{N}], D, E, Z,[L D Z],[W O R K],[\mathbb{N} F O])\)
CHARACTER (LEN=1) ::COM PZ
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) :: Z
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} \mathrm{Z}, \mathbb{N}\) FO
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D ,E,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zsteqr(char com pz, intn, double *d, double *e, doublecom plex *z, int ldz, int *info);
void zsteqr_64 (charcom pz, long n, double *d, double *e, doublecom plex *z, long ldz, long *info);

\section*{PURPOSE}
zsteqr computes all eigenvalues and, optionally, eigenvectors of a sym \(m\) etric tridiagonalm atrix using the im plicit Q L orQ \(R\) m ethod. The eigenvectors of a full or band com plex H erm itian m atrix can also be found ifCHETRD or CHPTRD orCHBTRD has been used to reduce thism atrix to tridiagonalform.

\section*{ARGUMENTS}

COMPZ (input)
\(=\mathrm{N}\) : C om pute eigenvahues only .
\(=\mathrm{V}\) ': Com pute eigenvalues and eigenvectors of the original \(H\) erm itian \(m\) atrix. O \(n\) entry, \(Z\) m ust contain the unitary \(m\) atrix used to reduce the originalm atrix to tridiagonal form . = ' I ': C om pute eigenvalues and eigenvectors of the tridiagonal m atrix. Z is initialized to the identity m atrix.

N (input) The order of the m atrix. \(\mathrm{N}>=0\).

D (input/output)
O n entry, the diagonal elem ents of the tridiagonal m atrix. On exit, if \(\mathbb{N} F O=0\), the eigenvalues in ascending order.

E (input/output)
O \(n\) entry, the ( \(n-1\) ) subdiagonal elem ents of the tridiagonal \(m\) atrix. On exit, E has been destroyed.

Z (input) On entry, if COMPZ \(=V\) ', then \(Z\) contains the unitary \(m\) atrix used in the reduction to tridiagonalform . On exit, if \(\mathbb{N F O}=0\), then if \(\mathrm{COMPZ}=\)

V', Z contains the orthonorm aleigenvectors of the original H erm itian \(m\) atrix, and if \(C O M P Z=' I\) ', \(Z\) contains the orthonorm al eigenvectors of the sym \(m\) etric tridiagonalm atrix. If \(C O M P Z=N\) ', then \(Z\) is not referenced.

LD \(Z\) (input)
The leading dim ension of the array Z . LD \(\mathrm{Z}>=1\), and if eigenvectors are desired, then LD Z >= \(\max (1, N)\).

W ORK (w orkspace)
dim ension (max (1,2*N-2)) IfCOM PZ = \(N\) ', then \(W\) ORK is not referenced.
\(\mathbb{N}\) FO (output)
= 0 : successfulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum enthad an illegalvalue
>0: the algorithm has failed to find all the eigenvalues in a total of \(30 *\) N iterations; if \(\mathbb{N}\) FO \(=i\), then ielem ents of \(E\) have not converged to zero; on exit, \(D\) and \(E\) contain the elem ents of a sym \(m\) etric tridiagonalm atrix which is unitarily sim ilar to the originalm atrix.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zstsv -com pute the solution to a com plex system of linear equations \(A * X=B\) where \(A\) is a \(H\) erm tian tridiagonal \(m\) atrix

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZSTSV (N,NRHS,L,D,SUBL,B,LDB, \mathbb{PIV,INFO)}}\mathbf{N},\textrm{N},\textrm{N}
DOUBLE COM PLEX L (*),D (*),SUBL (*),B (LDB,*)
\mathbb{NTEGER N,NRHS,LDB,}\mathbb{N}FO
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(*)}

```

```

DOUBLE COM PLEXL (*),D (*),SUBL (*),B (LDB,*)
\mathbb{NTEGER*8N,NRHS,LDB,INFO}
\mathbb{NTEGER** \mathbb{PIV (*)}}\mathbf{*}\mathrm{ ( }
F95 INTERFACE
SU BROUTINE STSV $\mathbb{N}, N R H S, L, D, S U B L, B,[L D B], \mathbb{P} \mathbb{I},[\mathbb{N} F O])$
COMPLEX (8),D $\mathbb{M}$ ENSION (:) ::L,D,SUBL
COM PLEX (8), D $\mathbb{M}$ ENSION (: : : : : $:$ B
$\mathbb{N}$ TEGER ::N,NRHS,LDB, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENS} \mathbb{O} N(:):: \mathbb{P} \mathbb{V}$
SUBROUTINE STSV_64 $\mathbb{N}, N R H S, L, D, S U B L, B,[L D B], \mathbb{P} \mathbb{I},[\mathbb{N} F O])$
COMPLEX (8),D $\mathbb{M}$ ENSION (:) ::L,D,SUBL
COM PLEX (8),D $\mathbb{M}$ ENSION (:,:) ::B
$\mathbb{N}$ TEGER (8) :: N,NRHS,LDB, $\mathbb{N} F O$
$\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{I} \operatorname{ENS} \operatorname{ION}(:):: \mathbb{P} \mathbb{V}$

```

\section*{C INTERFACE}
\#include < sunperfh>
void zstsv (intn, intnrhs, doublecom plex *l, doublecom plex
*d, doublecom plex *subl, doublecom plex *b, int ldb, int *ịìiv, int*info);
void zstsv_64 (long n, long nrhs, doublecom plex *l, doublecom plex *d, doublecom plex *subl, doublecom plex *b, long ldb, long *ipiv, long *info);

\section*{PURPOSE}
zstsv com putes the solution to a com plex system of linear equations \(A * X=B\) where \(A\) is a \(H\) erm tian tridiagonal \(m\) atrix.

\section*{ARGUMENTS}

N (input)
The order of them atrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides in \(B\).
L (input/output)
COM PLEX array, dim ension \(\mathbb{N}\) )
O n entry, the n-1 subdiagonalelem ents of the tridiagonal m atrix A. On exit, part of the factorization ofA.

D (input/output)
REA L array, dim ension (N)
O n entry, the n diagonalelem ents of the tridiagonalm atrix A. On exit, the \(n\) diagonalelem ents of the diagonalm atrix \(D\) from the factorization of \(A\).

SUBL (output)
COM PLEX array, dim ension \(\mathbb{N}\) )
On exit, part of the factorization of .
B (input/output)
The colum ns ofB contain the righthand sides.
LD B (input)
The leading dim ension of \(B\) as specified in a type
orD \(\mathbb{I M}\) ENSION statem ent.

IPIV (output)
\(\mathbb{N}\) TEGER array, dim ension \(\mathbb{N}\) )
O \(n\) exit, the pivot indices of the factorization.
\(\mathbb{I N F O}\) (output)
IN TEGER
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=i, D(k, k)\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix D is exactly singular and division by zero w illoccur if it is used to solve a system of equations.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zsthrf-com pute the factorization of a com plex Herm itian tridiagonalm atrix A

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZSTTRF N,L,D,SUBL,\mathbb{PIV,INFO)}}\mathbf{N},\mp@code{L}
D OU BLE COM PLEX L (*),D (*),SUBL (*)
INTEGER N,\mathbb{NFO}
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(})

```

```

D OUBLE COM PLEX L (*),D (*),SUBL (*)
INTEGER*8N,\mathbb{NFO}
\mathbb{NTEGER** \mathbb{PIV (*)}}\mathbf{*}/
F95 INTERFACE

```

```

    COM PLEX (8),D IM ENSION (:) ::L,D,SUBL
    \mathbb{NTEGER ::N,\mathbb{NFO}}0=0
    INTEGER,D IM ENSION (:) :: \mathbb{P IV}
    ```

```

    COM PLEX (8),D IM ENSION (:) ::L,D ,SUBL
    \mathbb{NTEGER (8)::N,\mathbb{NFO}}\mathbf{N}=0
    INTEGER (8),D IM ENSION (:) ::\mathbb{PIV}
    ```
C INTERFACE
    \#include <sunperfh>
void zsthrf(intn, doublecom plex *l, doublecom plex *d, doublecom plex *subl, int *ịiv, int *info);
void zsturf_64 (long n, doublecom plex *l, doublecom plex *d, doublecom plex *subl, long *ípiv, long *info);

\section*{PURPOSE}
zsturf com putes the \(L * D * L * * H\) factorization of a com plex H er\(m\) itian tridiagonalm atrix A.

\section*{ARGUMENTS}

N (input) \(\mathbb{N}\) TEGER
The order of them atrix A. N \(>=0\).

L (input/output)
COM PLEX aray, dim ension \((\mathbb{N})\)
O n entry, the n-1 subdiagonalelem ents of the tridiagonal m atrix A. On exit, part of the factorization of A .

D (input/output)
REAL array, dim ension \((\mathbb{N})\)
O n entry, the \(n\) diagonalelem ents of the tridiagonalm atrix A. On exit, the n diagonalelem ents of the diagonalm atrix D from the factorization of .

SUBL (output)
COM PLEX aray, dim ension \((\mathbb{N})\)
O n exit, part of the factorization of .

IP IV (output)
\(\mathbb{N}\) TEGER array, dim ension \((\mathbb{N})\)
O \(n\) exit, the pivot indices of the factorization.
\(\mathbb{I N} F O\) (output)
IN TEGER
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the i-th argum enthad an ille-
galvalue
\(>0:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{D}(\mathrm{k}, \mathrm{k})\) is exactly zero. The factorization has been com pleted, but the block
diagonalm atrix D is exactly singular and division
by zero w illoccur if it is used to solve a system
of equations.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zsturs - com putes the solution to a com plex system of linear equations \(A * X=B\)

\section*{SYNOPSIS}

```

DOUBLE COM PLEX L (*),D (*),SUBL (*),B (LDB,*)
INTEGERN,NRHS,LDB,INFO
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(*)}
SUBROUT\mathbb{NE ZSTTRS_64 N,NRHS,L,D,SUBL,B,LDB,\mathbb{P}\mathbb{IV,INFO)}}\mathbf{N},\textrm{N}
DOUBLE COM PLEX L (*),D (*),SUBL (*),B (LDB,*)
INTEGER*8N,NRHS,LDB,INFO
\mathbb{NTEGER*8 \mathbb{PIV (*)}}\mathbf{*}/

```
F95 INTERFACE
    SUBROUTINE STTRS \(\mathbb{N}, N R H S, L, D, S U B L, B,[L D B], \mathbb{P} \mathbb{I},[\mathbb{N} F O])\)
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::L,D,SUBL
    COM PLEX (8),D IM ENSION (: : : : : B
    \(\mathbb{N}\) TEGER ::N,NRHS,LDB, \(\mathbb{N}\) FO
    \(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V}\)
    SU BROUTINE STTRS_64 \(\mathbb{N}, N R H S, L, D, S U B L, B,[L D B], \mathbb{P} \mathbb{I},[\mathbb{N} F O])\)
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::L,D,SUBL
    COM PLEX (8), D IM ENSION (: : : : : B
    \(\mathbb{N}\) TEGER (8) :: N,NRHS,LD B, \(\mathbb{N} F O\)
    \(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{I} \operatorname{ENS} \mathbb{O} \mathrm{N}(:):: \mathbb{P} \mathbb{I}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zsttrs (intn, intnrhs, doublecom plex *l, doublecom plex *d, doublecom plex *subl, doublecom plex *b, int ldb, int *íiv, int *info);
void zsttrs_64 (long n, long nrhs, doublecom plex *l, doublecom plex *d, doublecom plex *subl, doublecom plex *b, long ldb, long *ịiv, long *info);

\section*{PURPOSE}
zsttrs com putes the solution to a com plex system of linear equations \(A * X=B\), where \(A\) is an \(N\) boy \(-N\) symm etric tridiagonalm atrix and X and B are N -by-N R H S m atriges.

\section*{ARGUMENTS}

N (input) \(\mathbb{N} T E G E R\)
The order of them atrix A. N >=0.

NRHS (input)
\(\mathbb{N}\) TEGER
The num ber of righthand sides, ie., the num ber of colum ns of the m atrix B.NRH S \(>=0\).

L (input) COM PLEX array, dim ension \((\mathbb{N}-1\) )
O n entry, the subdiagonalelem ents ofLL and D D.

D (input) COM PLEX aray, dim ension \((\mathbb{N})\)
O n entry, the diagonalelem ents ofD D .

SUBL (input)
COM PLEX aray, dim ension \((\mathbb{N}-2)\)
O n entry, the second subdiagonalelem ents of LL .

B (input/output)
COM PLEX amay, dim ension (LD B , NRHS)
On entry, the N -by-NRHS righthand side m atrix B.
On exit, if \(\mathbb{N F O}=0\), the N boy-NRHS solution
\(m\) atrix \(X\).

LD B (input)
IN TEGER
The leading dim ension of the array B. LD B >= \(\max (1, N)\)

IPIV (output)
\(\mathbb{I N}\) TEGER array, dim ension \(\mathbb{N}\) )
D etails of the interchanges and block pivot. If \(\mathbb{P} \mathbb{V}(\mathbb{K})>0,1\) by 1 pivot, and if \(\mathbb{P} \mathbb{I}(\mathbb{K})=K+1\) an interchange done; If \(\mathbb{P} \mathbb{I}(\mathbb{K})<0,2\) by 2 pivot, no interchange required.
\(\mathbb{I N F O}\) (output)
IN TEGER
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-\mathrm{k}\), the k -th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zSw ap -E xchange vectors \(x\) and \(y\).

\section*{SYNOPSIS}

```

DOUBLE COM PLEX X (*),Y (*)
\mathbb{NTEGERN,}\mathbb{NCX,\mathbb{NCY}}\mathbf{}\mathrm{ \}
SU BROUT\mathbb{NE ZSW AP_64 N,X,NNCX,Y, NNCY)}
DOUBLE COM PLEX X (*),Y (*)
INTEGER*8N,\mathbb{NCX,INCY}

```
F95 INTERFACE
    SU BROUTINE SW AP ( \(\mathbb{N}], X,[\mathbb{N} C X], Y,[\mathbb{N} C Y])\)
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::X,Y
    \(\mathbb{N} T E G E R:: N, \mathbb{N} C X, \mathbb{N} C Y\)
    SU BROUTINE SW AP_64 (N ],X, [ \(\mathbb{N} C X], Y,[\mathbb{N} C Y])\)
    COM PLEX (8),D \(\mathbb{I}\) ENSION (:) ::X,Y
    \(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{I N C X}, \mathbb{N} C Y\)
C INTERFACE
    \#include <sunperfh>
    void zsw ap (intn, doublecom plex *x, intincx, doublecom plex
        *y, intincy);
    void zsw ap_64 (long n, doublecom plex *x, long incx, doub-

\section*{PURPOSE}
zsw ap Exchange \(x\) and \(y\) where \(x\) and \(y\) are \(n-v e c t o r s\).

\section*{ARGUMENTS}

N (input)
On entry, \(N\) specifies the num ber of elem ents in the vector. N m ustbe at leastone for the subroutine to have any visible effect. U nchanged on exit.
X (input/output)
( \(1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C X)\) ). On entry, the
increm ented array \(X\) m ustcontain the vector \(x\). On
exit, the \(y\) vector.
\(\mathbb{N C X}\) (input)
On entry, \(\mathbb{N} C X\) specifies the increm ent for the elem ents of \(\mathrm{X} . \mathbb{N} C X\) m ustnotbe zero. U nchanged on exit.

Y (input/output)
( \(1+(\mathrm{n}-1) * \operatorname{abs}(\mathbb{N} C Y)\) ). On entry, the increm ented array \(Y \mathrm{~m}\) ustcontain the vectory. On exit, the x vector.
\(\mathbb{N} C Y\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of Y. \(\mathbb{N} C Y\) m ustnotbe zero. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zsycon -estim ate the reciprocalof the condition num ber (in the 1 -norm ) of a com plex sym \(m\) etric \(m\) atrix \(A\) using the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by CSY TRF

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)
INTEGERN,LDA,}\mathbb{N}F
\mathbb{NTEGER \mathbb{PIVOT(*)}}\mathbf{(})
DOUBLE PRECISION ANORM,RCOND
SUBROUT\mathbb{NE ZSYCON_64(UPLO,N,A,LDA, \mathbb{PIVOT,ANORM,RCOND,WORK,}}\mathbf{N},\textrm{N},\textrm{N}
\mathbb{NFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGER*8N,LDA,INFO}
INTEGER*8 \mathbb{PIVOT (*)}
DOUBLE PRECISION ANORM,RCOND

```

\section*{F95 INTERFACE}
```

SU BROUTINE SYCON (UPLO, $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{V} O T, A N O R M, R C O N D,[W O R K]$, [ $\mathbb{N} F O$ ])

```

CHARACTER (LEN=1)::UPLO
COMPLEX (8), D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: N, L D A, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)

SU BROUT \(\mathbb{N} E\) SYCON_64 (UPLO, \(\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I} O T, A N O R M, R C O N D,[\mathbb{N} O R K]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1)::UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : ::A
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} A, \mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{P} \mathbb{I V O T}\)
REAL (8) ::ANORM,RCOND

\section*{C INTERFACE}
\#include < sunperfh>
void zsycon (char uplo, intn, doublecom plex *a, int lda, int *ịívot, double anorm , double *rcond, int *info);
void zsycon_64 (charuple, long n, doublecom plex *a, long lda, long *ịpívot, double anorm, double *roond, long *info);

\section*{PURPOSE}
zsycon estim ates the reciprocal of the condition num ber (in
the 1 -norm ) of a com plex sym \(m\) etric \(m\) atrix \(A\) using the fac-
torization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by CSY TRF .

A n estim ate is obtained fornorm (inv (A )), and the reciprocal of the condition num ber is com puted as RCOND = 1 / ANORM * norm (inv (A))).

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) ': U ppertriangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\);
= L': Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\).
N (input) The order of the matrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input) The block diagonalm atrix \(D\) and the \(m\) ultipliers used to obtain the factorU orL as com puted by CSY TRF.

LDA (input)
The leading dim ension of the array A. LDA >=
\(\max (1, \mathbb{N})\).
PIVOT (input)
D etails of the interchanges and the block structure ofD as determ ined by CSY TRF.

ANORM (input)
The 1-norm of the originalm atrix A.
RCOND (output)
The reciprocal of the condition number of the
\(m\) atrix \(A\), com puted as RCOND \(=1 /(A N O R M * A \mathbb{N} V N M)\), where \(A \mathbb{N} V N M\) is an estim ate of the 1 -norm of inv (A) com puted in this routine.

W ORK (w orkspace)
dim ension \((2 * N)\)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zsymm -perform one of the \(m\) atrix-m atrix operations \(\quad \mathrm{C}:\) alpha*A *B + beta*C orC := alpha*B *A + beta*C

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZSYMM (SDE,UPLO,M,N,ALPHA,A,LDA,B,LDB,BETA,C,}
LD C )
CHARACTER * 1 SIDE,UPLO
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (LDA,*),B (LDB,*),C (LD C ,*)
INTEGERM,N,LDA,LDB,LDC
SU BROUT\mathbb{NE ZSYMM_64 (S\mathbb{DE,UPLO,M ,N,ALPHA,A,LDA,B,LDB,BETA,C,}}\mathbf{~},\textrm{L}
LD C)
CHARACTER * 1 SIDE,UPLO
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (LDA,*),B (LDB,*),C (LD C ,*)
INTEGER*8M ,N,LDA,LDB,LDC

```

\section*{F95 INTERFACE}
```

SU BROUTINE SYMM (SDE,UPLO, $\mathbb{M}$ ], $\mathbb{N}], A L P H A, A,[L D A], B,[L D B]$, BETA, C, [LDC])
CHARACTER (LEN=1) ::SDE,UPLO
COM PLEX (8) ::ALPHA,BETA
COM PLEX (8), D $\mathbb{M}$ ENSION (:,:) :: A, B, C
$\mathbb{I N}$ TEGER ::M , N,LDA,LDB,LDC
SUBROUTINE SYMM_64 (SDE, UPLO, M ], $\mathbb{N}], A L P H A, A,[L D A], B,[L D B]$, BETA, C, [LDC])

```

CHARACTER ( \([E N=1):: S D E, U P L O\)
COM PLEX (8) ::ALPHA,BETA
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : : A , B , C
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{LD} A, L D B, L D C\)

\section*{C INTERFACE}
\#include <sunperfh>
void zsym m (charside, charuplo, intm, intn, doublecom plex
*alpha, doublecom plex *a, int lda, doublecom plex
*b, int ldlo, doublecom plex *beta, doublecom plex
\({ }^{*} \mathrm{c}\), int ldc);
void zsym m _64 (char side, char uplo, long m , long n, doublecomplex *alpha, doublecom plex *a, long lda, doublecom plex *b, long ldlb, doublecom plex *beta, doublecom plex *c, long ldc);

\section*{PURPOSE}
zsymm perform sone of the \(m\) atrix \(m\) atrix operations \(C:=\) alpha*A *B + beta*C orC \(:=\) alpha*B*A + beta*C where alpha and beta are scalars, \(A\) is a sym \(m\) etric \(m\) atrix and \(B\) and \(C\) are \(m\) by \(n m\) atrices.

\section*{ARGUMENTS}

SID E (input)
On entry, SIDE specifies w hether the sym m etric \(m\) atrix A appears on the leftorright in the operation as follow s:
\(S \mathbb{D} E=\) L'or \(\mathrm{I}^{\prime} \mathrm{C}:=\) alpha*A *B + beta*C,
\(S D E=R\) 'or 'r' \(C:=\) alpha*B *A + beta* \(C\),

U nchanged on exit.

UPLO (input)
On entry, UPLO specifies whether the upper
or lower triangular part of the symm etric
\(m\) atrix \(A\) is to be referenced as follow \(s\) :
\(\mathrm{UPLO}=\mathrm{U}\) 'or L ' Only the upper triangularpart of the sym \(m\) etric \(m\) atrix is to be referenced.

UPLO = 'L 'or I' O nly the low er triangularpart
of the sym \(m\) etric \(m\) atrix is to be referenced.

U nchanged on exit.

M (input)
O n entry, M specifies the num ber of row s of the \(m\) atrix \(C . M>=0\). Unchanged on exit.

N (input)
O n entry, \(N\) specifies the num ber of colum ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\). U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha. U nchanged on exit.

A (input)
COM PLEX *16 aray ofD \(\mathbb{M} E N S I O N\) (LDA, ka ), where ka ism when \(S \mathbb{D} E=\mathrm{L}\) 'or \(\mathrm{I}^{\prime}\) and is n otherw ise.

Before entry w ith \(S \mathbb{D E}=\mathrm{L}\) 'or \({ }^{2}\) ', the \(m\) by \(m\) part of the aray A m ustcontain the sym \(m\) etric \(m\) atrix, such thatw hen \(U P L O=U\) 'or \(L^{\prime}\), the leading \(m\) by \(m\) uppertriangularpart of the array A m ust contain the upper triangular part of the symm etricm atrix and the strictly lower triangularpart of \(A\) is not referenced, and when UPLO = L' or I', the leading m by m low ertriangularpart of the array A m ust contain the low er triangular part of the sym \(m\) etric \(m\) atrix and the strictly upper triangular part of A is notreferenced.

Before entry with \(S \mathbb{D E}=\mathrm{R}\) 'or \(\mathrm{r}^{\prime}\) ', the n by \(n\) part of the aray A mustcontain the sym \(m\) etric \(m\) atrix, such thatw hen \(U P L O=U\) 'or 4 ', the leading \(n\) by \(n\) uppertriangularpart of the array A m ust contain the upper triangular part of the symm etricm atrix and the strictly lower triangularpart of \(A\) is not referenced, and \(w\) hen \(\mathrm{UPLO}=\mathrm{L}\) ' or I ', the leading n by n low er triangularpart of the array A must contain the lower triangular part of the sym \(m\) etric \(m\) atrix and the strictly upper triangular part of A is notreferenced.

U nchanged on exit.

LD A (input)

O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen SDE = L'or I' then LD A >= max ( \(1, \mathrm{~m}\) ), otherw ise LD \(\mathrm{A}>=\max (1, \mathrm{n})\). U nchanged on exit.

B (input)
COM PLEX *16 array ofD \(\mathbb{M}\) ENSION (LD B, n). Before
entry, the leading \(m\) by \(n\) part of the amay \(B\)
\(m\) ust contain the \(m\) atrix \(B\). Unchanged on exit.

LD B (input)
On entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program. LD \(B>=m a x(1, m)\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. W hen BETA is supplied as zero then C need notbe set on input. U nchanged on exit.

C (input/output)
COM PLEX *16 aray ofD \(\mathbb{I M} E N S I O N(L D C, n)\). Before entry, the leading \(m\) by \(n\) partof the array \(C\) \(m\) ustcontain the \(m\) atrix \(C\), exceptw hen beta is zero, in which case \(C\) need notbe seton entry. On exit, the array \(C\) is overw ritten by the \(m\) by \(n\) updated \(m\) atrix.

LD C (input)
On entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program.
LD C \(>=\max (1, m)\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zsyn \(2 k\)-perform one of the sym \(m\) etric rank \(2 k\) operations \(C\) \(:=\) alpha*A *B' + alpha*B*A ' + beta*C orC \(:=\) alpha*A *B + alpha*B *A + beta*C

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZSYR2K (UPLO,TRANSA,N,K,ALPHA,A,LDA,B,LDB,BETA,C,}
LD C )
CHARACTER * 1 UPLO,TRANSA
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (LDA,*),B (LDB,*),C (LDC,*)
\mathbb{N TEGER N,K,LDA,LD B,LDC}
SUBROUT\mathbb{NE ZSYR2K_64 (UPLO,TRANSA,N,K,ALPHA,A,LDA,B,LDB,BETA,}
C,LDC)

```
CHARACTER * 1 UPLO, TRANSA
DOUBLE COM PLEX ALPHA, BETA
D OUBLE COM PLEX A (LDA,*), B (LDB,*), C (LD C ,*)
\(\mathbb{I N}\) TEGER*8N,K,LDA,LDB,LDC

\section*{F95 INTERFACE}

SU BROUTINE SYR2K ©PLO, [TRANSA], \(\mathbb{N}],[K], A L P H A, A,[L D A], B,[L D B]\), BETA, C, [LDC])

CHARACTER (LEN=1) ::UPLO,TRANSA
COMPLEX (8) ::ALPHA,BETA
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A, B, C
\(\mathbb{N}\) TEGER ::N,K,LDA,LDB,LDC
SU BROUTINE SYR2K_64 (UPLO, [TRANSA], \(\mathbb{N}],[K], A L P H A, A,[L D A], B\),
[LD B],BETA, C, [LD C ])

CHARACTER (LEN=1) ::UPLO,TRANSA
COM PLEX (8) ::ALPHA,BETA
COM PLEX (8), D IM ENSION (:,:) ::A,B,C
\(\mathbb{N}\) TEGER (8) ::N , K, LDA ,LDB ,LDC

\section*{C INTERFACE}
\#include <sunperfh>
void zsyr2k (charuplo, chartransa, intn, int k, doublecom plex *alpha, doublecom plex *a, int lda, doublecom plex *b, int ldb, doublecom plex *beta, doublecom plex * \({ }^{\prime}\), int ldc);
void zsyr2k_64 (charuplo, chartransa, long n, long k, doublecom plex *alpha, doublecom plex *a, long lda, doublecom plex *b, long ldb, doublecom plex *beta, doublecom plex *c, long ldc);

\section*{PURPOSE}
zsyn2k perform s one of the sym \(m\) etric rank 2 k operations \(\mathrm{C}:=\) alpha*A *B'+ alpha*B*A'+ beta*C or C : alpha*A *B + alpha*B *A + beta*C where alpha and beta are scalars, C is an \(n\) by \(n\) symm etric \(m\) atrix and \(A\) and \(B\) are \(n\) by \(k\) m atrices in the first case and k by nm atrices in the second case.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the upper or lower triangular part of the array \(C\) is to be referenced as follow s:

UPLO = U'or L' Only the upper triangular partof \(C\) is to be referenced.

UPLO = L'or I' Only the lower triangular partof \(C\) is to be referenced.

U nchanged on exit.

TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} \quad C:=\) alpha*A* \(B^{\prime}+\) alpha*B*A ' + beta* \({ }^{*}\).

TRANSA \(=T^{\prime}\) or t' \(^{\prime}=\) alpha*A *B + alpha*B*A + beta*C.

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF95 \(\mathbb{N}\) TERFACE.

N (input)
O n entry, \(N\) specifies the order of the m atrix C. N m ust.be at least zero. U nchanged on exit.

K (input)
On entry w ith TRANSA \(=N\) 'or \(h\) ', \(K\) specifies the num ber of colum ns of the \(m\) atrioes \(A\) and \(B\), and on entry w th TRANSA \(=\) T' or \(t^{\prime}, \mathrm{K}\) specifies the num ber of row s of the m atriges A and B. K m ustbe at least zero. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha.
U nchanged on exit.

A (input)
COM PLEX *16 aray ofD \(\mathbb{I M} E N S I O N\) (LDA, ka ), where ka isk when TRANSA = N 'or h', and is n otherw ise. Before entry w ith \(\mathrm{TRANSA}=\mathrm{N}^{\prime}\) or
h', the leading \(n\) by k partof the amay A
\(m\) ustcontain the \(m\) atrix \(A\), otherw ise the leading k by n partof the amay \(A\) mustcontain the \(m\) atrix A. U nchanged on exit.

LDA (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen TRANSA \(=\mathrm{N}\) 'or h 'then LDA must be at least \(\max (1, n)\), otherw ise LD A m ustbe at least \(\max (1, k)\). U nchanged on exit.

B (input)
COM PLEX *16 aray ofD \(\mathbb{M}\) ENSION (LD B, kb ) , where kb isk when TRANSA = N 'or h', and is
n otherw ise. Before entry w ith \(\mathrm{TRANSA}=\mathrm{N}^{\prime}\) or
h ', the leading n by k partof the amay \(B\) \(m\) ustcontain the \(m\) atrix \(B\), otherw ise the leading k by n part of the aray \(B \mathrm{~m}\) ustcontain the \(m\) atrix \(B\). Unchanged on exit.

LD B (input)
O n entry, LD B specifies the firstdim ension of B as declared in the calling (sub) program. \(W\) hen TRANSA \(=N\) 'or \(h\) 'then LDB must be at least \(\max (1, n)\), otherw ise LD B m ustbe at least \(\max (1, k)\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. U nchanged on exit.
C (input/output)
COM PLEX *16 aray ofD \(\mathbb{M} E N S I O N\) (LD C , n ) .

Before entry w th UPLO = U 'or L ', the leading \(n\) by \(n\) upper triangular part of the array \(C\) \(m\) ustcontain the upper triangular part of the sym \(m\) etric \(m\) atrix and the strictly low ertriangularpartof C is not referenced. On exit, the upper triangularpart of the array \(C\) is overw ritten by the upper triangularpart of the updated \(m\) atrix.

Before entry w ith UPLO = L'or I', the leading \(n\) by \(n\) low er triangular part of the array \(C\) \(m\) ust contain the low er triangular part of the symm etric \(m\) atrix and the strictly upper triangularpartof C is not referenced. On exit, the low er triangularpart of the amay \(C\) is overw ritten by the low er triangularpart of the updated \(m\) atrix.

LD C (input)
O n entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program.
LD C must be at leastm ax ( \(1, \mathrm{n}\) ). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zsyrfs - im prove the com puted solution to a system of linear equations when the coefficientm atrix is sym \(m\) etric indefinite, and provides errorbounds and backw ard enror estim ates for the solution

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZSYRFS (UPLO,N,NRHS,A,LDA,AF,LDAF, IPIVOT,B,LDB,X,}
LDX,FERR,BERR,W ORK,W ORK 2,INFO)
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
\mathbb{NTEGER N,NRHS,LDA,LDAF,LDB,LDX,}\mathbb{N}FO
INTEGER \mathbb{PIVOT (*)}
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)
SUBROUT\mathbb{NE ZSYRFS_64 (UPLO,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,B,LDB,}}\mathbf{N},
X,LDX,FERR,BERR,W ORK,W ORK 2, \mathbb{NFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
\mathbb{NTEGER*8N,NRHS,LDA,LDAF,LD B,LDX,}\mathbb{N}FO
INTEGER*8 \mathbb{PIVOT (*)}
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE SYRFS (UPLO, N,NRHS,A, [LDA],AF, [LDAF], \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LD} \mathrm{B}]\), X, [LDX],FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1)::UPLO

COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{I M}\) ENSION (:,:) ::A,AF,B,X
\(\mathbb{N}\) TEGER :: N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2
SU BROUTINE SYRFS_64 (UPLO,N,NRHS,A, [LDA],AF, [LDAF], \(\mathbb{P} \mathbb{N O} T, B\), [LD B ], X, [LDX ],FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::UPLO
COM PLEX (8), D IM ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{I M}\) ENSION (:,:) ::A,AF,B,X
\(\mathbb{N}\) TEGER (8) ::N,NRHS,LDA,LDAF,LDB,LDX, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include < sunperfh>
void zsynfs (charuplo, intn, int nrhs, doublecom plex *a, int lda, doublecom plex *af, int ldaf, int *ipivot, doublecom plex *b, int ldb, doublecom plex *x, int ldx, double * ferr, double *berr, int *info);
void zsyrfs_64 (charuplo, long n, long nrhs, doublecom plex *a, long lda, doublecom plex *af, long ldaf, long *ịívot, doublecom plex *b, long ldb, doublecom plex *x, long ldx, double *ferr, double *ber, long *info);

\section*{PURPOSE}
zsyrfs im proves the com puted solution to a system of linear equations when the coefficientm atrix is sym \(m\) etric indefinite, and provides errorbounds and backw ard error estim ates forthe solution.

\section*{ARGUMENTS}

UPLO (input)
= U ': U ppertriangle ofA is stored;
= LL': Low ertriangle ofA is stored.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber
of colum ns of the \(m\) atrices \(B\) and \(X . N R H S>=0\).

A (input) The symm etric m atrix A. IfUPLO = U', the leading N -by -N uppertriangularpartofA contains the upper triangularpart of the \(m\) atrix \(A\), and the strictly low ertriangularpant of A is notreferenced. IfUPLO = L', the leading N -oy N lower triangularpart ofA contains the low er triangular part of them atrix A, and the strictly upper triangularpart of A is notreferenced.

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, N)\).
AF (input)
The factored form of them atrix A. AF contains the block diagonal \(m\) atrix D and the m ultipliers used to obtain the factorU orL from the factorization \(A=U * D * U * * T\) or \(A=L * D * L * * T\) as com puted by CSYTRF.

LD AF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, N)\).

IPIVOT (input)
D etails of the interchanges and the block structure ofD as determ ined by CSY TRF .
\(B\) (input) The righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

X (input/output)
O n entry, the solution \(m\) atrix \(X\), as com puted by CSY TRS. On exit, the im proved solution m atrix X .

LD X (input)
The leading dim ension of the array \(\mathrm{X} . \mathrm{LDX}>=\) \(\max (1, N)\).

\section*{FERR (output)}

The estim ated forw ard errorbound for each solution vector \(X(7)\) (the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution comesponding to \(X(\mathcal{i}), \operatorname{FERR}(\mathcal{)}\) is an estim ated upperbound forthe \(m\) agnitude of the largest ele\(m\) entin (X ( \()\)-X TRUE) divided by the m agnitude of the largestelem entin X ( 7 ). The estim ate is as
reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

BERR (output)
The com ponentw ise relative backw ard error of each solution vectorX ( \(j\) ) (ie., the sm allest relative change in any elem entof \(A\) or \(B\) thatm akes \(X(\mathcal{J})\) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{N}\) FO \(=-\) i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zsyrk -perform one of the sym \(m\) etric rank \(k\) operations \(C\) : alpha*A *A ' beta*C orC : alpha*A *A + beta*C

\section*{SYNOPSIS}
```

SUBROUTINE ZSYRK(UPLO,TRANSA,N,K,ALPHA,A,LDA,BETA,C,LDC)
CHARACTER * 1 UPLO,TRANSA
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (LDA,*),C (LDC ,*)
INTEGER N,K,LDA,LDC
SUBROUTINE ZSYRK_64 (UPLO,TRANSA,N,K,ALPHA,A,LDA,BETA,C,LDC)
CHARACTER * 1 UPLO,TRANSA
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX A (LDA,*),C (LDC,*)
INTEGER*8N,K,LDA,LDC

```

\section*{F95 INTERFACE}

SU BROUTINE SYRK (UPLO, [TRANSA], \(\mathbb{N}],[K], A L P H A, A,[L D A], B E T A, C\), [LD C])

CHARACTER (LEN=1) ::UPLO,TRANSA
COMPLEX (8) ::ALPHA,BETA
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) :: A , C
\(\mathbb{N} T E G E R:: N, K, L D A, L D C\)
SU BROUTINE SYRK_64 (UPLO, [TRANSA], \(\mathbb{N}],[K], A L P H A, A,[L D A], B E T A\), C, (LD C ])

CHARACTER (LEN=1) ::UPLO,TRANSA

COM PLEX (8) :: ALPHA,BETA
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : A , C
\(\mathbb{N}\) TEGER (8) :: \(N\), K, LDA , LD C

\section*{C INTERFACE}
\#include <sunperfh>
void zsyrk (charuplo, chartransa, intn, intk, doublecom plex *alpha, doublecom plex *a, int lda, doublecom plex *beta, doublecom plex * \({ }^{\text {c }}\), int ldc);
void zsydk_64 (char uplo, chartransa, long n, long k, doublecom plex *alpha, doublecomplex *a, long lda, doublecom plex *beta, doublecom plex *c, long ldc);

\section*{PURPOSE}
zsyik perform s one of the sym \(m\) etric rank \(k\) operations \(C:=\) alpha*A *A '+ beta*C orC := alpha*A *A + beta*C where alpha and beta are scalars, \(C\) is an \(n\) by \(n\) sym \(m\) etric \(m\) atrix and \(A\) is an \(n\) by \(k m\) atrix in the first case and a \(k\) by \(n\) \(m\) atrix in the second case.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whether the upper
or low er triangular part of the amay \(C\) is
to be referenced as follow s:
\(U P L O=U\) 'or \(G^{\prime}\) Only the upper triangular part of \(C\) is to be referenced.

UPLO = L'or I' Only the lower triangular part of \(C\) is to be referenced.

U nchanged on exit.

TRANSA (input)
O n entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} \mathrm{C}:=\) alpha*A *A ' + beta* C.

TRANSA \(=\) T'ort' \(\mathrm{C}:=\) alpha*A *A + beta*C.

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF95 \(\mathbb{I N T E R F A C E .}\)
N (input)
On entry, \(N\) specifies the order of the \(m\) atrix \(C\).
N m ustbe at least zero. U nchanged on exit.
\(K\) (input)
On entry with TRANSA \(=N\) 'or \(h\) ', \(K\) specifies the number of columns of the matrix \(A\), and on entry \(w\) th TRANSA \(=T^{\prime}\) or \(t^{\prime}, \mathrm{K}\) specifies the num ber of row sof the m atrix A. K m ust.be at least zero. U nchanged on exit.

ALPHA (input)
On entry, ALPHA specifies the scalar alpha.
U nchanged on exit.

A (input)
COM PLEX *16 anay ofD \(\mathbb{I M}\) ENSION (LDA, ka ), where ka isk when TRANSA = N 'or \(h\) ', and is
n otherw ise. Before entry w th TRANSA \(=\mathrm{N}^{\prime}\) or
h ', the leading n by k part of the array A
m ustcontain the \(m\) atrix \(A\), otherw ise the leading
k by n partof the array A must contain the \(m\) atrix A. U nchanged on exit.

LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program.
W hen TRANSA \(=\mathrm{N}\) 'or h 'then LDA must be at
least \(\max (1, \mathrm{n})\), otherw ise LDA m ust.be at least \(\max (1, k)\). U nchanged on exit.

BETA (input)
On entry, BETA specifies the scalar beta. U nchanged on exit.

C (input/output)
COM PLEX *16 anay ofD \(\mathbb{I}\) ENSION (LD C, n).
Before entry w ith UPLO = U 'or L', the leading \(n\) by \(n\) upper triangularpart of the array \(C\) \(m\) ustcontain the upper triangular part of the sym \(m\) etric \(m\) atrix and the strictly low ertriangularpartofC is not referenced. On exit, the upper triangularpart of the array \(C\) is overw ritten by the upper triangularpart of the updated \(m\) atrix.

Before entry with UPLO = L'or 1', the leading \(n\) by \(n\) low er triangularpart of the anray \(C\) \(m\) ustcontain the low er triangular part of the sym \(m\) etric \(m\) atrix and the strictly upper triangularpartof \(C\) is not referenced. On exit, the low er triangularpart of the array \(C\) is overw ritten by the low er triangularpart of the updated \(m\) atrix.

LD C (input)
On entry, LD C specifies the firstdim ension of C as declared in the calling (sub) program. LD C must be at leastmax(1,n). Unchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zsysv - com pute the solution to a com plex system of linear equations \(A * X=B\),

\section*{SYNOPSIS}
```

SUBROUTINE ZSYSV (UPLO,N,NRHS,A,LDA, \mathbb{PIV,B,LDB,W ORK,LW ORK,}
\mathbb{NFO)}
CHARACTER * 1 UPLO
D OUBLE COM PLEX A (LDA,*),B (LDB,*),W ORK (*)
INTEGERN,NRHS,LDA,LDB,LW ORK,\mathbb{NFO}
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(})
SU BROUTINE ZSY SV_64 (UPLO,N,NRHS,A,LDA,\mathbb{PIV ,B,LDB,W ORK,}
LW ORK,\mathbb{NFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*),W ORK (*)
\mathbb{NTEGER*8N,NRHS,LDA,LDB,LW ORK,INFO}
\mathbb{NTEGER*8 \mathbb{P IV (*)}}\mathbf{(*)}

```

\section*{F95 INTERFACE}

SU BROUTINE SYSV (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I}, B,[L D B],[W\) ORK], [LW ORK], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A,B
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, L W O R K, \mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}\)
SU BROUTINE SYSV_64 (UPLO, \(\mathbb{N}], \mathbb{N} R H S], A,[L D A], \mathbb{P} \mathbb{I}, B,[L D B],[W\) ORK],
\[
[L W \text { ORK ], [ } \mathbb{N} F O] \text { ) }
\]

CHARACTER (LEN=1) ::UPLO
COMPLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : : A , B
\(\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D B, L W O R K, \mathbb{N F O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zsysv (charuplo, intn, intnrhs, doublecom plex *a, int lda, int *ípívot, doublecom plex *b, int ldb, int *info);
void zsysv_64 (charuplo, long n, long nrhs, doublecom plex
*a, long lda, long *ípívot, doublecom plex *b, long ldlo, long *info);

\section*{PURPOSE}
zsysv com putes the solution to a com plex system of linear equations
\(A\) * \(X=B\), where \(A\) is an \(N\)-by-N symm etric \(m\) atrix and \(X\) and \(B\) are N -by-N RH S m atrices.

The diagonal pivoting \(m\) ethod is used to factorA as
\[
\begin{aligned}
& A=U * D * U * * T, \text { if } U P L O=U \prime \text {, or } \\
& A=L * D * L * * T, \text { if } U P L O=L \prime
\end{aligned}
\]
where \(U\) (orL) is a product of perm utation and unit upper (low er) triangular \(m\) atrioes, and \(D\) is sym \(m\) etric and block diagonalw ith 1 -by-1 and 2 -by- 2 diagonalblocks. The factored form of \(A\) is then used to solve the system of equations \(A * X=B\).

\section*{ARGUMENTS}

\section*{UPLO (input)}
= U :: U pper triangle ofA is stored;
\(=\mathrm{L}\) ': Low er triangle of A is stored.

N (input) The num ber of linearequations, i.e., the order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colm ns of them atrix B. NRHS \(>=0\).

A (input/output)
O n entry, the sym m etric m atrix A. If UPLO = U', the leading N -by -N uppertriangularpartofA contains the uppertriangular part of the \(m\) atrix \(A\), and the strictly low ertriangularpartofA is not referenced. If UPLO = L', the leading N -by-N low er triangularpart of A contains the low ertriangularpartof the m atrix A, and the strictly upper triangular part of A is not referenced.

On exit, if \(\mathbb{N F O}=0\), the block diagonalm atrix \(D\) and the \(m\) ultipliers used to obtain the factor \(U\) or L from the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) or \(\mathrm{A}=\) \(\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) as com puted by CSY TRF.
LDA (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

IP IV (output)
D etails of the interchanges and the block structure ofD , as determ ined by CSY TRF. If IP IV (k) > 0 , then row \(s\) and colum ns \(k\) and \(\mathbb{P}\) IV ( \(k\) ) were interchanged, and \(\mathrm{D}(\mathrm{k}, \mathrm{k})\) is a 1-by-1 diagonalblock. If UPLO \(=U\) 'and \(\mathbb{P} \mathbb{I V}(k)=\mathbb{P} \mathbb{I}(k-1)<0\), then row s and columns \(k-1\) and \(-\mathbb{P} \mathbb{I V}(k)\) were interchanged and \(D(k-1 k, k-1 k)\) is a \(2-b y-2\) diagonal block. IfU PLO = L'and \(\mathbb{P} \mathbb{I V}(k)=\mathbb{P} \mathbb{I}(k+1)<0\), then row \(s\) and \(c o l u m n s k+1\) and \(-\mathbb{P} \mathbb{I V}(k)\) w ere interchanged and \(\mathrm{D}(\mathrm{k} k+1, \mathrm{k} k+1)\) is a 2 -by-2 diagonal block.

B (input/output)
On entry, the N -by-NRH \(S\) righthand side \(m\) atrix \(B\). On exi, if \(\mathbb{N} F O=0\), the \(N\) by \(N\) RH solution \(m\) atrix X.

LD B (input)
The leading dim ension of the anay B . LD B >= max ( \(1, \mathbb{N}\) ).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, \mathrm{~W} O R K(1)\) retums the optim al LW ORK.

LW ORK (input)
The length ofW ORK. LW ORK >=1, and forbestperform ance LW ORK \(>=N * N B\), where \(N B\) is the optim al blocksize forC SY TRF.

IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N F O}=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=\mathrm{i}, \mathrm{D}(i, i)\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, so the solution could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zsysvx - use the diagonalpivoting factorization to com pute the solution to a com plex system of linearequations A * X = B,

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZSYSVX (FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,B,}}\mathbf{N},\mp@code{N},
LDB,X,LDX,RCOND,FERR,BERR,W ORK,LDW ORK,W ORK2,INFO)
CHARACTER * 1FACT,UPLO
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
INTEGER N,NRHS,LDA,LDAF,LDB,LDX,LDW ORK,\mathbb{NFO}
\mathbb{NTEGER IPIVOT (*)}
DOUBLE PRECISION RCOND
D OUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)
SUBROUT\mathbb{NE ZSYSVX_64(FACT,UPLO,N,NRHS,A,LDA,AF,LDAF,\mathbb{PIVOT,}}\mathbf{N},\mp@code{N},\mp@code{N}
B,LDB,X,LDX,RCOND,FERR,BERR,W ORK,LDW ORK,W ORK 2, INFO)
CHARACTER * 1 FACT,UPLO
DOUBLE COM PLEX A (LDA,*), AF (LDAF,*), B (LDB,*), X (LDX,*),
W ORK (*)
INTEGER*8N,NRHS,LDA,LDAF,LDB,LDX,LDW ORK,INFO
INTEGER*8 \mathbb{P IVOT (*)}
DOUBLE PRECISION RCOND
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUT \(\mathbb{N} E S Y S V X \mathbb{F A C T}, \mathrm{UPLO}, \mathrm{N}, \mathrm{NRHS}, \mathrm{A},[L D A], A F,[L D A F], \mathbb{P} \mathbb{I} O T\), B, [LDB], X, [LDX ],RCOND ,FERR,BERR, [W ORK ], [LD W ORK ], [W ORK 2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::FACT, UPLO
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{I M} E N S I O N(:,:):: A, A F, B, X\)
\(\mathbb{N}\) TEGER ::N,NRHS,LDA,LDAF,LDB,LDX,LDWORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)
REAL (8) ::RCOND
REAL (8), D \(\mathbb{I}\) ENSION (:) :: FERR,BERR,W ORK2

SU BROUTINE SYSVX_64 (FACT,UPLO,N,NRHS,A, [LDA],AF, [LDAF], \(\mathbb{P} I V O T, B,[L D B], X,[L D X], R C O N D, F E R R, B E R R,[W O R K],[L D W O R K]\), [ W ORK2], \([\mathbb{N} \mathrm{FO}]\) )

CHARACTER (LEN=1) ::FACT, UPLO
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A,AF,B,X
\(\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D A F, L D B, L D X, L D W O R K, \mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION(:)::\mathbb {P}\mathbb {V}OT}\)
REAL (8) :: RCOND
REAL (8), D \(\mathbb{I}\) ENSION (:) :: FERR, BERR,W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void zsysvx (char fact, char uplo, int n, int nins, doublecom plex *a, int lda, doublecom plex *af, int ldaf, int *ipivot, doublecom plex *b, int ldb, doublecom plex *x, intldx, double *rcond, double * ferr, double *berr, int *info);
void zsysvx_64 (char fact, char uplo, long n, long nrhs, doublecom plex *a, long lda, doublecom plex *af, long ldaf, long *ipivot, doublecom plex *b, long ldb, doublecom plex *x, long ldx, double *roond, double * ferr, double *berr, long *info);

\section*{PURPOSE}
zsysvx uses the diagonalpivoting factorization to com pute the solution to a com plex system of linear equations \(A * X=\) \(B\), where \(A\) is an \(N\) boy \(N\) sym \(m\) etric \(m\) atrix and \(X\) and \(B\) are \(N-\) by-N R H S m atrices.

E rrorbounds on the solution and a condition estim ate are also provided.

The follow ing steps are perform ed:
1. IfFACT \(=N\) ', the diagonalpivoting \(m\) ethod is used to
factorA.
The form of the factorization is
\[
\begin{aligned}
& A=U * D * U * * T, \text { if } U P L O=U ' \text {, or } \\
& A=L * D * L * * T, \text { if } U P L O=L \prime
\end{aligned}
\]
where \(U\) (orL) is a product of perm utation and unit upper (low er)
triangularm atrices, and \(D\) is sym \(m\) etric and block diagonalw th
1-by-1 and 2-by-2 diagonalblocks.
2. If som eD \((i, i)=0\), so thatD is exactly singular, then the routine
retums with \(\mathbb{I N} F O=\) i. O therw ise, the factored form of A is used
to estim ate the condition num ber of the \(m\) atrix \(A\). If the
reciprocal of the condition num ber is less than \(m\) achine precision,
\(\mathbb{N} F O=N+1\) is retumed as a waming, but the routine stillgoes on
to solve for \(X\) and com pute enror bounds as described below .
3.The system ofequations is solved for \(X\) using the factored form of A.
4. Iterative refinem ent is applied to im prove the com puted solution
\(m\) atrix and calculate error bounds and backw ard error estim ates
for it.

\section*{ARGUMENTS}

FACT (input)
Specifies w hether ornot the factored form of A has been supplied on entry. = F ': On entry, A F and \(\mathbb{P}\) IV OT contain the factored form of A. A, AF and \(\mathbb{P} \mathbb{V O T}\) will not be modified. = N : The \(m\) atrix A w illlbe copied to A F and factored.

UPLO (input)
= U :: U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The num ber of linearequations, ie., the order of them atrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atriges B and X . NRHS \(>=0\).

A (input) The symm etric matrix A. IfUPLO = U', the leading N -by -N uppertriangularpart of A contains the uppertriangularpart of the \(m\) atrix \(A\), and the strictly low ertriangularpart of A is notreferenced. IfUPLO = L', the leading N -by-N lower triangularpart ofA contains the low er triangular part of the m atrix A, and the strictly upper triangularpart ofA is notreferenced.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).
AF (input/output)
IfFACT \(=F^{\prime}\), then \(A F\) is an inputargum entand on entry contains the block diagonalm atrix \(D\) and the m ultipliers used to obtain the factor U orL from the factorization \(A=U * D * U * * T\) or \(A=L * D * L * * T\) as com puted by C SY TRF .

IfFACT \(=N\) ', then \(A F\) is an output argum ent and on exit retums the block diagonalm atrix D and the m ultipliers used to obtain the factorU or L from the factorization \(A=U * D * U * * T\) or \(A=\) L*D*L**T.

LDAF (input)
The leading dim ension of the array AF. LDAF >= \(\max (1, N)\).

IPIVOT (inputoroutput)
IfFACT = \(\mathrm{F}^{\prime}\), then \(\mathbb{P} \mathbb{I V O T}\) is an input argum ent and on entry contains details of the interchanges and the block structure of D , as determ ined by CSY TRF. If \(\mathbb{P} \mathbb{I V O T}(k)>0\), then row \(s\) and colum nsk and \(\mathbb{P} \mathbb{I V O T}(k)\) w ere interchanged and \(D(k, k)\) is a 1 boy-1 diagonal block. If \(\mathrm{UPLO}=\mathrm{U}\) ' and \(\mathbb{P} \operatorname{IV} \circ T(\mathrm{k})=\mathbb{P} \mathbb{I} \circ \mathrm{T}(\mathrm{k}-1)<0\), then row s and colum ns \(\mathrm{k}-1\) and - \(\mathbb{P}\) IVOT (k) were interchanged and \(\mathrm{D}(\mathrm{k}-\) \(1 \mathrm{k}, \mathrm{k}-1 \mathrm{k}\) ) is a 2-by-2 diagonalblock. If \(\mathrm{PLO}=\) \(\mathbb{L}\) 'and \(\mathbb{P} \mathbb{I V O T}(k)=\mathbb{P} \mathbb{V} O T(k+1)<0\), then row sand colum nsk+1 and -TP IVOT (k) w ere interchanged and D \((k: k+1, k: k+1)\) is a 2 -by-2 diagonalblock.

If \(\mathrm{FACT}=\mathrm{N}\) ', then \(\mathbb{P} \mathbb{I V O T}\) is an output argum ent and on exitcontains details of the interchanges and the block structure of \(D\), as determ ined by

CSY TRF.

B (input) The N -by -N RH S righthand side m atrix B .

LD B (input)
The leading dim ension of the array \(B\). LD B >= \(\max (1, N)\).

X (output)
If \(\mathbb{N} F O=0\) or \(\mathbb{N} F O=\mathrm{N}+1\), the N -by-NRHS solution \(m\) atrix \(X\).
LD X (input)
The leading dim ension of the array X . LD X >= \(\max (1, N)\).

RCOND (output)
The estim ate of the reciprocal condition num berof the \(m\) atrix \(A\). IfRCOND is less than the \(m\) achine precision (in particular, if RCOND \(=0\) ), the \(m\) atrix is singular to working precision. This condition is indicated by a retum code of \(\mathbb{N}\) FO > 0 .

\section*{FERR (output)}

The estim ated forw ard enrorbound for each solution vector \(X()\) ) the \(j\) th colum \(n\) of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{O}), \operatorname{FERR}(\mathcal{O})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{H})\)-XTRUE) divided by the magnitude of the largestelem ent in \(X(\mathcal{)}\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector \(X(\mathcal{)}\) ) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \()\) ) an exactsolution).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length ofW ORK. LDW ORK \(>=2 * N\), and for best perform ance LDW ORK \(>=N * N B\), where \(N B\) is the optim alblocksize forCSY TRF.

IfLDW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.

W ORK 2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{N N F O}=-\) i, the \(i\)-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N F O}=i\), and \(i\) is
\(<=\mathrm{N}: \mathrm{D}(i, i)\) is exactly zero. The factorization
has been completed but the factorD is exactly singular, so the solution and error bounds could not be com puted. RCOND \(=0\) is retumed. \(=\mathrm{N}+1\) : D is nonsingular, butRCOND is less than machine precision, \(m\) eaning that the \(m\) atrix is singular to w orking precision. Nevertheless, the solution and error bounds are com puted because there are a num berof situationsw here the com puted solution can bem ore accurate than the value ofRCOND w ould suggest.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zsytf2 -com pute the factorization of a com plex sym m etric \(m\) atrix A using the Bunch \(-K\) aufn an diagonalpivoting \(m\) ethod

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
D OUBLE COM PLEX A (LDA,*)
\mathbb{NTEGERN,LDA,}\mathbb{NNFO}
\mathbb{NTEGER \mathbb{PIV (*)}}\mathbf{(*)}

```

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*)
INTEGER*8N,LDA,INFO
\mathbb{NTEGER*8 \mathbb{PIV (*)}}\mathbf{*}\mathrm{ ( }

```

\section*{F95 INTERFACE}
```

SU BROUTINE SY TF2 (UPLO, $\mathbb{N}], A,[L D A], \mathbb{P} \mathbb{I}$, $[\mathbb{N} F O])$
CHARACTER (LEN=1)::UPLO
COMPLEX (8), D IM ENSION (:,:) ::A
$\mathbb{N}$ TEGER ::N,LDA, $\mathbb{N} F O$
$\mathbb{N} T E G E R, D \mathbb{I M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V}$
SU BROUTINE SYTF2_64 (UPLO, $\mathbb{N}$ ],A, [LDA ], $\mathbb{P} \mathbb{I V},[\mathbb{N} F O])$
CHARACTER (LEN=1)::UPLO
COM PLEX (8),D $\mathbb{I}$ ENSION (:,:) ::A

```
\(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} A, \mathbb{N}\) FO
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zsytf2 (charuplo, intn, doublecom plex *a, intlda, int *ịív, int *info);
void zsyt+2_64 (charuplo, long n, doublecom plex *a, long lda, long *ịiv, long *info);

\section*{PURPOSE}
zsytf2 com putes the factorization of a com plex symm etric \(m\) atrix A using the B unch \(K\) aufm an diagonalpivoting \(m\) ethod:
\[
A=U * D * U{ }^{\prime} \text { or } A=L * D * L^{\prime}
\]
where U (orL) is a productofperm utation and unit upper (low er) triangularm atriges, U 'is the transpose of U , and D is sym \(m\) etric and block diagonalw ith 1 -by-1 and 2 -by-2 diagonalblocks.

This is the unblocked version of the algorithm, calling Level2 BLAS.

\section*{ARGUMENTS}

\section*{UPLO (input)}

Specifies w hether the upper or low er triangular part of the sym \(m\) etric \(m\) atrix \(A\) is stored:
\(=\mathrm{U}\) ': Upper triangular
\(=\mathrm{L}\) ': Low ertriangular

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the sym m etric m atrix A. If UPLO = U', the leading \(n-b y-n\) uppertriangularpartofA contains the upper triangular part of the \(m\) atrix \(A\), and the strictly low er triangularpart of A is not referenced. If UPLO \(=\mathbb{L}\) ', the leading \(n-b y-n\) low er triangularpart of A contains the low ertriangularpart of the m atrix A, and the strictly upper triangularpart of \(A\) is notreferenced.

On exit, the block diagonalm atrix D and the mul
tipliers used to obtain the factorU orl (see below for further details).

LDA (input)
The leading dim ension of the anray A. LD A >= \(\max (1, \mathbb{N})\).

IPIV (output)
D etails of the interchanges and the block structure ofD. If \(\mathbb{P} \mathbb{I V}(k)>0\), then row \(s\) and colum ns \(k\) and \(\mathbb{P} \mathbb{I V}(k)\) were interchanged and \(D(k, k)\) is a 1 -by-1 diagonalblock. IfUPLO \(=U\) 'and \(\mathbb{P} \mathbb{I V}(k)\) \(=\mathbb{P} \mathbb{V}(k-1)<0\), then row \(s\) and colmm ns \(k-1\) and \(-\mathbb{P} \mathbb{V}(k)\) w ere interchanged and \(D(k-1 * k, k-1 *)\) is a 2 -by-2 diagonalblock. IfUPLO \(=\mathbb{L}\) 'and \(\mathbb{P} \mathbb{I V}(k)\)
\(=\mathbb{P} \mathbb{I}(k+1)<0\), then row sand colum ns \(k+1\) and \(-\mathbb{P} \mathbb{V}(k)\) w ere interchanged and \(D(k, k+1, k \mathrm{k}+1)\) is a 2-by-2 diagonalblock.
\(\mathbb{N F O}\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\mathrm{k}\), the k -th argum enthad an illegalvalue \(>0:\) if \(\mathbb{N} F O=k, D(k, k)\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, and division by zero w illoccur if it is used to solve a system of equations.

\section*{FURTHER DETAILS}

\section*{1-96 -B ased on m odifications by J.Lew is, Boeing Com puter}

\section*{Services}

Com pany
If U PLO \(=\mathrm{U}\) ', then \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}\) ', where
\(U=P(n) \star U(n) * \ldots * P(k) U(k) * \ldots\),
i.e., \(U\) is a productof term \(S P(k) * U(k)\), where \(k\) decreases from \(n\) to 1 in steps of 1 or2, and \(D\) is ablock diagonal \(m\) atrix \(w\) ith 1 -by -1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{V}(k)\), and \(U(k)\) is a unituppertriangularm atrix, such that if the diagonal block \(D(k)\) is of orders ( \(s=1\) or 2 ), then
\[
\begin{aligned}
& \text { ( I v 0 ) k-s } \\
& \mathrm{U}(\mathrm{k})=(0 \mathrm{I} 0) \mathrm{s} \\
& \text { ( } 0 \text { O I ) } \mathrm{n}-\mathrm{k} \\
& \mathrm{k}-\mathrm{s} \mathrm{~s} \mathrm{n}-\mathrm{k}
\end{aligned}
\]

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(1 k-\)
\(1, k)\). If \(s=2\), the upper triangle ofD ( \(k\) ) overw rites \(A(k-\) \(1, k-1)\), A \((k-1, k)\), and \(A(k, k)\), and \(v\) overw rites A ( 1 k- \(2, k-\) \(1 \mathrm{k})\).

If \(\operatorname{PLO}=\mathrm{L}\) ', then \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L}\) ', where
\(\mathrm{L}=\mathrm{P}(1) \star \mathrm{L}(1){ }^{*} \ldots * \mathrm{P}(k) \star \mathrm{L}(k)^{*} \ldots\),
i.e., \(L\) is a product of term \(S P(k) * L(k)\), where \(k\) increases from 1 to \(n\) in steps of 1 or 2, and \(D\) is a block diagonal \(m\) atrix \(w\) ith 1 -by -1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{V}(k)\), and \(L(k)\) is a unit low ertriangularm atrix, such that if the diagonal
block D (k) is of orders ( \(s=1\) or2), then
\[
\begin{aligned}
& \text { ( } \left.\begin{array}{llll}
\mathrm{I} & 0 & 0
\end{array}\right) \mathrm{k}-1 \\
& L(k)=\left(\begin{array}{lll}
0 & I & 0
\end{array}\right) s \\
& \text { ( } 0 \text { v I ) } n-k-s+1 \\
& \text { k-1 s n-k-s+1 }
\end{aligned}
\]

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites
\(A(k+1 n, k)\). If \(s=2\), the low er triangle ofD ( \(k\) ) overw rites A \((k, k), A(k+1, k)\), and \(A(k+1, k+1)\), and \(v\) overw rites A \((k+2 m, k k+1)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zsytrf-com pute the factorization of a com plex sym m etric \(m\) atrix A using the Bunch \(-K\) aufn an diagonalpivoting \(m\) ethod

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)
INTEGERN,LDA,LDWORK,\mathbb{NFO}
INTEGER \mathbb{PIVOT (*)}
SU BROUT\mathbb{NE ZSYTRF_64(UPLO,N,A,LDA, \mathbb{PIVOT,W ORK,LDW ORK,INFO)}}\mathbf{N}\mathrm{ (N,N}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)
INTEGER*8N,LDA,LDW ORK,INFO
INTEGER*8 \mathbb{PIVOT (*)}

```

\section*{F95 INTERFACE}

SU BROUTINE SY TRF (UPLO ,N,A, [LDA], \(\mathbb{P} \mathbb{I V O T}, \mathbb{W}\) ORK ], [LDW ORK ], [ \(\mathbb{N F O}])\)
CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER ::N,LDA,LDW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: \(\mathbb{P} \mathbb{V} O T\)
SU BROUTINE SYTRF_64 (UPLO,N,A, [LDA], \(\mathbb{P} \mathbb{I V O T}, \mathbb{W}\) ORK ], [LDW ORK ], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER (8) ::N,LDA,LDW ORK, \(\mathbb{N} F O\)
\(\mathbb{N} T E G E R(8), D \mathbb{I} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{I V O T}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zsytrf(charuplo, intn, doublecom plex *a, int lda, int
*ịívot, int*info);
void zsytrf_64 (charuplo, long n, doublecom plex *a, long lda, long *ịíivot, long *info);

\section*{PURPOSE}
zsytrf com putes the factorization of a com plex sym m etric \(m\) atrix \(A\) using the \(B\) unch- \(K\) aufn an diagonalpivoting \(m\) ethod. The form of the factorization is
\[
\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T} \text { or } \mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}
\]
where \(U\) (orL) is a productof perm utation and unit upper (low er) triangular \(m\) atrices, and \(D\) is sym \(m\) etric and block diagonalw th w th 1 -by-1 and 2 -by- 2 diagonalblocks.

This is the blocked version of the algorithm, calling Level 3 BLAS.

\section*{ARGUMENTS}

UPLO (input)
= U ': U ppertriangle ofA is stored;
= L': Low ertriangle ofA is stored.

N (input) The order of the m atrix A. N >=0.

A (input/output)
O n entry, the sym m etric m atrix A. If UPLO = U', the leading \(\mathrm{N}-\) by -N uppertriangularpartofA contains the upper triangular part of the \(m\) atrix \(A\), and the strictly low er triangularpart of A is not referenced. If UPLO = L', the leading N -by-N low er triangularpart of \(A\) contains the low ertriangularpartof them atrix A, and the strictly upper triangular part of A is not referenced.

On exit, the block diagonalm atrix D and the multipliers used to obtain the factorU orL (see below for further details).

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

\section*{IPIVOT (output)}

D etails of the interchanges and the block structure of D. If \(\mathbb{P I V O T}(k)>0\), then row sand columnsk and \(\mathbb{P I V O T}(k)\) were interchanged and \(D(k, k)\) is a \(1-b y-1\) diagonalblock. If \(U P L O=U^{\prime}\) and \(\mathbb{P} \mathbb{V} O T(k)=\mathbb{P} \mathbb{V} O T(k-1)<0\), then row \(s\) and colum ns \(k-1\) and - \(\mathbb{P I V O T}(k)\) were interchanged and D ( \(k-1 * k, k-1 k)\) is a \(2-b y-2\) diagonal block. If UPLO \(=\mathrm{L}\) 'and \(\mathbb{P I V O T}(k)=\mathbb{P} \mathbb{I V O T}(k+1)<0\), then row sand colum ns \(k+1\) and \(-\mathbb{P} \mathbb{V} O T(k)\) were interchanged and \(D(k k+1, k k+1)\) is a \(2-b y-2\) diagonal block.

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, \mathrm{~W}\) ORK (1) retums the optim al LDW ORK.

LDW ORK (input)
The length ofW ORK. LDW ORK >=1. Forbestperfor\(m\) ance LDW ORK >=N *NB, where NB is the block size retumed by \(\amalg A E N V\).

IfLDW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LDW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfinlexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N F O}=\mathrm{i}, \mathrm{D}(\mathrm{i}, \mathrm{i})\) is exactly zero. The factorization has been com pleted, but the block diagonalm atrix \(D\) is exactly singular, and division by zero w illoccur if it is used to solve a system of equations.

\section*{FURTHER DETAILS}

If \(\mathrm{ULO}=\mathrm{U}\) ', then \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U}\) ', where
\(U=P(n) \star U(n)^{\star} \ldots{ }^{\star} P(k) U(k)^{\star} \ldots\),
ie., \(U\) is a product ofterm \(\operatorname{sP}(k) * U(k)\), where \(k\) decreases from \(n\) to 1 in steps of 1 or 2 , and \(D\) is a block diagonal \(m\) atrix w ith 1 -by-1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I V O T}(\mathrm{k})\), and \(\mathrm{U}(\mathrm{k})\) is a unituppertriangularm atrix, such that if the diagonal block D (k) is of orders ( \(s=1\) or 2 ), then
\[
\begin{aligned}
& \text { ( I v 0 ) k-s } \\
& U(k)=\left(\begin{array}{lll}
0 & I
\end{array}\right) s \\
& \text { ( } 000 \text { I ) n-k } \\
& \mathrm{k}-\mathrm{s} \text { s n-k }
\end{aligned}
\]

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(1 k-\) \(1, k\) ). If \(s=2\), the upper triangle ofD \((k)\) overw rites \(A(k-\) \(1, k-1), A(k-1, k)\), and \(A(k, k)\), and \(V\) overw rites \(A(1 k-2, k-\) \(1 \mathrm{k})\).

If \(\mathrm{UPLO}=\mathrm{L}\) ', then \(A=\mathrm{L} * \mathrm{D} * \mathrm{~L}\) ', where
\(L=P(1) \star L(1)^{\star} \ldots * P(k) \star L(k) * \ldots\)
ie., \(L\) is a productofterm \(s P(k) * L(k)\), where \(k\) increases
from 1 to n in steps of 1 or 2 , and D is a block diagonal \(m\) atrix \(w\) th 1 -by -1 and 2 -by-2 diagonalblocks \(D(k) . P(k)\) is a perm utation \(m\) atrix as defined by \(\mathbb{P} \mathbb{I V O T}(k)\), and \(L(k)\) is a unitlow ertriangularm atrix, such that if the diagonal
block \(D(k)\) is of orders ( \(s=1\) or 2 ), then
\[
\begin{aligned}
& \text { ( } \left.\begin{array}{llll}
I & 0 & 0
\end{array}\right) k-1 \\
& L(k)=\left(\begin{array}{lll}
0 & I & 0
\end{array}\right) s \\
& \text { ( } 0 \mathrm{~V} \text { I ) } \mathrm{n}-\mathrm{k}-\mathrm{s}+1 \\
& \mathrm{k}-1 \text { s } \mathrm{n}-\mathrm{k}-\mathrm{s}+1
\end{aligned}
\]

If \(s=1, D(k)\) overw rites \(A(k, k)\), and \(v\) overw rites \(A(k+1 m, k)\). If \(s=2\), the low ertriangle ofD \((k)\) overw rites \(A(k, k), A(k+1, k)\), and \(A(k+1, k+1)\), and \(v\) overw rites A \((k+2 m, k: k+1)\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zsytri-com pute the inverse of a com plex sym \(m\) etric indefinte \(m\) atrix \(A\) using the factorization \(A=U * D * U * * T\) orA \(=\) L*D *L**T com puted by CSY TRF

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZSYTRI(UPLO,N,A,LDA,\mathbb{PIVOT,W ORK,INFO)}}\mathbf{N}\mathrm{ (N,}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)
NNTEGER N,LDA,\mathbb{NFO}
INTEGER \mathbb{PIVOT(*)}

```

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),W ORK (*)
INTEGER*8N,LDA,INFO
INTEGER*8 \mathbb{PIVOT (*)}

```
F95 INTERFACE
    SU BROUTINE SYTRI(UPLO,N,A, [LDA ], \(\mathbb{P} \mathbb{I} O T,[W O R K],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO
    COMPLEX (8),D IM ENSION (:) ::W ORK
    COM PLEX (8),D IM ENSION (:,:) ::A
    \(\mathbb{N}\) TEGER ::N,LDA, \(\mathbb{N} F O\)
    \(\mathbb{N} T E G E R, D \mathbb{M} \operatorname{ENSION}(:):: \mathbb{P} \mathbb{V} O T\)
    SU BROUTINE SY TRI_64 (UPLO,N,A, [LDA ], \(\mathbb{P} \mathbb{I V O T},[\mathbb{W}\) ORK ], [ \(\mathbb{N F O}\) ])
    CHARACTER (LEN=1)::UPLO

COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathrm{LD} \mathrm{A}, \mathbb{N} F \mathrm{O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I V O T}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zsytri(charuplo, intn, doublecom plex *a, int lda, int *ipivot, int*info);
void zsytri_ 64 (charuplo, long n, doublecom plex *a, long lda, long *ịíivot, long *info);

\section*{PURPOSE}
zsytricom putes the inverse of a com plex sym m etric indefinte \(m\) atrix \(A\) using the factorization \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\) L *D *L**T com puted by CSY TRF .

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) ': U ppertriangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\);
= \(\mathrm{L}^{\prime}\) : L Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D}\) * \(\mathrm{L} * * \mathrm{~T}\).

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input/output)
O n entry, the block diagonalm atrix D and the \(m\) ultipliers used to obtain the factorU orL as com puted by CSY TRF .

On exit, if \(\mathbb{N F F O}=0\), the (sym metric) inverse of the original m atrix. If \(\mathrm{UPLO}=\mathrm{U}\) ', the upper triangularpart of the inverse is form ed and the partofA below the diagonal is not referenced; if UPLO = L' the low er triangular part of the inverse is formed and the partofA above the diagonal is not referenced.

LD A (input)
The leading din ension of the array A. LD A >= \(\max (1, N)\).
\(\mathbb{P I V O T}\) (input)
D etails of the interchanges and the block structure ofD as determ ined by CSY TRF.

W ORK (w orkspace)
dim ension ( 2 * N )
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i, D(i, i)=0\); the \(m\) atrix is singular and its inverse could notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zsytre - solve a system of linearequations \(A * X=B\) with a complex symmetric \(m\) atrix \(A\) using the factorization \(A=\) \(\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by CSY TRF

\section*{SYNOPSIS}

```

CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGERN,NRHS,LDA,LDB,INFO
INTEGER \mathbb{PIVOT (*)}
SU BROUT\mathbb{NE ZSYTRS_64(UPLO,N,NRHS,A,LDA,\mathbb{PIVOT,B,LD B, NNFO)}}\mathbf{N}\mathrm{ (N,N}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGER*8N,NRHS,LDA,LDB,INFO
INTEGER*8 \mathbb{PIVOT (*)}

```
F95 INTERFACE
    SU BROUTINE SYTRS (UPLO,N,NRHS,A, [LDA], \(\mathbb{P} \mathbb{I V O T}, \mathrm{B},[\mathrm{LD} B],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO
    COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : ) ::A, B
    \(\mathbb{N}\) TEGER ::N,NRHS,LDA,LDB, \(\mathbb{N} F O\)
    \(\mathbb{N} T E G E R, D \mathbb{I} \operatorname{ENSION(:)::\mathbb {P}\mathbb {O}OT}\)
    SU BROUTINE SYTRS_64 (UPLO ,N,NRHS,A, [LDA ], \(\mathbb{P} \mathbb{I V} O T, B,[L D B],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO
    COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : ) :: A, B
\(\mathbb{N}\) TEGER (8) :: N , NRHS,LDA,LDB, \(\mathbb{N}\) FO
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{P} \mathbb{I} O\) T

\section*{C INTERFACE}
\#include <sunperfh>
void zsytrs (charuplo, intn, int nrhs, doublecom plex *a, int lda, int *ipívot, doublecom plex *b, int lalb, int*info);
void zsytrs_64 (charuplo, long n, long nrhs, doublecom plex
*a, long lda, long *ípivot, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
zsytrs solves a system of linearequations \(A * X=B\) with a complex symm etric \(m\) atrix \(A\) using the factorization \(A=\) \(\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\) orA \(=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\) com puted by CSY TRF.

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the details of the factorization
are stored as an upper or low er triangularm atrix.
\(=\mathrm{U}\) ': Uppertriangular, form is \(\mathrm{A}=\mathrm{U} * \mathrm{D} * \mathrm{U} * * \mathrm{~T}\);
\(=\mathrm{L}^{\prime}\) : Low ertriangular, form is \(\mathrm{A}=\mathrm{L} * \mathrm{D} * \mathrm{~L} * * \mathrm{~T}\).

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the m atrix B. NRH S \(>=0\).

A (input) The block diagonalm atrix D and the multipliers used to obtain the factorU orL as com puted by CSYTRF.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).
\(\mathbb{P} \mathbb{I V O T}\) (input)
D etails of the interchanges and the block structure ofD as determ ined by CSY TRF.

B (input/output)

O \(n\) entry, the right hand side \(m\) atrix \(B\). On exit, the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the aray B. LD B >= \(\max (1, N)\).
\(\mathbb{I N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztocon - estim ate the reciprocal of the condition num ber of a triangular band \(m\) atrix \(A\), in etherthe 1 -norm orthe infinity-norm

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZTBCON NORM,UPLO,DIAG,N,KD,A,LDA,RCOND,WORK,}
W ORK2,\mathbb{NFO)}
CHARACTER * 1NORM,UPLO,DIAG
DOUBLE COM PLEX A (LDA,*),W ORK (*)
INTEGERN,KD,LDA, \mathbb{NFO}
DOUBLE PRECISION RCOND
DOUBLE PRECISION W ORK 2 (*)
SUBROUT\mathbb{NE ZTBCON_64NORM,UPLO,DIAG,N,KD,A,LDA,RCOND,WORK,}
W ORK2,\mathbb{NFO)}
CHARACTER * 1 NORM,UPLO,DIAG
DOUBLE COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGER*8N,KD,LDA,INFO}
DOUBLE PRECISION RCOND
DOUBLE PRECISION W ORK 2 (*)

```

\section*{F95 INTERFACE}
```

SU BROUTINE TBCON $\mathbb{N} O R M, U P L O, D \mathbb{I} G, N, K D, A,[L D A], R C O N D,[W O R K]$, [W ORK 2], [ $\mathbb{N F O}$ ])
CHARACTER (LEN=1) ::NORM,UPLO,D IAG
COM PLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D $\mathbb{M}$ ENSION (:,:) ::A
$\mathbb{N} T E G E R:: N, K D, L D A, \mathbb{N} F O$

```

REAL (8) :: RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ORK2
SUBROUTINE TBCON_64 \(\mathbb{N} O R M, U P L O, D I A G, N, K D, A,[L D A], R C O N D\), [W ORK], [W ORK2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::NORM,UPLO,DIAG
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8), D IM ENSION (:,:) ::A
\(\mathbb{N}\) TEGER (8) ::N,KD,LDA, \(\mathbb{N} F O\)
REAL (8) :: RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void ztbcon (charnorm , charuplo, chardiag, intn, int kd, doublecom plex *a, int lda, double *roond, int *info);
void ztbcon_64 (charnorm , charuplo, chardiag, long n, long kd, doublecom plex *a, long lda, double *roond, long *info);

\section*{PURPOSE}
ztbcon estim ates the reciprocal of the condition num ber of a triangular band matrix \(A\), in either the 1 -norm or the infinity-norm .

The norm ofA is com puted and an estim ate is obtained for norm (inv (A )), then the reciprocal of the condition num ber is com puted as
\(\operatorname{RCOND}=1 /(\operatorname{nom}(A) * \operatorname{nom}(\operatorname{inv}(A)))\).

\section*{ARGUMENTS}
```

NORM (input)
Specifies w hether the 1-norm condition num ber or
the infinity-norm condition num ber is required:
= '1'or O ': 1-nom;
= I': Infinity-norm.
UPLO (input)
= U ': A is uppertriangular;
= LL':A is low ertriangular.

```
D IA G (input)
\(=N: A\) is non-unit triangular;
\(=\mathrm{U}\) ': A is unit triangular.
N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

KD (input)
The num berof superdiagonals or subdiagonals of the triangularband \(m\) atrix \(A . K D>=0\).

A (input) The upper or low er triangular band \(m\) atrix \(A\), stored in the firstkd+1 row sof the array. The \(j\) th column ofA is stored in the \(j\) th column of the array A as follow s: if UPLO = U',A (kd+1+i\(j, 7)=A(i, j)\) for \(\max (1, j \mathrm{kd})<=i<=j\) if UPLO \(=\) \(\left.L^{\prime}, A(1+i-j\rangle\right)=A(i, \gamma)\) for \(j<=i<=m\) in \((n, j+k d)\). IfD IA G = U', the diagonalelem ents of A are not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the array A. LDA >= K D +1 .

RCOND (output)
The reciprocal of the condition num ber of the \(m\) atrix \(A\), computed as RCOND \(=1 /\) (norm (A) * norm (inv (A))).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{I N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztom \(v\)-perform one of the \(m\) atrix-vectoroperations \(x:=\) \(A{ }^{*} x\), or \(x: A{ }^{*} x\), or \(x:=\operatorname{con} g\left(A^{\prime}\right)^{*} x\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZTBMV (UPLO,TRANSA,D IAG,N,K,A,LDA,Y, INCY)}
CHARACTER * 1 UPLO,TRANSA,DIAG
DOUBLE COM PLEXA (LDA,*),Y(*)
INTEGER N,K,LDA,INCY
SUBROUT\mathbb{NE ZTBM V_64(UPLO,TRANSA,D IAG,N,K,A,LDA,Y, NNCY)}
CHARACTER * 1 UPLO,TRANSA,D IAG
DOUBLE COM PLEXA (LDA,*),Y (*)
\mathbb{NTEGER*8N,K,LDA,INCY}

```
F95 INTERFACE
    SU BROUTINE TBMV (UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{N}], K, A,[L D A], Y,[\mathbb{N C Y}])\)
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
    COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::Y
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N} T E G E R:: N, K, L D A, \mathbb{N} C Y\)
    SU BROUTINE TBM V_64 (UPLO, [TRANSA ],D \(\mathbb{A} G, \mathbb{N}], K, A,[L D A], Y\),
        [ \(\mathbb{N} C Y\) ])
    CHARACTER (LEN=1)::UPLO,TRANSA,DIAG
    COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::Y
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N} T E G E R(8):: N, K, L D A, \mathbb{N} C Y\)

\section*{C INTERFACE}
\#include <sunperfh>
void ztom v (charuplo, chartransa, chardiag, intn, int k, doublecom plex *a, int lda, doublecom plex *y, int incy);
void ztom v_64 (charuplo, chartransa, char diag, long n, long k, doublecom plex *a, long lda, doublecom plex *y, long incy);

\section*{PURPOSE}
\(z t b m \mathrm{v}\) perform s one of the \(m\) atrix-vectoroperations \(\mathrm{x}:=\mathrm{A} * \mathrm{x}\), or \(x:=A\) * \(x\), or \(x:=\) con \(\dot{g}\left(A^{\prime}\right)^{*} x\) where \(x\) is an \(n\) elem ent vectorand \(A\) is an \(n\) by \(n\) unit, ornon-unit, upperorlow er triangularband \(m\) atrix, \(w\) ith ( \(k+1\) ) diagonals.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whetherthem atrix is an upper or low er triangularm atrix as follow s:

UPLO = U'or \(\mathrm{u}^{\prime} \mathrm{A}\) is an upper triangular \(m\) atrix.

UPLO = L' or 1' A is a lower triangular \(m\) atrix.

U nchanged on exit.
TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) 'or \(h^{\prime} \mathrm{x}:=\mathrm{A}{ }^{*} \mathrm{x}\).
TRANSA \(=\) T'ort' \(x:=A * x\).
TRANSA \(=\) C'ort \(^{\prime} \mathrm{x}:=\operatorname{conjg}\left(\mathrm{A}^{\prime}\right){ }^{\star} \mathrm{x}\).
U nchanged on exit.
TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
O n entry, D IA G specifies w hether ornotA is unit
triangular as follow \(s\) :
D \(\mathbb{A} G=U\) 'or 4 ' \(A\) is assum ed to be unit triangular.
\(D \mathbb{A} G=N\) 'or \(h^{\prime} A\) is notassumed to be unit triangular.

U nchanged on exit.
N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix \(A\). \(\mathrm{N}>=0\). U nchanged on exit.
\(K\) (input)
O n entry w ith UPLO \(=\) U 'or L', K specifies the num ber of super-diagonals of them atrix \(A\). On entry \(w\) th UPLO = L' or I', K specifies the num ber of sub-diagonals of the \(m\) atrix \(A . K>=0\). U nchanged on exit.

A (input)
Before entry w ith UPLO = U 'or L ', the leading ( \(k+1\) ) by n part of the array A mustcontain the upper triangularband part of the \(m\) atrix of coefficients, supplied colum \(n\) by colum \(n\), w ith the leading diagonal of the \(m\) atrix in row ( \(k+1\) ) of the aray, the firstsuper-diagonalstarting at position 2 in row \(k\), and so on. The top leftk by \(k\) triangle of the array A is not referenced. The follow ing program segm entw illtransfer an upper triangular band \(m\) atrix from conventional full \(m\) atrix storage to band storage:
\[
\begin{aligned}
& \text { DO } 20, J=1, N \\
& \text { M }=\mathrm{K}+1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{M} A X(1, \mathrm{~J}-\mathrm{K}), \mathrm{J} \\
& \quad \text { A }(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \text { CONTINE } \\
& 20 \text { CONTINUE }
\end{aligned}
\]

Before entry w ith UPLO = L 'or 1', the leading ( \(\mathrm{k}+1\) ) by n partof the array A m ust contain the low er triangularband part of the \(m\) atrix of coefficients, supplied colum \(n\) by colum n, w th the leading diagonal of the \(m\) atrix in row 1 of the array, the first sub-diagonal starting atposition 1 in row 2 , and so on. The bottom right \(k\) by \(k\) triangle of the array \(A\) is not referenced. The
follow ing program segm entw ill transfer a low er triangular band \(m\) atrix from conventional full
\(m\) atrix storage to band storage:

DO \(20, J=1, N\)
\(M=1-J\)
DO \(10, I=J, M \mathbb{N}(N, J+K)\)
\(A(M+I, J)=m \operatorname{atrix}(I, J)\)
10 CONTINUE
20 CONTINUE
\(N\) ote thatw hen DIA G = U 'or 4 'the elem ents of the array A corresponding to the diagonalelem ents of the \(m\) atrix are not referenced, butare assum ed to be unity. U nchanged on exit.
LD A (input)
O n entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD A \(>=\) ( \(\mathrm{k}+1\) ). U nchanged on exit.

Y (input/output)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) m ust contain the \(n\) elem ent vectorx. On exit, \(Y\) is overw ritten \(w\) th the tranform ed vector \(x\).
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{I N C Y}\) specifies the increm ent for the elem ents of \(. \mathbb{N} C Y<>0\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztbrfs - provide emrorbounds and backw ard emor estim ates for the solution to a system of linear equations w ith a triangularband coefficientm atrix

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZTBRFS (UPLO,TRANSA,D IAG,N,KD,NRHS,A,LDA,B,LDB,}
X,LDX,FERR,BERR,W ORK,W ORK2,\mathbb{NFO)}
CHARACTER * 1UPLO,TRANSA,DIAG
DOUBLE COM PLEX A (LDA,*),B (LDB,*),X (LDX ,*),W ORK (*)
INTEGERN,KD,NRHS,LDA,LDB,LDX, INFO
D OUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)
SUBROUT\mathbb{NE ZTBRFS_64 (UPLO,TRANSA,DIAG,N,KD,NRHS,A,LDA,B,}
LDB,X,LDX,FERR,BERR,W ORK,W ORK2, INFO)
CHARACTER * 1UPLO,TRANSA,DIAG
DOUBLE COM PLEX A (LDA,*),B (LDB,*),X (LDX,*),W ORK (*)
NNTEGER*8N,KD,NRHS,LDA,LDB,LDX, IN FO
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE TBRFS (UPLO, [TRANSA],D IAG,N,KD,NRHS,A, [LDA],B, [LDB], X, [LDX ],FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8), D IM ENSION (:,:) ::A, B, X
\(\mathbb{N} T E G E R:: N, K D, N R H S, L D A, L D B, L D X, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

SU BROUTINE TBRFS_64 (UPLO, [TRANSA],D \(\mathbb{A} G, N, K D, N R H S, A,[L D A]\),


CHARACTER (LEN=1) ::UPLO, TRANSA,D IA G
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A, B,X
\(\mathbb{N} T E G E R(8):: N, K D, N R H S, L D A, L D B, L D X, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK2

\section*{C INTERFACE}
\#include < sunperfh>
void ztbrfs (char uplo, char transa, chardiag, int n, int kd, int nrhs, doublecom plex *a, intlda, doublecom plex *b, int 1 db , doublecom plex *x, int ldx, double * ferrs, double *berr, int *info);
void ztorfs_64 (charuplo, chartransa, char diag, long n, long kd, long nrhs, doublecom plex *a, long lda, doublecom plex *b, long ldb, doublecom plex *x, long ldx, double *ferr, double *berr, long *info);

\section*{PURPOSE}
ztbrfs provides errorbounds and backw ard error estim ates forthe solution to a system of linear equations \(w\) th a triangularband coefficientm atrix.

The solution \(m\) atrix \(X\) m ustibe com puted by CTBTRS or some other \(m\) eans before entering this routine. CTBRFS does not do iterative refinem entbecause doing so cannotim prove the backw ard error.

\section*{ARGUMENTS}

\section*{UPLO (input)}
= U : A is uppertriangular;
= L ': A is low ertriangular.

TRANSA (input)
Specifies the form of the system of equations:
\(=\mathrm{N}: \mathrm{A} * \mathrm{X}=\mathrm{B} \quad\) N 0 transpose)
\(=T: A * * T * X=B \quad(T\) ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose)
TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
\(=\mathrm{N}\) : A is non-unit triangular;
\(=\mathrm{U}:\) : A is unit triangular.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
\(K D\) (input)
The num berof superdiagonals or subdiagonals of the triangularband \(m\) atrix \(A . K D>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of collm ns of the m atrices B and X. NRHS \(>=0\).
A (input) The upper or low er triangular band \(m\) atrix \(A\), stored in the firstkd+1 row sof the amay. The \(j\) th column ofA is stored in the \(j\) th column of the anay A as follow s: if UPLO = U',A (kd+1+i\(j, 7)=A(i, j)\) for \(\max (1, j \mathrm{kd})<=i<=j\) if UPLO \(=\) L', A \((1+i-j\rangle)=A(i, 7)\) for \(j<=i<=m\) in \((n, j+k d)\). IfD \(\mathbb{I A}=U\) ', the diagonalelem ents of A are not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the anay A. LDA >= K D +1.
\(B\) (input) The righthand side m atrix \(B\).

LD B (input)
The leading dim ension of the array B . LD B >= \(\max (1, N)\).
\(X\) (input) The solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the anay X . LD X >= \(\max (1, N)\).

FERR (output)
The estim ated forw ard errorbound for each solution vector \(X()\) (the \(j\) th column of the solution \(m\) atrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{H}), \operatorname{FERR}(1)\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(X(\mathcal{D})-X\) TRU \(E\) ) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{J})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true error.

BERR (output)

The com ponentw ise relative backw ard error of each solution vector \(X(\mathcal{)}\) (ie., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{(}\) ) an exactsolution).

W ORK (w orkspace) dim ension \(\left(2{ }^{*} \mathrm{~N}\right.\) )
W ORK 2 (w orkspace) dim ension (N)
\(\mathbb{N F O}\) (output)
= 0 : successfiulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztbsv - solve one of the system sofequations \(A * x=b\), or
\(A^{*} x=b, \operatorname{orcong}\left(A^{\prime}\right){ }^{*} x=b\)

\section*{SYNOPSIS}

SUBROUTINE ZTBSV (UPLO,TRANSA,D \(\mathbb{I A} G, N, K, A, L D A, Y, \mathbb{N} C Y)\)
CHARACTER * 1 UPLO, TRANSA, DIAG
DOUBLE COM PLEXA (LDA,*), Y (*)
\(\mathbb{N}\) TEGER \(N, K, L D A, \mathbb{N} C Y\)
SUBROUTINE ZTBSV_64(UPLO,TRANSA,D \(\mathbb{A} G, N, K, A, L D A, Y, \mathbb{N} C Y)\)

CHARACTER * 1 UPLO,TRANSA, D IAG
DOUBLE COM PLEXA (LDA, \(\left.{ }^{\star}\right)\), Y ( \({ }^{\star}\) )
\(\mathbb{N}\) TEGER* \(8 \mathrm{~N}, \mathrm{~K}, \mathrm{LD} A, \mathbb{N} C Y\)

\section*{F95 INTERFACE}

SU BROUTINE TBSV (UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{N}], K, A,[L D A], Y,[\mathbb{N C Y}])\)
CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) :: Y
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: N, K, L D A, \mathbb{N} C Y\)
SU BROUTINE TBSV_64 (UPLO, [TRANSA],D \(\mathbb{I} G, \mathbb{N}], K, A,[L D A], Y\), [ \(\mathbb{N} C Y\) ])

CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::Y
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R(8):: N, K, L D A, \mathbb{N} C Y\)

\section*{C INTERFACE}
\#include <sunperfh>
void ztosv (char uplo, chartransa, chardiag, intn, int k, doublecom plex *a, int lda, doublecom plex *y, int incy);
void ztosv_64 (charuplo, chartransa, char diag, long n, long \(k\), doublecom plex *a, long lda, doublecom plex *y, long incy);

\section*{PURPOSE}
ztbsv solves one of the system sofequations \(A * x=b\), or \(A^{*} x=b\), orconjg ( \(A^{\prime}\) ) * \(x=b\) where \(b\) and \(x\) are \(n\) elem ent vectors and \(A\) is an \(n\) by \(n\) unit, ornon-unit, upper or low er triangularband \(m\) atrix, \(w\) ith ( \(k+1\) ) diagonals.

No test forsingularity ornear-singularity is included in this routine. Such testsm ustbe penform ed before calling this routine.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies w hether the \(m\) atrix is an upper or low er triangularm atrix as follow s:

UPLO = U'or \(\mathrm{L'}^{\prime} \mathrm{A}\) is an upper triangular \(m\) atrix.

UPLO = L' or \(\mathrm{I}^{\prime}\) A is a lower triangular \(m\) atrix.

U nchanged on exit.
TRANSA (input)
On entry, TRANSA specifies the equations to be solved as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} A * x=b\).

TRANSA \(=\mathrm{T}^{\prime}\) or \(\mathrm{t}^{\prime} \mathrm{A}{ }^{*} \mathrm{x}=\mathrm{b}\).
TRANSA \(=C^{\prime}\) ort' cong \(\left(A^{\prime}\right) \star \mathrm{x}=\mathrm{b}\).

U nchanged on exit.

TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
O n entry, D IA G specifies w hether ornot A is unit triangular as follow s:

D \(\mathbb{I}\) G \(=U\) 'or 4 ' \(A\) is assum ed to be unit triangular.

D \(\mathbb{A} G=N\) 'or \(h\) ' \(A\) is notassumed to be unit triangular.

U nchanged on exit.
N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix \(A\). \(\mathrm{N}>=0\). U nchanged on exit.

K (input)
On entry with UPLO \(=U\) 'or U ', \(K\) specifies the num ber of super-diagonals of them atrix \(A\). On entry w th UPLO = L' or I', K specifies the num ber of sub-diagonals of the \(m\) atrix \(A . K>=0\). U nchanged on exit.

A (input)
Before entry w ith UPLO = U 'or L ', the leading ( \(k+1\) ) by \(n\) part of the array A m ust contain the upper triangularband part of the \(m\) atrix of coefficients, supplied colum \(n\) by colum \(n\), w ith the leading diagonal of the \(m\) atrix in row ( \(k+1\) ) of the aray, the firstsuper-diagonalstarting at position 2 in row \(k\), and so on. The top leftk by \(k\) triangle of the array A is not referenced. The follow ing program segm entw illtransfer an upper triangular band \(m\) atrix from conventional full m atrix storage to band storage:
\[
\begin{aligned}
& \text { DO } 20, J=1, N \\
& \mathrm{M}=\mathrm{K}+1-\mathrm{J} \\
& \mathrm{DO} 10, \mathrm{I}=\mathrm{M} A X(1, \mathrm{~J}-\mathrm{K}), \mathrm{J} \\
& \quad \text { A }(\mathrm{M}+\mathrm{I}, \mathrm{~J})=\mathrm{m} \operatorname{atrix}(\mathrm{I}, \mathrm{~J}) \\
& 10 \text { CONTINUE } \\
& 20 \mathrm{CONTINUE}
\end{aligned}
\]

Before entry w ith UPLO = L 'or 1', the leading ( \(k+1\) ) by \(n\) part of the array A m ust contain the low er triangularband part of the \(m\) atrix of coefficients, supplied column by colum n, w th the
leading diagonal of the \(m\) atrix in row 1 of the array, the first sub-diagonal starting atposition
1 in row 2 , and so on. The bottom right \(k\) by \(k\) triangle of the array \(A\) is not referenced. The follow ing program segm entw ill transfer a low er triangular band \(m\) atrix from conventional full \(m\) atrix storage to band storage:
\[
\text { D O } 20, \mathrm{~J}=1, \mathrm{~N}
\]
\(\mathrm{M}=1-\mathrm{J}\)
DO \(10, \mathrm{I}=\mathrm{J}, \mathrm{M} \mathbb{N}(\mathrm{N}, \mathrm{J}+\mathrm{K})\)
A \((M+I, J)=m \operatorname{atrix}(I, J)\)
10 CONTINUE
20 CONTINUE
\(N\) ote thatwhen D \(\mathbb{A} G=U\) 'or 4 'the elem ents of the array A corresponding to the diagonalelem ents of the \(m\) atrix are not referenced, but are assum ed to be unity. U nchanged on exit.

LDA (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program .LD A >=( \(\mathrm{k}+1\) ). U nchanged on exit.

Y (input/output)
\((1+(n-1) \star \operatorname{abs}(\mathbb{N} C Y))\). Before entry, the increm ented aray \(Y\) must contain the \(n\) elem ent righthand side vectorb. On exit, \(Y\) is overw ritten \(w\) th the solution vectorx.
\(\mathbb{N} C Y\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y\). \(\mathbb{N C Y}\) <> 0. U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztbtrs - solve a triangularsystem of the form \(A * X=B\), \(\mathrm{A} * * \mathrm{~T} * \mathrm{X}=\mathrm{B}\),orA \({ }^{*}{ }^{*} \mathrm{H} * \mathrm{X}=\mathrm{B}\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZTBTRS (UPLO,TRANSA,DIAG,N,KD,NRHS,A,LDA,B,LDB,}
\mathbb{NFO)}
CHARACTER * 1 UPLO,TRANSA,D IAG
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
\mathbb{NTEGER N,KD,NRHS,LDA,LDB,INFO}
SUBROUT\mathbb{NE ZTBTRS_64 (UPLO,TRANSA,DIAG,N,KD,NRHS,A,LDA,B,}
LDB,INFO)
CHARACTER * 1 UPLO,TRANSA,DIAG
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
\mathbb{NTEGER*8N,KD,NRHS,LDA,LDB,INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE TBTRS (UPLO, TRANSA, D IA G ,N,KD,NRHS,A, [LDA],B, [LDB], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
COM PLEX (8), D IM ENSION (:,:) ::A, B
\(\mathbb{N}\) TEGER :: N, KD, NRHS,LDA,LDB, \(\mathbb{N} F O\)
SUBROUTINE TBTRS_64 (UPLO, TRANSA, D \(\mathbb{A} G, N, K D, N R H S, A,[L D A], B\), [LDB], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : ) :: A, B
\(\mathbb{N}\) TEGER (8) ::N,KD,NRHS,LDA, LD B, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void ztotrs (charuplo, chartransa, chardiag, int n, int kd, int nihs, doublecom plex *a, int lda, doublecom plex *b, int ldb, int *info);
void ztbtrs_64 (charuplo, chartransa, char diag, long n, long kd, long nrhs, doublecom plex *a, long lda, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
ztbtres solves a triangular system of the form
\(w\) here \(A\) is a triangularband \(m\) atrix of order \(N\), and \(B\) is an N -by-NRHS matrix. A check ism ade to verify thatA is nonsingular.

\section*{ARGUMENTS}

UPLO (input)
= U ': A is upper triangular;
\(=\mathrm{L}\) : A is low ertriangular.

TRANSA (input)
Specifies the form of the system of equations:
\(=N\) ': A * \(\mathrm{X}=\mathrm{B} \quad\) N o transpose)
\(=T: A * * T * X=B \quad(T\) ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose)

D IA G (input)
= \(\mathrm{N}^{\prime}: \mathrm{A}\) is non-unit triangular;
\(=\mathrm{U}\) : A is unit triangular.

N (input) The order of the matrix A. \(\mathrm{N}>=0\).
KD (input)
The num berof superdiagonals or subdiagonals of the triangularband \(m\) atrix A. KD \(>=0\).

NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS \(>=0\).

A (input) The upper or low er triangular band \(m\) atrix A, stored in the first kd+1 row sofA. The jth column ofA is stored in the \(j\) th column of the array A as follow s: if UPLO = U',A (kd+1+i-j)
\(=A(i, j)\) form ax \((1, j \mathrm{jkd})<=\mathrm{i}<=\dot{j}\) if UPLO \(=\mathrm{L}^{\prime}\), A \((1+i-j-j)=A(i, j)\) for \(j=i<=m\) in \((n, j+k d)\). If
\(D \mathbb{I A}=U\) ', the diagonalelem ents of \(A\) are not referenced and are assum ed to be 1 .

LDA (input)
The leading dim ension of the aray A. LDA >= K D +1 .

B (input/output)
On entry, the righthand side m atrix B. On exit, if \(\mathbb{N} F O=0\), the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array \(B\). LD B >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the \(i\)-th diagonalelem ent of \(A\)
is zero, indicating that the \(m\) atrix is singular and the solutions X have notbeen com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztgevc - com pute som e orall of the rightand/or left generalized eigenvectors of a pair of com plex upper triangular \(m\) atrices \((A, B)\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZTGEVC (S\mathbb{DE,HOW MNY,SELECT,N,A,LDA,B,LDB,VL,LDVL,}}\mathbf{N},\textrm{L},\textrm{L}
VR,LDVR,MM,M,W ORK,RW ORK,INFO)
CHARACTER * 1SDE,HOW M NY
D OUBLE COM PLEX A (LDA,*),B (LDB,*), VL (LDVL,*), VR (LDVR,*),
W ORK (*)
\mathbb{NTEGER N,LDA,LDB,LDVL,LDVR,M M ,M , INFO}
LOG ICAL SELECT (*)
DOUBLE PRECISION RW ORK (*)
SUBROUT\mathbb{NE ZTGEVC_64 (SDE,HOW MNY,SELECT,N,A,LDA,B,LDB,VL,}
LDVL,VR,LDVR,MM,M,W ORK,RW ORK,INFO)
CHARACTER * 1SDE,HOW MNY
DOUBLE COM PLEX A (LDA,*),B (LDB,*), VL (LDVL,*), VR (LDVR,*),
W ORK (*)
\mathbb{N TEGER*8 N,LDA,LD B,LDVL,LDVR,M M ,M , INFO}
LOG ICAL*8 SELECT (*)
DOUBLE PRECISION RW ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE TGEVC (SDE,HOW MNY,SELECT, \(\mathbb{N}], A,[L D A], B,[L D B], V L\), [LDVL],VR, [LDVR],MM,M, \(\mathbb{W}\) ORK], \(\mathbb{R W} O R K],[\mathbb{N} F O])\)

CHARACTER (LEN=1)::SDE,HOW MNY
COM PLEX (8),D IM ENSION (:) ::W ORK

COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, B, VL,VR
\(\mathbb{N}\) TEGER :: N,LDA,LDB,LDVL,LDVR,MM,M, \(\mathbb{N} F O\)
LOG ICAL,D IM ENSION (:) ::SELECT
REAL (8),D \(\mathbb{M}\) ENSION (:) ::RW ORK

SU BROUTINE TGEVC_64 (SDE,HOW M NY, SELECT, \(\mathbb{N}], A,[L D A], B,[L D B]\), VL, [LDVL],VR, [LDVR],MM,M, \(\left[\begin{array}{l}\text { ORK }],[R W O R K],[\mathbb{N} F O])\end{array}\right.\)

CHARACTER (LEN=1) ::SDE,HOW M NY
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::A,B,VL,VR
\(\mathbb{N}\) TEGER (8) :: N ,LDA,LDB,LDVL,LDVR,MM,M, \(\mathbb{N}\) FO
LOG ICAL (8),D IM ENSION (:) ::SELECT
REAL (8), D IM ENSION (:) ::RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void ztgevc (char side, char how my, int *select, int n, doublecom plex *a, int lda, doublecom plex *b, int ldb, doublecom plex *vl, int ldvl, doublecom plex \({ }^{*}\) vr, int ldvr, intm \(m\), int \({ }^{m}\), int *info);
void ztgevc_64 (char side, charhow m ny, long *select, long n, doublecom plex *a, long lda, doublecom plex *b, long ldb, doublecom plex *vl, long ldvl, doublecom plex *vr, long ldvr, long m m, long *m , long *info);

\section*{PURPOSE}
ztgevc com putes som e orallof the right and/or left generalized eigenvectors of a pair of com plex upper triangular \(m\) atrices \((A, B)\).

The right generalized eigenvectorx and the leftgeneralized eigenvectory of \((A, B)\) corresponding to a generalized eigenvalue w are defined by:
\[
(A-w B) * x=0 \text { and } y^{\star *} H *(A-w B)=0
\]
\(w\) here \(\mathrm{y}^{* *}{ }_{\mathrm{H}}\) denotes the conjugate tranpose of y .

If an eigenvalue w is determ ined by zero diagonal elem ents of both \(A\) and \(B\), a unit vector is retumed as the comesponding eigenvector.

If alleigenvectors are requested, the routine \(m\) ay either retum the \(m\) atrices \(X\) and/or \(Y\) of rightor lefteigenvectors of \((A, B)\), or the products \(Z * X\) and/or \(Q * Y\), where \(Z\) and \(Q\) are input unitary \(m\) atrioes. If \((A, B) w\) as obtained from the gen-
eralized Schur factorization of an original pair ofm atrices
\((A 0, B 0)=Q * A * Z * * H, Q * B * Z * * H)\), then \(Z * X\) and \(Q * Y\) are the \(m\) atrioes of right or lefteigenvectors of A.

\section*{ARGUMENTS}

\section*{STDE (input)}
\(=\mathrm{R}\) ': com pute righteigenvectors only;
\(=\mathrm{L}\) ': com pute lefteigenvectors only;
\(=\mathrm{B}\) ': com pute both right and lefteigenvectors.
HOW MNY (input)
= A ': com pute all right and/or lefteigenvectors;
= B':com pute all right and/or left eigenvectors, and backtransform them using the inputm atrices supplied in VR and/orV L; = 5 : com pute selected right and/or lefteigenvectors, specified by the logicalaray SELECT.

\section*{SELECT (input)}

If HOW M NY = S', SELECT specifies the eigenvectors to be computed. If HOW MNY=A 'or B', SELECT is not referenced. To select the eigenvector comesponding to the jth eigenvalue, SELECT ( \(\ddagger\) m ust.be set to .TRUE..

N (input) The order of the m atrioes A and B. \(\mathrm{N}>=0\).

A (input) The uppertriangularm atrix A .
LD A (input)
The leading dimension of array A. LDA >= \(\max (1, \mathbb{N})\).

B (input) The upper triangularm atrix B. B m ust have real diagonalelem ents.

LD B (input)
The leading dimension of array B. LD B >= \(\max (1, \mathbb{N})\).

VL (input/output)
On entry, if \(S D E=L\) 'or \(B\) 'and HOW M NY = B', VL must contain an \(N\)-by N m atrix \(Q\) (usually the unitary \(m\) atrix \(Q\) of left Schurvectors retumed by CHGEQZ). On exit, if \(S \mathbb{D} E=\mathrm{L}\) 'or \(\mathrm{B}^{\prime}, \mathrm{VL}\) contains: if HOW MNY = A', the matrix Y of left
eigenvectors of ( \(A, B\) ); if HOW M NY = \(B\) ', the matrix Q *Y ; ifHOW M NY = \(S^{\prime}\), the left eigenvectors of ( \(A, B\) ) specified by SELECT, stored consecutively in the colum ns of V , in the same order as their eigenvalues. If \(S \mathbb{D} E=R \prime, V L\) is not referenced.

LDVL (input)
The leading dimension of array VL . LDVL >= \(\max (1, N)\) if \(S \mathbb{D} E=L\) 'or \(B^{\prime} ; L D V L>=1\) otherw ise.
VR (input/output)
On entry, ifS \(\mathrm{D}=\mathrm{R}\) 'or \(\mathrm{B}^{\prime}\) 'and HOW MNY = \(\mathrm{B}^{\prime}\), VR m ust contain an N -by \(-\mathrm{N} m\) atrix Q (usually the unitary \(m\) atrix \(Z\) of rightSchur vectors retumed by CHGEQZ). On exit, if \(S \mathbb{D} E=R\) 'or \(B\) ', VR contains: if H OW M NY = A', them atrix X of right eigenvectors of \((A, B)\); if HOW MNY = \(B\) ', the matrix \(Z * X\); if HOW M NY \(=S\) ', the right eigenvectors of ( \(A\), B ) specified by SELEC T, stored consecutively in the colum ns ofVR, in the same order as their eigenvalues. If \(S \mathbb{D} E=L\) ', VR is not referenced.

LDVR (input)
The leading dim ension of the array VR . LDVR >= \(\max (1, N)\) if \(S \mathbb{D} E=R\) 'or \(B^{\prime} ; L D V R>=1\) other wise.

M M (input)
The num ber of colum ns in the arays V L and/or VR. \(M \mathrm{M}>=\mathrm{M}\).

M (output)
The num berof colum ns in the amays \(V \mathrm{~L}\) and/or VR actually used to store the eigenvectors. If
HOW M NY = A 'or B', M is set to N. Each selected eigenvector occupies one colum \(n\).

W ORK (w orkspace)
dim ension ( \(2 * N\) )

RW ORK (w orkspace)
dim ension ( \(2 * N\) )
\(\mathbb{I N F O}\) (output)
= 0: successfulexit.
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ztgexc - reorder the generalized Schur decom position of a com plex matrix pair ( \(A, B\) ), using an unitary equivalence transform ation \((A, B):=Q\) * \((A, B)\) * \(Z\) ', so that the diagonalblock of \((A, B)\) w ith row index \(\mathbb{F S T}\) ism oved to row \(\mathbb{L} S T\)

\section*{SYNOPSIS}
```

SUBROUTINE ZTGEXC (N ANTQ,W ANTZ,N,A,LDA,B,LDB,Q,LDQ,Z,LD Z,
FST,|ST,\mathbb{NFO)}
DOUBLE COM PLEX A (LDA,*),B (LDB,*),Q (LDQ ,*), Z (LD Z,*)
INTEGER N,LDA,LD B,LDQ,LD Z, IFST, LST, INFO
LOGICALWANTQ,W ANTZ
SU BROUTINE ZTGEXC_64(N ANTQ,W ANTZ,N,A,LDA,B,LDB,Q,LDQ,Z,LD Z,
\mathbb{FST},|ST,\mathbb{NFO)}

```
D OUBLE COM PLEX A (LDA,*), B (LDB,*), Q (LDQ , *), Z (LD Z,*)
\(\mathbb{N}\) TEGER* \(8 \mathrm{~N}, \mathrm{LD} A, L D B, L D Q, L D Z, \mathbb{F S T}, \mathbb{L S T}, \mathbb{N} F O\)
LOG ICAL*8W ANTQ,W ANTZ

\section*{F95 INTERFACE}

SU BROUTINE TGEXC \(\mathbb{N}\) ANTQ,W ANTZ, \(\mathbb{N}], A,[L D A], B,[L D B], Q,[L D Q], Z\), \([\mathrm{LD} Z], \mathbb{F S T}, \mathbb{L} \mathrm{ST},[\mathbb{N F O}])\)

COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : :: A, \(\mathrm{B}, \mathrm{Q}, \mathrm{Z}\)
\(\mathbb{N} T E G E R:: N, L D A, L D B, L D Q, L D Z, \mathbb{F S T}, \mathbb{L} T, \mathbb{N} F O\)
LOGICAL ::WANTQ,WANTZ
SU BROUTINE TGEXC_64 (NANTQ,W ANTZ, \(\mathbb{N}], A,[L D A], B,[L D B], Q,[L D Q]\), Z, [LD Z], \(\mathbb{F S T}, \mathbb{L} S T,[\mathbb{N F O}])\)

COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A, B, Q , Z
\(\mathbb{N} \operatorname{TEGER}(8):: N, L D A, L D B, L D Q, L D Z, \mathbb{F} S T, \Pi S T, \mathbb{N} F O\) LOGICAL (8) ::W ANTQ,W ANTZ

\section*{C INTERFACE}
\#include <sunperfh>
void ztgexc (intw antq, intw antz, intn, doublecom plex *a, int lda, doublecom plex *b, int ldlo, doublecom plex * \(q\), int ldq, doublecom plex * \(z\), int ldz, int *ifst, int*ilst, int*info);
void ztgexc_64 (long w antar, long w antz, long n, doublecom plex
*a, long lda, doublecom plex *b, long ldlb, doublecom plex *q, long ldq, doublecom plex *z, long ldz, long *ifst, long *ilst, long *info);

\section*{PURPOSE}
ztgexc reorders the generalized Schur decom position of a complex matrix pair ( \(A, B\) ), using an unitary equivalence transform ation \((A, B):=Q *(A, B) * Z\) ', so that the diagonal block of \((A, B)\) w th row index \(\mathbb{F} S T\) ism oved to row \#ST.
( \(\mathrm{A}, \mathrm{B}\) ) m ustbe in generalized Schur canonical form, that is, \(A\) and \(B\) are both upper triangular.

Optionally, the m atriges Q and Z of generalized Schur vectors are updated.

Q (in) * A (in) * Z (in) \({ }^{\prime}=\mathrm{Q}\) (out) * A (out) * Z (out)'
Q (in) * B (in) \(\mathrm{I}_{\mathrm{Z}}\) (in) \({ }^{\prime}=\mathrm{Q}\) (out) * B (out) \(* \mathrm{Z}\) (out) \({ }^{\prime}\)

\section*{ARGUMENTS}

W ANTQ (input)

W ANTZ (input)

N (input) The order of the \(m\) atriges \(A\) and \(B . N>=0\).

A (input/output)
On entry, the uppertriangular \(m\) atrix \(A\) in the pair ( \(A, B\) ). On exit, the updated \(m\) atrix \(A\).

LD A (input)
The leading dim ension of the aray A. LD A >= \(\max (1, N)\).

B (input/output)
O \(n\) entry, the upper triangular \(m\) atrix \(B\) in the pair ( \(A, B\) ). On exit, the updated \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).

Q (input/output)
On entry, ifW ANTQ = TRUE , the unitary \(m\) atrix \(Q\).
On exit, the updated matrix \(Q\). If \(W\) ANTQ \(=\) FALSE., Q is not referenced.

LD Q (input)
The leading dim ension of the array \(Q\). LD \(Q>=1\); If \(\mathrm{W} A N T Q=. \operatorname{RUE}, \operatorname{LDQ}>=\mathrm{N}\).

Z (input/output)
On entry, ifW ANTZ = TRUE., the unitary \(m\) atrix \(Z\).
On exit, the updated matrix \(Z\). If \(W\) ANTZ \(=\) FALSE., Z is not referenced.

LD Z (input)
The leading dim ension of the array Z . LD \(\mathrm{Z}>=1\); If
W ANTZ = .TRUE.,LD Z >= N.
FST (input/output)
Specify the reordering of the diagonal blocks of ( \(A, B\) ). The block w th row index \(\mathbb{F} S T\) ism oved to row ILST, by a sequence of Sw apping betw een adjcentblocks.

ISST (input/output)
See the description of IFST .
\(\mathbb{N} F \mathrm{O}\) (output)
=0: Successfulexit.
<0: if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvahue.
=1: The transform ed \(m\) atrix pair ( \(A, B\) ) w ould be too far from generalized Schur form ; the problem is ill-conditioned. (A, B) m ay have been partially reordered, and ILST points to the first row of the current position of the block being \(m\) oved.

\section*{FURTHER DETAILS}

B ased on contributions by
Bo K agstrom and PeterPorom aa, D epartm ent of Com puting Science,
Um ea U niversity, S-901 87 Um ea, Sw eden.
[1] B . K agstrom ; A D irectM ethod forReordering Eigenvalues in the G eneralized RealSchurForm of a RegularM atrix Pair (A, B), in

M S.M oonen etal (eds), LinearA lgebra for Large Scale and

Real-T in e A pplications, K luw erA cadem ic Publ. 1993, pp 195-218.
[2] B . K agstrom and P.Porom aa; Com puting Eigenspaces w ith Specified

Eigenvalues of a RegularM atrix Pair (A, B) and Condition

Estim ation: Theory, A lgorithm s and Softw are, Report
UM \(\mathbb{N} F-94.04\), Departm ent of Com puting Science, Umea U níversity,

S-901 87 Um ea, Sw eden, 1994.A lso as LA PA CK W orking N ote 87.

To appear in N um erical A lgorithm s, 1996.
[3] B . K agstrom and P. Porom aa, LA PA C K -Style A lgorithm s and Softw are
for Solving the Generalized Sylvester Equation and Estim ating the

Separation betw een RegularM atrix Pairs, ReportUM IN F 9323,

D epartm entofC om puting Science, Um ea U niversity, S-901 87 Um ea,

Sw eden, Decem ber 1993, Revised A pril 1994, A lso as LAPACK w orking

N ote 75.To appear in A CM Trans. on \(M\) ath. Softw are, \(V\) ol \(22, N \circ 1\),
1996.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ztgsen - reorder the generalized Schur decom position of a complex matrix pair ( \(A, B\) ) (in term \(s\) of an unitary equivalence trans-form ation \(Q\) '* \((A, B)\) * \(Z\) ), so that a selected cluster of eigenvalues appears in the leading diagonalblocks of the pair ( \(A, B\) )

\section*{SYNOPSIS}

SU BROUTINE ZTGSEN (LOB B,W ANTQ,W ANTZ,SELECT,N,A,LDA,B,LDB, ALPHA, BETA, \(Q, L D Q, Z, L D Z, M, P L, P R, D \mathbb{F}, W\) ORK,LW ORK, \(\mathbb{I N} O R K\), LIN ORK, \(\mathbb{N} F O\) )

DOUBLE COMPLEX A (LDA, *), B (LDB, \(\left.{ }^{\star}\right)\), ALPHA (*), BETA (*),

\(\mathbb{N}\) TEGER IJOB,N,LDA,LDB,LDQ,LDZ,M,LWORK,LINORK, \(\mathbb{N} F O\) \(\mathbb{N} T E G E R \mathbb{I N}\) ORK (*)
LOGICALWANTQ,WANTZ
LOG ICAL SELECT (*)
DOUBLE PRECISION PL, PR
DOUBLE PRECISION D \(\mathbb{F}\) (*)
SU BROUTINE ZTGSEN_64 (LDB B, W ANTQ,W ANTZ,SELECT,N,A,LDA,B,LDB, ALPHA,BETA, \(Q, L D Q, Z, L D Z, M, P L, P R, D \mathbb{F}, W\) ORK,LW ORK, \(\mathbb{I W} O R K\), \(\left.\mathrm{L} \mathbb{I} \mathrm{N}^{\circ} \mathrm{OR}, \mathbb{N} F \mathrm{O}\right)\)

DOUBLE COMPLEX A (LDA, *), B (LDB,*), ALPHA (*), BETA (*), Q (LDQ, \(\left.{ }^{\star}\right), \mathrm{Z}(\mathrm{LD} Z, \star), \mathrm{W} O R K(*)\)
\(\mathbb{N}\) TEGER*8 IJOB,N,LDA,LDB,LDQ,LDZ, M, LW ORK, LIN ORK, \(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER*8 \(\mathbb{I N}\) ORK (*)
LOGICAL*8W ANTQ, W ANTZ

\section*{F95 INTERFACE}

SU BROUTINE TGSEN (LOB B , W ANTQ,W ANTZ, SELECT,N,A, [LDA], B, [LDB], \(A L P H A, B E T A, Q,[L D Q], Z,[L D Z], M, P L, P R, D \mathbb{E},[W O R K],[L W O R K]\), \([\mathbb{I W}\) ORK], [LIW ORK], [ \(\mathbb{N} F O])\)

COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::ALPHA,BETA,W ORK
COM PLEX (8), D \(\mathbb{M} E N S I O N(:,:):: A, B, Q, Z\)
\(\mathbb{N}\) TEGER : \(: \operatorname{IJOB}, N, L D A, L D B, L D Q, L D Z, M, L W O R K, L \mathbb{N} O R K\),
\(\mathbb{N}\) FO
\(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{I W}\) ORK
LOGICAL ::W ANTQ, W ANTZ
LOGICAL,D \(\mathbb{I M} E N S I O N(:):: S E L E C T\)
REAL (8) :: PL, PR
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D \(\mathbb{F}\)
SU BROUTINE TGSEN_64 (LOB B,W ANTQ,W ANTZ,SELECT,N,A, [LDA],B, [LD B], ALPHA,BETA, \(Q,[L D Q], Z,[L D Z], M, P L, P R, D \mathbb{E},[W O R K],[L W O R K]\), \([\mathbb{I W} O R K],[\llbracket \mathbb{W}\) ORK \(],[\mathbb{N} F O])\)

COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::ALPHA,BETA, W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : A , B, Q , Z
\(\mathbb{N} \operatorname{TEGER}\) (8) :: IJOB , N,LDA,LDB,LDQ,LDZ,M,LWORK,LIWORK,
\(\mathbb{N}\) FO
\(\mathbb{N} \operatorname{TEGER}(8), \mathrm{D} \mathbb{M} \operatorname{ENSION}(:):: \mathbb{I W} O R K\)
LOGICAL (8) ::W ANTQ,W ANTZ
LOGICAL (8), D \(\mathbb{M}\) ENSION (:) :: SELECT
REAL (8) :: PL, PR
REAL (8), D \(\mathbb{M}\) ENSION (:) ::D \(\mathbb{F}\)

\section*{C INTERFACE}
\#include <sunperfh>
void ztgsen (int ijob, intw antq, intw antz, int * select, int n, doublecom plex *a, intlda, doublecom plex *b, int ldb, doublecom plex *alpha, doublecom plex *beta, doublecom plex *q, int ldq, doublecom plex * \(z\), int \(1 d z\), int *m, double *pl, double *pr, double *dif, int *info);
void ztgsen_64 (long ijob, long w antq, long w antz, long * select, long n, doublecom plex *a, long lda, doublecom plex *b, long ldb, doublecom plex *alpha, doublecom plex *beta, doublecom plex *q, long ldq, doublecom plex *z, long ldz, long *m, double \({ }^{*} \mathrm{pl}\), double *pr, double *dif, long *info);

\section*{PURPOSE}
ztgsen reorders the generalized Schur decom position of a complex \(m\) atrix pair ( \(A, B\) ) (in term \(s\) of an unitary equivalence trans-form ation \(Q\) '* ( \(A, B\) ) * \(Z\) ), so that a selected cluster of eigenvalues appears in the leading diagonalblocks of the pair ( \(A, B\) ). The leading colum ns of \(Q\) and \(Z\) form unitary bases of the comesponding left and right eigenspaces (deflating subspaces). (A, B) m ust be in generalized Schur canonical form, that is, \(A\) and \(B\) are both upper triangular.

ZTG SEN also com putes the generalized eigenvalues
w ( \(\mathfrak{j})=A L P H A(\mathcal{j}) / B E T A(\mathcal{O})\)
of the reordered \(m\) atrix pair ( \(A, B\) ).

Optionally, the routine com putes estim ates of reciprocal condition num bers foreigenvalues and eigenspaces. These are D ifin [ \(A 11, B 11\) ), (A 22, B22)] and D ifl[ \((A 11, B 11)\), (A 22, B22)], i.e. the separation (s) betw een the \(m\) atrix pairs (A 11, B 11) and ( \(22, B 22\) ) that comespond to the selected cluster and the eigenvalues outside the cluster, resp., and norm sof "pro ections" onto left and right eigenspaces w r.t. the selected cluster in the \((1,1)\)-block.

\section*{ARGUMENTS}

\footnotetext{
LJOB (input)
Specifies w hether condition num bers are required for the cluster ofeigenvalues (PL and PR) orthe deflating subspaces (D ifiu and \(D\) ifl):
=0: Only reorderw r.t. SELEC T .N o extras.
=1: Reciprocalofnorm s of "pro jections" onto left and righteigenspaces w r.t. the selected cluster ( PL and PR). = 2 : U pperbounds on D ifu and D ifl.
F-norm -based estim ate
(D \(\mathbb{F}(1: 2))\).
\(=3\) : Estim ate ofD ifiu and \(D\) ifl. 1 -norm -based esti-
\(m\) ate
(D IF (1:2)). About5 tim es as expensive as IJO B \(=\)
2. \(=4: C\) om pute \(P L, P R\) and \(D \mathbb{F}\) (i.e. 0,1 and 2 above) : Econom ic version to get itall. \(=5\) : C om pute PL, PR and D FF (ie. 0, 1 and 3 above)
}

\section*{SELECT (input)}

SELEC T specifies the eigenvalues in the selected
cluster. To selectan eigenvalue w ( \()\), SELECT ( \()\)
m ustbe set to
N (input) The order of the m atrices A and \(\mathrm{B} . \mathrm{N}>=0\).
A (input/output)
On entry, the upper triangularm atrix A, in generalized Schur canonical form. On exit, A is overw rilten by the reordered \(m\) atrix A.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

B (input/output)
O n entry, the upper triangularm atrix \(B\), in generalized Schur canonical form. On exit, \(B\) is overw ritten by the reordered \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the anay \(\mathrm{B} . \operatorname{LD} \mathrm{B}>=\) \(\max (1, \mathbb{N})\).

A LPHA (output)
The diagonalelem ents of \(A\) and \(B\), respectively, when the pair \((A, B)\) has been reduced to generalized Schur form. A LPHA (i),BETA (i) \(i=1, \ldots, N\) are the generalized eigenvalues.

BETA (output)
See the description of A LPH A .
Q (input/output)
On entry, if \(W\) ANTQ = TRUE., \(Q\) is an \(N\)-by \(N\) \(m\) atrix. On exit, \(Q\) has been postm ultiplied by the leftunitary transform ation \(m\) atrix which reorder ( \(\mathrm{A}, \mathrm{B}\) ); The leading \(M\) colum ns ofe form orthonor\(m\) albases for the specified pair of left eigenspaces (deflating subspaces). IfW ANTQ = FALSE., \(Q\) is notreferenced.

LDQ (input)
The leading dim ension of the array \(\mathrm{Q} . \mathrm{LD} \mathrm{Q}>=1\). IfW ANTQ = TRUE., LDQ >=N.

Z (input/output)

On entry, if \(W\) ANTZ = .TRUE., \(Z\) is an \(N\) Hy \(-N\) \(m\) atrix. O \(n\) exit, \(Z\) has been postm ultiplied by the leftunitary transform ation \(m\) atrix which reorder (A, B); The leading M colum ns of \(Z\) form orthonor\(m\) albases forthe specified pair of left eigenspaces (deflating subspaces). IfW ANTZ = FALSE., \(Z\) is notreferenced.

LD Z (input)
The leading \(d\) im ension of the array \(Z . L D Z>=1\). IfW ANTZ = .TRUE , LDZ \(>=N\).

M (output)
The dim ension of the specified pair of left and righteigenspaces, (deflating subspaces) \(0<=\mathrm{M}<=\) N.

PL (output)
IF \(\mathrm{IHOB}=1,4\), or \(5, \mathrm{PL}, \mathrm{PR}\) are low er bounds on
the reciprocal of the norm of "projections" onto leftand right eigenspace \(w\) ith respect to the selected chuster.
\(0<\mathrm{PL}, \mathrm{PR}<=1\). IfM \(=0\) orM \(=\mathrm{N}, \mathrm{PL}=\mathrm{PR}=1\) 。
If IO \(B=0,2\), or \(3 \mathrm{PL}, \mathrm{PR}\) are not referenced.

PR (output)
See the description ofPL .

D IF (output)
If IOO B \(>=2, D \mathbb{F}(1: 2)\) store the estim ates ofD ifu and \(D\) iff.
If \(I W B=2\) or \(4, D \mathbb{F}(1: 2)\) are \(F\)-norm -based upper
bounds on
\(D\) ifu and \(D\) ifl. If \(I O O B=3\) or \(5, D \mathbb{F}(1: 2)\) are \(1-\) norm - based estim ates of D ifu and D ifl, com puted using reversed comm unication w ith CLACON. IfM \(=\) 0 orN,\(D \mathbb{F}(1: 2)=F\)-norm ( \([A, B]\) ). If \(I J O B=0\) or \(1, D \mathbb{F}\) is notreferenced.

W ORK (w orkspace)
If \(I \mathrm{OOB}=0, \mathrm{~W} O R \mathrm{~K}\) is not referenced. O therw ise, on exit, if \(\mathbb{N F O}=0, \mathrm{~W}\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the amay \(W\) ORK.LW ORK \(>=1\) If \(\mathrm{IJOB}=1,2\) or \(4, \mathrm{LW} O R K>=2 \star M *(\mathbb{N} M)\) If \(\mathrm{IJOB}=3\) or 5 , LW ORK \(>=4 * M *(\mathbb{N}-\mathrm{M})\)

IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
IV ORK (w orkspace/output)
If \(\mathrm{IJO} B=0, \mathbb{I V} O R K\) is not referenced. O therw ise, on exit, if \(\mathbb{N} F O=0, \mathbb{I V} O R K(1)\) retums the optim al LIV ORK.

LIN ORK (input)
The dim ension of the anray \(\mathbb{I W} O R K . L \mathbb{I W} O R K>=1\). If \(\mathrm{IJO} \mathrm{B}=1,2\) or \(4, \mathrm{~L} \mathbb{I N}\) ORK \(>=\mathrm{N}+2\); If IJOB \(=3\) or \(5, L \mathbb{I N}\) ORK >= MAX \(\mathbb{N}+2,2 * M * \mathbb{N}+\mathrm{M})\) );

If LIV ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of the \(\mathbb{I W}\) ORK array, retums this value as the first entry of the \(\mathbb{I W} O R K\) array, and no errorm essage related to LIIN ORK is issued by X ERBLA.
\(\mathbb{I N F O}\) (output)
=0: Successfulexit.
\(<0:\) If \(\mathbb{N}\) FO \(=-\) - , the \(i\)-th argum enthad an illegal
value.
\(=1\) : Reordering of ( \(A, B\) ) failed because the transform ed \(m\) atrix pair ( \(A, B\) ) w ould be too far from generalized Schur form ; the problem is very ill-conditioned. (A, B) \(m\) ay have been partially reordered. If requested, 0 is retumed in D \(\mathbb{F}\) (*), \(P L\) and \(P R\).

\section*{FURTHER DETAILS}

ZTG SEN firstcollects the selected eigenvalues by com puting unitary \(U\) and \(W\) thatm ove them to the top left comerof ( \(A\), B ). In otherw ords, the selected eigenvalues are the eigenvalues of (A11, B11) in
\[
\begin{gathered}
U *(A, B) * W=(A 11 A 12)(B 11 B 12) n 1 \\
(0 \text { A 22),(0 B22) n2 } \\
n 1 \text { n2 n1 n2 }
\end{gathered}
\]
where \(\mathrm{N}=\mathrm{n} 1+\mathrm{n} 2\) and U ' m eans the conjugate transpose of U . The first \(n 1\) colum ns of \(U\) and \(W\) span the specified pair of left and righteigenspaces (deflating subspaces) of ( \(A, B\) ).

If ( \(\mathrm{A}, \mathrm{B}\) ) has been obtained from the generalized real Schur decom position of a \(m\) atrix pair \((C, D)=Q *(A, B) * Z\) ', then the reordered generalized Schur form of ( \(\mathrm{C}, \mathrm{D}\) ) is given by
\[
(C, D)=(Q * U) \star(U *(A, B) \star W) \star(Z * W) ',
\]
and the firstn1 colum ns of Q * U and \(\mathrm{Z} * \mathrm{~W}\) span the corresponding deflating subspaces of ( \(C, D\) ) \(Q\) and \(Z\) store \(Q * U\) and Z *W , resp.).
N ote that if the selected eigenvalue is sufficiently illconditioned, then its value \(m\) ay differ significantly from its value before reordering.

The reciprocalcondition num bers of the left and right eigenspaces spanned by the firstn1 colum ns ofU and \(W\) (or \(\mathrm{Q} * \mathrm{U}\) and \(\mathrm{Z} * \mathrm{~W}) \mathrm{m}\) ay be retumed in \(\mathrm{D} \boldsymbol{F}(1: 2)\), comesponding to \(D\) ifu and \(D\) ifl, resp.

The D ifu and D iflare defined as:
ifu \([\) A 11, B11), (A 22, B22) \(]=\) sigm a-m in ( Zu )
and
where sigm a-m in \((\mathrm{Zu})\) is the sm allest singular value of the
\[
\left(2 \star_{\mathrm{n}} 1 *_{\mathrm{n}} 2\right)-\text { by }-\left(2 \star_{\mathrm{n}} 1 *_{\mathrm{n}} 2\right) \mathrm{m} \text { atrix }
\]
\(\mathrm{u}=[\mathrm{kron}(\mathrm{In} 2, \mathrm{~A} 11)-\mathrm{kron}(\mathrm{A} 22\) ', In1) ]
[kron(In2, B11) kron (B22', In1)].

H ere, \(\operatorname{In} x\) is the identily \(m\) atrix of size \(n x\) and A 22 ' is the transpose of A 22. kron ( \(X\), \(Y\) ) is the \(K\) roneckerproduct betw een the \(m\) atrices \(X\) and \(Y\).

W hen D IF (2) is sm all, sm allchanges in (A, B) can cause large changes in the deflating subspace. A \(n\) approxim ate (asym ptotic) bound on them axim um angularemor in the com puted deflating subspaces is PS * norm ( \((A, B)\) ) /D \(\mathbb{F}(2)\),
where EPS is the \(m\) achine precision.

The reciprocal norm of the pro jectors on the leftand right eigenspaces associated \(w\) ith (A 11, B 11) m ay be retumed in PL and PR. They are com puted as follow s. First we com pute L and \(R\) so that \(P\) * \((A, B){ }^{\star} Q\) is block diagonal, w here
\[
=(I-\Psi) n 1 \quad Q=(I R) n 1
\]
\[
(0 \mathrm{I}) \mathrm{n} 2 \text { and }(0 I) \mathrm{n} 2
\]
n1 n2
n1 n2
and \((\amalg, R)\) is the solution to the generalized Sylvester equation \(11 * \mathrm{R}-\mathrm{L} * \mathrm{~A} 22=-\mathrm{A} 12\)
 norm \((\mathbb{R}) * * 2+1) * *(-1 / 2)\). A n approxim ate (asym ptotic) bound on the average absolute error of the selected eigenvalues is EPS * norm ( \(\mathrm{A}, \mathrm{B})\) )/PL.

There are also globalemorbounds which valid forperturbations up to a certain restriction: A low erbound \((x)\) on the sm allest \(F\)-norm ( \(\mathbb{E}, F\) ) forw hich an eigenvalue of (A 11, B 11) \(m\) ay \(m\) ove and coalesce \(w\) th an eigenvalue of (A 22,B22) under perturbation \((\mathbb{E}, F)\), (ie. \((A+E, B+F)\), is
\(\mathrm{x}=\)


A \(n\) approxim ate bound on \(x\) can be com puted from \(D \mathbb{F}(1: 2)\), PL and PR .

If \(y=(F-\) norm \((E, F) / x)<=1\), the angles betw een the perturbed (L', R) and unperturbed ( \(L, R\) ) left and right deflating subspaces associated \(w\) ith the selected chuster in the \((1,1)\)-blocks can be bounded as
\(m a x-a n g l e(L, L)<=\arctan (y * P L /(1-y *(1-P L *\) PL)** (1/2))
\(\max \operatorname{angle}(R, R)<=\arctan (y * P R /(1-y *(1-P R *\) PR ) ** (1/2))

See LA PA CK U ser's G uide section 4.11 or the follow ing references form ore inform ation.

N ote that if the default \(m\) ethod for com puting the Frobenius-norm - based estim ate D \(\mathbb{F}\) is not wanted (see CLATDF), then the param eter \(\mathbb{D} \mathbb{F} \mathbb{B}\) (see below) should be changed from 3 to 4 (routine CLATDF (INOB \(=2 \mathrm{w}\) ill be used)). See C TG SY L form ore details.

B ased on contributions by
Bo K agstrom and PeterPorom aa, D epartm ent of C om puting Scienœ,
U m ea U niversity, S-901 87 Um ea, Sw eden.

References
\(========\)
[1] B . K agstrom ; A D irectM ethod forReordering Eigenvalues in the

G eneralized RealSchurForm of a RegularM atrix Pair (A, B), in

M S.M oonen etal (eds), LinearA lgebra for Large Scale and

Real-T in e A pplications, K luw erA cadem ic Publ. 1993, pp 195-218.
[2] B.K agstrom and P.Porom aa; Com puting Eigenspaces w ith Specified

Eigenvalues of a RegularM atrix Pair ( \(A, B\) ) and Condition

Estim ation: Theory, A lgonithm s and Softw are, Report
UM \(\mathbb{N} F-94.04\), Departm ent of Com puting Science, Um ea U niversity, S-901 87 Um ea, Sw eden, 1994. A lso as LA PA CK W orking N ote 87.

To appear in N um ericalA lgorithm \(\mathrm{S}, 1996\).
[3] B . K agstrom and P . Porom aa, LA PA CK -Style A lgorithm s and Softw are
for Solving the Generalized Sylvester Equation and Estim ating the Separation betw een RegularM atrix Pairs, ReportUM \(\mathbb{N}\) F 9323, D epartm entofC om puting Science, Um ea U niversity, S-901 87 Um ea, Sw eden, D ecem ber 1993, Revised A pril 1994, A lso as LAPACK w orking

N ote 75.To appear in A CM Trans. on M ath. Softw are, V ol 22, N O 1, 1996.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztgsja - com pute the generalized singular value decom position (G SVD ) of tw o com plex upper triangular (ortrapezoidal) \(m\) atrices \(A\) and \(B\)

\section*{SYNOPSIS}
```

SU BROUTINE ZTGSJA (OOBU,NOBV,JOBQ,M,P,N,K,L,A,LDA,B,LD B,
TOLA,TOLB,ALPHA,BETA,U,LDU,V,LDV,Q,LDQ,W ORK,NCYCLE,
\mathbb{NFO)}

```
CHARACTER * 1 JOBU, \(0 \mathrm{OBV}, ~ J O B Q\)
DOUBLE COMPLEX A (LDA,*), B (LDB,*), U (LDU,*), V (LDV,*),
Q (LDQ,*), W ORK (*)
\(\mathbb{N}\) TEGER M, \(\mathrm{P}, \mathrm{N}, \mathrm{K}, \mathrm{L}, \mathrm{LD} A, L D B, L D U, L D V, L D Q, N C Y C L E, \mathbb{N} F O\)
DOUBLE PRECISION TOLA, TOLB
DOUBLE PRECISION ALPHA (*), BETA (*)
SU BROUTINE ZTGSJA_64 (JOBU, JOBV, JOBQ,M,P,N,K,L,A,LDA,B,LDB,
    TOLA, TOLB, ALPHA,BETA, U,LDU,V,LDV,Q,LDQ,W ORK,NCYCLE,
    \(\mathbb{N} F O\) )
CHARACTER * 1 JOBU, JOBV , JOBQ
DOUBLE COM PLEX A (LDA,*), B (LDB,*), U (LDU ,*), V (LDV ,*),
Q (LDQ, \(\left.{ }^{*}\right), \mathrm{W}\) ORK (*)
\(\mathbb{N} T E G E R * 8 \mathrm{M}, \mathrm{P}, \mathrm{N}, \mathrm{K}, \mathrm{L}, \mathrm{LD} A, L D B, L D U, L D V, L D Q, N C Y C L E\),
\(\mathbb{N} F O\)
DOUBLE PRECISION TOLA, TOLB
DOUBLE PRECISION ALPHA (*),BETA (*)

F95 INTERFACE
SU BROUTINE TGSJA (JOBU, JOBV, JOBQ, \(\mathbb{M}],[\mathbb{P}], \mathbb{N}], K, L, A,[L D A], B\), [LDB],TOLA, TOLB, ALPHA,BETA, \(\mathrm{U},[\mathrm{LD} \mathrm{U}], \mathrm{V},[\mathrm{LDV}], \mathrm{Q},[\mathrm{LD} Q]\),
[W ORK],NCYCLE, [ \(\mathbb{N} F O\) ])

COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:)::A, B, U, V, Q
\(\mathbb{N} T E G E R:: M, P, N, K, L, L D A, L D B, L D U, L D V, L D Q, N C Y C L E\), \(\mathbb{N}\) FO
REAL (8) ::TOLA,TOLB
REAL (8),D \(\mathbb{I M}\) ENSION (:) ::A LPHA,BETA
SU BROUTINE TGSJA_64 (JOBU, JOBV, \(\mathrm{JOBQ}, \mathbb{M}], \mathbb{P}], \mathbb{N}], K, L, A,[L D A]\), B, [LD B ], TOLA, TOLB, A LPHA, BETA, U, [LDU ], V, [LDV], \(\mathrm{Q},[\mathrm{LD} \mathrm{Q}]\), [W ORK],NCYCLE, [NFO])

COMPLEX (8),D \(\mathbb{D}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : ::A, \(B, U, V, Q\)
\(\mathbb{N}\) TEGER (8) ::M, \(\mathrm{P}, \mathrm{N}, \mathrm{K}, \mathrm{L}, \mathrm{LDA}, \mathrm{LD} B, L D \mathrm{U}, \mathrm{LDV}, \mathrm{LD} Q, N C Y-\)
CLE, \(\mathbb{I N F O}\)
REAL (8) ::TOLA,TOLB
REAL (8), D \(\mathbb{M}\) ENSION (:) ::ALPHA,BETA

\section*{C INTERFACE}
\#include <sunperfh>
void ztgsja (char jobu, char jobv, char jobq, intm , int p, int \(n\), int \(k\), int l, doublecom plex *a, int lda, doublecom plex *b, int ldb, double tola, double tolb, double *alpha, double *beta, doublecom plex *u, int ldu, doublecom plex *v, int ldv, doublecom plex *q, intldq, int*ncycle, int*info);
void ztgsja_64 (char jंbu, char jobv, char jobq, long m, long p, long \(n\), long \(k\), long \(l\), doublecom plex *a, long lda, doublecom plex *b, long ldb, double tola, double tolb, double *alpha, double *beta, doublecom plex *u, long ldu, doublecom plex *v, long ldv, doublecom plex *q, long ldq, long *ncycle, long *info);

\section*{PURPOSE}
ztgsja com putes the generalized singular value decom position (GSVD) of tw o complex upper triangular (or trapezoidal) \(m\) atrices \(A\) and \(B\).

On entry, it is assum ed thatm atrices \(A\) and \(B\) have the follow ing form s , which m ay be obtained by the preprocessing subroutine CGGSVP from a generalM by -N m atrix A and \(\mathrm{P}-b y-\mathrm{N}\)
m atrix B ：
```

            NK工 K L
    A = K (0 A12 A 13) ifM K-L >= 0;
L (0 0 A 23)
M K-工(0 0 0 )
N-K\dashv K L
A = K (0 A12 A13) ifM K L<0;
M-K(0 0 A 23)
N-K\dashvK L
B=L(0 0 B13}
P-工(0 0 0 )

```
where the \(K\) boy \(K\) m atrix A 12 and \(L\) boy－ m atrix B 13 are non－ singularupper triangular；A 23 is L－by－L upper triangular if \(M K-\rfloor>=0\) ，otherw ise A 23 is \((M-K)\)－by -4 uppertrapezoidal． On exit，
\(U^{*} A * Q=D 1^{*}(0 R), \quad V{ }^{*} B * Q=D 2^{*}(0 R)\),
where \(U, V\) and \(Q\) are unitary \(m\) atrices，\(Z\)＇denotes the conju－ gate transpose of \(Z, R\) is a nonsingular upper triangular \(m\) atrix，and D 1 and D 2 are＇＂diagonal＂m atriges，which are of the follow ing structures：

IfM \(K-工>=0\) ，
\[
\begin{aligned}
& \text { K L } \\
& \text { D } 1=K(I O) \\
& \text { L ( } 0 \text { C ) } \\
& M \text { K- ( } 00 \text { ) } \\
& \text { K L } \\
& D 2=L \quad(0 \mathrm{~S}) \\
& \mathrm{P}-\left(\begin{array}{ll}
0 & 0
\end{array}\right)
\end{aligned}
\]

N K孔 K L
\((0 R)=K \quad(0 \quad R 11 R 12) K\)
\(\mathrm{L}\left(\begin{array}{lll}0 & \mathrm{R} 22\end{array}\right) \mathrm{L}\)
where
\[
\begin{aligned}
& C=\operatorname{diag}(A L P H A(K+1), \ldots, A L P H A(K+L)), \\
& S=\operatorname{diag}(\operatorname{BETA}(K+1), \ldots, B E T A(K+L)), \\
& C * * 2+S * * 2=I .
\end{aligned}
\]
\(R\) is stored in \(A(1: K+L, N-K+1 \mathbb{N})\) on exit．

IfM K \(\mathrm{K}<0\),

> K M K K + L M
> \(\mathrm{D} 1=\mathrm{K}(\mathrm{I} 00\) )
> M K ( 0 C 0 )
> K M K K + L M
> \(\mathrm{D} 2=\mathrm{M} \mathrm{K}(0 \mathrm{~S} 0)\)
> K+L-M (0 O I )
> P- ( 0000 )

N K- \(\mathrm{K} \quad \mathrm{M}\) K \(\mathrm{K}+\mathrm{L} \mathrm{M}\)
\(M-K(0 \quad 0 \quad R 22 R 23)\)
\(\mathrm{K}+\mathrm{L} \mathrm{M}\) (0 0 (0 R33)
where
C \(=\operatorname{diag}(A L P H A(K+1), \ldots, A L P H A M))\),
\(S=\operatorname{diag}(\) BETA \((K+1), \ldots, B E T A(M))\),
\(\mathrm{C} * * 2+\mathrm{S} * * 2=\mathrm{I}\).
\(R=(R 11 R 12 R 13)\) is stored in A (1 M ,N K \(-\mathrm{L}+1 \mathbb{N}\) ) and R 33
is stored
( 0 R22R23)
in \(B(M-K+1: L, N+M-K-\perp+1 \mathbb{N})\) on exit.

The com putation of the unitary transform ation \(m\) atrioes \(U, V\) or \(Q\) is optional. These \(m\) atrioes \(m\) ay eitherbe form ed explicitly, or they \(m\) ay be postm ultiplied into input \(m\) atrices \(\mathrm{U} 1, \mathrm{~V} 1\), orQ 1 .

CTGSJA essentially uses a variant of \(K\) ogbetliantz algorithm to reduce \(m\) in ( \(L, M-K\) )-by-L triangular (ortrapezoidal) \(m\) atrix A 23 and \(L\)-by- \(m\) atrix B 13 to the form :
U 1 *A \(13 * \mathrm{Q} 1=\mathrm{C} 1 * \mathrm{R} 1\); V 1 *B13*Q \(1=\mathrm{S} 1 * \mathrm{R} 1\), where \(\mathrm{U} 1, \mathrm{~V} 1\) and Q 1 are unitary \(m\) atrix, and Z 'is the conjugate transpose of Z. C1 and S1 are diagonalm atrices satisfying
\(\mathrm{C} 1 * * 2+\mathrm{S} 1 * * 2=\mathrm{I}\),
and \(R 1\) is an \(L-b y-L\) nonsingular uppertriangularm atrix.

\section*{ARGUMENTS}
\(J 0 \mathrm{BU}\) (input)
\(=\mathrm{U}:\) : U m ustcontain a unitary matrix U 1 on entry, and the productU 1*U is retumed; = I': U is initialized to the unitm atrix, and the unitary \(m\) atrix \(U\) is retumed; \(=N\) : \(U\) is not com puted.
\(=\mathrm{V}^{\prime}\) : V m ustcontain a unitary matrix V1 on entry, and the product V 1*V is retumed; = I ': V is initialized to the unitm atrix, and the unitary m atrix V is retumed; \(=\mathrm{N}: \mathrm{V}\) is not com puted.

JOBQ (input)
\(=\mathrm{Q}\) ': Q m ustcontain a unitary matrix Q 1 on entry, and the product Q 1*Q is retumed; = I': Q is initialized to the unitm atrix, and the unitary m atrix Q is retumed; \(=\mathrm{N}\) : Q is not com puted.

M (input) The num ber of row s of the \(m\) atrix \(A . M>=0\).
\(P\) (input) The num ber of row s of the \(m\) atrix \(B . ~ P>=0\).

N (input) The num ber of colum ns of the \(m\) atrioes \(A\) and \(B\). \(N\) \(>=0\).
\(K\) (input) \(K\) and \(L\) specify the subblocks in the input \(m\) atrices \(A\) and \(B\) :
\(A 23=A(K+1 \mathbb{M} \mathbb{N}(K+L, M) \mathbb{N}-1+1 \mathbb{N})\) and \(B 13=\) \(B(1: L, N-\perp+1 \mathbb{N})\) of \(A\) and \(B, w h o s e G S V D\) is going to be com puted by CTGSJA. See the Further D etails section below .

L (input) See the description of K .

A (input/output)
On entry, the \(M\) boy -N matrix \(A\). On exit, \(A(\mathbb{N}-\) \(\mathrm{K}+1 \mathbb{N}, 1 \mathbb{M} \mathbb{N}(\mathrm{~K}+\mathrm{L}, \mathrm{M})\) ) contains the triangular \(m\) atrix \(R\) orpart ofR. See Punpose for details.

LD A (input)
The leading dim ension of the anay \(A . L D A>=\) \(\max (1, M)\).

B (input/output)
On entry, the \(\mathrm{P}-b y-N \mathrm{~N}\) atrix B. On exit, if neces sary, \(B(K+1: N+M-K+1 \mathbb{N}\) ) contains a partofR . See Purpose for details.

LD B (input)
The leading dim ension of the aray \(\mathrm{B} . \mathrm{LD} \mathrm{B}>=\) \(\max (1, P)\).

TOLA (input)
TOLA and TOLB are the convergence criteria for the Jacobi- K ogbetliantz iteration procedure. Generally, they are the sam e as used in the preprocessing step, say TOLA \(=M A X(M, N) \star\) norm (A) \({ }^{*} M\) ACHEPS, \(T O L B=M A X(P, N)^{\star}\) norm \((B) \star\) M ACHEPS.

TO LB (input)
See the description of TO LA .

A LPHA (output)
On exi, ALPHA and BETA contain the generalized singularvalue pairs of \(A\) and \(B\); \(A\) LPHA \((1: K)=1\), \(\operatorname{BETA}(1: K)=0\), and ifM \(K-L>=0\), A LPHA \((K+1 \mathbb{K}+L)\)
\(=\operatorname{diag}(\mathrm{C})\),
BETA \((\mathbb{K}+1 \mathrm{~K}+\mathrm{L})=\operatorname{diag}(\mathrm{S})\), or if \(\mathrm{M}-\mathrm{K}-<0\),
ALPHA \((\mathbb{K}+1 \mathrm{M})=\mathrm{C}, \mathrm{ALPHA}(\mathrm{M}+1 \mathrm{~K}+\mathrm{L})=0\)
BETA \((\mathbb{K}+1 \mathrm{M})=\mathrm{S}, \operatorname{BETA}(\mathrm{M}+1: K+\mathrm{L})=1\). Furtherm ore, if \(\mathrm{K}+\mathrm{L}<\mathrm{N}\), ALPHA \((\mathrm{K}+\mathrm{L}+1 \mathbb{N})=0\)
\(\operatorname{BETA}(K+L+1 \mathbb{N})=0\).

BETA (output)
See the description of A LPH A .

U (input) On entry, if \(\mathrm{JOBU}=\mathrm{U}\) ', U mustcontain a matrix
U 1 (usually the unitary \(m\) atrix retumed by
CGGSVP). On exit, if \(J O B U=\) ' 1 , \(U\) contains the unitary matrix \(U\); if \(J O B U=U\) ', \(U\) contains the product \(\mathrm{U} 1 * \mathrm{U}\). If \(\mathrm{JOB} \mathrm{B}=\mathrm{N}\) ', U is notreferenced.

LD U (input)
The leading dim ension of the array U. LD U >= \(m\) ax \((1, M)\) if \(J O B U=U ' ; L D U>=1\) otherw ise.

V (input) On entry, if \(\mathrm{JO} \mathrm{BV}=\mathrm{V}\) ', V m ust contain a matrix V1 (usually the unitary matrix retumed by CGGSVP). On exit, if \(J O B V=I^{\prime}, V\) contains the unitary \(m\) atrix \(V\); if \(J O B V=V\) ', \(V\) contains the product \(\mathrm{V} 1 * \mathrm{~V}\). If \(\mathrm{JO} \mathrm{BV}=\mathrm{N}, \mathrm{V}\) is not referenced.

LD V (input)
The leading dim ension of the aray \(V\). LD \(V\) >= \(\max (1, \mathrm{P})\) if \(\mathrm{JOBV}=\mathrm{V} ;\);LDV >=1 otherw ise.

Q (input) \(O n\) entry, if \(J O B Q=Q\) ', Q m ustcontain a matrix Q 1 (usually the unitary matrix retumed by CGGSVP). On exit, if \(J O B Q=' I\) ', \(Q\) contains the unitary matrix \(Q\); if \(J O B Q=Q\) ', \(Q\) contains the product \(Q 1 * Q\). If \(J O B Q=N\) ', \(Q\) is not referenced.

LD Q (input)
The leading dim ension of the array \(Q . L D Q>=\) \(\max (1, N)\) if \(\mathcal{O} B Q=Q ; L D Q>=1\) otherw ise.

W ORK (w orkspace)
dim ension \((2 * N)\)

NCYCLE (output)
The num ber of cycles required for convergence.
\(\mathbb{N F O}\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue.
= 1: the procedure does not converge after M A X IT cycles.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ztgsna -estim ate reciprocalcondition num bers for specified
eigenvalues and/oreigenvectors of a \(m\) atrix pair ( \(A, B\) )

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZTGSNA (JOB,HOW MNT,SELECT,N,A,LDA,B,LDB,VL,LDVL,}
VR,LDVR,S,D\mathbb{F},MM,M,WORK,LWORK,I\mathbb{ORK,INFO)}
CHARACTER * 1 JOB,HOW MNT
DOUBLE COM PLEXA (LDA,\star),B (LDB,*),VL (LDVL,\star),VR (LDVR,*),
W ORK (*)
\mathbb{N TEGER N,LDA,LDB,LDVL,LDVR,MM ,M ,LW ORK,INFO}
INTEGER IN ORK (*)
LOG ICAL SELECT (*)
DOUBLE PRECISION S (*),D \mathbb{F (*)}
SU BROUT\mathbb{NE ZTGSNA_64(JOB,HOW MNT,SELECT,N,A,LDA,B,LDB,VL,}

```

```

CHARACTER * 1 JOB,HOW MNT
D OU BLE COM PLEX A (LDA,*),B (LDB,*), VL (LDVL,*), VR (LDVR,*),
W ORK (*)
\mathbb{NTEGER*8N,LDA,LDB,LDVL,LDVR,MM ,M,LW ORK, INFO}
INTEGER*8 IN ORK (*)
LOG ICAL*8 SELECT (*)
DOUBLE PRECISION S (*),D \mathbb{F (*)}

```
F95 INTERFACE
    SU BROUTINE TGSNA (JOB,HOW MNT, SELECT, \(\mathbb{N}\) ],A, [LDA], B, [LDB],VL,
    [LDVL],VR, [LDVR],S,D \(\mathbb{F}, M M, M,[W O R K],[L W O R K],[\mathbb{W} O R K]\),
    [ \(\mathbb{N} F O\) ])

CHARACTER ( \(几 E N=1\) ) : : JOB , HOW M NT
COM PLEX (8), D \(\mathbb{M} E N S I O N(:):\) W ORK
COM PLEX (8), D \(\mathbb{I M} E N S I O N\) (:,:) ::A, B,VL,VR
\(\mathbb{N} T E G E R:: N, L D A, L D B, L D V L, L D V R, M M, M, L W O R K, \mathbb{N} F O\)
\(\mathbb{N}\) TEGER,D \(\mathbb{M}\) ENSION (:) :: \(\mathbb{I W}\) ORK
LOG ICAL, D \(\mathbb{I M}\) ENSION (:) :: SELECT
REAL (8), D \(\mathbb{M}\) ENSION (:) :: S, D \(\mathbb{F}\)

SU BROUTINE TGSNA_64 (JOB,HOW M NT,SELECT, \(\mathbb{N}], A,[L D A], B,[L D B], V L\), \([\) LDVL], VR, [LDVR], S, D \(\mathbb{F}, M M, M,[W O R K],[L W O R K],[\mathbb{W} O R K]\), [ \(\mathbb{N} \mathrm{FO}]\) )

CHARACTER (LEN=1) :: JOB, HOW M NT
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{I M} E N S I O N\) (:,:) ::A, B,VL,VR
\(\mathbb{N}\) TEGER (8) :: N , LD A , LD B , LDVL, LDVR , M M , M , LW ORK , \(\mathbb{N} F O\)
\(\mathbb{N} \operatorname{TEGER}\) (8), D \(\mathbb{M} \operatorname{ENSION(:)::\mathbb {IW}ORK}\)
LOGICAL (8), D IM ENSION (:) :: SELECT
REAL (8), D \(\mathbb{M}\) ENSION (:) ::S,D \(\mathbb{F}\)

\section*{C INTERFACE}
\#include <sunperfh>
void ztgsna (char job, char how m nt, int *select, intn, doublecom plex *a, int lda, doublecom plex *b, int ldb, doublecom plex \({ }^{*}\) vl, int ldvl, doublecom plex \({ }^{*}\) vr, int ldvr, double *s, double *dif, intm m, int *m, int*info);
void ztgsna_64 (char jं.b, charhow mnt, long *select, long n, doublecom plex *a, long lda, doublecom plex *b, long ldb, doublecom plex *vl, long ldvl, doublecom plex
*Vr, long ldvr, double *s, double *dif, long m m , long *m, long *info);

\section*{PURPOSE}
ztgsna estim ates reciprocalcondition num bers for specified eigenvalues and/or eigenvectors of a m atrix pair ( \(A, B\) ).
( \(\mathrm{A}, \mathrm{B}\) ) m ustbe in generalized Schur canonical form, that is,
\(A\) and \(B\) are both upper triangular.

\section*{ARGUMENTS}

Specifies w hether condition num bers are required foreigenvalues (S) oreigenvectors (DF):
\(=\mathrm{E}\) ': foreigenvalues only (S);
= V ': foreigenvectors only ( \(\mathbb{P}\) );
\(=B\) ': forboth eigenvalues and eigenvectors ( S and \(D \mathbb{F}\) ).

HOW MNT (input)
= A ': com pute condition num bers for all eigenpairs;
\(=S^{\prime}\) : com pute condition num bers for selected eigenpairs specified by the array SELEC T .

SELECT (input)
If HOW MNT = S', SELECT specifies the eigenpairs
for which condition num bers are required. To select condition num bers for the comesponding j th eigenvalue and/oreigenvector, SELECT ( ) ) must be set to TRUE.. IfHOW MNT=A', SELECT is not referenced.

N (input) The order of the square \(m\) atrix pair ( \(A, B\) ). \(N \quad>=\) 0.

A (input) The uppertriangularm atrix \(A\) in the pair \((A, B)\).
LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, \mathbb{N})\).
\(B\) (input) The uppertriangularm atrix \(B\) in the pair \((A, B)\).
LD B (input)
The leading dim ension of the aray \(\mathrm{B} . \mathrm{LD} \mathrm{B}>=\) \(\max (1, N)\).

VL (input)
If \(\mathrm{JOB}=\mathrm{E}\) 'or B ', VL mustcontain left eigenvectors of \((A, B)\), comesponding to the eigenpairs specified by HOW M NT and SELECT. The eigenvectors
\(m\) ust be stored in consecutive colum ns of V , as retumed by CTGEVC. If JOB \(=V^{\prime}, V L\) is not referenced.

LDVL (input)
The leading din ension of the amray VL.LD VL >=1; and If \(J 0 B=E\) 'or \(B ', L D V L>=N\).

VR (input)
If \(J 0 B=E\) 'or \(B\) ', VR m ust contain right eigen-
vectors of ( \(A, B\) ), comesponding to the eigenpairs specified by H OW M NT and SELECT . The eigenvectors \(m\) ust be stored in consecutive colum ns ofVR, as retumed by CTGEVC. If JOB \(=V\) ', \(V R\) is not referenced.

LDVR (input)
The leading dim ension of the array VR.LD VR >= 1;
If \(\mathrm{JOB}=\mathrm{E}\) 'or B ', LDVR \(>=\mathrm{N}\).
S (output)
If \(J O B=E\) ' or \(B^{\prime}\) ', the reciprocal condition
num bers of the selected eigenvalues, stored in consecutive elem ents of the aray. If \(\mathrm{JO} \mathrm{B}=\mathrm{V}\) ', \(S\) is not referenced.

D \(\mathbb{F}\) (output)
If \(\mathrm{JOB}=\mathrm{V}\) ' or B ', the estim ated reciprocal condition numbers of the selected eigenvectors, stored in consecutive elem ents of the array. If the eigenvalues cannot be reordered to com pute D \(\mathbb{F}(\mathcal{j}), \mathrm{D} \mathbb{F}(\mathrm{y})\) is set to 0 ; this can only occur w hen the true value w ould be very sm allanyw ay. Foreach eigenvalue/vector specified by SELECT, D \(\mathbb{F}\) stores a Frobenius norm -based estim ate of D ifl. If \(\mathrm{JOB}=\mathrm{E}, \mathrm{D} \mathbb{F}\) is not referenced.

M M (input)
The num berofelem ents in the anays \(S\) and \(D \mathbb{F} . M M\) \(>=\mathrm{M}\).

M (output)
The num berof elem ents of the arrays \(S\) and D \(\mathbb{F}\) used to store the specified condition num bers; for each selected eigenvalue one elem ent is used. If HOWMNT=A', M is setto \(N\).

W ORK (w orkspace)
If \(\mathrm{JOB}=\mathrm{E}\) ', \(\mathrm{W} O \mathrm{OR}\) is not referenced. O therw ise, on exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the amay \(W\) ORK. LW ORK \(>=1\). If JOB \(=V\) 'or \(B '\) 'LW ORK \(>=2 * N * N\).

IW ORK (w orkspace)
dim ension \((\mathbb{N}+2)\) If \(J O B=E^{\prime}, \mathbb{I N} O R K\) is not referenced.
\(\mathbb{N F O}\) (output)

\section*{FURTHER DETAILS}

The reciprocal of the condition num ber of the \(i\)-th generalized eigenvalue \(w=(a, b)\) is defined as
\(S(\mathbb{I})=\left(v^{A} u|* * 2+|v B u| * * 2)^{* *}(1 / 2) /\right.\)
(norm (u)*norm (v))
\(w\) here \(u\) and \(v\) are the rightand lefteigenvectors of ( \(A, B\) ) comesponding to \(\mathrm{w} ;|z|\) denotes the absolute value of the com plex num ber, and norm (u) denotes the 2-norm of the vector \(u\). The pair \((a, b)\) comesponds to an eigenvaluew \(=a \neq\) \(v A u(N B u)\) of the \(m\) atrix pair ( \(A, B\) ). If both \(a\) and \(b\) equal zero, then \((A, B)\) is singular and \(S(I)=-1\) is retumed.

A \(n\) approxim ate errorbound on the chordal distance betw een the i-th computed generalized eigenvalue \(w\) and the comesponding exacteigenvalue lam bda is
\[
\text { chord }(w, \operatorname{lam} \text { bda) }<=\text { EPS * nom }(A, B) / S(I),
\]
where EPS is the \(m\) achine precision.

The reciprocal of the condition num ber of the right eigenvector \(u\) and lefteigenvectorv corresponding to the generalized eigenvalue w is defined as follow s. Suppose
\[
\begin{array}{rl}
(\mathrm{A}, \mathrm{~B}) & =(\mathrm{a} *)(\mathrm{b} *) 1 \\
& (0 \mathrm{~A} 22),(0 \mathrm{~B} 22) \mathrm{n}-1 \\
1 & \mathrm{n}-1 \\
1 \mathrm{n}-1
\end{array}
\]

Then the reciprocalcondition num berD \(\mathbb{F}(\mathbb{I})\) is
\[
\mathrm{D} \text { ifl }[(\mathrm{a}, \mathrm{~b}), \text { (A 22, B 22)] = sigm a-m in ( } \mathrm{Z} 1 \text { ) }
\]
where sigm a-m in (Z) denotes the sm allest singular value of
```

Z l= [kron (a, In-1) -kron(1,A 22) ]

```
[ kron (b, In-1) kron (1, B22) ].

H ere In - 1 is the identity \(m\) atrix of size \(n-1\) and \(X\) ' is the conjugate transpose of X . \(\mathrm{kron}(\mathrm{X}, \mathrm{Y})\) is the K roneckerproduct.betw een the \(m\) atrices \(X\) and \(Y\).

W e approxim ate the sm allestsingularvalue of Zl w th an
upperbound. This is done by C LA TD F.
A \(n\) approxim ate errorbound for com puted eigenvector \(V L\) (i) orVR(i) is given by
```

EPS * norm (A,B )/D \mathbb{F (i).}

```

See ref. [2-3] form ore details and further references.
\(B\) ased on contributions by
Bo K agstrom and PeterPorom aa, D epartm ent of Com puting Science,
U m ea U niversity, S-901 87 Um ea, Sw eden.

\section*{References}

\section*{==========}
[1] B .K agstrom ; A D irectM ethod forReordering Eigenvalues in the

G eneralized RealSchurForm of a RegularM atrix Pair (A, B), in

M S.M oonen etal (eds), LinearA lgebra for Large Scale and

Real-T im e A pplications, K luw erA cadem ic Publ. 1993, pp 195-218.
[2] B.K agstrom and P.Porom aa; C om puting Eigenspaces w th Specified

Eigenvalues of a RegularM atrix Pair (A, B) and Condition

Estim ation: Theory, A lgonithm s and Softw are, Report
UM \(\mathbb{I N} F-94.04\), D epartm ent of Com puting Science, Umea U niversity,

S-901 87 Um ea, Sw eden, 1994.A lso as LA PA CK W orking N ote 87.

To appear in N um ericalA lgonithm s, 1996.
[3] B. K agstrom and P. Porom aa, LA PA CK -Style A lgorithm \(s\) and Softw are
for Solving the Generalized Sylvester Equation and Estim ating the

Separation betw een RegularM atrix Pairs, ReportUM \(\mathbb{N}\) F 93 23,

D epartm entofC om puting Science, Um ea U niversity, S-901 87 Um ea,

Sw eden, D ecem ber 1993, Revised A pril 1994, A lso as LAPACK W orking

N ote 75.
To appear in A CM Trans. on M ath. Softw are, V ol22, \(\mathrm{N} \circ\) 1, 1996.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ztgsyl-solve the generalized Sylvesterequation

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZTGSYL (TRANS,IJOB,M,N,A,LDA,B,LDB,C,LDC,D,LDD,}
E,LDE,F,LDF,SCALE,D \mathbb{F,W ORK,LW ORK,IN ORK,INFO)}
CHARACTER * 1 TRANS
DOUBLE COM PLEX A (LDA,*), B (LDB,*), C (LDC,*), D (LDD,*),
E (LDE,*),F (LDF,*),W ORK (*)
\mathbb{NTEGER LDOB,M,N,LDA,LDB,LDC, LDD, LDE, LDF, LW ORK,}
\mathbb{NFO}
INTEGER 㱐ORK (*)
DOUBLE PRECISION SCALE,D F
SU BROUT\mathbb{NE ZTGSYL_64 (TRANS,IJOB,M,N,A,LDA,B,LDB,C,LDC,D,}
LDD,E,LDE,F,LDF,SCALE,D \mathbb{F,W ORK,LW ORK,IN ORK,INFO)}
CHARACTER * 1 TRANS
DOUBLE COM PLEX A (LDA,*), B (LDB,*), C (LDC,*), D (LDD ,*),
E (LDE,*),F (LDF,*),W ORK (*)
\mathbb{N TEGER*8 IJOB,M ,N,LDA,LDB,LDC,LDD ,LDE, LD F, LW ORK,}
\mathbb{NFO}
INTEGER*8 IN ORK (*)
DOUBLE PRECISION SCALE,D F

```

\section*{F95 INTERFACE}

SU BROUTINE TGSYL (TRANS, IJOB, \(\mathbb{M}], \mathbb{N}], A,[L D A], B,[L D B], C,[L D C]\), D, [LDD ], E, [LDE],F, [LDF],SCALE,D \(\mathbb{F},\left[\begin{array}{l}\text { W ORK ], [LW ORK ], [IW ORK ], }\end{array}\right.\) [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1)::TRANS
COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::A, B, C,D,E,F
\(\mathbb{N} T E G E R:: I J O B, M, N, L D A, L D B, L D C, L D D, L D E, L D F, L W O R K\),
\(\mathbb{N}\) FO
\(\mathbb{N} T E G E R, D \mathbb{I M}\) ENSION (:) :: \(\mathbb{I N}\) ORK
REAL (8) ::SCALE,D \(\mathbb{F}\)

SU BROUTINE TGSY L_64 (TRANS, IJO B, M ], \(\mathbb{N}\) ], A, [LD A ], B , [LD B ], C , [LDC],D, [LDD ],E, [LDE],F, [LDF],SCALE,D \(\mathbb{F},\left[\begin{array}{l}\text { W ORK ], [LW ORK], }\end{array}\right.\) [ \(\mathbb{I N}\) ORK], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::TRANS
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8), D IM ENSION (:,:) ::A,B,C,D,E,F
\(\mathbb{N}\) TEGER (8) :: IDOB,M,N,LDA, LDB, LD C, LDD, LDE, LDF,
LW ORK, \(\mathbb{N} F \mathrm{O}\)
\(\mathbb{N} T E G E R(8), D \mathbb{I M} E N S I O N(:):: \mathbb{I N}\) ORK
REAL (8) :: SCALE,D \(\mathbb{F}\)

\section*{C INTERFACE}
\#include <sunperfh>
void ztgsyl(chartrans, int ijob, intm, int n, doublecom plex *a, int lda, doublecom plex *b, int ldb, doublecom plex *c, int ldc, doublecom plex *d, int ldd, doublecom plex *e, int lde, doublecom plex *f, int ldf, double *scale, double *dif, int *info);
void ztgsyl 64 (chartrans, long ij.j. long m , long n, doublecom plex *a, long lda, doublecom plex *b, long ldb, doublecom plex *c, long ldc, doublecom plex *d, long ldd, doublecom plex *e, long lde, doublecom plex *f, long ldf, double *scale, double *dif, long *info);

\section*{PURPOSE}
ztgsylsolves the generalized Sylvesterequation:
\[
\begin{aligned}
& A * R-L * B=\text { scale * C } \\
& D * R-L * E=\text { scale } * F
\end{aligned}
\]
where \(R\) and \(L\) are unknow \(n m\) by-n \(m\) atrices, ( \(A, D\) ), ( \(B\), E) and ( \(C, F\) ) are given \(m\) atrix pairs of size \(m \rightarrow b y-m, n-b y-n\) and \(m\)-by-n, respectively, w ith com plex entries. A, B, D and E are upper triangular (i.e., \((A, D)\) and \((B, E)\) in generalized Schur form ).

The solution ( \(\mathrm{R}, \mathrm{L}\) ) overw rites ( \(\mathrm{C}, \mathrm{F}\) ). 0 <= SCALE <= 1 is an output scaling factor chosen to avoid overflow .

In \(m\) atrix notation (1) is equivalent to solve \(Z x=s c a l e * b\), where \(Z\) is defined as
\[
\begin{aligned}
Z= & {\left[\operatorname{kron}(\mathbb{I n}, A)-\operatorname{kron}\left(B^{\prime}, \mathbb{I}\right)\right] } \\
& {\left[\operatorname{kron}(\mathbb{I n}, D) \operatorname{dron}\left(E^{\prime}, \mathbb{I}\right)\right] }
\end{aligned}
\]

Here Ix is the identity m atrix of size x and X 'is the conjugate transpose of X . K ron \((X, Y)\) is the K roneckerproduct betw een the \(m\) atrioes \(X\) and \(Y\).

If TRAN \(S=C\) ', \(y\) in the conjugate transposed system \(Z\) * \(y=\) scale*b is solved for, which is equivalent to solve forR and \(L\) in
\[
\begin{align*}
& A^{\prime} * R+D^{\prime *} L=\text { scale * C }  \tag{3}\\
& \text { R * } \mathrm{B}^{\prime}+\mathrm{L} \text { * } \mathrm{E} \text { '= scale * } \mathrm{F}
\end{align*}
\]

This case (TRANS = C) is used to com pute an one-norm -based estim ate of \(D\) if \([(A, D),(B, E)]\), the separation betw een the \(m\) atrix pairs \((A, D)\) and \((B, E)\), using C LA CON .

If \(\mathrm{IJO} \mathrm{B}>=1, \mathrm{CTG}\) SY L com putes a F robenius norm -based esti\(m\) ate ofD if \([(A, D),(B, E)]\). That is, the reciprocalof a low er bound on the reciprocal of the sm allest singular value of \(Z\).

This is a level-3 BLA S algorithm .

\section*{ARGUMENTS}

TRANS (input)
\(=\mathrm{N}\) ': solve the generalized sylvester equation
(1).
= C':solve the "conjugate transposed" system
(3).

IJOB (input)
Specifies whatkind of functionality to be per-
form ed. \(=0\) : solve (1) only .
\(=1\) : The functionality of 0 and 3 .
\(=2\) : The functionality of 0 and 4 .
\(=3: 0\) nly an estim ate ofD if \([(A, D),(B, E)]\) is computed. (look ahead strategy is used). \(=4: 0 \mathrm{nly}\) an estim ate of \(D\) if \([(A, D),(B, E)]\) is com puted.
(CGECON on sub-system S is used). N ot referenced if TRANS = C'.
\(M\) (input) The order of them atrices \(A\) and \(D\), and the row dim ension of the \(m\) atrioes \(C, F, R\) and \(L\).
\(N\) (input) The order of the \(m\) atriges \(B\) and \(E\), and the colum \(n\) dim ension of the \(m\) atriges \(C, F, R\) and \(L\).

A (input) The uppertriangularm atrix A.

LD A (input)
The leading dim ension of the array A.LDA >= \(\max (1, M)\).

B (input) The uppertriangularm atrix B.

LD B (input)
The leading dim ension of the aray \(\mathrm{B} . \mathrm{LD} \mathrm{B}>=\) \(\max (1, N)\).

C (input/output)
On entry, \(C\) contains the right-hand-side of the first \(m\) atrix equation in ( 1 ) or ( \(\beta\) ). On exit, if IJO B \(=0,1\) or \(2, C\) has been overw rilten by the solution R. If IJOB \(=3\) or 4 and TRANS \(=N\) ', C holds \(R\), the solution achieved during the com putation of the \(D\) ifestim ate.

LD C (input)
The leading dim ension of the array \(C . \operatorname{LDC}>=\) \(\max (1, M)\).

D (input) The uppertriangularm atrix D.

LD D (input)
The leading dim ension of the aray \(D . \operatorname{LDD}>=\) \(\max (1, M)\).

E (input) The upper triangularm atrix E.

LD E (input)
The leading dim ension of the array \(\mathrm{E} . \mathrm{LDE}>=\) \(\max (1, N)\).

F (input/output)
On entry, \(F\) contains the right-hand-side of the second \(m\) atrix equation in (1) or (3). On exit, if IJO B \(=0,1\) or \(2, F\) has been overw ritten by the solution L. If IJOB=3 or 4 and TRANS \(=N^{\prime}, F\) holds L , the solution achieved during the com putation of the \(D\) ifestim ate.

LD F (input)
The leading dim ension of the array F. LDF >= \(m a x(1, M)\).

SCALE (output)
On exitSCALE is the reciprocalof a low er bound of the reciprocal of the \(D\) if-function, ie. SCA LE is an upperbound ofD if \([(A, D),(B, E)]=\) sigm a\(m\) in ( \(Z\) ), where \(Z\) as in (2). If \(I O B=0\) orTRAN \(S=\) \(C^{\prime}\),SCALE is not referenced.

D \(\mathbb{F}\) (output)
On exitSCA LE is the reciprocalofa lower bound of the reciprocal of the \(D\) if-function, i.e. SCA LE is an upperbound ofD if \([(A, D),(B, E)]=\) sigm a\(m\) in ( \(Z\) ), where \(Z\) as in (2). If IJO \(B=0\) orTRAN \(S=\) \(C\) ',SCALE is not referenced.

W ORK (w orkspace)
If \(I J O B=0, W O R K\) is not referenced. O therw ise, on exit, if \(\mathbb{N} F O=0\) then \(W\) ORK (1) retums the optim allW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. LW ORK > = 1. If \(\mathrm{IJOB}=1\) or 2 and TRANS \(=\mathrm{N}^{\prime}\), LW ORK >= \(2 * \mathrm{M} * \mathrm{~N}\) 。

IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim alsize of theW ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.

IV ORK (w orkspace)
If \(I O B=0, \mathbb{I N} O R K\) is notreferenced.
\(\mathbb{N} F O\) (output)
=0: successfulexit
\(<0:\) If \(\mathbb{N}\) FO \(=-\) - , the \(i\)-th argum enthad an illegal value.
>0: \((A, D)\) and \((B, E)\) have com \(m\) on or very close eigenvalues.

\section*{FURTHER DETAILS}

B ased on contributions by
Bo K agstrom and PeterPorom aa, D epartm ent of Com puting
Science,
Um ea U niversity, S-901 87 Um ea, Sw eden.
[1] B . K agstrom and P . Porom aa, LA PA C K -Style A lgorithm \(s\) and Softw are
for Solving the Generalized Sylvester Equation and Estim ating the

Separation betw een RegularM atrix Pairs, ReportUM INF 9323,

D epartm ent of C om puting Scienœ, Um ea U niversity, S-901
87 Um ea,
Sw eden, D ecem ber 1993, Revised A pril 1994, A lso as LAPACK W orking

N ote 75. To appear in A CM Trans. on M ath. Softw are, \(V\) ol 22,

No 1, 1996.
[2] B .K agstrom ,A Perturbation A nalysis of the G eneralized Sylvester

Equation (AR-LB,DR-LE) =(C,F), SIAM J. M atrix Anal.

A ppl, 15 (4):1045-1060, 1994.
[3] B.K agstrom and L. W estin, G eneralized Schur M ethods w ith

C ondition Estim ators for Solving the G eneralized Sylvester

Equation, \(\mathbb{E E E}\) Transactions on A utom atic C ontrol, V ol. 34,No.7,

July 1989, pp 745-751.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztpcon -estim ate the reciprocal of the condition num ber of a packed triangular \(m\) atrix \(A\), in either the 1 -norm or the infinity-norm

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZTPCON NORM,UPLO,DIAG,N,A,RCOND,W ORK,W ORK2,\mathbb{NFO)}}\mathbf{N}\mathrm{ (N,}
CHARACTER * 1 NORM,UPLO,DIAG
DOUBLE COM PLEX A (*),W ORK (*)
\mathbb{NTEGER N,INFO}
DOUBLE PRECISION RCOND
DOUBLE PRECISION W ORK 2 (*)
SUBROUT\mathbb{NE ZTPCON_64 NORM,UPLO,DIAG,N,A,RCOND,WORK,W ORK2,}
\mathbb{NFO)}
CHARACTER * 1 NORM,UPLO,DIAG
DOUBLE COM PLEX A (*),W ORK (*)
INTEGER*8N,\mathbb{NFO}
DOUBLE PRECISION RCOND
DOUBLE PRECISION W ORK2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE TPCON \(\mathbb{N} O R M, U P L O, D \mathbb{I} G, N, A, R C O N D,[\mathbb{W} O R K],[W O R K 2]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1)::NORM,UPLO,D IA G
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::A,W ORK
\(\mathbb{N} T E G E R:: N, \mathbb{N F O}\)
REAL (8) :: RCOND
REAL (8),D IM ENSION (:) ::W ORK2

SU BROUTINE TPCON_64 \(\mathbb{N} O R M, U P L O, D \mathbb{I} G, N, A, R C O N D,[\mathbb{O} O R K],[W O R K 2]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::NORM,UPLO,D IA G
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::A,W ORK
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{N}, \mathbb{N} F O\)
REAL (8) :: RCOND
REAL (8),D IM ENSION (:) ::W ORK2

\section*{C INTERFACE}
\#include <sunperfh>
void ztpcon (charnorm, charuplo, chardiag, int n, doublecom plex *a, double *roond, int *info);
void ztpcon_64 (charnorm , char uplo, char diag, long n, doublecom plex *a, double *roond, long *info);

\section*{PURPOSE}
ztpcon estim ates the reciprocal of the condition num ber of a packed triangular matrix \(A\), in either the 1-norm orthe infinity-norm .

The norm ofA is com puted and an estim ate is obtained for norm (inv (A )), then the reciprocal of the condition num ber is com puted as
\(\operatorname{RCOND}=1 /(\operatorname{norm}(A) *\) norm (inv (A))).

\section*{ARGUMENTS}

NORM (input)
Specifies whether the 1-norm condition number or the infinity-norm condition num ber is required:
= 1 'or \(0^{\prime}\) : 1-nom ;
= I': Infinity-norm .
UPLO (input)
\(=\mathrm{U}\) : A is uppertriangular;
\(=\mathrm{L}\) ': A is low ertriangular.
D \(\mathbb{I A} G\) (input)
\(=\mathrm{N}: \mathrm{A}\) is non-unit triangular;
\(=\mathrm{U}\) ': A is unit triangular.
N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).

A (input) The upper or low er triangular m atrix A, packed colum nw ise in a linear array. The jth colum n of A is stored in the aray A as follow s: if UPLO = \(U^{\prime}, A(i+(j-1) * j 2)=A(i, j)\) for \(1<=i<=j\); if \(\mathrm{UPLO}=\mathrm{L}, \mathrm{A}(i+(j-1) *(2 n-j) / 2)=A(i, j)\) for \(j=i<=n\). IfD \(\mathbb{A} G=U\) ', the diagonalelem ents of \(A\) are not referenced and are assum ed to be 1 .

RCOND (output)
The reciprocal of the condition num ber of the \(m\) atrix \(A\), computed as RCOND \(=1 /(\) noim (A) * norm (inv (A))).
W ORK (w orkspace)
dim ension \((2 * N)\)

W ORK 2 (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{I N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztpm \(v\)-perform one of the \(m\) atrix-vectoroperations \(x\) := A *x, orx \(=A{ }^{*} x\), orx \(:=\operatorname{cong}\left(A^{\prime}\right){ }^{*} x\)

\section*{SYNOPSIS}

SU BROUTINE ZTPMV (UPLO,TRANSA,D \(\mathbb{I} G, N, A, Y, \mathbb{N} C Y)\)
CHARACTER * 1 UPLO, TRANSA, D IAG
DOUBLE COM PLEX A (*), Y (*)
\(\mathbb{N} T E G E R N, \mathbb{N C Y}\)
SU BROUTINE ZTPMV_64 (UPLO, TRANSA, D \(\mathbb{I A} G, N, A, Y, \mathbb{N C Y})\)
CHARACTER * 1 UPLO, TRANSA, D IA G
D OUBLE COM PLEX A (*), Y (*)
\(\mathbb{N} T E G E R * 8 N, \mathbb{N C Y}\)

\section*{F95 INTERFACE}

SU BROUTINE TPMV (UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{N}], A, Y,[\mathbb{N} C Y])\)
CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
COMPLEX (8),D IM ENSION (:) ::A, Y
\(\mathbb{N} T E G E R:: N, \mathbb{N C Y}\)
SU BROUTINE TPM V_64 (UPLO, [TRANSA ],D \(\mathbb{I A G}, \mathbb{N}], A, Y,[\mathbb{N} C Y])\)
CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::A, Y
\(\mathbb{N} T E G E R(8):: N, \mathbb{N C Y}\)
C INTERFACE
\#include <sunperfh>
void ztpm v (charuplo, chartransa, chardiag, int n, doublecom plex *a, doublecom plex *y, int incy);
void ztpm v_64 (charuplo, char transa, char diag, long n, doublecom plex *a, doublecom plex *y, long incy);

\section*{PURPOSE}
ztpm v perform s one of the \(m\) atrix-vector operations \(x: A * x\), or \(x:=A{ }^{*} x\), or \(x:=\) con \(\dot{g}\left(A^{\prime}\right)^{*} x\) where \(x\) is an \(n\) elem ent vectorand \(A\) is an \(n\) by \(n\) unit, ornon-unit, upper or low er triangularm atrix, supplied in packed form .

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies whetherthem atrix is an upper or low er triangularm atrix as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or G ' A is an upper triangular \(m\) atrix.

UPLO = L' or '1' A is a lower triangular \(m\) atrix.

U nchanged on exit.
TRANSA (input)
On entry, TRANSA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} \mathrm{x}:=\mathrm{A}{ }^{*} \mathrm{x}\).

TRANSA \(=\) T'ort' \(x:=A * x\).

U nchanged on exit.

TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
On entry, D IA G specifies whether ornotA is unit triangular as follow s:

D \(\mathbb{I}\) G \(=\) U 'or L ' \(A\) is assum ed to be unit tri-
angular.
\(D \mathbb{A G}=N\) 'or \(h\) ' \(A\) is notassum ed to be unit triangular.

U nchanged on exit.

N (input)
O n entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.

A (input)
( \((n *(n+1)) / 2)\). Before entry with UPLO = \(U\) ' or \(L\) ', the amay A m ustcontain the upper triangularm atrix packed sequentially, colum \(n\) by colum \(n\), so thatA (1) contains a (1, 1), A (2) and \(A(3)\) contain \(a(1,2)\) and \(a(2,2)\) respectively, and so on. Before entry w ith UPLO = L' or I', the anray A m ust contain the low er triangular m atrix packed sequentially, colum \(n\) by colum \(n\), so thatA (1) contains a (1,1),A(2) and \(A(3)\) contain \(a(2,1)\) and \(a(3,1)\) respec tively, and so on. N ote thatw hen D IA G \(=U U^{\prime}\) or \(G\) ', the diagonal elem ents of A are notreferenced, butare assum ed to be unity. U nchanged on exit.

Y (input/output)
\((1+(n-1) * a b s(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) must contain the \(n\) elem ent vectorx. O n exit, \(Y\) is overw ritten \(w\) ith the tranform ed vector \(x\).
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(. \mathbb{N} C Y<>0\). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztprfs - provide emrorbounds and backw ard error estim ates forthe solution to a system of linearequations \(w\) th a triangular packed coefficientm atrix

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZTPRFS (UPLO,TRANSA,D IAG,N,NRHS,A,B,LDB,X,LDX,}
FERR,BERR,W ORK,W ORK2,INFO)
CHARACTER * 1 UPLO,TRANSA,DIAG
DOUBLE COM PLEX A (*),B (LDB ,*),X (LDX ,*),W ORK (*)
INTEGERN,NRHS,LDB,LDX,\mathbb{NFO}
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)
SUBROUT\mathbb{NE ZTPRFS_64 (UPLO,TRANSA,D IAG,N,NRHS,A,B,LDB,X,LDX,}
FERR,BERR,W ORK,W ORK2,NFO)
CHARACTER * 1 UPLO,TRANSA,DIAG
DOUBLE COM PLEX A (*),B (LDB,*),X (LDX ,*),W ORK (*)
INTEGER*8N,NRHS,LDB,LDX,\mathbb{NFO}
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE TPRFS (UPLO, [TRANSA],D IA G ,N,NRHS,A,B, [LDB],X, [LDX ], FERR, BERR, [W ORK], [W ORK 2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
COM PLEX (8),D IM ENSION (:) ::A,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : ) : : B, X
\(\mathbb{N} T E G E R:: N, N R H S, L D B, L D X, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

SU BROUTINE TPRFS_64 (UPLO, [TRANSA],D \(\mathbb{I A G}, N, N R H S, A, B,[L D B], X\), [LDX],FERR,BERR, [W ORK], [W ORK 2], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::UPLO, TRANSA,D IA G
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::A,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) :: B , X
\(\mathbb{N} T E G E R(8):: N, N R H S, L D B, L D X, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void ztprfs (char uplo, char transa, chardiag, int n, int nihs, doublecom plex *a, doublecom plex *b, int ldb, doublecom plex *x, int ldx, double *ferr, double *bers, int*info);
void ztprfs_64 (charuplo, chartransa, char diag, long n, long nihs, doublecom plex *a, doublecom plex *b, long ldb, doublecom plex *x, long ldx, double * ferr, double *berr, long *info);

\section*{PURPOSE}
ztprfs provides errorbounds and backw ard error estim ates forthe solution to a system of linear equations \(w\) th a triangular packed coefficientm atrix.

The solution m atrix X m ustbe com puted by CTPTRS or some other \(m\) eans before entering this routine. CTPRFS does not do iterative refinem entbecause doing so cannotim prove the backw ard error.

\section*{ARGUMENTS}

\section*{UPLO (input)}
= U ': A is uppertriangular;
= LL': A is low ertriangular.

TRANSA (input)
Specifies the form of the system of equations:
\(=\mathrm{N}\) ': A * \(\mathrm{X}=\mathrm{B} \quad\) N \(\circ\) transpose)
\(=T\) ': A ** \(T\) * \(\mathrm{X}=\mathrm{B} \quad\) ( r ranspose)
= C ': A **H * X = B (C onjugate transpose)

TRANSA is defaulted to \(N\) 'forF95 \(\mathbb{I N}\) TERFACE.

D IA G (input)
\(=\mathrm{N}\) : A is non-unit triangular;
\(=\mathrm{U}:\) : A is unit triangular.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of the \(m\) atrices \(B\) and \(X\). NRH \(S>=0\).

A (input) The upper or low er triangular m atrix A, packed colum nw ise in a linear amay. The jth colum \(n\) of A is stored in the array A as follow s: if UPLO = \(U^{\prime}, A(i+(j-1) * j 2)=A(i, j)\) for \(1<=i<=j\); if UPLO = L', A ( \(\left.i+(j-1)^{*}(2 n-j) / 2\right)=A(i, j)\) for \(j=i<=n\). IfD \(\mathbb{A} G=U\) ', the diagonalelem ents of \(A\) are not referenced and are assum ed to be 1 .
\(B\) (input) The righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the aray \(B\). LD B >= \(\max (1, N)\).

X (input) The solution matrix X .
LD X (input)
The leading dim ension of the array X . LD X >= \(\max (1, \mathbb{N})\).

FERR (output)
The estim ated forw ard enrorbound for each solution vector \(X()\) ) the \(j\) th column of the solution matrix X). If XTRUE is the true solution corresponding to \(X(\mathcal{H}), \operatorname{FERR}(\mathcal{)})\) is an estim ated upperbound for the \(m\) agnitude of the largest ele\(m\) ent in ( \(\mathrm{X}(\mathcal{1})-\mathrm{XTRUE}\) ) divided by the \(m\) agnitude of the largestelem ent in \(X(\mathcal{j})\). The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slight overestim ate of the true emor.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each solution vector X ( \()\) (i.e., the sm allest relative change in any elem entofA orB thatm akes X (i) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )

W ORK2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztpsv -solve one of the system sofequations \(A * x=b\), or \(A^{*} \mathrm{x}=\mathrm{b}\), orcong \(\left(\mathrm{A}^{\prime}\right)^{*} \mathrm{x}=\mathrm{b}\)

\section*{SYNOPSIS}

SU BROUTINE ZTPSV (UPLO,TRANSA,D \(\mathbb{I} G, N, A, Y, \mathbb{N} C Y\) )
CHARACTER * 1 UPLO, TRANSA, D IA G
DOUBLE COM PLEX A ( \({ }^{*}\) ), Y (*)
\(\mathbb{I N} T E G E R N, \mathbb{N} C Y\)
SU BROUTINE ZTPSV_64 (UPLO ,TRANSA,D \(\mathbb{I} G, N, A, Y, \mathbb{N} C Y\) )
CHARACTER * 1 UPLO, TRANSA, D IA G
DOUBLE COM PLEXA (*), Y (*)
\(\mathbb{I N} T E G E R * 8 \mathrm{~N}, \mathbb{N} C Y\)

\section*{F95 INTERFACE}

SU BROUTINE TPSV (UPLO, [TRANSA ], D \(\mathbb{A} G, \mathbb{N}], A, Y,[\mathbb{N C Y}])\)
CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
COMPLEX (8),D IM ENSION (:) ::A, Y
\(\mathbb{N} T E G E R:: N, \mathbb{N C Y}\)
SU BROUTINE TPSV_64 (UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{N}], A, Y,[\mathbb{N} C Y])\)
CHARACTER (LEN=1)::UPLO,TRANSA,D IAG COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::A, Y
\(\mathbb{N} T E G E R(8):: N, \mathbb{N C Y}\)

\section*{C INTERFACE}
\#include <sunperfh>
void ztpsv (charuplo, chartransa, chardiag, int n, doublecom plex *a, doublecom plex *y, int incy);
void ztpsv_64 (charuplo, chartransa, char diag, long n, doublecom plex *a, doublecom plex *y, long incy);

\section*{PURPOSE}
ztpsv solves one of the system sofequations \(A * x=b\), or \(A^{\prime}{ }^{*} x=b, \operatorname{orcon} \dot{g}\left(A^{\prime}\right) * x=b\) where \(b\) and \(x\) are \(n\) elem ent vectors and A is an n by \(n\) unit, ornon-unit, upper or low er triangularm atrix, supplied in packed form .

N o test forsingularity or near-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.

\section*{ARGUMENTS}

UPLO (input)
On entry, UPLO specifies w hether the \(m\) atrix is an upper or low er triangularm atrix as follow s:

UPLO = U'or L ' A is an upper triangular \(m\) atrix.

UPLO = L' or \(\mathrm{I}^{\prime}\) A is a lower triangular \(m\) atrix.

U nchanged on exit.
TRANSA (input)
On entry, TRANSA specifies the equations to be solved as follow s:

TRANSA \(=N\) 'or \(h\) ' \(A * x=b\).

TRANSA \(=\) T'ort' \(A * x=b\).


U nchanged on exit.
TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.

On entry, D IA G specifies whether ornotA is unit triangular as follow s:

D \(\mathbb{A G G}=\mathrm{U}\) 'or \(\mathrm{L}^{\prime} \mathrm{A}\) is assum ed to be unit triangular.
\(D \mathbb{A} G=N\) 'or \(h\) ' \(A\) is notassumed to be unit triangular.

U nchanged on exit.

N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.
A (input)
\(\left(\left(\mathrm{n}^{*}(\mathrm{n}+1)\right) / 2\right)\). Before entry w th \(\mathrm{UPLO}=\) \(U\) ' or U ', the array A m ustcontain the upper triangularm atrix packed sequentially, colum \(n\) by colum n , so that A (1) contains a ( 1,1 ), A (2) and \(A(3)\) contain \(a(1,2)\) and \(a(2,2)\) respectively, and so on. Before entry w ith UPLO = \(\mathrm{L}^{\prime}\) or 1 ', the amay A m ust contain the low er triangular \(m\) atrix packed sequentially, colum \(n\) by colum n, so thatA (1) contains a (1,1), A (2) and \(A\) ( 3 ) contain a \((2,1)\) and a \((3,1)\) respectively, and so on. N ote thatw hen D IA G \(=\) U' or G ', the diagonal elem ents of A are notreferenced, but are assum ed to be unity. Unchanged on exit.

Y (input/output)
\((1+(n-1) * a b s(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y \mathrm{~m}\) ust contain the n elem ent righthand side vectorb. O \(n\) exit, \(Y\) is overw ritten \(w\) ith the solution vector \(x\).
\(\mathbb{N} C Y\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of. \(\mathbb{I N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ztptri-com pute the inverse of a com plex upper or low er triangularm atrix A stored in packed form at

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZTPTRI(UPLO,D IAG,N,A, INFO)}
CHARACTER * 1 UPLO,DIAG
DOUBLE COM PLEX A (*)
INTEGERN,\mathbb{NFO}
SU BROUT\mathbb{NE ZTPTRI_64 (UPLO,D IA G ,N ,A , IN FO )}
CHARACTER * 1 UPLO,DIAG
DOUBLE COM PLEX A (*)
\mathbb{NTEGER*8N,INFO}
F95 INTERFACE

```

```

CHARACTER (LEN=1)::UPLO,D IA G
COM PLEX (8),D IM ENSION (:) ::A
INTEGER::N,\mathbb{NFO}
SU BROUTINE TPTRI_64 (UPLO,D IA G ,N,A, [\mathbb{NFO ])}
CHARACTER (LEN=1)::UPLO,D IAG
COM PLEX (8),D IM ENSION (:) ::A
\mathbb{NTEGER (8)::N,\mathbb{NFO}}\mathbf{N}=\mp@code{N}

```
void ztptri(charuplo, chardiag, intn, doublecom plex *a, int*info);
void ztptri_64 (charuplo, chardiag, long n, doublecom plex *a, long *info);

\section*{PURPOSE}
ztptricom putes the inverse of a com plex upper or low er triangularm atrix A stored in packed form at.

\section*{ARGUMENTS}
```

UPLO (input)
= U ': A is upper triangular;
= L': A is low ertriangular.

```

D IA G (input)
\(=\mathrm{N}: \mathrm{A}\) is non-unit triangular;
\(=\mathrm{U}\) ': A is unit triangular.

N (input) The order of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input/output)
On entry, the upper or low ertriangularm atrix A, stored colum nw ise in a linearanay. The jth colum \(n\) ofA is stored in the array A as follow \(s\) : if \(U P L O=U ', A(i+(j-1) * j 2)=A(i, 7)\) for \(1<=\mathrm{i}<=\dot{j}\) ifUPLO = L', A (i+ \((\mathfrak{j} 1)^{*}((2 * \mathrm{n}-\mathrm{j} / 2)=\) A \((i, 1)\) for \(j<=i<=n\). See below for further details. On exit, the (triangular) inverse of the original \(m\) atrix, in the sam e packed storage format.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
<0: if \(\mathbb{I N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue
\(>0\) : if \(\mathbb{N F O}=\mathrm{i}, \mathrm{A}(i, i)\) is exactly zero. The triangular \(m\) atrix is singular and its inverse can notbe com puted.

\section*{FURTHER DETAILS}

A triangularm atrix A can be transferred to packed storage using one of the follow ing program segm ents:
\(\mathrm{UPLO}=\mathrm{U} ': \quad \mathrm{UPLO}=\mathrm{L}^{\prime}:\)
J \(=1\)
DO \(2 \mathrm{~J}=1\), N
\(\pi=1\)
DO \(2 \mathrm{~J}=1\), N
D○ \(1 \mathrm{I}=1\), J
DO \(1 \mathrm{I}=\mathrm{J}, \mathrm{N}\)
\(A(J C+I-1)=A(I, J) \quad A(J C+I-J)=\)
A (I, N)
\(\begin{array}{ll}1 & \text { CONTINUE } \\ \mathbb{C}=\mathrm{C}+\mathrm{J} & \mathbb{C O N T I N U E} \\ & \mathbb{C}=\mathrm{J}+\mathrm{N}-\mathrm{J}+\end{array}\)
1
2 CONTINUE \(\quad 2\) CONTINUE

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztptrs - solve a triangularsystem of the form \(A * X=B\), \(\mathrm{A} * * \mathrm{~T} * \mathrm{X}=\mathrm{B}\),orA \({ }^{*}{ }^{*} \mathrm{H} * \mathrm{X}=\mathrm{B}\),

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZTPTRS (UPLO,TRANSA,D IAG,N,NRHS,A,B,LDB,INFO)}
CHARACTER * 1UPLO,TRANSA,DIAG
DOUBLE COM PLEXA (*),B (LDB,*)
INTEGER N,NRHS,LDB,INFO
SU BROUT\mathbb{NE ZTPTRS_64(UPLO,TRANSA,D IAG,N,NRHS,A ,B,LD B, INFO)}
CHARACTER * 1 UPLO,TRANSA,D IAG
DOUBLE COM PLEXA (*),B (LDB,*)
INTEGER*8N,NRHS,LDB,INFO

```
F95 INTERFACE
    SU BROUTINE TPTRS (UPLO, TRANSA, D \(\mathbb{I A G}, N, N R H S, A, B,[L D B],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::A
    COM PLEX (8), D IM ENSION (:,:) :: B
    \(\mathbb{N}\) TEGER ::N,NRHS,LDB, \(\mathbb{N}\) FO
    SU BROUTINE TPTRS_64 (UPLO,TRANSA,D \(\mathbb{I A G}, N, N R H S, A, B,[L D B],[\mathbb{N} F O])\)
    CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
    COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::A
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::B
    \(\mathbb{N}\) TEGER (8) :: N,NRHS,LD B, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void ztptrs (charuplo, chartransa, chardiag, int \(n\), int nrhs, doublecom plex *a, doublecom plex *b, int ldb, int*info);
void ztptrs_64 (charuple, chartransa, char diag, long n, long nihs, doublecom plex *a, doublecom plex *b, long lalb, long *info);

\section*{PURPOSE}
ztptes solves a triangular system of the form
where A is a triangularm atrix of orderN stored in packed
form at, and B is an N boy-NRH S m atrix. A check ism ade to verify that \(A\) is nonsingular.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}:\) A is upper triangular;
\(=\mathbb{L}^{\prime}: A\) is low er triangular.

TRANSA (input)
Specifies the form of the system of equations:
\(=N: A * X=B \quad\) ( \(o\) transpose)
\(=T\) ': \(A * * T * X=B \quad\) ( ranspose)
\(=C: A * * H * X=B \quad\) (C onjugate transpose)

D IA G (input)
\(=\mathrm{N}\) : A is non-unit triangular;
\(=U\) : A is unittriangular.

N (input) The order of them atrix \(\mathrm{A} . \mathrm{N}>=0\).

NRHS (input)
The num ber of righthand sides, ie., the num ber of colum ns of the m atrix B. NRHS \(>=0\).

A (input) The upper or low er triangular matrix A, packed colum nw ise in a linear array. The jth colum n of A is stored in the array A as follow s: ifUPLO = \(U ', A(i+(j-1) * j 2)=A(i, j)\) for \(1<=i<=j\) if \(\mathrm{UPLO}=\mathrm{L}, \mathrm{A}(i+(j 1) \star(2 \star \mathrm{n}-\mathrm{j} / 2)=A(i, j)\) for \(j=i<=n\).

B (input/output)
On entry, the righthand side \(m\) atrix B. On exit, if \(\mathbb{N} F O=0\), the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array B . LD B >= \(\max (1, N)\).
\(\mathbb{I N} F O\) (output)
= 0 : successfinlexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the \(i\)-th diagonalelem ent of \(A\) is zero, indicating that the \(m\) atrix is singular and the solutions \(X\) have notbeen com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

ztrans - transpose and scale source m atrix

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZTRANS (PLACE,SCALE,SOURCE,M ,N,DEST)}
CHARACTER * 1 PLACE
DOUBLE COM PLEX SCALE
DOUBLE COM PLEX SOURCE (*),DEST (*)
INTEGERM,N
SUBROUT\mathbb{NE ZTRANS_64(PLACE,SCALE,SOURCE,M,N,DEST)}
CHARACTER * 1 PLACE
DOUBLE COM PLEX SCALE
DOUBLE COM PLEX SOURCE (*),DEST (*)
INTEGER*8M ,N
F95 INTERFACE
SUBROUTINE TRANS ([PLACE],SCALE,SOURCE,M,N, DEST])
CHARACTER (LEN=1) ::PLACE
COMPLEX (8) ::SCALE
COM PLEX (8),D IM ENSION (:) ::SOURCE,DEST
\mathbb{NTEGER ::M,N}
SU BROUT\mathbb{NE TRANS_64(PLACE],SCALE,SOURCE,M ,N, DDET ])}
CHARACTER (LEN=1)::PLACE
COM PLEX (8) ::SCALE
COM PLEX (8),D IM ENSION (:) ::SOURCE,DEST
\mathbb{NTEGER (8) ::M ,N}

```

\section*{C INTERFACE}
\#include < sunperfh>
void ztrans (charplace, doublecom plex *scale, doublecom plex
*source, intm, intn, doublecom plex *dest);
void ztrans_64 (charplace, doublecom plex *scale, doublecom plex *source, long m, long n, doublecomplex *dest);

\section*{PURPOSE}
ztrans scales and transposes the source \(m\) atrix. The \(2 \times \mathrm{x} 1\) result is w ritten into SO U RCE when PLACE = 'I'or 'i', and DEST when PLACE = 0 'or \(\mathrm{b}^{\prime}\) '.

PLACE = 'I'or 'i': SOURCE = SCALE *SOURCE'
PLACE = O'orb':DEST = SCALE * SOURCE'

\section*{ARGUMENTS}

PLACE (input)
Type of transpose. 'I'or i'for in-place, \(\mathrm{O}^{\prime}\) or b'for out-of-place. 'T' is default.

SCALE (input)
Scale factor on the SO U RCE m atrix.

SO URCE (input/output)
on input. A ray of \((\mathbb{N}, \mathrm{M})\) on output if in-place transpose.

M (input)
\(N\) um ber of row \(s\) in the SO U RCE \(m\) atrix on input.

N (input)
N um ber of colum ns in the SOU RCE m atrix on input.
DEST (output)
Scaled and transposed SOURCE m atrix if out-ofplace transpose. N ot referenced if in-place transpose.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztroon -estim ate the reciprocal of the condition num ber of a triangular \(m\) atrix \(A\), in either the 1 -norm or the infinity-norm

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZTRCON NORM,UPLO,DIAG,N,A,LDA,RCOND,W ORK,W ORK2,}
INFO)
CHARACTER * 1NORM,UPLO,DIAG
DOUBLE COM PLEX A (LDA,*),W ORK (*)
\mathbb{NTEGERN,LDA,}\mathbb{N}FO
DOUBLE PRECISION RCOND
DOUBLE PRECISION W ORK 2 (*)
SUBROUT\mathbb{NE ZTRCON_64 NORM,UPLO,DIAG,N,A,LDA,RCOND,W ORK,W ORK2,}
\mathbb{NFO)}
CHARACTER * 1NORM,UPLO,DIAG
DOUBLE COM PLEX A (LDA,*),W ORK (*)
INTEGER*8N,LDA,INFO
DOUBLE PRECISION RCOND
DOUBLE PRECISION W ORK 2 (*)

```

\section*{F95 INTERFACE}

SU BROUTINE TRCON \(\mathbb{N} O R M, U P L O, D \mathbb{I} G, N, A,[L D A], R C O N D,[W\) ORK ], [W ORK 2], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1)::NORM,UPLO,DIAG
COM PLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: N, L D A, \mathbb{N} F O\)

REAL (8) :: RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ORK2

SU BROUTINE TRCON_64 \(\mathbb{N} O R M, U P L O, D \mathbb{I} G, N, A,[L D A], R C O N D,[W O R K]\), [W ORK2], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::NORM,UPLO,DIAG
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R(8):: N, L D A, \mathbb{N} F O\)
REAL (8) :: RCOND
REAL (8),D \(\mathbb{M}\) ENSION (:) ::W ORK 2

\section*{C INTERFACE}
\#include <sunperfh>
void ztroon (charnorm , charuplo, chardiag, int n, doublecom plex *a, int lda, double *rcond, int *info);
void ztroon_64 (charnom , char uplo, char diag, long n, doublecom plex *a, long lda, double *rcond, long *info);

\section*{PURPOSE}
ztrcon estim ates the reciprocal of the condition num ber of a triangular \(m\) atrix \(A\), in either the 1 -norm or the infinity norm.

The norm ofA is com puted and an estim ate is obtained for norm (inv (A )), then the reciprocal of the condition num ber is com puted as
```

RCOND = 1 / (nom (A ) * nom (inv (A)) ).

```

\section*{ARGUMENTS}
```

NORM (input)
Specifies w hether the 1-nom condition number or
the infinity-norm condition num ber is required:
= I 'or O ': 1-nom ;
= I':}\quad\mathrm{ Infinity-norm .
UPLO (input)
= U ': A is uppertriangular;
= LL':A is low ertriangular.

```
D IA G (input)
\(=N^{\prime}\) : A is non-unit triangular;
\(=\mathrm{U}\) : A is unit triangular.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).

A (input) The triangularm atrix A . If \(\mathrm{PLO}=\mathrm{U}\) ', the leading N -by -N upper triangularpart of the amay A contains the upper triangular \(m\) atrix, and the strictly low ertriangular partofA is not referenced. IfUPLO = L', the leading N -by-N lower triangular part of the array A contains the low er triangularm atrix, and the strictly upper triangular part ofA is not referenced. IfD \(\mathbb{I A} G=U\) ', the diagonalelem ents ofA are also not referenced and are assum ed to be 1 .
LDA (input)
The leading dim ension of the array A. LDA >= \(\max (1, N)\).

RCOND (output)
The reciprocal of the condition num ber of the \(m\) atrix \(A\), computed as RCOND \(=1 /(\) norm (A) * norm (inv (A))).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension \((\mathbb{N})\)
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ztrevc - com pute som e or all of the right and/or lefteigen-
vectors of a com plex upper triangularm atrix \(T\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZTREVC (SDE,HOW MNY,SELECT,N,T,LDT,VL,LDVL,VR,}
LDVR,MM,M,WORK,RWORK,INFO)
CHARACTER * 1SDDE,HOW M NY
DOUBLE COM PLEX T (LDT,*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
INTEGER N,LDT,LDVL,LDVR,MM,M,\mathbb{NFO}
LOG ICAL SELECT (*)
DOUBLE PRECISION RW ORK (*)
SU BROUT\mathbb{NE ZTREVC_64 (SDE ,HOW MNY,SELECT,N,T,LDT,VL,LDVL,VR,}
LDVR,MM,M,W ORK,RW ORK,\mathbb{NFO)}
CHARACTER * 1SDE,HOWMNY
DOUBLE COM PLEX T (LDT,*),VL (LDVL,*),VR (LDVR,*),W ORK (*)
\mathbb{NTEGER*8N,LDT,LDVL,LDVR,MM,M,NNFO}
LOG ICAL*8 SELECT (*)
DOUBLE PRECISION RW ORK (*)

```

\section*{F95 INTERFACE}

SU BROUTINE TREVC (SDE,HOW M NY, SELECT, \(\mathbb{N}], T,[L D T], V L,[L D V L], V R\), [LDVR],MM,M,[WORK], RW ORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::SDE,HOW M NY
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::T,VL,VR
\(\mathbb{N}\) TEGER ::N,LDT,LDVL,LDVR,MM,M, \(\mathbb{N} F O\)

LOG ICAL,D IM ENSION (:) :: SELECT
REAL (8),D \(\mathbb{M}\) ENSION (:) ::RW ORK
SU BROUTINE TREVC_64 (SIDE,HOW MNY,SELECT, \(\mathbb{N}], T,[L D T], V L,[L D V L]\), VR, [LDVR], MM, M, [W ORK], RW ORK], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) ::SDE,HOW MNY
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8),D \(\mathbb{I}\) ENSION (:,:) ::T,VL,VR
\(\mathbb{N} T E G E R(8):: N, L D T, L D V L, L D V R, M M, M, \mathbb{N F O}\)
LOG ICAL (8), D IM ENSION (:) :: SELECT
REAL (8),D \(\mathbb{M}\) ENSION (:) ::RW ORK

\section*{C INTERFACE}
\#include <sunperfh>
void ztrevc (char side, char how my, int *select, int n, doublecom plex *t, int ldt, doublecom plex *vl, int ldvl, doublecom plex *vr, int ldvr, intm \(m\), int *m, int*info);
void ztrevc_64 (charside, charhow m ny, long *select, long n, doublecom plex *t, long ldt, doublecom plex *vl, long ldvl, doublecom plex *vr, long ldvr, long m m, long *m, long *info);

\section*{PURPOSE}
ztrevc com putes som e or all of the right and/or left eigenvectors of a com plex upper triangularm atrix T .

The righteigenvectorx and the left eigenvector \(y\) of \(T\) comesponding to an eigenvalue \(w\) are defined by:
\[
\mathrm{T}^{*} \mathrm{x}=\mathrm{w} * \mathrm{x}, \quad \mathrm{y} * \mathrm{~T}=\mathrm{w}^{*} \mathrm{y}^{\prime}
\]
where y'denotes the conjugate transpose of the vectory.

If alleigenvectors are requested, the routine m ay either retum the \(m\) atrices \(X\) and/or \(Y\) of rightor lefteigenvectors of \(T\), or the products \(Q * X\) and/or \(Q * Y\), where \(Q\) is an input unitary
\(m\) atrix. If \(T\) w as obtained from the Schur factorization of an original matrix \(A=Q * T * Q\) ', then \(Q * X\) and \(Q * Y\) are the \(m\) atrices of right or lefteigenvectors of \(A\).

\section*{ARGUMENTS}

STDE (input)
= R ': com pute righteigenvectors only;
= L ': com pute lefteigenvectors only;
= B ': com pute both right and lefteigenvectors.

HOW M NY (input)
= A ': com pute all right and/or left eigenvec-
tors;
= B ': com pute all right and/or left eigenvectors, and backtransform them using the input \(m\) atrices supplied in VR and/orV L; = S ': com pute selected right and/or left eigenvectors, specified by the logicalamay SELEC T .

SELECT (input/output)
If HOW M NY = S', SELECT specifies the eigenvectors to be com puted. IfHOW M NY = A 'or B', SELECT is not referenced. To select the eigenvector corresponding to the jth eigenvalue, SELECT (j) mustbe setto .TRUE ..

N (input) The order of the m atrix \(\mathrm{T} . \mathrm{N}>=0\).

T (input/output)
The upper triangularm atrix T . T ism odified, but restored on exit.

LD T (input)
The leading dim ension of the aray T. LD T >= \(\max (1, N)\).

VL (input/output)
On entry, ifSDE \(=\mathrm{L}\) 'or B 'and HOW MNY = B',
VL must contain an \(N\) boy N matrix Q (usually the unitary matrix \(Q\) of Schur vectors retumed by CHSEQR). On exit, if \(S \mathbb{D} E=\) L'or B',VL contains: if HOW MNY = A', the matrix Y of left eigenvectors of T ; L is low ertriangular. The ith column VL (i) of VL is the eigenvector corresponding to \(T(i, i)\). if HOW MNY \(=B\) ', the \(m\) atrix \(Q\) *Y ; if H OW M N Y = S', the lefteigenvectors of T specified by SELEC \(T\), stored consecutively in the colum ns of VL, in the same order as their eigenvalues. If \(S \mathbb{D} E=R\) ', \(V L\) is not referenced.

LDVL (input)
The leading dim ension of the array VL. LDVL >= \(\max (1, \mathbb{N})\) if \(\mathrm{S} \mathbb{D} \mathrm{E}=\mathrm{L}\) 'or B'; LDVL >= 1 otherw ise.

VR (input/output)
On entry, if \(S \mathbb{D} E=R\) 'or \(B\) 'and HOW M NY = B', VR must contain an N -by -N m atrix Q (usually the unitary matrix \(Q\) of Schur vectors retumed by CHSEQR). On exit, if \(S \mathbb{D} E=R\) 'or \(B\) ', VR contains: ifHOW MNY = A', the matrix \(X\) of right eigenvectors of \(T\); \(V R\) is uppertriangular. The ith column VR (i) of VR is the eigenvector corresponding to \(\mathrm{T}(\mathrm{i}, \mathrm{i})\). if HOW M NY = B ', the \(m\) atrix \(Q * X\); if HOW M NY \(=S\) ', the right eigenvectors of \(T\) specified by SELECT, stored consecutively in the colum ns of \(V R\), in the sam e order as their eigenvalues. If \(\mathrm{SD} E=\mathrm{L}\) ', VR is not referenced.

LDVR (input)
The leading dim ension of the aray VR. LDVR >= \(\max (1, N)\) if \(S \mathbb{D}=R\) 'or \(B\) '; LDVR \(>=1\) otherw ise.

M M (input)
The num ber of colum ns in the arrays VL and/or VR. M M >= M .

M (output)
The num ber of colum ns in the arrays VL and/or VR actually used to store the eigenvectors. If HOW M NY = A 'or B',M is set to N. Each selected eigenvector occupies one colum \(n\).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )

RW ORK (w orkspace)
dim ension \(\mathbb{N}\) )
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{FURTHER DETAILS}

The algorithm used in this program is basically backw ard (forw ard) substitution, \(w\) th scaling to \(m\) ake the the code robustagainstpossible overflow .

Each eigenvector is nom alized so that the elem ent of largest \(m\) agnitude has \(m\) agnitude 1 ; here the \(m\) agnitude of a com plex num ber \((x, y)\) is taken to be \(|x|+|y|\).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztrexc - reorder the Schur factorization of a com plex \(m\) atrix
\(\mathrm{A}=\mathrm{Q} * \mathrm{~T} * \mathrm{Q} * * \mathrm{H}\), so that the diagonalelem entof T w th row index \(\mathbb{F S T}\) ism oved to row \(\mathbb{L S T}\)

\section*{SYNOPSIS}

```

CHARACTER * 1 COMPQ
DOUBLE COM PLEX T (LDT,*),Q (LDQ,*)
\mathbb{NTEGERN,LDT,LDQ,\mathbb{FST,}|ST,INFO}

```

```

CHARACTER * 1 COMPQ
DOUBLE COM PLEX T (LDT,*),Q (LDQ,*)
\mathbb{NTEGER*8 N,LDT,LDQ,\mathbb{FST},|ST,INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE TREXC (COM PQ, \(\mathbb{N}], T,[L D T], Q,[L D Q], \mathbb{F} S T, \mathbb{L} S T,[\mathbb{N} F O])\)
CHARACTER (LEN=1) ::COMPQ
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::T, Q
\(\mathbb{N}\) TEGER ::N,LDT,LDQ, \(\mathbb{F S T}, \mathbb{L S T}, \mathbb{N} F O\)
SU BROUTINE TREXC_64 (COMPQ, \(\mathbb{N}], T,[L D T], Q,[L D Q], \mathbb{F S T}, \mathbb{L} S T,[\mathbb{N F O}])\)
CHARACTER (LEN=1) ::COMPQ
COM PLEX (8), D IM ENSION (: : : : : T, Q
\(\mathbb{N}\) TEGER (8) ::N,LD T,LDQ, \(\mathbb{F S T}, \mathbb{L S T}, \mathbb{N} F O\)
\#include < sunperfh>
void ztrexc (char com pq, intn, doublecom plex *t, int ldt, doublecom plex *q, int ldq, int ifst, int ilst, int *info);
void ztrexc_64 (charcom pq, long n, doublecom plex *t, long ldt, doublecom plex *q, long ldq, long ifst, long ilst, long *info);

\section*{PURPOSE}
ztrexc reorders the Schur factorization of a com plex \(m\) atrix \(\mathrm{A}=\mathrm{Q} * \mathrm{~T} * \mathrm{Q} * * \mathrm{H}\), so that the diagonalelem entof T w th row index \(\mathbb{F} S T\) ism oved to row \(\Pi\) ST.
The Schurform \(T\) is reordered by a unitary sim ilarity transform ation \(Z * * H * T * Z\), and optionally the \(m\) atrix \(Q\) of Schurvectors is updated by postm ultplying itw ith Z .

\section*{ARGUMENTS}

COMPQ (input)
\(=\mathrm{V}\) : update the matrix Q of Schurvectors;
\(=\mathrm{N}\) ': do notupdate Q .
N (input) The order of the m atrix \(\mathrm{T} \cdot \mathrm{N}>=0\).

T (input/output)
On entry, the upper triangularm atrix T. On exit, the reordered upper triangularm atrix.

LD T (input)
The leading dim ension of the array \(T\). LD \(T>=\) \(\max (1, N)\).

Q (input) \(O n\) entry, if \(C O M P Q=V\) ', them atrix \(Q\) of Schur
vectors. On exit, if COMPQ = V', Q has been postm ultiplied by the unitary transform ation \(m\) atrix \(Z\) which reorders \(T\). IfC \(O M P Q=N^{\prime}, Q\) is not referenced.

LD Q (input)
The leading dim ension of the array \(Q\). LDQ >= \(\max (1, \mathbb{N})\).

FST (input)
Specify the reordering of the diagonalelem ents of

T: The elem ent with row index \(\mathbb{F} S T\) ism oved to row UST by a sequence of transpositions betw een adjłcent elem ents. \(1<=\mathbb{F} S T<=\mathrm{N} ; 1<=\mathbb{L} S T<=\) N .

UST (input)
See the description of IFST .
\(\mathbb{N} F O\) (output)
= 0 : successfinlexit
\(<0\) : if \(\mathbb{N F O}=-i\), the \(i\)-th argum ent had an illegal value

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztrm \(m\)-perform one of them atrix \(m\) atrix operations \(B:=\) alpha*op (A )*B, orB :=alpha*B *op (A) where alpha is a scalar, \(B\) is an \(m\) by \(n m\) atrix, \(A\) is a unit, or non-unit, upper or low er triangularm atrix and op (A) is one of op (


\section*{SYNOPSIS}
```

SUBROUTINE ZTRMM (S\mathbb{DE,UPLO,TRANSA,DIAG,M,N,ALPHA,A,LDA,B,}

```
    LD B )
CHARACTER * 1 SDE E, UPLO,TRANSA, D IAG
DOUBLE COM PLEX ALPHA
DOUBLE COM PLEX A (LDA,*), B (LDB,*)
\(\mathbb{N} T E G E R M, N, L D A, L D B\)
SU BROUTINE ZTRMM_64 (STDE, UPLO,TRANSA,D IA G,M,N,ALPHA,A,LDA,B,
    LD B)
CHARACTER * 1 SDE, UPLO,TRANSA,D IAG
D OUBLE COM PLEX ALPHA
DOUBLE COM PLEXA (LDA,*), B (LDB, \(\left.{ }^{*}\right)\)
\(\mathbb{N}\) TEGER*8M,N,LDA,LDB

\section*{F95 INTERFACE}

SU BROUTINE TRMM (SDE E, UPLO, [TRANSA ],D IA G, \(\mathbb{M}], \mathbb{N}], A L P H A, A,[L D A]\), B, [LD B])

CHARACTER (LEN=1)::SDE,UPLO,TRANSA,D IA G
COM PLEX (8) ::ALPHA
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A,B
\(\mathbb{N} T E G E R:: M, N, L D A, L D B\)

SU BROUTINE TRM M _64 (S \(\mathbb{D} E, U P L O,[T R A N S A], D \mathbb{I} G, \mathbb{M}], \mathbb{N}], A L P H A, A\), [LD A ], B, [LD B])

CHARACTER (LEN=1)::SDE,UPLO,TRANSA,D IA G
COMPLEX (8) ::ALPHA
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) :: A, B
\(\mathbb{N}\) TEGER (8) ::M , N ,LDA,LDB

\section*{C INTERFACE}
\#include <sunperfh>
void ztrm m (char side, char uplo, char transa, chardiag, int m , intn, doublecom plex *alpha, doublecom plex *a, int lda, doublecom plex *b, int ldb);
void ztrm m _64 (char side, charuplo, chartransa, char diag, long m , long n , doublecom plex *alpha, doublecom plex *a, long lda, doublecom plex *b, long ldb);

\section*{PURPOSE}
ztrm \(m\) perform sone of the \(m\) atrix \(m\) atrix operations \(B:=\) alpha*op ( A ) B , or \(\mathrm{B}:=\) alpha* B *op ( A ) where alpha is a scalar, \(B\) is an \(m\) by \(n m\) atrix, \(A\) is a unit, or non-unit, upper or low er triangularm atrix and op (A ) is one of op ( \(\mathrm{A})=\mathrm{A} \operatorname{orop}(\mathrm{A})=\mathrm{A}^{\prime} \operatorname{orop}(\mathrm{A})=\operatorname{conjg}\left(\mathrm{A}^{\prime}\right)\)

\section*{ARGUMENTS}

STDE (input)
On entry, SDE specifies whether op (A) m ultiplies \(B\) from the leftor right as follow \(s\) :

SDEE L'or I' B := alpha*op (A )*B.


U nchanged on exit.
UPLO (input)
On entry, UPLO specifies whether them atrix A is an upper or low er triangularm atrix as follow \(s\) :

UPLO = U'or 4 ' A is an upper triangular \(m\) atrix.

UPLO = L' or 1' A is a lower triangular
m atrix.

U nchanged on exit.

TRANSA (input)
On entry, TRAN SA specifies the form of op (A ) to be used in the \(m\) atrix \(m\) ultiplication as follow \(s\) :

TRANSA \(=N^{\prime}\) 'or \(h^{\prime}\) op (A) \()=A\).

TRANSA \(=\) T'ort'op \((A)=A\).

TRANSA \(=C^{\prime}\) or \(C^{\prime} o p(A)=c o n \dot{g}\left(A^{\prime}\right)\).

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.
D IA G (input)
O n entry, D IA G specifies w hether ornotA is unit triangular as follow s:
\(D \mathbb{A G}=U\) 'or \(U^{\prime} A\) is assum ed to be unit triangular.

D IA G \(=N\) 'or \(h^{\prime} A\) is notassum ed to be unit triangular.

U nchanged on exit.

M (input)
O \(n\) entry, \(M\) specifies the num ber of row \(s\) of \(B . M\) \(>=0\). U nchanged on exit.

N (input)
O n entry, \(N\) specifies the num ber of colum ns of \(B\). \(\mathrm{N}>=0\). U nchanged on exit.

ALPHA (input)
On entry, A LPH A specifies the scalar alpha.W hen alpha is zero then \(A\) is notreferenced and \(B\) need notbe setbefore entry. U nchanged on exit.

A (input)
COM PLEX *16 aray ofD \(\mathbb{I M}\) ENSION (LDA,k), where \(k\) is \(m\) when \(S \mathbb{D E}=\mathbb{L}\) 'or I' and is \(n\) when \(S \mathbb{D E}=\mathrm{R}\) 'or \(\mathrm{r}^{\prime}\).

Before entry w th UPLO = U'or L', the leading \(k\) by \(k\) upper triangularpart of the aray \(A\) \(m\) ustcontain the upper triangularm atrix and the
strictly low ertriangularpartofA is not referenced.

Before entry with UPLO = L'or 1', the leading \(k\) by \(k\) low er triangularpart of the array A m ust contain the low er triangularm atrix and the strictly uppertriangularpartofA is not referenced.

N ote thatw hen \(\mathrm{D} \mathbb{I A} G=\mathrm{U}\) ' or U ', the diagonal elem ents of A are not referenced either, butare assum ed to be unity.

U nchanged on exit.
LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. W hen SDE \(=\) L'or 1 'then LD \(A>=\max (1, M)\), when \(S \mathbb{D} E\) \(=R^{\prime}\) or 'r'then LD A \(>=\max (1, N)\). U nchanged on exit.

B (input/output)
COM PLEX *16 aray ofD \(\mathbb{I M}\) ENSION (LD B,n). Before
entry, the leading \(M\) by \(N\) partof the array \(B\) m ust contain the \(m\) atrix \(B\), and on exit is overw rilten by the transform ed \(m\) atrix.

LD B (input)
O n entry, LD B specifies the firstdim ension of B as declared in the calling subprogram. LD B m ust be at leastm ax ( \(1, \mathrm{M}\) ). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

ztm v -perform one of them atrix-vectoroperations x :=

```
\(A{ }^{*} \mathrm{x}\), or \(\mathrm{x}:=\mathrm{A}{ }^{*} \mathrm{x}\), orx \(:=\operatorname{cong}\left(\mathrm{A}^{\prime}\right){ }^{*} \mathrm{x}\)

\section*{SYNOPSIS}
```

SUBROUTINE ZTRMV (UPLO,TRANSA,D IAG,N,A,LDA,Y, INCY)
CHARACTER * 1UPLO,TRANSA,DIAG
DOUBLE COM PLEX A (LDA,*),Y (*)
INTEGERN,LDA,}\mathbb{N}C
SU BROUT\mathbb{NE ZTRM V_64 (UPLO,TRANSA,D IAG ,N,A ,LDA,Y , INCY)}
CHARACTER * 1UPLO,TRANSA,DIAG
DOUBLE COM PLEXA (LDA,*),Y (*)
\mathbb{NTEGER*8N,LDA,}\mathbb{N}CY

```

\section*{F95 INTERFACE}

SU BROUTINE TRMV (UPLO, [TRANSA],D \(\mathbb{I A G}, \mathbb{N}], A,[L D A], Y,[\mathbb{N} C Y])\)
CHARACTER (LEN=1) ::UPLO,TRANSA,D IAG
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::Y
COM PLEX (8), D \(\mathbb{I M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER ::N,LDA, \(\mathbb{N} C Y\)

SU BROU TINE TRM V_64 (UPLO, [TRANSA ], D \(\mathbb{I A} G, \mathbb{N}], A,[L D A], Y,[\mathbb{N} C Y])\)
CHARACTER (LEN=1)::UPLO,TRANSA,D IA G
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::Y
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER (8) ::N,LDA, \(\mathbb{N} C Y\)

\section*{C INTERFACE}
\#include <sunperfh>
void ztrm v (charuplo, chartransa, chardiag, int n, doublecom plex *a, int lda, doublecom plex *y, int incy);
void ztrm v_64 (charuplo, chartransa, char diag, long n, doublecom plex *a, long lda, doublecom plex *y, long incy);

\section*{PURPOSE}
\(z\) trm \(v\) perform s one of the \(m\) atrix-vector operations \(x:=A{ }^{*} x\), or \(x:=A{ }^{*} x\), or \(x:=\) con \(\dot{g}\left(A^{\prime}\right) \star x w h e r e x\) is an \(n\) elem ent vectorand \(A\) is an \(n\) by \(n\) unit, ornon-unit, upper or low er triangularm atrix.

\section*{ARGUMENTS}

UPLO (input)
O n entry, UPLO specifies w hether the \(m\) atrix is an upper or low er triangularm atrix as follow \(s\) :
\(\mathrm{UPLO}=\mathrm{U}\) 'or G ' \(A\) is an upper triangular m atrix.
\(\mathrm{UPLO}=\mathrm{L}\) ' or \(\mathrm{I}^{\prime}\) A is a lower triangular \(m\) atrix.

U nchanged on exit.

TRANSA (input)
On entry, TRAN SA specifies the operation to be perform ed as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} x:=A * x\).

TRANSA \(=\) T'ort' \(x:=A{ }^{*} x\).

TRANSA \(=\) C'ort' \(\mathrm{x}:=\operatorname{conjg}\left(A^{\prime}\right) \star \mathrm{x}\).

U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)

On entry, D IA G specifies whether ornotA is unit triangular as follow s:

D \(\mathbb{A G G}=\mathrm{U}\) 'or \(\mathrm{L}^{\prime} \mathrm{A}\) is assum ed to be unit triangular.
\(D \mathbb{A} G=N\) 'or \(h\) ' \(A\) is notassumed to be unit triangular.

U nchanged on exit.

N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.
A (input)
Before entry w ith UPLO = U 'or L ', the leading n by n upper triangularpart of the array A m ust contain the upper triangular \(m\) atrix and the strictly low ertriangularpartofA is not referenced. Before entry w ith UPLO = L'or I', the leading \(n\) by \(n\) low er triangularpart of the anray A \(m\) ust contain the low ertriangularm atrix and the strictly uppertriangularpartofA is not referenced. N ote thatw hen D IAG \(=\mathrm{U}\) ' or L ', the diagonal elem ents of A are not referenced either, but are assum ed to be unity. U nchanged on exit.

LDA (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program . LD A \(>=\) \(m a x(1, n)\). U nchanged on exit.

Y (input/output)
\((1+(n-1) * \operatorname{abs}(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) must contain the \(n\) elem ent vectorx. On exit, \(Y\) is overw rilten \(w\) th the tranform ed vector \(x\).
\(\mathbb{N C Y}\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents ofY. \(\mathbb{N C Y}\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztrifs - provide errorbounds and backw ard enror estim ates for the solution to a system of linear equations w ith a triangular coefficientm atrix

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZTRRFS (UPLO,TRANSA,D IAG,N,NRHS,A,LDA,B,LDB,X,}
LDX,FERR,BERR,W ORK,W ORK2,INFO)
CHARACTER * 1UPLO,TRANSA,DIAG
DOUBLE COM PLEX A (LDA,*),B (LDB,*),X (LDX,*),W ORK (*)
INTEGERN,NRHS,LDA,LDB,LDX,\mathbb{NFO}
DOUBLE PRECISION FERR (*),BERR (*),W ORK 2 (*)
SUBROUT\mathbb{NE ZTRRFS_64 (UPLO,TRANSA,D IAG,N,NRHS,A,LDA,B,LDB,X,}
LDX,FERR,BERR,W ORK,W ORK2,INFO)

```
CHARACTER * 1 UPLO, TRANSA, DIAG
DOUBLE COM PLEXA (LDA, *), B (LDB, \(\left.{ }^{\star}\right), \mathrm{X}(\mathbb{L D} X, \star), \mathrm{W} O R K\left({ }^{\star}\right)\)
\(\mathbb{N} T E G E R * 8 N, N R H S, L D A, L D B, L D X, \mathbb{N} F O\)
DOUBLE PRECISION FERR (*), BERR (*), W ORK 2 (*)

\section*{F95 INTERFACE}

SU BROUTINE TRRFS (UPLO, [TRANSA ], D IA G,N,NRHS,A, [LDA ], B, [LDB ], X, [LDX],FERR,BERR, [W ORK], [W ORK2], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1)::UPLO ,TRANSA,D IA G
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8), D IM ENSION (:,:) ::A, B, X
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, L D X, \mathbb{N} F O\)
REAL (8),D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

SU BROUTINE TRRFS_64 (UPLO, [TRANSA ], D \(\mathbb{I A G}, N, N R H S, A,[L D A], B,[L D B]\), \(\mathrm{X},[\operatorname{LLX}], F E R R, B E R R,[\mathbb{W}\) ORK \(],[\mathbb{W}\) ORK2], [ \(\mathbb{N} F O])\)

CHARACTER (LEN=1) ::UPLO, TRANSA,D IA G
COMPLEX (8),D IM ENSION (:) ::W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : A, B, X
\(\mathbb{N}\) TEGER ( 8 ) :: N,NRHS,LDA,LDB,LDX, \(\mathbb{N} F O\)
REAL (8), D \(\mathbb{M}\) ENSION (:) ::FERR,BERR,W ORK 2

\section*{C INTERFACE}
\#include < sunperfh>
void ztmfs (char uple, char transa, chardiag, int n, int nihs, doublecom plex *a, int lda, doublecom plex *b, int ldb, doublecom plex *x, int ldx , double *ferr, double *berr, int *info);
void ztrnfs_64 (charuplo, chartransa, char diag, long n, long nihs, doublecom plex *a, long lda, doublecom plex *b, long ldb, doublecom plex *x, long ldx, double *ferr, double *berr, long *info);

\section*{PURPOSE}
ztrnfs provides errorbounds and backw ard error estim ates for the solution to a system of linear equations \(w\) th a triangular coefficientm atrix.

The solution \(m\) atrix \(X\) m ustbe com puted by CTRTRS or some other \(m\) eans before entering this routine. CTRRFS does not do iterative refinem entbecause doing so cannot im prove the backw ard error.

\section*{ARGUMENTS}

\section*{UPLO (input)}
\(=\mathrm{U}\) : A is uppertriangular;
= L' ': A is low ertriangular.
TRANSA (input)
Specifies the form of the system of equations:
\(=N^{\prime}: A * X=B \quad\) N o transpose)
\(=T\) : A ** \(\mathrm{T} * \mathrm{X}=\mathrm{B} \quad\) ( T ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose)
TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
\(=\mathrm{N}: A\) is non-unit triangular;
\(=U\) ': A is unit triangular.

N (input) The order of the matrix A. \(\mathrm{N}>=0\).

NRHS (input)
The num ber of right hand sides, ie., the num ber of colum ns of the m atrices B and X . NRH S >=0.

A (input) The triangularm atrix A. If \(\mathrm{U} P L O=\mathrm{U}\) ', the leading N -by- N upper triangularpart of the aray A contains the upper triangular \(m\) atrix, and the strictly low er triangular part of \(A\) is not referenced. IfUPLO = L ', the leading N -by- N low er triangular part of the amray A contains the low ertriangularm atrix, and the strictly upper triangularpart ofA is not referenced. IfD IA G = U', the diagonal elem ents of A are also not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).
\(B\) (input) The righthand side \(m\) atrix \(B\).

LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, N)\).
\(X\) (input) The solution \(m\) atrix \(X\).

LD X (input)
The leading dim ension of the array X. LD X >= \(\max (1, N)\).

FERR (output)
The estim ated forw ard emorbound for each solution vectorX (i) the \(j\) th colum \(n\) of the solution \(m\) atrix \(X)\). If \(X T R U E\) is the true solution comesponding to \(X(\mathcal{i}), \operatorname{FERR}(\mathcal{)}\) is an estim ated upperbound forthe \(m\) agnitude of the largest ele\(m\) entin (X ( \()\)-X TRUE) divided by the magninude of the largestelem entin X ( 7 ) . The estim ate is as reliable as the estim ate forRCOND, and is alm ost alw ays a slightoverestim ate of the true error.

\section*{BERR (output)}

The com ponentw ise relative backw ard error of each
solution vectorX (i) (i.e., the sm allest relative change in any elem entofA orB thatm akes X ( \(\mathcal{(})\) an exactsolution).

W ORK (w orkspace)
dim ension ( \(2 * \mathrm{~N}\) )
W ORK 2 (w orkspace)
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0: successfulexit
< 0: if \(\mathbb{N}\) FO = -i, the i-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ztrsen - reorder the Schur factorization of a com plex m atrix
\(A=Q * T * Q * * H\), so that a selected clusterofeigenvalues appears in the leading positions on the diagonal of the upper triangularm atrix \(T\), and the leading colum ns of form an orthonorm albasis of the corresponding right invariant subspace

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZTRSEN (OOB,COMPQ,SELECT,N,T,LDT,Q,LDQ,W,M,S,}
SEP,W ORK,LW ORK,INFO)
CHARACTER * 1 JOB,COMPQ
DOUBLE COM PLEX T (LDT,*),Q (LDQ,*),W (*),W ORK (*)
\mathbb{NTEGER N,LDT,LDQ,M,LW ORK,INFO}
LOG ICAL SELECT (*)
DOUBLE PRECISION S,SEP
SUBROUTINE ZTRSEN_64(JOB,COM PQ,SELECT,N,T,LDT,Q,LDQ,W ,M ,S,
SEP,W ORK,LW ORK,INFO)
CHARACTER * 1 JOB,COMPQ
DOUBLE COM PLEX T (LDT,*),Q (LDQ ,*),W (*),W ORK (*)
INTEGER*8N,LDT,LDQ,M,LW ORK, INFO
LOG ICAL*8 SELECT (*)
DOUBLE PRECISION S,SEP

```

F95 INTERFACE
SU BROUTINE TRSEN (JOB,COMPQ,SELECT, \(\mathbb{N}], T,[L D T], Q,[L D Q], W, M\), S, SEP, [W ORK ], [LW ORK], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) :: JOB ,COMPQ
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::W ,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::T,Q
\(\mathbb{N} T E G E R:: N, L D T, L D Q, M, L W O R K, \mathbb{N} F O\)
LO G ICAL,D IM ENSION (:) ::SELECT
REAL (8) :: S, SEP

SU BROUTINE TRSEN_64 (JO B,COM PQ, SELECT, \(\mathbb{N}], T,[L D T], Q,[L D Q], W\), \(\mathrm{M}, \mathrm{S}, \mathrm{SEP},[\mathbb{W}\) ORK], [WW ORK ], [ \(\mathbb{N} F \mathrm{O}])\)

CHARACTER (LEN=1)::JOB,COMPQ
COMPLEX (8),D IM ENSION (:) ::W ,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::T,Q
\(\mathbb{N}\) TEGER (8) ::N,LDT,LDQ,M,LW ORK, \(\mathbb{N} F O\)
LO G ICAL (8), D IM ENSIO N (:) ::SELECT
REAL (8) :: S, SEP

\section*{C INTERFACE}
\#include <sunperfh>
void ztrsen (char job, char com pq, int *select, intn, doublecom plex *t, int ldt, doublecom plex *q, int ldq, doublecom plex *w, int *m, double *s, double *sep, int*info);
void ztrsen_64 (char job, char com pq, long *select, long n, doublecom plex *t, long ldt, doublecom plex *q, long ldq, doublecom plex *w , long *m , double \({ }^{*}\) s, double *sep, long *info);

\section*{PURPOSE}
ztrsen reorders the Schur factorization of a com plex m atrix \(\mathrm{A}=\mathrm{Q} * \mathrm{~T} * \mathrm{Q} * * \mathrm{H}\), so that a selected clusterofeigenvalues appears in the leading positions on the diagonal of the upper triangularm atrix \(T\), and the leading colum ns ofQ form an orthonorm albasis of the comesponding right invariant subspace.

Optionally the routine com putes the reciprocal condition num bers of the cluster ofeigenvalues and/or the invariant subspace.

\section*{ARGUMENTS}
\(J O B\) (input)
Specifies w hether condition num bers are required for the cluster of eigenvalues (S) or the invari-
ant subspace (SEP):
= N ': none;
= E': foreigenvalues only (S);
= V ': for invariant subspace only (SEP);
= B ': forboth eigenvalues and invariant subspace (S and SEP).

COMPQ (input)
= V ': update the m atrix Q ofSchurvectors;
= N ': do notupdate Q .

\section*{SELECT (input)}

SELEC T specifies the eigenvalues in the selected
cluster. To select the \(j\) th eigenvalue, SELEC T (i) mustbe setto .TRUE ..

N (input) The order of the m atrix \(\mathrm{T} \cdot \mathrm{N}>=0\).
T (input/output)
O n entry, the uppertriangularm atrix T. On exit, \(T\) is overw rilten by the reordered \(m\) atrix \(T, w\) ith the selected eigenvalues as the leading diagonal elem ents.

LD T (input)
The leading dim ension of the array T. LD T >= \(\max (1, N)\).

Q (input) \(O n\) entry, if \(C O M P Q=V\) ', the \(m\) atrix \(Q\) of Schur vectors. On exit, if COMPQ = V', Q has been postm ultiplied by the unitary transform ation \(m\) atrix which reorders \(T\); the leading \(M\) colum ns of Q form an orthonorm al basis for the specified invariant subspace. If \(C O M P Q=N\) ', Q is not referenced.

LD Q (input)
The leading dim ension of the array \(\mathrm{Q} . \mathrm{LD} \mathrm{Q}>=1\); and ifCOM \(P Q=V\) ', LD \(Q>=N\).

W (output)
The reordered eigenvalues of \(T\), in the sam e order as they appear on the diagonalof \(T\).

M (output)
The dim ension of the specified invariant subspace. \(0<=\mathrm{M}<=\mathrm{N}\).

S (output)
If \(\mathrm{JOB}=\mathrm{E}\) 'or \(\mathrm{B}^{\prime}\) ', S is a lower bound on the reciprocal condition num ber for the selected clus-
ter ofeigenvalues. S cannot underestim ate the true reciprocal condition num berby \(m\) ore than a factorof sqit \(\mathbb{N}\) ). If \(M=0\) or \(N, S=1\). If \(J 0 B=\) N 'or V ', S is not referenced.

SEP (output)
If \(\mathrm{JO} \mathrm{B}=\mathrm{V}\) 'or B ', SEP is the estim ated reciprocal condition num ber of the specified invariant subspace. IfM \(=0\) orN, \(\mathrm{SEP}=\) norm ( T ). If \(\mathrm{JOB}=\) N 'or E',SEP is not referenced.
W ORK (w orkspace)
If \(\mathrm{JOB}=\mathrm{N}\) ', W ORK is not referenced. O therw ise, on exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the aray \(W\) ORK. If \(J O B=N\) ', LW ORK >=1; if \(\mathcal{O} O B=E \prime\), LW \(O R K=M * \mathbb{N}-M)\); if \(J O B\) \(=V\) 'or \(B\) ', LW ORK \(>=2 * M * \mathbb{N}-M)\).

IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N F O}\) (output)
= 0: successfulexit
<0: if \(\mathbb{I N}\) FO = -i, the i-th argum ent had an illegalvalue

\section*{FURTHER DETAILS}

C TRSEN first collects the selected eigenvalues by com puting a unitary transform ation \(Z\) to \(m\) ove them to the top left comerofT. In otherw ords, the selected eigenvalues are the eigenvalues of T11 in:
\[
\begin{gathered}
\mathrm{Z} \text { * } \mathrm{T} * \mathrm{Z}=(\mathrm{T} 11 \mathrm{~T} 12) \mathrm{n} 1 \\
(0 \mathrm{~T} 22) \mathrm{n} 2 \\
\mathrm{n} 1 \mathrm{n} 2
\end{gathered}
\]
\(w\) here \(N=n 1+n 2\) and \(Z\) ' \(m\) eans the conjugate transpose of \(Z\). The firstn1 colum ns of \(Z\) span the specified invariantsubspace of \(T\).

If \(T\) hasbeen obtained from the Schur factorization of a \(m\) atrix \(A=Q * T * Q\) ', then the reordered Schur factorization of \(A\) is given by \(\left.A=Q * Z)^{*}\left(Z{ }^{*} T * Z\right)^{*} Q * Z\right)\) ', and the first \(n 1\) colum ns of Q * Z span the comesponding invariant subspace of
A.

The reciprocal condition num ber of the average of the eigenvalues of T11 may be retumed in S.S lies betw een 0 (very badly conditioned) and 1 (very w ellconditioned). It is com puted as follow s. Firstw e com pute \(R\) so that
\[
\begin{gathered}
P=\left(\begin{array}{l}
\text { I R }) ~ n 1 ~ \\
(00) n 2 \\
n 1 n 2
\end{array}\right.
\end{gathered}
\]
is the projector on the invariant subspace associated with T11. R is the solution of the Sylvesterequation:
\[
\mathrm{T} 11 * \mathrm{R}-\mathrm{R} * \mathrm{~T} 22=\mathrm{T} 12 .
\]

LetF-norm M) denote the Frobenius-norm of M and 2-norm (M) denote the tw o-norm of . Then \(S\) is com puted as the low er bound
\[
(1+F-\text { norm }(R) * * 2)^{* *}(-1 / 2)
\]
on the reciprocalof2-nom (P), the true reciprocal condition number. S cannotunderestim ate \(1 / 2\)-norm (P) by m ore than a factorof sqrt \(\mathbb{N}\) ).

A \(n\) approxim ate errorbound for the com puted average of the eigenvalues of T11 is
```

EPS * norm (T)/S

```
where EPS is the m achine precision.
The reciprocal condition num ber of the right invariant subspace spanned by the firstn1 colum nsof (orofQ *Z) is retumed in SEP. SEP is defined as the separation of T11 and T 22 :
```

sep(T11,T22 ) = sigm a-m in (C )

```
where sigm a-m in (C) is the sm allest singularvalue of the \(\mathrm{n} 1 * \mathrm{n} 2-b y-n 1 * n 2 \mathrm{~m}\) atrix
\[
C=\operatorname{kprod}(I(n 2), T 11)-k p r o d(t r a n s p o s e(T 22), I(n 1))
\]

I( \(m\) ) is an \(m\) by \(m\) identity \(m\) atrix, and kprod denotes the \(K\) ronecker product. W e estim ate sigm a-m in (C) by the reciprocalofan estim ate of the 1 -norm of inverse (C). The true reciprocal 1-norm of inverse (C ) cannotdiffer from sigm a\(m\) in (C) by \(m\) ore than a factor of sqrt (n1*n2).

W hen SEP is sm all, sm all changes in \(T\) can cause large changes in the invariantsubspace. A \(n\) approxim ate bound on the \(m\) axim um angularerrorin the com puted right invariant subspace is

EPS * norm (T) /SEP

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztrsm - solve one of the \(m\) atrix equations op (A )*X = alpha*B,orX *op (A ) = alpha*B

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZTRSM (SDE,UPLO,TRANSA,D IAG,M,N,ALPHA,A,LDA,B,}
LD B )
CHARACTER * 1SDEE,UPLO,TRANSA,D IAG
DOUBLE COM PLEX ALPHA
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGERM,N,LDA,LDB

```

```

    LD B)
    ```
CHARACTER * 1 SDE, UPLO,TRANSA,D IA G
DOUBLE COM PLEX ALPHA
DOUBLE COM PLEXA (LDA,*), B (LDB, \(\left.{ }^{*}\right)\)
\(\mathbb{N}\) TEGER*8 M , N , LD A , LD B

\section*{F95 INTERFACE}

SU BROUTINE TRSM (SDE, UPLO, [TRANSA ],D \(\mathbb{I A G}, \mathbb{M}], \mathbb{N}], A L P H A, A,[L D A]\), B, [LD B])

CHARACTER (LEN=1)::SDE,UPLO,TRANSA,D IAG
COM PLEX (8) ::ALPHA
COM PLEX (8), D IM ENSION (:,:) ::A,B
\(\mathbb{N} T E G E R:: M, N, L D A, L D B\)
SU BROUTINE TRSM_64 (SDE,UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{M}], \mathbb{N}], A L P H A, A\), [LDA], B, [LDB])

CHARACTER (LEN=1) :: SDE \(\operatorname{CHPLO}\),TRANSA,D IA G
COM PLEX (8) :: ALPHA
COM PLEX (8), D \(\mathbb{M}\) ENSION (: : : : : : A , B
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{LD} A, L D B\)

\section*{C INTERFACE}
\#include <sunperfh>
void ztrsm (charside, charuplo, chartransa, chardiag, int \(m\), intn, doublecom plex *alpha, doublecom plex *a, int lda, doublecom plex *b, int ldb);
void ztrsm _64 (charside, charuplo, chartransa, char diag, long \(m\), long \(n\), doublecomplex *alpha, doublecom plex *a, long lda, doublecom plex *b, long ldb);

\section*{PURPOSE}
ztrem solves one of the \(m\) atrix equations op (A )*X = alpha*B, or \(X *\) op (A ) \(=\) alpha*B w here alpha is a scalar, \(X\) and \(B\) are \(m\) by \(n m\) atriges, \(A\) is a unit, ornon-unit, upper or low ertriangularm atrix and op (A ) is one of
\(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{cong}(\) \(\left.A^{\prime}\right)\).

Them atrix \(X\) is overw ritten on \(B\).

\section*{ARGUMENTS}

SDE (input)
On entry, SID E specifies w hetherop (A ) appears on the left or right ofX as follow s:
\(S \mathbb{D} E=\mathbb{L}\) 'or \(\mathrm{I}^{\prime}\) op (A ) *X = a pha*B.
\(S \mathbb{D} E=R^{\prime}\) or \(r^{\prime} X^{*}\) op (A) \(=\) alpha*B.

U nchanged on exit.

UPLO (input)
O n entry, UPLO specifies w hether the m atrix A is an upper or low er triangularm atrix as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or G ' \(A\) is an upper triangular
m atrix.
\(\mathrm{UPLO}=\mathrm{L}\) ' or I' A is a lower triangular \(m\) atrix.

U nchanged on exit.

TRANSA (input)
On entry, TRAN SA specifies the form ofop (A ) to be used in the \(m\) atrix \(m\) ultiplication as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime}\) op (A) \()=A\).

TRANSA = T'or \(t^{\prime} \mathrm{op}(A)=A\) '.

TRANSA \(=C^{\prime}\) or \(C^{\prime} o p(A)=c o n j g\left(A^{\prime}\right)\).
U nchanged on exit.

TRANSA is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
O n entry, D IA G specifies w hether ornotA is unit triangular as follow s:

D IA G = U 'or \(\mathrm{U}^{\prime}\) A is assum ed to be unit triangular.
\(D \mathbb{A G}=N\) 'or \(h^{\prime} A\) is notassum ed to be unit triangular.

U nchanged on exit.

M (input)
O \(n\) entry, \(M\) specifies the num ber of row s of \(B . M\) \(>=0\). U nchanged on exit.

N (input)
O n entry, \(N\) specifies the num ber of colum ns of \(B\).
\(\mathrm{N}>=0\). U nchanged on exit.

ALPHA (input)
On entry, A LPH A specifies the scalar alpha. W hen alpha is zero then \(A\) is notreferenced and \(B\) need notbe setbefore entry. U nchanged on exit.

A (input)
COMPLEX *16 aray ofD \(\mathbb{I}\) ENSION (LDA, k) , \(w\) here \(k\) is \(m\) when \(S \mathbb{D E}=\mathbb{L}\) 'or \(\mathbb{I}^{\prime}\) and is \(n\) when \(S \mathbb{D} E=R\) 'or \(r^{\prime}\) '.

Before entry \(w\) th \(U P L O=U\) 'or \(u\) ', the leading \(k\) by \(k\) upper triangularpart of the array A m ustcontain the upper triangularm atrix and the strictly low ertriangularpartofA is not referenced.

Before entry with UPLO = L'or 1', the leading \(k\) by \(k\) low er triangularpart of the array \(A\) \(m\) ust contain the low er triangularm atrix and the strictly uppertriangular partofA is not referenced.
\(N\) ote thatw hen \(D \mathbb{A} G=U\) ' or L ', the diagonal elem ents of A are not referenced either, butare assum ed to be unity.

U nchanged on exit.
LD A (input)
On entry, LD A specifies the first dim ension of A as declared in the calling (sub) program. W hen \(S \mathbb{D} E=\mathrm{L}\) 'or I ' then LD \(A>=\mathrm{max}(1, \mathrm{M})\), when \(S \mathbb{D} E\) \(=R^{\prime}\) or ' \(r^{\prime}\) then LD \(A>=m a x(1, N)\). U nchanged on exit.

B (input/output)
COM PLEX *16 array ofD \(\mathbb{M}\) ENSION (LDB, n). B efore entry, the leading \(M\) by \(N\) part of the amay \(B m\) ust contain the righthand side \(m\) atrix \(B\), and on exit is overw rilten by the solution \(m\) atrix \(X\).

LD B (input)
On entry, LD B specifies the firstdim ension of B as declared in the calling subprogram. LD B >= max (1,M). U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ztrsna - estim ate reciprocal condition num bers for specified eigenvalues and/or right eigenvectors of a com plex upper triangularm atrix T (orofany m atrix \(\mathrm{Q} * \mathrm{~T} * \mathrm{Q} * * \mathrm{H}\) w ith Q unitary)

\section*{SYNOPSIS}
```

SUBROUTINE ZTRSNA (JOB,HOW MNY,SELECT,N,T,LDT,VL,LDVL,VR,LDVR,
S,SEP,MM ,M ,W ORK,LDW ORK,W ORK1, NNFO)
CHARACTER * 1 JOB,HOW MNY
DOUBLE COMPLEX T (LDT,*), VL (LDVL,*), VR (LDVR,*),
W ORK (LDW ORK,*)
\mathbb{NTEGERN,LDT,LDVL,LDVR,MM,M,LDW ORK,INFO}
LOG ICAL SELECT (*)
DOUBLE PRECISION S (*),SEP (*),W ORK 1 (*)
SUBROUT\mathbb{NE ZTRSNA_64(JOB,HOW MNY,SELECT,N,T,LDT,VL,LDVL,VR,}
LDVR,S,SEP,MM ,M ,W ORK,LDW ORK,W ORK 1, INFO)
CHARACTER * 1 OOB,HOW MNY
DOUBLE COMPLEX T (LDT,*), VL (LDVL,*), VR (LDVR,*),
W ORK (LDW ORK,*)
NNTEGER*8N,LDT,LDVL,LDVR,MM ,M,LDW ORK,INFO
LOG ICAL*8 SELECT (*)
DOUBLE PRECISION S (*),SEP (*),W ORK 1 (*)

```
F95 INTERFACE
    SU BROUTINE TRSNA (JOB,HOW M NY, SELECT, \(\mathbb{N}], T,[L D T], V L,[L D V L], V R\),
    [LDVR],S,SEP, MM,M,[WORK], [LDW ORK], [WORK1], [ \(\mathbb{N} F O]\) )

CHARACTER (LEN=1) :: JOB,HOW M NY
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::T,VL,VR,W ORK
\(\mathbb{N} T E G E R:: N\), LD T, LDVL,LDVR, M M , M , LDW ORK , \(\mathbb{N} F O\)
LOG ICAL, D \(\mathbb{I M} E N S I O N\) (:) ::SELECT
REAL (8), D \(\mathbb{M}\) ENSION (:) :: S, SEP,W ORK 1

SU BROUTINE TRSNA_64 (OBB,HOW M NY, SELECT, \(\mathbb{N}], T,[L D T], V L,[L D V L]\), VR, [LDVR],S,SEP,MM,M,[WORK],[LDWORK],[WORK1],[NFO])

CHARACTER (LEN=1) :: JOB,HOW M NY
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::T,VL,VR,W ORK
\(\mathbb{N}\) TEGER (8) :: N , LD T , LDVL, LDVR, M M , M, LDW ORK , \(\mathbb{N} F O\)
LOGICAL (8), D \(\mathbb{M}\) ENSION (:) :: SELECT
REAL (8), D \(\mathbb{M}\) ENSION (:) :: S, SEP, W ORK 1

\section*{C INTERFACE}
\#include <sunperfh>
void ztrsna (char job, char how my, int *select, intn, doublecom plex *t, int ldt, doublecom plex *vl, int ldvl, doublecom plex *vr, int ldvr, double *s, double *sep, intm \(m\), int *m, int ldw ork, int *info);
void ztrsna_64 (char j.b, charhow m ny, long *select, long n, doublecom plex *t, long ldt, doublecom plex *vl, long ldvl, doublecom plex *vr, long ldvr, double *s, double *sep, long mm, long *m, long ldw ork, long *info);

\section*{PURPOSE}
ztrsna estim ates reciprocalcondition num bers for specified eigenvahues and/or right eigenvectors of a com plex upper triangularm atrix T (orofany \(m\) atrix \(Q * T * Q * * H\) with \(Q\) uni tary).

\section*{ARGUMENTS}

JOB (input)
Specifies w hethercondition num bers are required
foreigenvahues (S) oreigenvectors (SEP):
\(=\mathrm{E}\) ': foreigenvahues only (S);
\(=V^{\prime}\) : foreigenvectors only (SEP);
\(=B\) ': forboth eigenvalues and eigenvectors ( \(S\) and SEP).

H OW M NY (input)
= 'A ': com pute condition num bers for all eigen-
pairs;
= S ': com pute condition num bers for selected eigenpairs specified by the array SELEC T .

SELECT (input)
If HOW M NY = S',SELECT specifies the eigenpairs
for which condition num bers are required. To
selectcondition num bers for the \(j\) th eigenpair,
SELECT ( \(\boldsymbol{\jmath}\) ) mustbe set to TRUE.. If HOW M NY = A',
SELECT is not referenced.

N (input) The order of the m atrix \(\mathrm{T} . \mathrm{N}>=0\).
T (input) The upper triangularm atrix T .
LD T (input)
The leading dim ension of the array T. LD T >= \(\max (1, \mathbb{N})\).

VL (input)
If \(\mathrm{O} \mathrm{B}=\mathrm{E}\) 'or B ', VL mustcontain left eigenvectors of \(T\) (orofany \(Q * T * Q * * H\) with \(Q\) unitary), corresponding to the eigenpairs specified by HOW M NY and SELECT.The eigenvectors m ustbe stored in consecutive columns of \(V L\), as retumed by \(C H S E I N\) orCTREVC. If \(J B=V\) ', \(V L\) is notreferenced.

LDVL (input)
The leading dim ension of the array VL. LD V L >=1; and if \(J 0 B=E\) 'or \(B ', L D V L>=N\).

VR (input)
If \(J 0 B=E\) 'or \(B\) ', VR m ust contain right eigenvectors of \(T\) (or of any \(Q * T * Q * * H\) with \(Q\) unitary), corresponding to the eigenpairs specified by HOW M NY and SELECT. The eigenvectors m ustbe stored in consecutive colum ns of VR, as retumed by \(C H S E I N\) orCTREVC. If \(O B=V\) ', VR is notreferenced.

LDVR (input)
The leading dim ension of the array VR. LD V R >=1; and if \(J 0 B=E\) 'or \(B ', L D V R>=N\).

S (output)
If \(\mathrm{JOB}=\mathrm{E}\) ' or B ', the reciprocal condition num bers of the selected eigenvalues, stored in consecutive elem ents of the aray. Thus \(S(\mathcal{)}\), \(\operatorname{SEP}(\mathcal{j})\), and the \(j\) th colum ns of VL and VR all comespond to the sam e eigenpair (butnot in gen-
eral the \(j\) th eigenpair, unless alleigenpairs are selected). If \(\mathrm{OB}=\mathrm{V}^{\prime}, \mathrm{S}\) is notreferenced.

SEP (output)
If \(\mathrm{OB}=\mathrm{V}\) 'or B', the estim ated reciprocalcondition numbers of the selected eigenvectors, stored in consecutive elem ents of the aray. If \(\mathrm{JOB}=\mathrm{E}\) ', SEP is notreferenced.
M M (input)
The num berofelem ents in the anays S (if \(\mathrm{OB}=\) \(E^{\prime}\) or B') and/orSEP (if \(\mathrm{OB}=\mathrm{V}\) 'or B).M M \(>=\mathrm{M}\) 。

M (output)
The num ber ofelem ents of the arays \(S\) and/or SEP
actually used to store the estim ated condition num bers. If HOW M NY = \(A\) ', M is setto \(N\).

W ORK (w orkspace)
dim ension (LDW ORK, \(N+1\) ) If \(O B=E\) ', WORK is not referenced.

LDW ORK (input)
The leading dim ension of the array \(W\) ORK. LDW ORK \(>=1\); and if \(\mathrm{OB}=\mathrm{V}\) 'or \(\mathrm{B}^{\prime}\), LDW ORK \(>=\mathrm{N}\).

W ORK1 (w orkspace)
dimension \(\mathbb{N}\) ) If OB = E', W ORK1 is not referenced.
\(\mathbb{I N F O}\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}

The reciprocal of the condition num ber of an eigenvalue lam boda is defined as
\[
S(\operatorname{lam} \text { bda })=f^{*} u \mid /(\text { norm }(u) * \text { norm }(v))
\]
\(w\) here \(u\) and \(v\) are the right and left eigenvectors of \(T\) corresponding to lam boda; v'denotes the conjugate transpose of \(v\), and norm (u) denotes the Euclidean norm . These reciprocal condition num bers alw ays lie betw een zero (very badly conditioned) and one (very w ell conditioned). If \(n=1\), \(S\) (lam bda) is defined to be 1 .

A \(n\) approxim ate errorbound for a com puted eigenvalue \(W\) (i) is
given by
EPS * norm (T) /S (i)
where EPS is the m achine precision.
The reciprocal of the condition num ber of the right eigenvectoru conesponding to lam bda is defined as follow s. Suppose
\[
\begin{gathered}
\mathrm{T}=\binom{\operatorname{lam} \mathrm{bda} \mathrm{c})}{\left(\begin{array}{cc}
\mathrm{T} 22
\end{array}\right)}
\end{gathered}
\]

Then the reciprocalcondition num ber is

SEP (lam bda, T22 \()=\) sigm a-m in (T22 -lam bda*I \()\)
where sigm a-m in denotes the sm allest singular value. \(W\) e approxim ate the sm allest singularvalue by the reciprocal of an estim ate of the one-norm of the inverse of T22 \(\operatorname{lam}\) bda* I . If \(\mathrm{n}=1, \operatorname{SEP}(1)\) is defined to be abs \((\mathrm{T}(1,1))\).

A \(n\) approxim ate errorbound for a com puted right eigenvector VR (i) is given by
EPS * norm (I) /SEP (i)

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztrsv - solve one of the system sofequations \(A * x=b\), or \(A^{*} x=b, \operatorname{orcon} \dot{g}\left(A^{\prime}\right){ }^{*} x=b\)

\section*{SYNOPSIS}

SU BROUTINE ZTRSV (UPLO,TRANSA,D \(\mathbb{A} G, N, A, L D A, Y, \mathbb{N} C Y)\)
CHARACTER * 1 UPLO, TRANSA, D IA G
DOUBLE COM PLEXA (LDA, *), Y (*)
\(\mathbb{N}\) TEGER \(N, L D A, \mathbb{N} C Y\)
SUBROUTINE ZTRSV_64 (UPLO, TRANSA,DIAG,N,A,LDA,Y, \(\mathbb{N} C Y\) )

CHARACTER * 1 UPLO,TRANSA, D IAG
DOUBLE COM PLEXA (LDA, \(\left.{ }^{\star}\right)\), Y ( \({ }^{\star}\) )
\(\mathbb{N}\) TEGER*8N,LDA, \(\mathbb{N} C Y\)

\section*{F95 INTERFACE}

SU BROUTINE TRSV (UPLO, [TRANSA],D \(\mathbb{I A} G, \mathbb{N}], A,[L D A], Y,[\mathbb{N C Y}])\)
CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) :: Y
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: N, L D A, \mathbb{N} C Y\)
SU BROUTINE TRSV_64 (UPLO, [TRANSA],D \(\mathbb{A} G, \mathbb{N}], A,[L D A], Y,[\mathbb{N} C Y])\)
CHARACTER (LEN=1)::UPLO,TRANSA,D IAG
COMPLEX (8),D IM ENSION (:) ::Y
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R(8):: N, L D A, \mathbb{N} C Y\)

\section*{C INTERFACE}
\#include <sunperfh>
void ztrsv (charuplo, chartransa, chardiag, int n, doublecom plex *a, int lda, doublecom plex *y, int incy);
void ztrsv_64 (charuplo, chartransa, char diag, long n, doublecom plex *a, long lda, doublecom plex *y, long incy);

\section*{PURPOSE}
ztrsv solves one of the system s ofequations \(A * x=b\), or \(A^{*}{ }^{\prime} \mathrm{x}=\mathrm{b}\), orcon \(\mathrm{g}_{\mathrm{g}}\left(\mathrm{A}^{\prime}\right)^{\star} \mathrm{x}=\mathrm{b}\) where b and x are n elem ent vectors and \(A\) is an \(n\) by \(n\) unit, ornon-unit, upper or low er triangularm atrix.

N o testforsingularity or near-singularity is included in this routine. Such testsm ustbe perform ed before calling this routine.

\section*{ARGUMENTS}

UPLO (input)
O n entry, UPLO specifies w hether the \(m\) atrix is an upper or low er triangularm atrix as follow s:
\(\mathrm{UPLO}=\mathrm{U}\) 'or \(\mathrm{U}^{\prime} \mathrm{A}\) is an upper triangular \(m\) atrix .
\(\mathrm{UPLO}=\mathrm{L}\) ' or \(\mathrm{I}^{\prime}\) ' A is a lower triangular m atrix.

U nchanged on exit.

TRANSA (input)
On entry, TRAN SA specifies the equations to be solved as follow s:

TRANSA \(=N^{\prime}\) or \(h^{\prime} A{ }^{*} \mathrm{x}=\mathrm{b}\).

TRANSA = T'ort'A *x = b.

TRANSA \(=C^{\prime}\) or \(t^{\prime} \operatorname{cong}\left(A^{\prime}\right) \star x=b\).

U nchanged on exit.

TRANSA is defaulted to N 'forF95 \(\mathbb{I N}\) TERFACE.

D IA G (input)
On entry, D IA G specifies whether ornotA is unit triangular as follow s:

D \(\mathbb{I}\) G \(=U\) 'or 4 ' \(A\) is assum ed to be unit triangular.
\(D \mathbb{A} G=N\) 'or \(h\) ' \(A\) is notassum ed to be unit triangular.

U nchanged on exit.

N (input)
O \(n\) entry, \(N\) specifies the order of the \(m\) atrix A. \(\mathrm{N}>=0\). U nchanged on exit.

A (input)
Before entry \(w\) ith UPLO = U 'or L ', the leading \(n\) by \(n\) upper triangular part of the array A must contain the upper triangular \(m\) atrix and the strictly low ertriangularpartofA is not referenced. Before entry w ith UPLO = L 'or I', the leading \(n\) by \(n\) low er triangularpart of the anay A m ust contain the low er triangularm atrix and the strictly uppertriangularpartofA is not referenced. N ote thatw hen D \(\mathbb{A G}=\mathrm{U}\) ' or L ', the diagonal elem ents of A are not referenced either, butare assum ed to be unity. U nchanged on exit.

LD A (input)
On entry, LD A specifies the firstdim ension of A as declared in the calling (sub) program. LD A \(>=\) \(m a x(1, n)\). U nchanged on exit.

Y (input/output)
\((1+(n-1) * a b s(\mathbb{N} C Y))\). Before entry, the increm ented array \(Y\) must contain the \(n\) elem ent righthand side vectorb. On exit, \(Y\) is overw ritten \(w\) ith the solution vectorx.
\(\mathbb{N} C Y\) (input)
On entry, \(\mathbb{N} C Y\) specifies the increm ent for the elem ents of \(Y . \mathbb{N} C Y\) <> 0 . U nchanged on exit.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztrsyl-solve the com plex Sylvesterm atrix equation

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZTRSYL (TRANA,TRANB,ISGN,M,N,A,LDA,B,LDB,C,LDC,}
SCALE,INFO)
CHARACTER * 1 TRANA,TRANB
DOUBLE COM PLEX A (LDA,*),B (LDB,*),C (LD C ,*)
\mathbb{N TEGER ISGN,M,N,LDA,LDB,LDC,NNFO}
DOUBLE PRECISION SCALE
SUBROUTINE ZTRSYL_64(IRANA,TRANB,ISGN,M,N,A,LDA,B,LDB,C,
LDC,SCALE, INFO)
CHARACTER * 1 TRANA,TRANB
DOUBLE COM PLEX A (LDA,*),B (LDB,*),C (LDC,*)
\mathbb{NTEGER*8 ISGN,M,N,LDA,LDB,LDC,}\mathbb{N}FO
DOUBLE PRECISION SCALE

```

\section*{F95 INTERFACE}
```

SU BROUTINE TRSYL (TRANA,TRANB, ISGN, $\mathbb{M}], \mathbb{N}], A,[L D A], B,[L D B], C$, [LDC],SCALE, [ $\mathbb{N} F O$ ])
CHARACTER (LEN=1) ::TRANA,TRANB
COMPLEX (8),D $\mathbb{M}$ ENSION (: : : : :: A, B, C
$\mathbb{N} T E G E R:: \mathbb{I S G N}, \mathrm{M}, \mathrm{N}, L D A, L D B, L D C, \mathbb{N} F O$
REAL (8) :: SCALE
SU BROUTINE TRSY L_64 (TRANA,TRANB, ISGN, $\mathbb{M}], \mathbb{N}], A,[L D A], B,[L D B]$, C, [LDC],SCALE, [ $\mathbb{N} F O]$ )

```

CHARACTER (LEN=1) ::TRANA,TRANB
COM PLEX (8), D IM ENSION (:,:) ::A, B, C
\(\mathbb{N}\) TEGER (8) :: \(\operatorname{ISG} \mathrm{N}, \mathrm{M}, \mathrm{N}, \mathrm{LD} A, L D B, L D C, \mathbb{N}\) FO
REAL (8) :: SCALE

\section*{C INTERFACE}
\#include <sunperfh>
void ztrsyl(chartrana, chartranb, int isgn, intm, int n, doublecom plex *a, intlda, doublecom plex *b, int ldlb, doublecom plex *c, int ldc, double *scale, int *info);
void ztrsyl 64 (chartrana, chartranb, long isgn, long m, long \(n\), doublecom plex *a, long lda, doublecom plex *b, long ldb, doublecom plex *C, long ldc, double *scale, long *info);

\section*{PURPOSE}
ztrsylsolves the com plex Sylvesterm atrix equation:
\(o p(A) * X+X * o p(B)=\) scale \({ }^{*} C\) or
op (A ) *X -X *op (B) \(=\) scale* \(C\),
where op \((A)=A\) or \(A * * H\), and \(A\) and \(B\) are both uppertriangular. \(A\) is \(M\) boy -M and B is N boy -N ; the righthand side C and the solution X are M -by -N ; and scale is an output.scale factor, set < = 1 to avoid overflow in X .

\section*{ARGUMENTS}

TRANA (input)
Specifies the option op (A):
\(=N^{\prime}: o p(A)=A \quad\) (Notranspose)
\(=C: \operatorname{op}(A)=A * * H \quad\) (C onjugate transpose)

TRANB (input)
Specifies the option op ( \(B\) ) :
\(=N^{\prime}: o p(B)=B \quad\) (N o transpose)
\(=C\) ': op \((B)=B * * H \quad\) (C onjugate transpose)

ISG N (input)
Specifies the sign in the equation:
\(=+1\) : solve op (A ) \({ }^{\text {( } X}+X\) *op \((B)=\) scale \(^{\star} C\)
\(=-1\) : solve op (A ) \({ }^{\star} X-X *\) op \((B)=\) scale \({ }^{\star} C\)
\(M\) (input) The order of the m atrix \(A\), and the num berof row \(s\) in the m atrioes X and \(\mathrm{C} . \mathrm{M}>=0\).
\(N\) (input) The orderof the \(m\) atrix \(B\), and the num ber of colum ns in the \(m\) atrices \(X\) and C. \(\mathrm{N}>=0\).

A (input) The upper triangularm atrix A.

LD A (input)
The leading dim ension of the aray A. LDA >= max (1, M).
\(B\) (input) The upper triangularm atrix \(B\).
LD B (input)
The leading dim ension of the array B. LD B >= \(\max (1, \mathbb{N})\).

C (input/output)
On entry, the \(M\)-by -N righthand side \(m\) atrix C . On exi, \(C\) is overw ritten by the solution \(m\) atrix \(X\).

LD C (input)
The leading dim ension of the aray C. LD C >= \(\max (1, M)\)

SCALE (output)
The scale factor, scale, set < = 1 to avoid overflow in X.
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
< 0 : if \(\mathbb{N N F O}=-\) i, the \(i\)-th argum ent had an illegalvalue
\(=1\) : A and B have common or very close eigenvalues; perturbed values w ere used to solve the equation (but the \(m\) atrices \(A\) and \(B\) are unchanged).

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztrti2 -com pute the inverse of a complex upper or low er triangularm atrix

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZTRTI2(UPLO,DIAG,N,A,LDA,INFO)}
CHARACTER * 1 UPLO,D IAG
DOUBLE COM PLEXA (LDA,*)
\mathbb{NTEGERN,LDA, INFO}
SUBROUTINE ZTRTI2_64(UPLO,D IAG,N,A,LDA, INFO)
CHARACTER * 1 UPLO,DIAG
DOUBLE COM PLEX A (LDA,*)
INTEGER*8N,LDA,INFO
F95 INTERFACE
SUBROUT\mathbb{NE TRTI2 (UPLO,D IAG, N ],A ,[LDA],[NFO])}
CHARACTER (LEN=1) ::UPLO,D IA G
COM PLEX (8),D IM ENSION (:r:) ::A
\mathbb{NTEGER ::N,LDA,INFO}
SUBROUT\mathbb{NE TRTI2_64 (UPLO,D IAG, N ],A,[LDA ], [NNFO])}
CHARACTER (LEN=1)::UPLO,D IA G
COM PLEX (8),D IM ENSION (:,:) ::A
\mathbb{NTEGER (8)::N,LDA,}\mathbb{NFO}

```
C INTERFACE
    \#include <sunperfh>
void ztrti2 (char uplo, chardiag, intn, doublecom plex *a, intlda, int*info);
void ztrti2_64 (charuplo, chardiag, long n, doublecom plex *a, long lda, long *info);

\section*{PURPOSE}
ztrti2 com putes the inverse of a com plex upper or low er triangularm atrix.

This is the Level2 B LAS version of the algorithm .

\section*{ARGUMENTS}

UPLO (input)
Specifies w hether the m atrix A is upper or low er
triangular. = U ': U pper triangular
= L': Low ertriangular

D IA G (input)
Specifies w hether ornot the m atrix A is unittriangular. \(=\mathrm{N}\) ': N on-unittriangular
\(=\mathrm{U}\) ': Unittriangular

N (input) The order of the m atrix A. N \(>=0\).

A (input/output)
O n entry, the triangularm atrix A. If \(\mathrm{U} P \mathrm{O}=\mathrm{U}\) ', the leading \(n\) by \(n\) uppertriangularpart of the aray A contains the uppertriangularm atrix, and the strictly low er triangular part of A is not referenced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading n by n low er triangular part of the array A contains the low ertriangularm atrix, and the strictly upper triangularpart ofA is not referenced. IfD \(\mathbb{I A} G=\) U', the diagonal elem ents of A are also not referenced and are assum ed to be 1 .

On exit, the (triangular) inverse of the original \(m\) atrix, in the sam e storage form at.

LD A (input)
The leading dim ension of the amay A. LDA >= \(\max (1, N)\).

IN FO (output)
= 0: successfulexit
< 0: if \(\mathbb{N N}\) FO \(=-\mathrm{k}\), the k -th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztrtri-com pute the inverse of complex upper or low er triangularm atrix A

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZTRTRI(UPLO,D IAG,N,A,LDA, NNFO)}
CHARACTER * 1 UPLO,D IAG
DOUBLE COM PLEXA (LDA,*)
\mathbb{NTEGERN,LDA,}\mathbb{N}FO
SU BROUT\mathbb{NE ZTRTRI_64(UPLO,D IAG,N,A ,LDA, INFO)}
CHARACTER * 1 UPLO,D IAG
DOUBLE COM PLEXA (LDA,*)
INTEGER*8N,LDA,INFO

```
F95 INTERFACE
    SU BROUTINE TRTRI(UPLO, D \(\mathbb{A} G, N, A,[L D A],[\mathbb{N} F O])\)
    CHARACTER (LEN=1) ::UPLO,D IA G
    COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : ) ::A
    \(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathrm{LDA}, \mathbb{N} F O\)
    SUBROUTINE TRTRI_64 (UPLO, D IA G , N, A , [LDA ], [ \(\mathbb{N} F O\) ])
    CHARACTER (LEN=1) ::UPLO,D IA G
    COMPLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N}\) TEGER (8) :: \(\mathrm{N}, \mathrm{LD} A, \mathbb{N} F O\)
C INTERFACE
    \#include <sunperfh>
void ztrtri(charuplo, chardiag, intn, doublecom plex *a, int lda, int *info);
void ztitri_ 64 (charuplo, chardiag, long n, doublecom plex *a, long lda, long *info);

\section*{PURPOSE}
ztrtricom putes the inverse of a com plex upper or low er triangularm atrix A.

This is the Level3 B LA S version of the algorithm .

\section*{ARGUMENTS}

UPLO (input)
= U : : A is uppertriangular;
= L': A is low er triangular.

D IA G (input)
= N ': A is non-unit triangular;
\(=U\) : A is unittriangular.

N (input) The order of the m atrix A. \(\mathrm{N}>=0\).

A (input/output)
O n entry, the triangularm atrix A. If \(\mathrm{PLO}=\mathrm{U}\) ', the leading N -by N uppertriangularpart of the array A contains the upper triangularm atrix, and the strictly low er triangular partofA is not referenced. If UPLO = L', the leading N -by-N low er triangular part of the array A contains the low er triangularm atrix, and the strictly upper triangularpart ofA is not referenced. IfD \(\mathbb{I A} G=\) U', the diagonal elem ents of A are also not referenced and are assum ed to be 1 . On exit, the (triangular) inverse of the original \(m\) atrix, in the sam e storage form at.

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an ille-
galvalue
\(>0:\) if \(\mathbb{N} F O=i, A(i, i)\) is exactly zero. The triangular \(m\) atrix is singular and its inverse can notbe com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
ztrtes - solve a triangular system of the form \(A * X=B\), \(\mathrm{A} *{ }^{*} \mathrm{~T} * \mathrm{X}=\mathrm{B}\),orA \({ }^{*}{ }^{*} \mathrm{H} * \mathrm{X}=\mathrm{B}\),

\section*{SYNOPSIS}
```

SU BROUTINE ZTRTRS (UPLO,TRANSA,DIAG,N,NRHS,A,LDA,B,LDB,\mathbb{NFO)}
CHARACTER * 1UPLO,TRANSA,DIAG
DOUBLE COM PLEX A (LDA,*),B (LDB,*)
INTEGERN,NRHS,LDA,LDB,INFO
SUBROUT\mathbb{NE ZTRTRS_64 (UPLO,TRANSA,DIAG,N,NRHS,A,LDA,B,LDB,}
INFO)

```
CHARACTER * 1 UPLO, TRANSA, DIAG
D OUBLE COM PLEX A (LDA,*), B (LDB,*)
\(\mathbb{N}\) TEGER*8N,NRHS,LDA,LDB, \(\mathbb{N} F O\)

\section*{F95 INTERFACE}

SU BROUTINE TRTRS (UPLO, [TRANSA ], D IA G ,N,NRHS,A, [LDA],B, [LDB], [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
COM PLEX (8),D IM ENSION (:,:) ::A, B
\(\mathbb{N} T E G E R:: N, N R H S, L D A, L D B, \mathbb{N} F O\)

SU BROUTINE TRTRS_64 (UPLO, [TRANSA ],D \(\mathbb{I A} G, N, N R H S, A,[L D A], B,[L D B]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::UPLO,TRANSA,D IA G
COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : ) ::A, B
\(\mathbb{N} T E G E R(8):: N, N R H S, L D A, L D B, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void ztrtrs (charuplo, chartransa, chardiag, int \(n\), int nıhs, doublecom plex *a, int lda, doublecom plex *b, int ldb, int*info);
void ztrtrs_64 (charuplo, chartransa, char diag, long n, long nihs, doublecom plex *a, long lda, doublecom plex *b, long ldb, long *info);

\section*{PURPOSE}
ztates solves a triangular system of the form where \(A\) is a triangularm atrix of order \(N\), and \(B\) is an \(N\) -by-NRHS m atrix. A check ism ade to verify that A is nonsingular.

\section*{ARGUMENTS}

UPLO (input)
\(=\mathrm{U}: \mathrm{A}\) is uppertriangular;
= L': A is low ertriangular.
TRANSA (input)
Specifies the form of the system of equations:
\(=N\) ': A * \(\mathrm{X}=\mathrm{B} \quad\) ( o otranspose)
\(=T: A * * T * X=B \quad(T\) ranspose)
\(=\mathrm{C}: \mathrm{A} * * \mathrm{H} * \mathrm{X}=\mathrm{B} \quad\) (C onjugate transpose)
TRANSA is defaulted to N 'forF \(95 \mathbb{I N}\) TERFACE.

D IA G (input)
\(=N^{\prime}: A\) is non-unit triangular;
\(=\mathrm{U}\) : A is unit triangular.

N (input) The order of the \(m\) atrix \(\mathrm{A} . \mathrm{N}>=0\).
NRHS (input)
The num ber of righthand sides, i.e., the num ber of colum ns of them atrix B. NRHS \(>=0\).

A (input) The triangularm atrix A. If \(\mathrm{PLO}=\mathrm{U}\) ', the lead-
ing N -by -N upper triangularpart of the array A
contains the upper triangular \(m\) atrix, and the strictly low ertriangularpartofA is not refer-
enced. If \(\mathrm{UPLO}=\mathrm{L}\) ', the leading N -by N lower triangular part of the array A contains the low er triangularm atrix, and the strictly upper triangular part ofA is not referenced. IfD \(\mathbb{I A} G=U\) ', the diagonalelem ents of \(A\) are also not referenced and are assum ed to be 1 .

LD A (input)
The leading dim ension of the array A. LD A >= \(\max (1, N)\).

B (input/output)
On entry, the right hand side m atrix B. On exit, if \(\mathbb{N} F O=0\), the solution \(m\) atrix \(X\).

LD B (input)
The leading dim ension of the array \(B\). LD B \(>=\) \(\max (1, N)\).
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue
\(>0:\) if \(\mathbb{N} F O=i\), the \(i\)-th diagonalelem ent of \(A\) is zero, indicating that the \(m\) atrix is singular and the solutions \(X\) have notbeen com puted.

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ztzrqf-routine is deprecated and has been replaced by routine C TZRZF

\section*{SYNOPSIS}

```

    DOUBLE COM PLEX A (LDA,*),TAU (*)
    INTEGERM,N,LDA,\mathbb{NFO}
    SUBROUT\mathbb{NE ZTZRQF_64M,N,A,LDA,TAU,INFO)}
    DOUBLE COM PLEX A (LDA,*),TAU (*)
    INTEGER*8M,N,LDA,INFO
    F95 INTERFACE
SUBROUT\mathbb{NE TZRQF (M ], N ],A, [LDA],TAU, [NFO])}
COM PLEX (8),D IM ENSION (:) ::TAU
COM PLEX (8),D IM ENSION (:,:) ::A
\mathbb{NTEGER ::M,N,LDA,NNFO}

```

```

    COM PLEX (8),D IM ENSION (:) ::TAU
    COM PLEX (8),D IM ENSION (:,:) ::A
    \mathbb{NTEGER (8)::M,N,LDA,INFO}
    ```
C INTERFACE
    \#include <sunperfh>
void ztzrqf(intm , intn, doublecom plex *a, int lda, doublecom plex *tau, int *info);
void ztzrqf_64 (long m , long n, doublecom plex *a, long lda, doublecom plex *tau, long *info);

\section*{PURPOSE}
ztzrof routine is deprecated and has been replaced by routine CTZRZF.

CTZRQF reduces the M -by-N ( \(\mathrm{M}<=\mathrm{N}\) ) com plex upper trapezoidal \(m\) atrix \(A\) to upper triangular form by \(m\) eans of unitary transform ations.

The upper trapezoidalm atrix \(A\) is factored as
\[
A=\left(\begin{array}{ll}
R & 0
\end{array}\right)^{*} Z
\]
\(w\) here \(Z\) is an \(N\) boy \(-N\) unitary \(m\) atrix and \(R\) is an \(M\) boy \(M\) upper triangularm atrix.

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(A . M>=0\).

N (input) The num ber of colum ns of the m atrix \(A . N>=M\).

A (input/output)
O n entry, the leading M -by-N upper trapezoidal part of the array A m ustcontain the \(m\) atrix to be factorized. On exit, the leading M -by M upper triangularpart of A contains the upper triangular \(m\) atrix \(R\), and elem ents \(M+1\) to \(N\) of the first \(M\) row \(s\) of \(A\), w th the anray \(T A U\), represent the unitary \(m\) atrix \(Z\) as a product of \(M\) elem entary reflectors.

LDA (input)
The leading dim ension of the aray A. LDA >= \(\max (1, M)\).

TAU (output)
The scalar factors of the elem entary reflectors.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthad an ille-

\section*{FURTHER DETAILS}

The factorization is obtained by H ouseholder'sm ethod. The k th transform ation \(m\) atrix, Z ( \(k\) ), w hose conjugate transpose is used to introduce zeros into the ( \(m-k+1\) ) th row of \(A\), is given in the form
\[
\left.\begin{array}{c}
\mathrm{Z}(\mathrm{k})=\left(\begin{array}{ll}
\mathrm{I} & 0
\end{array}\right), \\
\left(\begin{array}{l}
\mathrm{O}
\end{array} \mathrm{~T}(\mathrm{k})\right.
\end{array}\right),
\]
where
\[
\begin{gathered}
T(k)=I-\tan * u(k) * u(k))^{\prime} \quad u(k)=\left(\begin{array}{ll}
1
\end{array}\right), \\
\left(\begin{array}{l}
0
\end{array}\right) \\
(z(k))
\end{gathered}
\]
tau is a scalar and \(z(k)\) is an ( \(n-m\) ) elem ent vector. tau and \(\mathrm{z}(\mathrm{k})\) are chosen to annihilate the elem ents of the kth row of .

The scalartau is retumed in the \(k\) th elem entofTA \(U\) and the vectoru ( \(k\) ) in the \(k\) th row of \(A\), such that the elem ents of \(z(k)\) are in \(a(k, m+1), \ldots, a(k, n)\). The elem ents ofR are retumed in the upper triangularpartofA.
\(Z\) is given by
\[
Z=Z(1) * Z(2) * \ldots * Z(m) .
\]

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
ztzrzf-reduce the M -by \(-\mathrm{N} \quad(\mathrm{M}<=\mathrm{N}\) ) complex upper trapezoidal \(m\) atrix A to upper triangular form by \(m\) eans of unitary transform ations

\section*{SYNOPSIS}

SUBROUTINE ZTZRZFM,N,A,LDA,TAU,WORK,LWORK, \(\mathbb{N} F O\) )
DOUBLE COM PLEX A (LDA ,*),TAU (*),W ORK (*) \(\mathbb{I N}\) TEGERM,N,LDA,LW ORK, \(\mathbb{N} F O\)

SU BROUTINE ZTZRZF_64 M,N,A,LDA,TAU,W ORK,LW ORK, \(\mathbb{N} F O\) )
DOUBLE COM PLEX A (LDA ,*), TAU (*), W ORK (*)
\(\mathbb{N}\) TEGER*8M,N,LDA,LWORK, \(\mathbb{N} F O\)

\section*{F95 INTERFACE}

SU BROUTINE TZRZF ( \(\mathbb{M}], \mathbb{N}], A,[L D A], T A U,[W O R K],[L W O R K],[\mathbb{N} F O])\)
COM PLEX (8),D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER ::M,N,LDA,LW ORK, \(\mathbb{N} F O\)
SU BROUTINE TZRZF_64 (M ], \(\mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L W\) ORK ], \([\mathbb{N} F O])\)
COM PLEX (8),D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
COM PLEX (8),D \(\operatorname{IM}\) ENSIDN (: : : : : A
\(\mathbb{N}\) TEGER (8) ::M , N,LDA,LW ORK, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include < sunperfh>
void ztzrzf(intm , intn, doublecom plex *a, int lda, doublecom plex *tau, int *info);
void ztzrzf_64 (long m, long n, doublecom plex *a, long lda, doublecom plex *tau, long *info);

\section*{PURPOSE}
ztzrzf reduces the M -by -N ( \(\mathrm{M}<=\mathrm{N}\) ) com plex uppertrapezoidal \(m\) atrix \(A\) to upper triangular form by \(m\) eans of unitary transform ations.

The upper trapezoidalm atrix A is factored as
\[
A=\left(\begin{array}{ll}
R & 0
\end{array}\right) * Z,
\]
where \(Z\) is an \(N\)-by \(N\) unitary \(m\) atrix and \(R\) is an \(M\) by \(-M\) upper triangularm atrix.

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{A} . \mathrm{M}>=0\).

N (input) The num ber of collm ns of the m atrix \(\mathrm{A} . \mathrm{N}>=0\).
A (input/output)
On entry, the leading \(\mathrm{M}-\mathrm{by}-\mathrm{N}\) upper trapezoidal part of the array A m ust contain the m atrix to be factorized. On exit, the leading M -by -M upper triangularpart ofA contains the upper triangular \(m\) atrix \(R\), and elem ents \(M+1\) to \(N\) of the first \(M\) row s of \(A\), w ith the array TA \(U\), represent the unitary \(m\) atrix \(Z\) as a productof \(M\) elem entary reflectors.

LD A (input)
The leading dim ension of the aray A. LDA >= max (1, M).

TAU (output)
The scalar factors of the elem entary reflectors.
W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

The dimension of the array \(W\) ORK. LW ORK >= \(m\) ax \((1, M)\). Foroptim um perform anœ \(L W O R K>=M * N B\), w here NB is the optim alblocksize.

If LW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{N F O}=-\) i, the \(i\)-th argum ent had an illegal value

\section*{FURTHER DETAILS}

B ased on contributions by
A. Petitet, C om puterScience D ept., U niv . of Tenn., K noxville, U SA

The factorization is obtained by H ouseholdersm ethod. The \(k\) th transform ation \(m\) atrix, \(Z(k)\), which is used to introduce zeros into the ( \(m-k+1\) )th row ofA, is given in the form
\[
\begin{gathered}
\mathrm{Z}(\mathrm{k})=\left(\begin{array}{ll}
\mathrm{I} & 0
\end{array}\right), \\
\binom{\mathrm{O}}{\mathrm{~T}(\mathrm{k})}
\end{gathered}
\]
where
\[
\begin{gathered}
\left.T(k)=I-\tan { }^{*} u(k) * u(k)\right), \quad u(k)=\left(\begin{array}{ll}
1
\end{array}\right), \\
\left(\begin{array}{l}
0
\end{array}\right) \\
(z(k))
\end{gathered}
\]
tau is a scalar and \(z(k)\) is an ( \(n-m\) ) elem ent vector. tau and \(\mathrm{z}(\mathrm{k})\) are chosen to annihilate the elem ents of the kth row of .

The scalartau is retumed in the kth elem entofTAU and the vectoru ( \(k\) ) in the kth row of A, such that the elem ents of \(z(k)\) are in \(a(k, m+1), \ldots, a(k, n)\). The elem ents ofR are retumed in the upper triangularpartofA.
\(Z\) is given by
\[
Z=Z(1) * Z(2) * \ldots * Z(m) .
\]

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zung2l-generate an \(m\) by \(n\) com plex \(m\) atrix \(Q\) w th orthonormalcolum ns,

\section*{SYNOPSIS}
```

SUBROUTINE ZUNG2L M,N,K,A,LDA,TAU,W ORK,INFO)
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGERM,N,K,LDA,INFO
SU BROUTINE ZUNG2L_64M,N,K,A,LDA,TAU,W ORK,INFO)
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGER*8M,N,K,LDA,\mathbb{NFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNG 2L \(\mathbb{M}, \mathbb{N}], \mathbb{K}], A,[L D A], T A U,[W O R K],[\mathbb{N} F O])\)
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, K, L D A, \mathbb{N} F O\)
SU BROUTINE UNG2L_64 \(\mathbb{M}, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{W} O R K],[\mathbb{N} F O])\)
COM PLEX (8),D \(\mathbb{I}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : ) ::A
\(\mathbb{N} \operatorname{TEGER}\) (8) ::M , N , K , LDA, \(\mathbb{N F}\) O

\section*{C INTERFACE}
\#include <sunperfh>
void zung2l(intm, intn, int , doublecom plex *a, int lda,
void zung21 64 (long m, long \(n\), long \(k\), doublecomplex *a, long lda, doublecom plex *tau, long *info);

\section*{PURPOSE}
zung21L generates an \(m\) by \(n\) com plex \(m\) atrix \(Q w i t h\) orthonorm al colum ns , which is defined as the lastn colum ns of a product ofk elem entary reflectors of orderm
\[
\mathrm{Q}=\mathrm{H}(\mathrm{k}) \ldots \mathrm{H}(2) \mathrm{H}(1)
\]
as retumed by CGEQ LF .

\section*{ARGUMENTS}

M (input) The num ber of row s of the \(m\) atrix \(Q . M>=0\).

N (input) The num ber of colum ns of the \(m\) atrix \(Q . \mathrm{M}>=\mathrm{N}>=\) 0.

K (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q . N>=K>=0\).

A (input/output)
On entry, the \((n-k+i)\)-th colum \(n m\) ust contain the vector which defines the elem entary reflector \(H\) (i), for \(i=1,2, \ldots, k\), as retumed by CGEQLF in the last \(k\) colum ns of its array argum entA. On exit, the \(m\) boy-n m atrix \(Q\).

LD A (input)
The first dimension of the array A. LDA >= \(\max (1, M)\).

TAU (input)
TA U (i) m ust contain the scalar factor of the ele\(m\) entary reflector H (i), as retumed by CGEQ LF.
```

W ORK (w orkspace)

```
dim ension \(\mathbb{N}\) )
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N} F O=-i\), the \(i\)-th argum enthas an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zung2r-generate an \(m\) by \(n\) com plex \(m\) atrix \(Q\) w th orthonor\(m\) alcolum ns,

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZUNG 2R M,N,K,A,LDA,TAU,W ORK,INFO)}
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGERM,N,K,LDA,INFO
SUBROUTINE ZUNG2R_64M,N,K,A,LDA,TAU,W ORK,\mathbb{NFO)}
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGER*8M,N,K,LDA,INFO

```
F95 INTERFACE
    SU BROUTINE UNG 2R \(\mathbb{M}, \mathbb{N}],[K], A,[L D A], T A U,[W O R K],[\mathbb{N} F O])\)
    COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
    COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
    \(\mathbb{N} T E G E R:: M, N, K, L D A, \mathbb{N} F O\)
    SU BROUTINE UNG 2R_64 \(\mathbb{M}, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{W} O R K],[\mathbb{N F O}])\)
    COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU ,W ORK
    COM PLEX (8),D IM ENSION (:,:) ::A
    \(\mathbb{N} T E G E R(8):: M, N, K, L D A, \mathbb{N F O}\)
C INTERFACE
    \#include <sunperfh>
    void zung \(2 r\) (intm , intn, int , doublecom plex *a, int lda,
void zung2r_64 (long m, long n, long k, doublecom plex *a, long lda, doublecom plex *tau, long *info);

\section*{PURPOSE}
zung2rR generates an \(m\) by \(n\) com plex \(m\) atrix \(Q w\) th orthonorm al colum ns, which is defined as the firstn 00 lm ns of a product ofk elem entary reflectors of orderm
\[
Q=H(1) H(2) \ldots H(k)
\]
as retumed by CGEQRF.

\section*{ARGUMENTS}

M (input) The num ber of row sof the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).
N (input) The num ber of colum ns of the \(m\) atrix \(\mathrm{Q} . \mathrm{M}>=\mathrm{N}>=\) 0.
\(K\) (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q . N>=K>=0\).

A (input/output)
On entry, the \(i\)-th columnm ustcontain the vector which defines the elem entary reflectort (i), for i \(=1,2, \ldots, k\), as retumed by CGEQRF in the first \(k\) colum ns of its array argum entA. On exit, the \(m\) by \(n m\) atrix \(Q\).

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CGEQRF.
```

W ORK (w orkspace)

```
dim ension (N)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthas an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zungbr-generate one of the com plex unitary \(m\) atrices \(Q\) or \(P * * H\) determ ined by CGEBRD when reducing a com plex \(m\) atrix A to bidiagonal form

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZUNGBR(NECT,M,N,K,A,LDA,TAU,W ORK,LW ORK,INFO)}
CHARACTER * 1VECT
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGER M,N,K,LDA,LW ORK,\mathbb{NFO}

```

```

CHARACTER * 1VECT
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
\mathbb{N}TEGER*8M,N,K,LDA,LW ORK,\mathbb{NFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNGBR (NECT, M, \(\mathbb{N}], K, A,[L D A], T A U,[W O R K],[L W ~ O R K]\), [ \(\mathbb{N}\) FO ])

CHARACTER (LEN=1) ::VECT
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU ,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, K, L D A, L W O R K, \mathbb{N} F O\)
SU BROUTINE UNGBR_64 NECT, M, \(\mathbb{N}], K, A,[L D A], T A U,[W O R K],[L W ~ O R K]\), [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::VECT
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU ,W ORK

\section*{C INTERFACE}
\#include <sunperfh>
void zungbr(charvect, intm, intn, int \(k\), doublecom plex
*a, int lda, doublecom plex *tau, int *info);
void zungbr_64 (charvect, long m, long n, long k, doublecom -
plex *a, long lda, doublecom plex *tau, long
*info);

\section*{PURPOSE}
zungbr generates one of the com plex unitary m atrioes \(Q\) or \(\mathrm{P} * * \mathrm{H}\) determ ined by CGEBRD when reducing a com plex \(m\) atrix \(A\) to bidiagonal form : \(A=Q * B * P * * H . Q\) and \(P * * H\) are defined as products ofelem entary reflectors \(H\) (i) orG (i) respectively .

IfVECT = Q', A is assum ed to have been an \(M\) boy \(K \mathrm{~K}\) atrix, and \(Q\) is of order \(M\) :
ifm \(>=k, Q=H(1) H(2) \ldots H(k)\) and CUNGBR retums the firstn colum \(n s\) of \(Q\), where \(m>=n>=k\); ifm \(<\mathrm{k}, \mathrm{Q}=\mathrm{H}(1) \mathrm{H}(2) \ldots \mathrm{H}(\mathrm{m}-1)\) and CUNGBR retums \(Q\) as an \(M\) boy \(-M\) matrix.

IfVECT \(=\mathrm{P}\) ', A is assum ed to have been a K -by -N m atrix, and \(P * * H\) is oforderN:
if \(k<n, P^{* *} H=G(k) \ldots G(2) G(1)\) and \(C U N G B R\) retums the firstm row sof \(\mathrm{P} * * \mathrm{H}\), where \(\mathrm{n}>=\mathrm{m}>=\mathrm{k}\); if \(k>=n, P * * H=G(n-1) \ldots G(2) G(1)\) and CUNGBR retums \(\mathrm{P} * * \mathrm{H}\) as an N boy -N m atrix.

\section*{ARGUMENTS}

\section*{VECT (input)}

Specifies w hether the m atrix \(Q\) orthem atrix \(P * * H\)
is required, as defined in the transform ation
applied by CGEBRD :
= Q ': generate Q ;
\(=P^{\prime}:\) generate \(P^{* *} H\).
\(M\) (input) The num ber of row s of the \(m\) atrix \(Q\) orP** \(H\) to be retumed. \(\mathrm{M}>=0\).

N (input) The num ber of \(\propto \mathrm{llm}\) ns of the \(m\) atrix \(Q\) or \(P * * H\) to be retumed. \(\mathrm{N}>=0\). IfVECT \(=Q\) ', \(\mathrm{M}>=\mathrm{N}>=\) \(m\) in \((M, K) ;\) if \(V E C T=P^{\prime}, N>=M>=m\) in \((N, K)\).
\(K\) (input) IfVECT = \(Q\) ', the num berof colum ns in the original \(M\) boy \(K\) matrix reduced by CGEBRD. IfVECT = P ', the num ber of row s in the original K -by N m atrix reduced by CGEBRD. \(\mathrm{K}>=0\).

A (input/output)
O \(n\) entry, the vectors w hich define the elem entary reflectors, as retumed by CGEBRD. On exit, the M toy \(-\mathrm{N} m\) atrix Q orP**H.
LD A (input)
The leading dim ension of the array A.LD A >=M.
TAU (input)
\((m\) in \((M, K))\) ifVECT \(=Q^{\prime}(m\) in \((N, K))\) ifVECT \(=P^{\prime}\)
TAU (i) m ustcontain the scalar factor of the ele\(m\) entary reflectorH (i) orG (i), which determ ines Q or \(P * * H\), as retumed by CGEBRD in its anray argu\(m\) entTAUQ orTAUP.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array W ORK. LW ORK >= \(\max (1, m\) in \(M, N))\). Foroptim um perform ance LW ORK \(>=\) \(m\) in \(M, N) \star N B\), where \(N B\) is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
<0: if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zunghr-generate a complex unitary matrix \(Q\) which is defined as the productof \(\mathbb{H}\) I-HO elem entary reflectors of order \(N\), as retumed by CGEH RD

\section*{SYNOPSIS}

```

DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)

```


```

DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)

```

F95 INTERFACE
    SUBROUTINE UNGHR ( \(\mathbb{N}], \mathbb{I} O, \mathbb{H} I, A,[L D A], T A U,[\mathbb{W}\) ORK ], [LW ORK ], [ \(\mathbb{N} F O]\) )
    COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU ,W ORK
    COM PLEX (8),D \(\mathbb{I}\) ENSION (:,:) ::A
    \(\mathbb{N} T E G E R:: N, \mathbb{L} O, \mathbb{H} I, L D A, L W O R K, \mathbb{N} F O\)
    SU BROUTINE UNGHR_64 ( \(\mathbb{N}], \mathbb{H}, \mathbb{H} I, A,[L D A], T A U,[W O R K],[L W\) ORK ],
        [ \(\mathbb{N}\) FO ])
    COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU ,W ORK
    COM PLEX (8),D IM ENSION (:,:) ::A
    \(\mathbb{N} T E G E R(8):: N, \mathbb{L} O, \mathbb{H} \mathrm{I}, \mathrm{LD} A, L W O R K, \mathbb{N} F O\)
C INTERFACE
    \#include < sunperfh>
void zunghr(intn, intilo, int ihi, doublecom plex *a, int lda, doublecom plex *tau, int *info);
void zunghr_64 (long n, long oill long ihi, doublecom plex *a, long lda, doublecom plex *tau, long *info);

\section*{PURPOSE}
zunghrgenerates a com plex unitary m atrix \(Q\) which is defined as the product of \(\mathbb{H}\) I-ILO elem entary reflectors of orderN, as retumed by C G EH RD :
\(Q=H\) ( 0 ) H ( i (م+1) . . H (ihi-1).

\section*{ARGUMENTS}

N (input) The order of the m atrix \(\mathrm{Q} . \mathrm{N}>=0\).

IIO (input)
\(\mathbb{I} O\) and \(\mathbb{H}\) Im usthave the sam e values as in the previous call of CGEHRD.Q is equal to the unit \(m\) atrix except in the subm atrix
Q (ilo+1: ihi, ilo+1: ihi). \(1<=\mathbb{I} 0<=\mathbb{H} I<=N\), if \(\mathrm{N}>0 ; ~ \Pi \mathrm{O}=1\) and \(\mathrm{H} \mathrm{I}=0\), if \(\mathrm{N}=0\).

IH I (input)
See the description of \(\mathbb{H}\) I.

A (input/output)
O n entry, the vectors w hich define the elem entary reflectors, as retumed by CGEHRD. On exit, the N boy-N unitary m atrix Q .

LD A (input)
The leading dim ension of the aray A. LDA >= \(\max (1, N)\).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectorH (i), as retumed by CGEHRD.

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al LW ORK.

The dim ension of the array \(W\) ORK. LW ORK >= \(\mathbb{H} I-\mathbb{H O}\). For optim um perform ance LW ORK \(>=(\mathbb{H} I-\mathbb{H} O)^{*} N B\), where NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim al size of theW ORK ancay, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
<0: if \(\mathbb{N F O}=-\) i, the \(i\)-th argum ent had an illegal value

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zung12 - generate an \(m\)-by-n com plex \(m\) atrix \(Q\) w th orthonorm alrow s,

\section*{SYNOPSIS}
```

SUBROUTINE ZUNGL2M,N,K,A,LDA,TAU,W ORK,INFO)
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGERM,N,K,LDA,INFO
SU BROUT\mathbb{NE ZUNGL2_64M,N,K,A,LDA,TAU,W ORK,INFO)}
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGER*8M,N,K,LDA,\mathbb{NFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNGL2 ( \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{N} O R K],[\mathbb{N} F O])\)
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, K, L D A, \mathbb{N} F O\)
SU BROUTINE UNGL2_64 ( \(\mathbb{M}], \mathbb{N}],[\mathbb{K}], A,[L D A], T A U,[\mathbb{W} O R K],[\mathbb{N} F O])\)
COMPLEX (8),D \(\mathbb{I}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : ) ::A
\(\mathbb{N} \operatorname{TEGER}\) (8) ::M , N , K , LDA, \(\mathbb{N F}\) O

\section*{C INTERFACE}
\#include <sunperfh>
void zungl2 (intm , intn, intk, doublecom plex *a, int lda,

\section*{PURPOSE}
zungl2 generates an \(m\)-by-n com plex \(m\) atrix \(Q w\) ith orthonorm al row S , w hich is defined as the firstm row s of a productofk elem entary reflectors of ordern
\[
Q=H(k)^{\prime} \ldots H(2) \cdot H(1) '
\]
as retumed by CGELQF.

\section*{ARGUMENTS}

M (input) The num ber of row sof the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).
N (input) The num ber of C lum ns of the m atrix \(\mathrm{Q} . \mathrm{N}>=\mathrm{M}\).
\(K\) (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q . M>=K>=0\).

A (input/output)
On entry, the \(i\)-th row must contain the vector which defines the elem entary reflectort (i), for i
\(=1,2, \ldots, k\), as retumed by CG ELQ F in the first k row sof its amay argum entA. On exit, them by \(n\) \(m\) atrix \(Q\).

LD A (input)
The first dim ension of the array A. LDA >= \(\max (1, M)\).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CG ELQ F.

W ORK (w orkspace)
dim ension M)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthas an illegalvahue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zunglq - generate an M -by N com plex m atrix Q w th orthonorm alrow s,

\section*{SYNOPSIS}
```

SU BROUTINE ZUNGLQ M,N,K,A,LDA,TAU,W ORK,LW ORK,\mathbb{NFO)}
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGERM,N,K,LDA,LW ORK,INFO
SUBROUT\mathbb{NE ZUNGLQ_64M ,N,K,A,LDA,TAU,W ORK,LW ORK,INFO)}
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
\mathbb{N}TEGER*8M,N,K,LDA,LW ORK,\mathbb{NFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNGLQ \(M, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{N} O R K],[L W ~ O R K],[\mathbb{N} F O])\)
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, K, L D A, L W\) ORK, \(\mathbb{N} F O\)
SU BROUTINE UNGLQ_64 \(M, \mathbb{N}],[K], A,[L D A], T A U,[W O R K],[L W O R K]\), [ \(\mathbb{N}\) FO ])

COM PLEX (8),D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R(8):: M, N, K, L D A, L W O R K, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include < sunperfh>
void zunglq (intm , intn, int \(k\), doublecom plex *a, int lda, doublecom plex *tau, int *info);
void zunglq_64 (long m, long \(n\), long \(k\), doublecomplex *a, long lda, doublecom plex *tau, long *info);

\section*{PURPOSE}
zunglq generates an M -by -N com plex \(m\) atrix Q w ith orthonorm al row S , w hich is defined as the firstM row s of a product of K elem entary reflectors of orderN
\[
Q=H(k)^{\prime} \ldots H(2)^{\prime} H(1)^{\prime}
\]
as retumed by CGELQF.

\section*{ARGUMENTS}

M (input) The num ber of row sof the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the \(m\) atrix \(\mathrm{Q} \cdot \mathrm{N}>=\mathrm{M}\).
\(K\) (input) The num ber of elem entary reflectors w hose product
defines the \(m\) atrix \(Q . M>=K>=0\).
A (input/output)
On entry, the \(i\)-th row must contain the vector which defines the elem entary reflectorH (i), for i \(=1,2, \ldots, k\), as retumed by CG ELQ \(F\) in the first \(k\) row sof its amay argum entA. On exit, the \(M\) boy N \(m\) atrix \(Q\).

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CG ELQ F.

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the array W ORK. LW ORK >= max (1,M). Foroptim um perform anœ LW ORK \(>=M\) *NB,
w here NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0 : successfiulexit;
\(<0:\) if \(\mathbb{I N F O}=-i\), the \(i\) th argum enthas an illegalvałue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zungql-generate an M -by N com plex m atrix Q w ith orthonormalcolum ns,

\section*{SYNOPSIS}

SU BROUTINE ZUNGQL \(M, N, K, A, L D A, T A U, W O R K, L W O R K, \mathbb{N} F O\) )
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
\(\mathbb{N}\) TEGER \(M, N, K, L D A, L W O R K, \mathbb{N} F O\)
SU BROUTINE ZUNGQL_64 \(M, N, K, A, L D A, T A U, W\) ORK,LW ORK, \(\mathbb{N} F O\) )

DOUBLE COM PLEX A (LDA ,*), TAU (*),W ORK (*)
\(\mathbb{N}\) TEGER*8 M , N , K , LDA , LW ORK , \(\mathbb{N}\) FO

\section*{F95 INTERFACE}

SU BROUTINE UNGQL M, \(\mathbb{N}],[K], A,[L D A], T A U,[\mathbb{N} O R K],[L W ~ O R K],[\mathbb{N} F O])\)
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, K, L D A, L W\) ORK, \(\mathbb{N} F O\)
SU BROUTINE UNGQL_64 M, \(\mathbb{N}],[K], A,[L D A], T A U,[W O R K],[L W O R K]\), [ \(\mathbb{N} F O\) ])

COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:r:) ::A
\(\mathbb{N} T E G E R(8):: M, N, K, L D A, L W O R K, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include < sunperfh>
void zungql(intm , intn, int \(k\), doublecom plex *a, int lda, doublecom plex *tau, int *info);
void zungql 64 (long m, long \(n\), long \(k\), doublecomplex *a, long lda, doublecom plex *tau, long *info);

\section*{PURPOSE}
zungqlgenerates an M -by -N com plex \(m\) atrix Q w ith orthonorm al colum ns, w hich is defined as the lastN colum ns of a product ofK elem entary reflectors of orderM
\[
Q=H(k) \ldots H(2) H(1)
\]
as retumed by C G EQ LF .

\section*{ARGUMENTS}

M (input) The num ber of row sof the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the matrix Q.M >= \(\mathrm{N} \quad>=\) 0.

K (input) The num ber of elem entary reflectors w hose product defines the m atrix \(\mathrm{Q} \cdot \mathrm{N}>=\mathrm{K}>=0\).

A (input/output)
O \(n\) entry, the \((n-k+i)\)-th colum \(n m\) ust contain the vector which defines the elem entary reflector \(H\) (i), for \(i=1,2, \ldots, k\), as retumed by CGEQLF in the last \(k\) colum ns of its aray argum entA. On exit, the M boy -N m atrix Q .

LD A (input)
The first dimension of the array A. LDA >= \(\max (1, M)\).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectorH (i), as retumed by CG EQ LF .

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, \mathrm{~W}\) ORK ( 1 ) retums the optim al
LW ORK.

LW ORK (input)
The dim ension of the array W ORK. LW ORK >=
\(\max (1, N)\). Foroptim um penform ance LW ORK \(>=N * N B\), where NB is the optim alblocksize.

If LW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent has an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zungqr-generate an M -by N com plex m atrix Q w th orthonormalcolum ns,

\section*{SYNOPSIS}

SU BROUTINE ZUNGQR \(M, N, K, A, L D A, T A U, W O R K \mathbb{N}, L W O R K \mathbb{N}, \mathbb{N} F O\) )
DOUBLE COM PLEX A (LDA ,*),TAU (*),W ORK \(\mathbb{N}\) (*) \(\mathbb{N}\) TEGER \(M, N, K, L D A, L W O R K \mathbb{N}, \mathbb{N} F O\)

SU BROUTINE ZUNGQR_64 M,N,K,A,LDA,TAU,WORK \(\mathbb{N}, L W O R K \mathbb{N}, \mathbb{N} F O)\)
DOUBLE COM PLEX A (LDA , *), TAU (*), WORK \(\mathbb{N}\) (*)
\(\mathbb{N}\) TEGER*8 M , N, K, LDA, LW ORK \(\mathbb{N}, ~ \mathbb{N} F O\)

\section*{F95 INTERFACE}

SU BROUTINE UNGQR \(M, \mathbb{N}],[K], A,[L D A], T A U,[W O R K \mathbb{N}],[L W O R K \mathbb{N}]\), [ \(\mathbb{N} F O\) ])

COMPLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK \(\mathbb{N}\)
COM PLEX (8),D IM ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, K, L D A, L W O R K \mathbb{N}, \mathbb{N} F O\)
SU BROUTINE UNGQR_64 \(\mathbb{M}, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{W} O R K \mathbb{N}],[L W O R K \mathbb{N}]\), [ \(\mathbb{N}\) FO ])

COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK \(\mathbb{N}\)
COMPLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{K}\), LDA, LW ORK \(\mathbb{N}, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zungqr(intm, intn, int \(k\), doublecom plex *a, int lda, doublecom plex *tau, int *info);
void zungqr_64 (long m, long \(n\), long \(k\), doublecomplex *a, long lda, doublecom plex *tau, long *info);

\section*{PURPOSE}
zungqrgenerates an \(M\) boy -N com plex \(m\) atrix \(Q\) w th orthonorm al colum ns, which is defined as the first N colum ns of a product of K elem entary reflectors of orderM
\(\mathrm{Q}=\mathrm{H}(1) \mathrm{H}(2) \ldots \mathrm{H}(\mathrm{k})\)
as retumed by CGEQRF.

\section*{ARGUMENTS}

M (input) The num ber of row sof the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the matrix \(\mathrm{Q} . \mathrm{M}>=\mathrm{N} \quad>=\) 0.

K (input) The num ber of elem entary reflectors w hose product defines the m atrix \(\mathrm{Q} \cdot \mathrm{N}>=\mathrm{K}>=0\).

A (input/output)
On entry, the i-th colum \(n m\) ustcontain the vector which defines the elem entary reflectorH (i), for \(i\)
\(=1,2, \ldots, k\), as retumed by CGEQRF in the first k colum ns of its array argum entA. On exit, the \(M-\) by \(-\mathrm{N} m\) atrix Q .

LDA (input)
The first dimension of the array A. LDA >= \(\max (1, M)\).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflector \(H\) (i), as retumed by CGEQRF.

W ORK \(\mathbb{N}\) (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK \(\mathbb{N}(1)\) retums the optim alLW ORK \(\mathbb{N}\).

LW ORK \(\mathbb{N}\) (input)
The dim ension of the anray \(W\) ORK \(\mathbb{N} . \operatorname{LW}\) ORK \(\mathbb{N}>=\)
\(\max (1, N)\). For optim um perform ance LW ORK \(\mathbb{N}>=\) \(\mathrm{N} * \mathrm{NB}\), where NB is the optim alblocksize.

If LW ORK \(\mathbb{N}=-1\), then a workspace query is assum ed; the routine only calculates the optim al size of the \(W\) ORK \(\mathbb{N}\) array, retums this value as the firstentry of the \(W\) ORK \(\mathbb{N}\) array, and no emor \(m\) essage related to LW ORK \(\mathbb{N}\) is issued by XERBLA.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
< 0: if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthas an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zungr2 - generate an \(m\) by \(n\) com plex \(m\) atrix \(Q\) with orthonorm alrow s,

\section*{SYNOPSIS}
```

SUBROUTINE ZUNGR2M,N,K,A,LDA,TAU,W ORK,INFO)
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGERM,N,K,LDA,INFO
SUBROUT\mathbb{NE ZUNGR2_64M,N,K,A,LDA,TAU,W ORK,INFO)}
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGER*8 M ,N,K,LDA,INFO

```

\section*{F95 INTERFACE}

SU BROUTINE UNGR2 ( \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{N} O R K],[\mathbb{N} F O])\)
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, K, L D A, \mathbb{N} F O\)
SU BROUTINE UNGR2_64 ( \(\mathbb{M}], \mathbb{N}],[\mathbb{K}], A,[L D A], T A U,[\mathbb{W} O R K],[\mathbb{N} F O])\)
COM PLEX (8),D \(\mathbb{I}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : ) ::A
\(\mathbb{N} \operatorname{TEGER}\) (8) ::M , N , K , LDA, \(\mathbb{N F}\) O

\section*{C INTERFACE}
\#include < sunperfh>
void zungr2 (intm, intn, intk, doublecom plex *a, int lda,
void zungr2_64 (long m, long n, long k, doublecom plex *a, long lda, doublecom plex *tau, long *info);

\section*{PURPOSE}
zungr2 generates an \(m\) by \(n\) com plex \(m\) atrix \(Q\) w th orthonorm al row S , w hich is defined as the lastm row s of a product ofk elem entary reflectors of ordern
\[
Q=H(1) \cdot H(2)^{\prime} \ldots H(k)^{\prime}
\]
as retumed by CGERQF.

\section*{ARGUMENTS}

M (input) The num ber of row s of the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).
N (input) The num ber of C lum ns of the m atrix \(\mathrm{Q} . \mathrm{N}>=\mathrm{M}\).
\(K\) (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q . M>=K>=0\).

A (input/output)
O \(n\) entry, the ( \(m-k+i\) )-th row \(m\) ustcontain the vector which defines the elem entary reflector H (i), fori \(=1,2, \ldots, k\), as retumed by CGERQF in the lastk row sof its amay argum entA. O n exit, the \(m-b y-n m\) atrix \(Q\).

LD A (input)
The first dim ension of the array A. LDA >= \(\max (1, M)\).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CG ERQF.

W ORK (w orkspace)
dim ension (M)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0\) : if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthas an illegalvahue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zungrq - generate an M -by N com plex m atrix Q w ith orthonormalrow s,

\section*{SYNOPSIS}

```

DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGERM,N,K,LDA,LW ORK,\mathbb{NFO}
SUBROUT\mathbb{NE ZUNGRQ_64M,N,K,A,LDA,TAU,W ORK,LW ORK, INFO)}
D OUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
\mathbb{N}TEGER*8M,N,K,LDA,LW ORK,\mathbb{NFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNGRQ \(M, \mathbb{N}],[K], A,[L D A], T A U,[W O R K],[L W O R K],[\mathbb{N} F O])\)
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D IM ENSION (:,:) ::A
\(\mathbb{N} T E G E R:: M, N, K, L D A, L W\) ORK, \(\mathbb{N} F O\)

SU BROUTINE UNGRQ_64 \(M, \mathbb{N}],[K], A,[L D A], T A U,[\mathbb{W} O R K],[L W O R K]\), [ \(\mathbb{N} F O\) ])

COM PLEX (8),D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R(8):: M, N, K, L D A, L W O R K, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include < sunperfh>
void zungrq (intm, intn, intk, doublecom plex *a, int lda, doublecom plex *tau, int *info);
void zungrq_64 (long m, long n, long k, doublecom plex *a, long lda, doublecom plex *tau, long *info);

\section*{PURPOSE}
zungrq generates an \(M\) boy -N com plex \(m\) atrix \(Q \mathrm{w}\) ith orthonorm al row \(S\), which is defined as the lastM row s of a productofK elem entary reflectors of orderN
\[
Q=H(1)^{\prime} H(2)^{\prime} \ldots H(k)^{\prime}
\]
as retumed by CGERQF.

\section*{ARGUMENTS}

M (input) The num ber of row sof the m atrix \(\mathrm{Q} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the \(m\) atrix \(\mathrm{Q} \cdot \mathrm{N}>=\mathrm{M}\).
\(K\) (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q . M>=K>=0\).

A (input/output)
On entry, the ( \(m-k+i\) )-th row \(m\) ust contain the vector which defines the elem entary reflector \({ }^{H}\) (i), fori=1,2,..,k, as retumed by CGERQF in the lastk row sof its array argum entA. O n exit, the M toy-N m atrix Q .

LD A (input)
The first dim ension of the array A. LDA >= max (1, M).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CGERQF.

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dimension of the anay \(W\) ORK. LW ORK >= \(m a x(1, M)\). Foroptim um perform anœ LW ORK \(>=M\) *NB,
w here NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{I N F O}=-i\), the i-th argum enthas an illegalvałue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zungtr-generate a complex unitary matrix \(Q\) which is defined as the product of \(n-1\) elem entary reflectors of order N , as retumed by CH ETRD

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZUNGTR (UPLO,N,A,LDA,TAU,W ORK,LW ORK,INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
INTEGERN,LDA,LW ORK,NNFO

```
SU BROUTINE ZUNGTR_64 (UPLO,N,A,LDA,TAU,W ORK,LW ORK, \(\mathbb{N} F O\) )
CHARACTER * 1 UPLO
DOUBLE COM PLEX A (LDA,*),TAU (*),W ORK (*)
\(\mathbb{N}\) TEGER*8N,LDA,LW ORK, \(\mathbb{N} F O\)

\section*{F95 INTERFACE}

SU BROUTINE UNGTR (UPLO, \(\mathbb{N}], A,[L D A], T A U,[W O R K],[L W O R K],[\mathbb{N} F O])\)
CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU ,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N}\) TEGER :: \(\mathrm{N}, \mathrm{LDA}, \mathrm{LW}\) ORK, \(\mathbb{N} F O\)

SU BROUTINE UNGTR_64 (UPLO, \(\mathbb{N}], A,[L D A], T A U,[\mathbb{W} O R K],[L W O R K],[\mathbb{N F O}])\)
CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A
\(\mathbb{N} T E G E R(8):: N, L D A, L W O R K, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zungtr(charuplo, intn, doublecom plex *a, int lda, doublecom plex *tau, int*info);
void zungtr_64 (char uplo, long n, doublecom plex *a, long lda, doublecom plex *tau, long *info);

\section*{PURPOSE}
zungtrgenerates a com plex unitary \(m\) atrix \(Q\) which is defined as the product of \(n-1\) elem entary reflectors of order \(N\), as retumed by CHETRD :
if \(U P L O=U ', Q=H(n-1) \ldots H(2) H(1)\),


\section*{ARGUMENTS}

UPLO (input)
= U ':U ppertriangle of A contains elem entary
reflectors from CHETRD; = L':Low ertriangle of A
contains elem entary reflectors from CHETRD.
N (input) The order of the m atrix \(\mathrm{Q} . \mathrm{N}>=0\).
A (input/output)
On entry, the vectors which define the elem entary reflectors, as retumed by CHETRD. On exit, the N boy-N unitary m atrix Q .

LD A (input)
The leading dim ension of the array A .LD A \(>=\mathrm{N}\).
TAU (input)
TAU (i) m ust contain the scalar factor of the elem entary reflectort (i), as retumed by CH ETRD.

W ORK (w orkspace)
On exit, if \(\mathbb{N F F O}=0, \mathrm{~W}\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. LW ORK \(>=N-1\).
For optim um perform ance \(L W\) ORK \(>=(\mathbb{N}-1)^{\star} N B\), where

NB is the optim alblocksize.

IfLW O RK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0\) : if \(\mathbb{N} F O=-i\), the \(i\) th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zunm br-VECT = Q ', CUNM BR overw rites the general com plex M -by-N m atrix C w ith \(S \mathbb{D} E=\mathrm{L} \mathrm{S}^{2} \mathrm{D} E=\mathrm{R}\) 'TRANS \(=\mathrm{N}^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZUNM BR NECT,SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,}
W ORK,LW ORK,INFO)
CHARACTER * 1VECT,SIDE,TRANS
DOUBLE COM PLEX A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
\mathbb{NTEGER M,N,K,LDA,LDC,LW ORK,NNFO}
SUBROUTINE ZUNM BR_64 NECT,SDDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,
W ORK,LW ORK,\mathbb{NFO)}
CHARACTER * 1VECT,SIDE,TRANS
D OUBLE COM PLEX A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
\mathbb{NTEGER*8M,N,K,LDA,LDC,LW ORK, INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNM BR \(N E C T, S \mathbb{D} E,[T R A N S], \mathbb{M}], \mathbb{N}], K, A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::VECT,SDE,TRANS
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU, W ORK
COMPLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER ::M , N, K,LDA,LDC,LW ORK, \(\mathbb{N} F O\)
SU BROUTINE UNMBR_64 (NECT,SDE, [TRANS], \(\mathbb{M}], \mathbb{N}], K, A,[L D A], T A U\), C, [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::VECT,SDE,TRANS

\section*{C INTERFACE}
\#include < sunperfh>
void zunm br(charvect, char side, char trans, intm, int n, int k , doublecom plex *a, int lda, doublecom plex *tau, doublecom plex *c, int ldc, int *info);
void zunm br_64 (charvect, charside, char trans, long m, long \(n\), long \(k\), doublecom plex *a, long lda, doublecom plex *tau, doublecom plex *c, long ldc, long *info);

\section*{PURPOSE}
zunm brVECT = Q ', CUNM BR overw rites the generalcom plex M by \(-\mathrm{N} m\) atrix \(\mathrm{C} w\) ith
```

        \(S \mathbb{D E}=\mathrm{L}^{\prime} \quad \mathrm{SDE}=\) R'TRANS \(^{\prime} \mathrm{N}^{\prime}:\)
    $Q * C \quad C * Q T R A N S=C: \quad Q * * H * C \quad C *$
Q**H

```
IfVECT = P', CUNM BR overw rites the general com plex M -by-N
\(m\) atrix \(C\) w ith
    \(S \mathbb{D} E=L^{\prime} \quad S \mathbb{D} E=R^{\prime}\)
TRANS = N': \(\quad \mathrm{P}\) * C \(\quad \mathrm{C}\) * P
TRANS \(=C\) : \(\quad P * * H * C \quad C * P * * H\)

H ere Q and \(\mathrm{P} * * \mathrm{H}\) are the unitary m atrioes determ ined by CGEBRD when reducing a com plex m atrix A to bidiagonal form : \(A=Q * B * P * * H . Q\) and \(P * * H\) are defined as products of ele\(m\) entary reflectors \(H\) (i) and \(G\) (i) respectively.

Letnq \(=m\) if \(S \mathbb{D} E=L\) 'and \(n q=n\) if \(S \mathbb{D} E=R\) '. Thus \(n q\) is the order of the unitary matrix \(Q\) or \(P * * H\) that is applied.

IfVECT = \(Q\) ', A is assum ed to have been an \(N Q\)-by \(K\) m atrix:
if \(n q>=k, Q=H(1) H(2) \ldots H(k)\);
ifnq \(<k, Q=H(1) H(2) \ldots H(n q-1)\).

IfVECT = \(\mathrm{P}^{\prime}, \mathrm{A}\) is assum ed to have been a \(\mathrm{K}-\) by -N Q m atrix:
ifk < nq, P = G (1) G (2) ... G (k);
if \(k>=n q, P=G(1) G(2) \ldots G(n q-1)\).

\section*{ARGUMENTS}
```

VECT (input)
= Q ': apply Q orQ **H ;
= P': apply P orP**H .

```

SID E (input)
\(=L^{\prime}:\) apply \(Q, Q * * H, P\) orP**H from the Left;
\(=R\) ': apply \(Q, Q * * H, P\) orP**H from the Right.

TRANS (input)
\(=\mathrm{N}:\) : N o transpose, apply Q orP;
\(=\mathrm{C}\) : C onjugate transpose, apply \(\mathrm{Q}^{* *} \mathrm{H}\) orP**H .

TRAN \(S\) is defaulted to \(N\) 'forF \(95 \mathbb{I N}\) TERFACE.

M (input) The num ber of row s of the \(m\) atrix \(C . M>=0\).
N (input) The num ber of colum ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\).

K (input) IfVECT = Q ', the num ber of colum ns in the original \(m\) atrix reduced by CGEBRD. IfVECT = \(P\) ', the num ber of row \(S\) in the originalm atrix reduced by CGEBRD. \(\mathrm{K}>=0\) 。

\(\mathrm{VECT}=\mathrm{P}\) 'The vectors w hich define the elem entary reflectors H (i) and G (i), whose products determ ine the \(m\) atrioes \(Q\) and \(P\), as returned by C G EBRD.

LD A (input)
The leading dim ension of the array A. If VECT = Q', LDA >= max (1,nq); if VECT = P',LDA >= \(m\) ax \((1, m\) in \((n q, K))\).

TAU (input)
TAU (i) m ustcontain the scalar factor of the ele\(m\) entary reflectorH (i) orG (i) which determ ines Q orP, as retumed by CGEBRD in the array argum ent TAUQ orTAUP.

C (input/output)
On entry, the \(M\) boy \(-\mathrm{N} m\) atrix C . On exit, C is overw rilten by Q * C or \(\mathrm{Q} * * \mathrm{H} * \mathrm{C}\) or C \(\mathrm{Q} * * \mathrm{H}\) or C * Q or \(\mathrm{P} * \mathrm{C}\) or \(\mathrm{P} * * \mathrm{H}\) * C or C * P or \(\mathrm{C}{ }^{\mathrm{P}}{ }^{* *} \mathrm{H}\).

LD C (input)
The leading dim ension of the array C.LDC >= \(\max (1, M)\).

W ORK (w orkspace)

On exit, if \(\mathbb{N F} F=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. IfSDE \(=L\) ', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LW ORK >= \(\max (1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=L '\), and \(L W O R K>=M * N B\) if \(S \mathbb{D} E=R\) ', where NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfiulexit
<0: if \(\mathbb{I N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zunm hr-overw rite the general com plex M -by -N m atrix C w ith \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime}\) TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUTINE ZUNMHR(S\mathbb{DE,TRANS,M ,N,\mathbb{LO,}\mathbb{H}I,A,LDA,TAU,C,LDC,}
W ORK,LW ORK,INFO)
CHARACTER * 1SIDE,TRANS
DOUBLE COM PLEX A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
INTEGER M ,N,\mathbb{LO,\mathbb{HI},LDA,LDC,LW ORK,INFO}

```

```

    LDC,W ORK,LW ORK,\mathbb{NFO)}
    CHARACTER * 1SIDE,TRANS
D OUBLE COM PLEX A (LDA,*),TAU (*),C (LDC,*),W ORK (*)

```


\section*{F95 INTERFACE}

SU BROUTINE UNM HR (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}], \mathbb{Z}, \mathbb{H} I, A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SDE,TRANS
COM PLEX (8),D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (: : : : : : A , C
\(\mathbb{N} T E G E R:: M, N, \mathbb{L O}, \mathbb{H} I, L D A, L D C, L W O R K, \mathbb{N} F O\)
SUBROUTINE UNMHR_64 (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}], \mathbb{L} O, \mathbb{H} I, A,[L D A], T A U\), C, [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SIDE,TRANS

COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER (8) ::M ,N, \(\mathbb{L} O, \mathbb{H} \mathrm{I}, \mathrm{LD} A, L D C, L W O R K, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zunm hr(char side, chartrans, intm, int \(n\), int ilo, int ini, doublecom plex *a, int lda, doublecom plex *tau, doublecom plex *c, int ldc, int *info);
void zunm hr_64 (char side, char trans, long m, long n, long ilo, long ihi, doublecom plex *a, long lda, doublecom plex *tau, doublecom plex *c, long ldc, long *info);

\section*{PURPOSE}
zunm hroverw rites the generalcom plex M -by -N m atrix C w ith TRANS \(=\mathrm{C}: \quad \mathrm{Q} * \mathrm{H}_{\mathrm{H}}{ }^{*} \mathrm{C} \quad \mathrm{C} * \mathrm{Q} *{ }^{*} \mathrm{H}\)
\(w\) here \(Q\) is a com plex unitary \(m\) atrix of ordernq, \(w\) th nq \(=m\) if \(S \mathbb{D} E=\mathrm{L}\) 'and \(\mathrm{nq}=\mathrm{n}\) if \(S \mathbb{D} E=R\) '. Q is defined as the productof IH I-HO elem entary reflectors, as retumed by CGEHRD:


\section*{ARGUMENTS}

SIDE (input)
\(=\mathrm{L}\) ': apply Q orQ \({ }^{* *} \mathrm{H}\) from the Left;
\(=\mathrm{R}\) ': apply Q orQ **H \(_{\mathrm{H}}\) from the R ight.

TRANS (input)
\(=\mathrm{N}\) ': apply Q (Notranspose)
= C ': apply Q **H (C onjugate transpose)
TRANS is defaulted to \(N\) 'forF \(95 \mathbb{N}\) TERFACE .

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{C} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix C. \(\mathrm{N}>=0\).
ㅍO (input)
\(\mathbb{H O}\) and \(\mathbb{H}\) Im usthave the sam e values as in the previous call of CGEHRD.Q is equal to the unit
\(m\) atrix except in the subm atrix
Q (ilo+1: :hi,ilo+1: :hi). IfS \(\mathbb{D} E=\mathrm{L}\) ', then \(1<=\) \(\mathbb{H}<=\mathbb{H} I<=M\), if \(M>0\), and \(\mathbb{H O}=1\) and \(\mathbb{H} I=\) 0 , if \(M=0\); if \(S \mathbb{D} E=R\) ', then \(1<=\mathbb{L O}<=\mathbb{H} I\) \(<=N\), if \(N>0\), and \(\mathbb{H} O=1\) and \(\mathbb{H} I=0\), if \(N=0\).

IH I (input)
See the description of IIO .

A (input) (LDA, M) if \(S \mathbb{D} E=L^{\prime}(\mathbb{L D A}, N)\) if \(S \mathbb{D} E=R^{\prime} T\) The vectors w hich define the elem entary reflectors, as retumed by CGEHRD.

LD A (input)
The leading dim ension of the anay A. LDA >= \(\max (1, M)\) ifSD \(E=L ; L D A>=\max (1, N)\) if \(S \mathbb{D} E=\) R.

TAU (input)
\((M-1)\) if \(S \mathbb{D} E=L^{\prime}(N-1)\) if \(S \mathbb{D E}=R^{\prime} T A U(i)\)
\(m\) ust contain the scalar factor of the elem entary
reflectorH (i), as retumed by CGEHRD.
C (input/output)
On entry, the M -by-N matrix C. On exit, C is overw rilten by \(\mathrm{Q} * \mathrm{C}\) or \(\mathrm{Q} * * \mathrm{H} * \mathrm{C}\) or \(\mathrm{C} \mathrm{Q}^{*} * \mathrm{H}_{\mathrm{H}}\) orC \({ }^{\mathrm{Q}} \mathrm{Q}\).

LD C (input)
The leading dim ension of the aray C. LD C >= max (1, M).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK ( 1 ) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. If \(S \mathbb{D} E=L\) ', LW ORK >= max ( \(1, \mathrm{~N}\) ); if \(S \mathbb{D} E=R\) ', LW ORK \(>=\) \(\max (1, M)\). Foroptim um perform ance LW ORK >= N *NB if \(S \mathbb{D} E=L\) ', and LW ORK \(>=M * N B\) if \(S \mathbb{D} E=R\) ', where NB is the optim alblocksize.

IfLW O RK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
\(=0\) : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthad an illegalvahue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zunm 12 -overw rite the general com plex \(m\)-by-n matrix \(C\) with \(Q\) * \(C\) if \(S \mathbb{D} E=\mathbb{L}\) 'and TRANS \(=N\) ', or \(Q * C\) if \(S \mathbb{D} E=\) L'and TRANS = C', or \(C * Q\) if \(S \mathbb{D} E=R '\) and TRANS = \(N\) ', or \(C\) * \(Q\) 'if \(S \mathbb{D E}=R\) 'and TRANS \(=C^{\prime}\),

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZUNM L2(S\mathbb{DE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{M},\textrm{M},\textrm{L}
\mathbb{NFO)}

```
CHARACTER * 1 SIDE,TRANS
DOUBLE COM PLEXA (LDA, *), TAU (*), C (LDC,\(\left.^{\star}\right), W\) ORK (*)
\(\mathbb{N}\) TEGER M, N, K,LDA,LDC, \(\mathbb{N} F O\)

SU BROUTINE ZUNM L2_64 (SDE,TRANS, M,N,K,A,LDA,TAU, C,LDC,WORK, \(\mathbb{N} F O\) )

CHARACTER * 1 SIDE,TRANS

\(\mathbb{N}\) TEGER*8 M , N , K , LDA , LD C , \(\mathbb{N}\) FO

\section*{F95 INTERFACE}

SU BROUTINE UNM L2 (SDE,TRANS, M ], \(\mathbb{N}],[K], A,[L D A], T A U, C,[L D C]\), [W ORK], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::SDE,TRANS
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU ,W ORK
COMPLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER :: M , N, K, LDA, LD C, \(\mathbb{N} F O\)
SU BROUT \(\mathbb{N} E\) UNM L2_64 (S \(\mathbb{D} E, T R A N S, \mathbb{M}], \mathbb{N}], \mathbb{K}], A,[L D A], T A U, C\),
[LDC], [WORK], [NFO])

CHARACTER ( \(\amalg E N=1\) ) : : SDE E ,TRANS
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::TAU ,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: \(:\) :) : : A , C
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LD} A, L D C, \mathbb{N F O}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zunm 12 (charside, chartrans, int \(m\), int \(n\), int \(k\), doublecom plex *a, int lda, doublecom plex *tau, doublecom plex *c, int ldc, int*info);
void zunm 12_64 (charside, chartrans, long m, long n, long k, doublecom plex *a, long lda, doublecom plex *tau, doublecom plex *c, long ldc, long *info);

\section*{PURPOSE}
zunm 12 overw rites the general com plex m -by-n m atrix C w ith
\(w\) here \(Q\) is a com plex untary \(m\) atrix defined as the product ofk elem entary reflectors
\[
\mathrm{Q}=\mathrm{H}(\mathrm{k})^{\prime} \ldots \mathrm{H}(2)^{\prime} \mathrm{H}(1)^{\prime}
\]
as retumed by CGELQF.Q is of orderm ifSIDE = L'and of ordern if \(S \mathbb{D} E=R\).

\section*{ARGUMENTS}

SIDE (input)
= L ': apply Q orQ 'from the Left
\(=R\) ': apply Q or Q 'from the R ight

TRANS (input)
\(=\mathrm{N}^{\prime}:\) apply \(\mathrm{Q} \quad(\mathbb{N}\) o transpose)
= C ': apply Q ' (C onjugate transpose)

M (input) The num ber of row s of the \(m\) atrix \(C . M>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\).

K (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q\). IfSID \(E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0\);
ifS \(\mathbb{D} E=R{ }^{\prime}, N>=K>=0\).

A (input) (LDA,M) if \(S \mathbb{D} E=L \prime\), (LDA,N) if \(S D E=R^{\prime}\) The i-th row \(m\) ust contain the vectorw hich defines the elem entary reflector \(H\) ( \(i\) ), for \(i=1,2, \ldots, k\), as retumed by CGELQF in the firstk row sof its array argum entA. A ism odified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A. LD A >= max \((1, K)\).

TAU (input)
TAU (i) must contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CG ELQ F.

C (input/output)
On entry, them -by \(n \mathrm{~m}\) atrix C . On exit, C is overw rilten by Q * C or \(\mathrm{Q}{ }^{*} \mathrm{C}\) orC \({ }^{*} \mathrm{Q}\) 'orC \({ }^{*} \mathrm{Q}\).

LD C (input)
The leading dim ension of the aray C. LD C >= \(m a x(1, M)\).

W ORK (w orkspace)
\((\mathbb{N})\) if \(\left.S \mathbb{D} E=L^{\prime}, M\right)\) if \(S \mathbb{D} E=R^{\prime}\)
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
< 0 : if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zunm lq -overw rite the general com plex M -by-N m atrix C w ith \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime}\) TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZUNM LQ (S\mathbb{DE,TRANS,M ,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{T},\textrm{T},\textrm{T}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
DOUBLE COM PLEX A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
\mathbb{NTEGER M,N,K,LDA,LDC,LW ORK,NNFO}
SUBROUT\mathbb{NE ZUNM LQ_64(SDDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SDDE,TRANS
DOUBLE COM PLEX A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
\mathbb{NTEGER*8M,N,K,LDA,LDC,LW ORK, INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNM LQ (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C,[L D C]\), [W ORK], [LW ORK ], [ \(\mathbb{N} F \mathrm{O}\) ])

CHARACTER (LEN=1) ::SDE,TRANS
COM PLEX (8),D \(\mathbb{I M} E N S I O N(:):: T A U, W\) ORK
COMPLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER ::M,N,K,LDA,LDC,LW ORK, \(\mathbb{N} F O\)
SUBROUTINE UNMLQ_64 (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [NFO])

CHARACTER (LEN=1) ::SDE,TRANS

COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER ( 8 ) :: M , N , K, LDA, LD C, LW ORK, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zunm lq (char side, chartrans, int \(m\), int \(n\), int \(k\), doublecom plex *a, int lda, doublecom plex *tau, doublecom plex *c, int ldc, int*info);
void zunm lq_64 (char side, chartrans, long m, long n, long k, doublecom plex *a, long lda, doublecom plex *tau, doublecom plex *C, long ldc, long *info);

\section*{PURPOSE}
zunm lq overw rites the general com plex M -by -N m atrix C w ith TRANS = C : \(\quad Q * * H * C \quad C * Q * *\)
where \(Q\) is a com plex unitary \(m\) atrix defined as the product ofk elem entary reflectors
\[
Q=H(k)^{\prime} \ldots H(2)^{\prime} H(1) '
\]
as retumed by CGELQF.Q is oforderM ifSIDE = L'and of orderN ifsIDE = R'.

\section*{ARGUMENTS}

SIDE (input)
\(=\mathrm{L}\) ': apply Q orQ \({ }^{* *} \mathrm{H}\) from the Left;
\(=\mathrm{R}\) ': apply Q orQ **H \(_{\mathrm{H}}\) from the R ight.

TRANS (input)
= N ': N o transpose, apply Q ;
= C ': C onjugate transpose, apply Q **H .
TRANS is defaulted to \(N\) 'forF95 \(\mathbb{N}\) TERFACE.

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{C} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix C. \(\mathrm{N}>=0\).
\(K\) (input) The num ber of elem entary reflectors \(w\) hose product defines the \(m\) atrix \(Q\). If \(S \mathbb{D} E=L ', M \quad=K>=0\); if \(S \mathbb{D} E=R \prime, N>=K>=0\).
 \(i\)-th row m ustcontain the vectorw hich defines the elem entary reflector \(H\) ( \(i\) ), for \(i=1,2, \ldots, k\), as retumed by CGELQF in the firstk row sof its array argum entA. A ism odified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, K)\).

TAU (input)
TAU (i) m ustcontain the scalar factor of the ele\(m\) entary reflectorH (i), as retumed by CG ELQ F.

C (input/output)
On entry, the \(M-b y-N m\) atrix \(C\). On exit, \(C\) is overw ritten by Q * C or \(\mathrm{Q} * * \mathrm{H} * \mathrm{C}\) or C * \(\mathrm{Q} * \mathrm{H}\) or C * Q .

LD C (input)
The leading dim ension of the array \(C . \operatorname{LDC}>=\) \(\max (1, M)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the amay W ORK. IfSIDE = L', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LWORK >= \(\max (1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D E} L^{\prime}\) ', and LW ORK \(>=M * N B\) if \(S \mathbb{D} E=R\) ',
w here N B is the optim alblocksize.

If LW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zunm ql-overw rite the general com plex M -by -N m atrix C w th \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R{ }^{\prime}\) TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZUNM QL (S\mathbb{DE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{M},\textrm{T},\textrm{T}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
DOUBLE COM PLEX A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
\mathbb{NTEGER M,N,K,LDA,LDC,LW ORK,NNFO}
SUBROUT\mathbb{NE ZUNMQL_64 (SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SDDE,TRANS
DOUBLE COM PLEX A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
\mathbb{NTEGER*8M,N,K,LDA,LDC,LW ORK, INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNM QL (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C,[L D C]\), [W ORK], [LW ORK], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::SDE,TRANS
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COMPLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A C
\(\mathbb{N}\) TEGER ::M,N,K,LDA,LDC,LW ORK, \(\mathbb{N} F O\)
SUBROUTINE UNM QL_64 (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SDE,TRANS

COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER ( 8 ) :: M , N , K, LDA, LD C, LW ORK, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zunm ql(charside, chartrans, int \(m\), int \(n\), int \(k\), doublecom plex *a, int lda, doublecom plex *tau, doublecom plex *c, int ldc, int*info);
void zunm ql_64 (char side, chartrans, long m, long n, long k, doublecom plex *a, long lda, doublecom plex *tau, doublecom plex *c, long ldc, long *info);

\section*{PURPOSE}
zunm qloverw rites the general com plex M -by -N m atrix C w ith TRANS = C : \(\quad Q * * H * C \quad C * Q * *\)
where \(Q\) is a com plex unitary \(m\) atrix defined as the product ofk elem entary reflectors
\[
Q=H(k) \ldots H(2) H(1)
\]
as retumed by CGEQ LF . Q is oforderM ifSIDE= L'and of orderN if \(S \mathbb{D} E=R\).

\section*{ARGUMENTS}

STDE (input)
\(=\mathrm{L}\) ': apply Q orQ \({ }^{* *} \mathrm{H}\) from the Left;
\(=\mathrm{R}\) ': apply Q orQ **H \(_{\mathrm{H}}\) from the R ight.

TRANS (input)
\(=\mathrm{N}\) ': N o transpose, apply Q ;
= C ': Transpose, apply Q **H .
TRANS is defaulted to \(N\) 'forF \(95 \mathbb{N}\) TERFACE .

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{C} . \mathrm{M}>=0\).
\(N\) (input) The num ber of colum ns of the m atrix C. \(\mathrm{N}>=0\).
\(K\) (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q\). If \(S \mathbb{D} E=L ', M \quad=K>=0\); if \(S \mathbb{D} E=R \prime, N>=K>=0\).

A (input) The i-th colum \(n\) must contain the vector which defines the elem entary reflector \(H\) (i), for \(i=\) \(1,2, \ldots, k\), as retumed by CGEQ LF in the last \(k\) colum ns of its array argum entA. A ism odified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A . If \(\mathrm{S} \mathbb{D} \mathrm{E}=\) L', LDA \(>=\max (1, M)\); if \(S D E=R '\) LDA \(>=\) \(\max (1, \mathbb{N})\).

TAU (input)
TAU (i) \(m\) ustcontain the scalar factorof the ele\(m\) entary reflectort (i), as retumed by CG EQ LF .

C (input/output)
On entry, the \(M\) boy \(-N m\) atrix C. On exit, \(C\) is overw ritten by \(\mathrm{Q} * \mathrm{C}\) or \(\mathrm{Q} * \mathrm{H}^{*} \mathrm{C}\) or \(\mathrm{C} \mathrm{Q}_{\mathrm{Q}}{ }^{* *} \mathrm{H}\) or C Q .

LD C (input)
The leading dim ension of the array C.LD C >= \(\max (1, \mathrm{M})\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK ( 1 ) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the aray \(W\) ORK. IfSDE \(=\mathrm{L}\) ', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LW ORK \(>=\) \(\max (1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=L\) ', and LW ORK \(>=M * N B\) if \(S \mathbb{D} E=R\) ', where NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zunm qr-overw rite the general com plex M -by -N m atrix C w th \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime}\) TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZUNMQR(SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
DOUBLE COM PLEX A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
\mathbb{NTEGER M,N,K,LDA,LDC,LW ORK,NNFO}
SUBROUT\mathbb{NE ZUNM QR_64 (S\mathbb{DE,TRANS,M ,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{T},\textrm{L},\textrm{L}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
DOUBLE COM PLEX A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
\mathbb{NTEGER*8M,N,K,LDA,LDC,LW ORK, INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNM QR (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C,[L D C]\), [W ORK], [LW ORK], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::SDE,TRANS
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COMPLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER ::M ,N,K,LDA,LDC,LW ORK, \(\mathbb{N} F O\)
SUBROUTINE UNM QR_64 (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SDE,TRANS

COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER ( 8 ) :: M , N , K, LDA, LD C, LW ORK, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zunm qr(char side, chartrans, int \(m\), int \(n\), int \(k\), doublecom plex *a, int lda, doublecom plex *tau, doublecom plex *c, int ldc, int*info);
void zunm qr_64 (char side, char trans, long m, long n, long k, doublecom plex *a, long lda, doublecom plex *tau, doublecom plex *c, long ldc, long *info);

\section*{PURPOSE}
zunm qroverw rites the general com plex M -by -N m atrix C w th TRANS \(=C: \quad Q * * H * C \quad C * Q * H\)
where \(Q\) is a com plex unitary \(m\) atrix defined as the product ofk elem entary reflectors
\[
Q=H(1) H(2) \ldots H(k)
\]
as retumed by CGEQRF.Q is oforderM ifSIDE = L'and of orderN if \(S \mathbb{D} E=R\).

\section*{ARGUMENTS}

STDE (input)
\(=\mathrm{L}\) ': apply Q orQ \({ }^{* *} \mathrm{H}\) from the Left;
\(=\mathrm{R}\) ': apply Q orQ **H \(_{\mathrm{H}}\) from the R ight.

TRANS (input)
= N ': N o transpose, apply Q ;
= C ': C onjugate transpose, apply Q **H .
TRANS is defaulted to \(N\) 'forF \(95 \mathbb{N}\) TERFACE .

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{C} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix C. \(\mathrm{N}>=0\).
\(K\) (input) The num ber of elem entary reflectors \(w\) hose product defines the \(m\) atrix \(Q\). If \(S \mathbb{D} E=L ', M \quad=K>=0\); if \(S \mathbb{D} E=R \prime, N>=K>=0\).

A (input) The i-th colum \(n\) must contain the vector which defines the elem entary reflector \(H\) (i), for \(i=\) \(1,2, \ldots, k\), as retumed by CGEQRF in the first \(k\) colum ns of its array argum entA. A ism odified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A . If \(\mathrm{S} \mathbb{D} \mathrm{E}=\) L', LDA >= \(\max (1, M)\); if \(S \mathbb{D E}=R^{\prime}, L D A>=\) \(\max (1, N)\).

TAU (input)
TAU (i) \(m\) ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CGEQRF.

C (input/output)
On entry, the \(M\) by -N matrix C. On exit, C is overw rilten by \(\mathrm{Q} * \mathrm{C}\) or \(\mathrm{Q} * * \mathrm{H} * \mathrm{C}\) or \(\mathrm{C} \mathrm{Q}_{\mathrm{Q}}{ }^{* *} \mathrm{H}\) orC \({ }^{\mathrm{Q}} \mathrm{Q}\).

LD C (input)
The leading dim ension of the aray C. LD C >= max (1,M).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the array \(W\) ORK. IfSDE \(=\mathrm{L}\) ', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LW ORK >= \(\max (1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=L '\), and \(L W\) ORK \(>=M * N B\) if \(S \mathbb{D} E=R\) ', where NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim al size of
the W ORK array, retums this value as the first
entry of the W ORK array, and no errorm essage
related to LW ORK is issued by XERBLA.
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum ent had an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zunm r2 -overw rite the general com plex \(m\)-by- \(n m\) atrix \(C\) with
\(Q\) * \(C\) if \(S \mathbb{D} E=\mathbb{L}\) 'and TRANS \(=N\) ', or \(Q * C\) if \(S \mathbb{D} E=\) L'and TRANS = C', or \(C * Q\) ifS \(\mathbb{D} E=R\) ' and TRANS = \(N\) ', or \(C * Q\) 'if \(S \mathbb{D E}=R\) 'and TRANS = \(C^{\prime}\),

\section*{SYNOPSIS}
```

SU BROUT\mathbb{NE ZUNMR2(S\mathbb{DE,TRANS,M ,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{T},\textrm{L}
\mathbb{NFO )}

```
CHARACTER * 1 SIDE,TRANS

\(\mathbb{N}\) TEGER M, N, K,LDA,LDC, \(\mathbb{N} F O\)
SU BROUTINE ZUNMR2_64 (SDE,TRANS, M,N,K,A,LDA,TAU, C,LDC,W ORK,
    \(\mathbb{N} F O\) )
CHARACTER * 1 SIDE,TRANS

\(\mathbb{N}\) TEGER*8 M , N , K , LDA , LD C , \(\mathbb{N}\) FO

\section*{F95 INTERFACE}

SUBROUTINE UNMR2 (SDE, TRANS, M ], \(\mathbb{N}],[K], A,[L D A], T A U, C,[L D C]\), [W ORK ], [ \(\mathbb{N} F O\) ])

CHARACTER (LEN=1) ::SDE,TRANS
COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU ,W ORK
COMPLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER :: M , N, K, LDA, LD C, \(\mathbb{N} F O\)
SUBROUTINE UNMR2_64 (SDE,TRANS, \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C\),
[LDC], [WORK], [NFO])

CHARACTER ( \(\amalg E N=1\) ) : : SDE E ,TRANS
COM PLEX (8), D \(\mathbb{M}\) ENSION (:) ::TAU ,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (: \(:\) :) : : A , C
\(\mathbb{N} \operatorname{TEGER}(8):: \mathrm{M}, \mathrm{N}, \mathrm{K}, \mathrm{LD} A, L D C, \mathbb{N F O}\)

\section*{C INTERFACE}
\#include <sunperfh>
void zunm r 2 (charside, chartrans, int \(m\), int \(n\), int \(k\), doublecom plex *a, int lda, doublecom plex *tau, doublecom plex * C , int ldc, int *info);
void zunm 22 _64 (char side, chartrans, long m, long n, long k, doublecom plex *a, long lda, doublecom plex *tau, doublecom plex *c, long ldc, long *info);

\section*{PURPOSE}
zunm \(r 2\) overw rites the general com plex \(m\) boy \(-n m\) atrix \(C\) with
\(w\) here \(Q\) is a com plex untary \(m\) atrix defined as the product ofk elem entary reflectors
\[
\mathrm{Q}=\mathrm{H}(1)^{\prime} \mathrm{H}(2)^{\prime} \ldots \mathrm{H}(\mathrm{k})^{\prime}
\]
as retumed by CGERQF.Q is of orderm ifSIDE \(=\mathrm{L}\) 'and of ordern if \(S \mathbb{D} E=R^{\prime}\).

\section*{ARGUMENTS}

SIDE (input)
= L ': apply Q orQ 'from the Left
\(=R\) ': apply Q or Q 'from the R ight

TRANS (input)
\(=\mathrm{N}^{\prime}:\) apply \(\mathrm{Q} \quad(\mathbb{N}\) o transpose)
= C ': apply Q ' (C onjugate transpose)

M (input) The num ber of row s of the \(m\) atrix \(C . M>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\).

K (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q\). IfSID \(E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0\);
ifS \(\mathbb{D} E=R{ }^{\prime}, N>=K>=0\).

A (input) ( \(L D A, M\) ) if \(S D E=L \prime\) ( \(L D A, N\) ) if \(S D E=R^{\prime} T\) he i-th row \(m\) ustcontain the vectorw hich defines the elem entary reflector H ( i ), for \(i=1,2, \ldots, \mathrm{k}\), as retumed by CGERQF in the lastk row s of its anay argum entA. A is modified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A. LD A >= max \((1, K)\).

TAU (input)
TAU (i) must contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CG ERQF.

C (input/output)
On entry, them -by \(n \mathrm{~m}\) atrix C . On exit, C is overw rilten by Q * C or \(\mathrm{Q}{ }^{*} \mathrm{C}\) orC \({ }^{*} \mathrm{Q}\) 'orC \({ }^{*} \mathrm{Q}\).

LD C (input)
The leading dim ension of the array C. LD C >= max (1, M).

W ORK (w orkspace)
\((\mathbb{N})\) if \(\left.S \mathbb{D} E=L^{\prime}, M\right)\) if \(S \mathbb{D} E=R^{\prime}\)
\(\mathbb{I N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N} F O=-\) i, the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zunm rq -overw rite the general com plex M -by -N m atrix C w th \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime}\) TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZUNM RQ (SDE,TRANS,M,N,K,A,LDA,TAU,C,LDC,W ORK,}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
DOUBLE COM PLEX A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
\mathbb{NTEGER M,N,K,LDA,LDC,LW ORK,NNFO}
SUBROUT\mathbb{NE ZUNM RQ_64 (S\mathbb{DE,TRANS,M ,N,K,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{T},\textrm{L},\textrm{L}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,TRANS
DOUBLE COM PLEX A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
INTEGER*8M,N,K,LDA,LDC,LW ORK,\mathbb{NFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNMRQ (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C,[L D C]\), \(\mathbb{W}\) ORK ], [LW ORK ], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::SDE,TRANS
COMPLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A C
\(\mathbb{N}\) TEGER ::M ,N,K,LDA,LDC,LW ORK, \(\mathbb{N} F O\)
SUBROUTINE UNMRQ_64 (SDE, [TRANS], \(\mathbb{M}], \mathbb{N}],[K], A,[L D A], T A U, C\), [LDC ], [W ORK ], [LW ORK ], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::SDE,TRANS

COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER ( 8 ) :: M , N , K, LDA, LD C, LW ORK, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zunm rq (char side, chartrans, int \(m\), int \(n\), int \(k\), doublecom plex *a, int lda, doublecom plex *tau, doublecom plex *c, int ldc, int*info);
void zunm rq_64 (char side, char trans, long m, long n, long k, doublecom plex *a, long lda, doublecom plex *tau, doublecom plex *c, long ldc, long *info);

\section*{PURPOSE}
zunm rq overw rites the general com plex M -by -N m atrix C w th TRANS = C : \(\quad Q * * H * C \quad C * Q * * H\)
where \(Q\) is a com plex unitary \(m\) atrix defined as the product ofk elem entary reflectors
\[
\mathrm{Q}=\mathrm{H}(1) \mathrm{H}(2)^{\prime} \ldots \mathrm{H}(\mathrm{k})^{\prime}
\]
as retumed by CGERQF.Q is oforderM ifSIDE \(=\mathbb{L}\) 'and of orderN ifSIDE = R'.

\section*{ARGUMENTS}

SIDE (input)
\(=\mathrm{L}\) ': apply Q orQ \({ }^{* *} \mathrm{H}\) from the Left;
\(=\mathrm{R}\) ': apply Q orQ **H \(_{\mathrm{H}}\) from the R ight.

TRANS (input)
= N ': N o transpose, apply Q ;
= C ': Transpose, apply Q **H .
TRANS is defaulted to \(N\) 'forF \(95 \mathbb{N}\) TERFACE .

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{C} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix C. \(\mathrm{N}>=0\).
\(K\) (input) The num ber of elem entary reflectors w hose product defines the \(m\) atrix \(Q\). If \(S \mathbb{D} E=L ', M \quad=K>=0\); if \(S \mathbb{D} E=R \prime, N>=K>=0\).

A (input) ( \(L D A, M\) ) if \(S \mathbb{D} E=\mathbb{L}\) ', (LDA, \(N\) ) if \(S \mathbb{D} E=R^{\prime}\) The \(i\)-th row m ustcontain the vectorw hich defines the elem entary reflector \(H\) ( \(i\) ), for \(i=1,2, \ldots, k\), as retumed by C GERQF in the lastk row s of its array argum entA. A is modified by the routine but restored on exit.

LD A (input)
The leading dim ension of the array A. LDA >= \(\max (1, K)\).

TAU (input)
TAU (i) m ust contain the scalar factor of the ele\(m\) entary reflectorH (i), as retumed by CGERQF.

C (input/output)
On entry, the \(M-b y-N m\) atrix \(C\). On exit, \(C\) is overw rilten by Q * C or \(\mathrm{Q} * \mathrm{H} * \mathrm{C}\) or C * \(\mathrm{Q} * \mathrm{H}\) or C * Q .

LD C (input)
The leading dim ension of the array \(C . \operatorname{LDC}>=\) \(\max (1, M)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the amay W ORK. IfSIDE = L', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LWORK >= \(\max (1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=L\) ', and LW ORK \(>=M * N B\) if \(S \mathbb{D} E=R\) ', w here N B is the optim alblocksize.

If LW ORK = -1 , then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK anay, retums this value as the first
entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.
\(\mathbb{N} F O\) (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-\) i, the i-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS
- FURTHER DETAILS

\section*{NAME}
zunm rz -overw rite the general com plex \(M\) by \(-\mathrm{N} m\) atrix \(C\) with \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime}\) TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZUNMRZ (S\mathbb{DE,TRANS,M,N,K,L,A,LDA,TAU,C,LDC,W ORK,}}\mathbf{M},\textrm{L},\textrm{L} LW ORK, $\mathbb{N} F O$ )

```

CHARACTER * 1 SDE,TRANS
D OUBLE COM PLEX A (LDA , *), TAU (*), C (LDC, \(\left.{ }^{*}\right), \mathrm{W} O R K(*)\)
\(\mathbb{N}\) TEGERM,N,K,L,LDA,LDC,LW ORK, \(\mathbb{N} F O\)
SUBROUTINE ZUNMRZ_64 (SDE,TRANS,M,N,K,L,A,LDA,TAU,C,LDC, W ORK,LW ORK, \(\mathbb{N} F O\) )

CHARACTER * 1 SIDE,TRANS

\(\mathbb{N}\) TEGER*8M,N,K,L,LDA,LDC,LWORK, \(\mathbb{N} F O\)

\section*{F95 INTERFACE}

SU BROUTINE ZUNMRZ (SDE,TRANS,M,N,K,L,A,LDA,TAU,C,LDC, W ORK,LW ORK, \(\mathbb{N} F\) ) )

CHARACTER (LEN=1) ::SIDE,TRANS
COM PLEX (8),D IM ENSION (:) ::TAU,W ORK
COM PLEX (8), D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER :: M , N, K, L,LDA,LD C,LWORK, \(\mathbb{N}\) FO
SUBROUTINE ZUNMRZ_64 (SDE,TRANS,M,N,K,L,A,LDA,TAU,C,LDC, W ORK,LW ORK, \(\mathbb{N} F O\) )

CHARACTER (LEN=1) ::SIDE,TRANS
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A,C
\(\mathbb{N}\) TEGER (8) :: M , N , K,L,LDA,LDC,LW ORK, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zunm rz (char side, chartrans, intm, intn, intk, int l, doublecom plex *a, int lda, doublecom plex *tau, doublecom plex *c, int ldc, int*info);
void zunm rz_64 (char side, chartrans, long m, long n, long
k, long l, doublecom plex *a, long lda, doublecom plex *tau, doublecom plex *c, long ldc, long
*info);

\section*{PURPOSE}
zunm rz overw rites the general com plex M -by -N m atrix C w ith TRANS = C : \(\quad \mathrm{Q} * * \mathrm{H} * \mathrm{C} \quad \mathrm{C} * \mathrm{Q} * * \mathrm{H}\)
where \(Q\) is a com plex unitary \(m\) atrix defined as the product ofk elem entary reflectors
\[
Q=H(1) H(2) \ldots H(k)
\]
as retumed by CTZRZF. Q is of orderM ifSDE \(=\mathbb{L}\) 'and of order \(N\) if \(S \mathbb{D} E=R\) '.

\section*{ARGUMENTS}

SIDE (input)
\(=\mathrm{L}\) ': apply Q orQ \({ }^{* *} \mathrm{H}\) from the Left;
= R ': apply Q orQ **H from the Right.

TRANS (input)
= N ': N o transpose, apply Q ;
= C ': C onjugate transpose, apply Q **H .
M (input) The num ber of row s of the \(m\) atrix \(C . M>=0\).
\(N\) (input) The num ber of colum ns of the m atrix C. \(\mathrm{N}>=0\).
\(K\) (input) The num ber of elem entary reflectors \(w\) hose product defines them atrix Q . If \(S \mathbb{D} E=\mathbb{L}, \mathrm{M}>=\mathrm{K}>=0\); if \(S \mathbb{D} E=R \prime, N>=K>=0\).

L (input) The num ber of colum ns of the m atrix A containing the \(m\) eaningfilpart of the \(H\) ouseholder reflectors. If \(S \mathbb{D} E=L ', M>=L>=0\), if \(S \mathbb{D} E=R \prime, N>=L\) \(>=0\) 。

A (input) (LDA, M) if \(S \mathbb{D} E=L \prime\), (LDA, \(N\) ) if \(S \mathbb{D} E=R^{\prime}\) The \(i\)-th row m ustcontain the vectorw hich defines the elem entary reflector \(H\) (i), for \(i=1,2, \ldots, k\), as retumed by CTZRZF in the lastk row s of its array argum entA. A is modified by the routine but restored on exit.

LDA (input)
The leading dimension of the array A. LDA >= \(\max (1, K)\).

TAU (input)
TAU (i) m ustcontain the scalar factor of the elem entary reflectorH (i), as retumed by CTZRZF.

C (input/output)
On entry, the M boy- N matrix C . On exit, C is overw ritten by Q * C or \(\mathrm{Q} * * \mathrm{H} * \mathrm{C}\) or \(\mathrm{C} \mathrm{Q}^{*}{ }^{*} \mathrm{H}\) or \(\mathrm{C} * \mathrm{Q}\).

LD C (input)
The leading dim ension of the aray \(C . \operatorname{LDC}>=\) \(\max (1, M)\).

W ORK (w orkspace)
On exit, if \(\mathbb{N F O}=0, W\) ORK (1) retums the optim al
LW ORK.

LW ORK (input)
The dim ension of the aray \(W\) ORK. IfSIDE = L',
LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R \prime\) LW ORK \(>=\) \(m\) ax \((1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=\mathbb{L}\) ', andLW ORK \(>=M * N B\) if \(S \mathbb{D} E=R \prime\), w here N B is the optim alblocksize.

IfLW O RK \(=-1\), then a w orkspace query is assum ed;
the routine only calculates the optim alsize of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by XERBLA.

INFO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N ~ F O ~}=-\) i, the i-th argum enthad an illegalvalue

\section*{FURTHER DETAILS}
\(B\) ased on contributions by
A. Petitet, C om puter S cience D ept., U niv . of Tenn ., K noxville, U SA

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zunm tr-overw rite the general com plex \(M\)-by -N m atrix C with \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime}\) TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZUNM TR (SDDE,UPLO,TRANS,M,N,A,LDA,TAU,C,LDC,W ORK,}
LW ORK,\mathbb{NFO)}
CHARACTER * 1SIDE,UPLO,TRANS
DOUBLE COM PLEX A (LDA,*),TAU (*),C (LDC ,*),W ORK (*)
INTEGER M ,N,LDA,LDC,LW ORK,INFO
SU BROUTINE ZUNM TR_64 (S\mathbb{DE,UPLO,TRANS,M ,N,A,LDA,TAU,C,LDC,}
W ORK,LW ORK,INFO)
CHARACTER * 1STDE,UPLO,TRANS
DOUBLE COM PLEX A (LDA,*),TAU (*),C (LDC,*),W ORK (*)
INTEGER*8M,N,LDA,LDC,LW ORK,\mathbb{N FO}

```

\section*{F95 INTERFACE}

SU BROUTINE UNM TR (SDE,UPLO, [TRANS], \(\mathbb{M}], \mathbb{N}], A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [ \(\mathbb{N F O}]\) )

CHARACTER (LEN=1) ::SDE,UPLO,TRANS
COM PLEX (8),D \(\mathbb{M}\) ENSION (:) ::TAU,W ORK
COMPLEX (8),D \(\mathbb{M}\) ENSION (: : : : : : A , C
\(\mathbb{N}\) TEGER ::M , N,LDA,LDC,LW ORK, \(\mathbb{N} F\) O
SU BROUTINE UNM TR_64 (SDE, UPLO, [TRANS], \(\mathbb{M}], \mathbb{N}], A,[L D A], T A U, C\), [LDC], [W ORK], [LW ORK], [NFO])

CHARACTER (LEN=1) ::SDE,UPLO,TRANS

COM PLEX (8),D \(\mathbb{I M}\) ENSION (:) ::TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::A, C
\(\mathbb{N}\) TEGER (8) ::M , N,LDA,LDC,LW ORK, \(\mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zunm tr (char side, charuplo, chartrans, intm, int n, doublecom plex *a, int lda, doublecom plex *tau, doublecom plex *c, int ldc, int*info);
void zunm tr_64 (charside, charuplo, char trans, long m, long n, doublecom plex *a, long lda, doublecom plex *tau, doublecom plex *c, long ldc, long *info);

\section*{PURPOSE}
zunm troverw rites the general com plex M -by -N m atrix C w ith TRANS = C : \(\quad Q * * H * C \quad C * Q * *\)
where \(Q\) is a com plex unitary \(m\) atrix of ordernq, \(w\) th \(n q=m\) if \(S \mathbb{D} E=\Sigma\) 'and \(n q=n\) if \(S \mathbb{D} E=R '^{\prime} . Q\) is defined as the product of nq-1 elem entary reflectors, as retumed by CHETRD :
if \(\mathrm{UPLO}=\mathrm{U}\) ', \(\mathrm{Q}=\mathrm{H}(\mathrm{nq}-1) \ldots\). \(\mathrm{H}(2) \mathrm{H}(1)\);
if UPLO = L', Q = H (1) H (2) ...H (nq-1).

\section*{ARGUMENTS}

SIDE (input)
\(=\mathrm{L}\) ': apply Q orQ \({ }^{* *} \mathrm{H}\) from the Left;
= R ': apply Q orQ **H from the Right.
UPLO (input)
= U :: U ppertriangle of A contains elem entary
reflectors from CHETRD ; = L':Low er triangle of A
contains elem entary reflectors from CHETRD.

TRANS (input)
= N ': N o transpose, apply Q ;
= C ': C onjugate transpose, apply Q **H .

TRANS is defaulted to \(N\) 'forF \(95 \mathbb{N}\) TERFACE .

M (input) The num ber of row s of the \(m\) atrix \(\mathrm{C} . \mathrm{M}>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\).
A (input) ( \(L D A, M\) ) if \(S \mathbb{D} E=L^{\prime}(L D A, N)\) if \(S \mathbb{D} E=R^{\prime}\) The vectors w hich define the elem entary reflectors, as retumed by CHETRD.

LD A (input)
The leading dim ension of the anay A. LDA >= \(\max (1, M)\) ifS \(\mathbb{D} E=L ; L D A>=m a x(1, N)\) if \(S \mathbb{D} E=\) R.

TAU (input)
\((M-1)\) ifSTDE \(=\mathbb{L}(N-1)\) if \(S \mathbb{D E}=R^{\prime} T A U\) (i) \(m\) ust contain the scalar factor of the elem entary reflectorH (i), as retumed by CHETRD .

C (input/output)
On entry, the \(M\) boy -N m atrix C . On exit, C is overw ritten by \(\mathrm{Q} * \mathrm{C}\) or \(\mathrm{Q} * * \mathrm{H} * \mathrm{C}\) or \(\mathrm{C} * \mathrm{Q} *{ }^{*} \mathrm{H}\) or \(\mathrm{C} * \mathrm{Q}\).

LD C (input)
The leading dim ension of the array C. LD C >= \(m\) ax (1, M).

W ORK (w orkspace)
On exit, if \(\mathbb{N} F O=0, W\) ORK (1) retums the optim al LW ORK.

LW ORK (input)
The dim ension of the anay \(W\) ORK. If \(S \mathbb{D} E=L\) ', LW ORK >= max ( \(1, N\) ); if \(S \mathbb{D} E=R\) ', LW ORK >= \(m a x(1, M)\). Foroptim um perform ance LW ORK \(>=N * N B\) if \(S \mathbb{D} E=L^{\prime}\), and LW ORK \(>=M\) *NB ifSDE \(=R\) ', where NB is the optim alblocksize.

IfLW ORK \(=-1\), then a w orkspace query is assum ed; the routine only calculates the optim al size of the W ORK array, retums this value as the first entry of the W ORK array, and no errorm essage related to LW ORK is issued by X ERBLA.
\(\mathbb{N F O}\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zupgtr-generate a complex unitary matrix \(Q\) which is defined as the product ofn-1 elem entary reflectors \(H\) (i) of ordern, as retumed by CH PTRD using packed storage

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZUPGTR (UPLO,N,AP,TAU,Q,LDQ,W ORK,INFO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEX AP (*),TAU (*),Q (LDQ ,*),W ORK (*)
NNTEGERN,LDQ,INFO
SU BROUT\mathbb{NE ZUPGTR_64 (UPLO,N,AP,TAU,Q,LDQ,W ORK, IN FO)}
CHARACTER * 1 UPLO
DOUBLE COM PLEXAP (*),TAU (*),Q (LDQ ,*),W ORK (*)
\mathbb{N TEGER*8 N,LDQ,INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UPGTR (UPLO, \(\mathbb{N}], A P, T A U, Q,[L D Q],[W O R K],[\mathbb{N} F O])\)
CHARACTER (LEN=1) ::UPLO
COMPLEX (8),D IM ENSION (:) ::AP,TAU,W ORK
COM PLEX (8),D IM ENSION (:,:) :: Q
\(\mathbb{N} T E G E R:: N, L D Q, \mathbb{N} F O\)

SU BROUTINE UPG TR_64 (UPLO , \(\mathbb{N}\) ],AP, TAU , \(\mathrm{Q},[\mathrm{LD} \mathrm{Q}],[\mathbb{W}\) ORK ], [ \(\mathbb{N} F \mathrm{FO}])\)
CHARACTER (LEN=1) ::UPLO
COM PLEX (8),D IM ENSION (:) ::AP,TAU,W ORK
COM PLEX (8),D \(\mathbb{M}\) ENSION (:,:) ::Q
\(\mathbb{N} T E G E R(8):: N, L D Q, \mathbb{N} F O\)

\section*{C INTERFACE}
\#include <sunperfh>
void zupgtr(charuplo, intn, doublecom plex *ap, doublecom plex *tau, doublecom plex *q, int ldq, int*info);
void zupgtr_64 (charuplo, long n, doublecom plex *ap, doublecom plex *tau, doublecom plex *q, long ldq, long *info);

\section*{PURPOSE}
zupgtrgenerates a com plex unitary \(m\) atrix \(Q\) which is defined as the product of n-1 elem entary reflectors \(H\) (i) of ordern, as retumed by CH PTRD using packed storage:
if \(U P L O=U ', Q=H(n-1) \ldots\) (2) \(H(1)\),
if \(U P L O=L ', Q=H(1) H(2) \ldots H(n-1)\).

\section*{ARGUMENTS}

UPLO (input)
= U ':U ppertriangular packed storage used in previous call to CHPTRD; = L ': Low er triangular packed storage used in previous call to CH PTRD .

N (input) The order of the m atrix \(\mathrm{Q} . \mathrm{N}>=0\).

AP (input)
The vectors w hich define the elem entary reflectors, as retumed by CHPTRD.

TAU (input)
TAU (i) \(m\) ust contain the scalar factor of the ele\(m\) entary reflectort (i), as retumed by CH PTRD.

Q (output)
The N toy -N unitary m atrix Q .
LDQ (input)
The leading dim ension of the array \(Q . L D Q>=\) \(\max (1, \mathbb{N})\).

W ORK (w orkspace)
dim ension \((\mathbb{N}-1)\)
\(\mathbb{N}\) FO (output)
= 0: successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-i\), the \(i\)-th argum enthad an illegalvalue

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
zupm tr-overw rite the general com plex \(M\)-by N m atrix C with \(S \mathbb{D} E=L^{\prime} S \mathbb{D} E=R^{\prime}\) TRANS \(=N^{\prime}\)

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZUPM TR (S\mathbb{DE,UPLO,TRANS,M,N,AP,TAU,C,LDC,W ORK,}}\mathbf{N},
\mathbb{NFO)}
CHARACTER * 1SIDE,UPLO,TRANS
DOUBLE COM PLEX AP (*),TAU (*),C (LDC ,*),W ORK (*)
\mathbb{NTEGER M,N,LDC,INFO}
SU BROUTINE ZUPM TR_64(SDE,UPLO,TRANS,M,N,AP,TAU,C,LDC,W ORK,
INFO)
CHARACTER * 1SIDE,UPLO,TRANS
DOUBLE COM PLEX AP (*),TAU (*),C (LDC ,*),W ORK (*)
\mathbb{NTEGER*8M,N,LDC,INFO}

```

\section*{F95 INTERFACE}

SU BROUTINE UPM TR (SDE, UPLO, [TRANS], \(\mathbb{M}], \mathbb{N}], A P, T A U, C,[L D C]\), [ \(W\) ORK], [ \(\mathbb{N F O}\) ])

CHARACTER (LEN=1) ::SDE,UPLO,TRANS
COMPLEX (8), D \(\mathbb{M}\) ENSION (:) ::AP,TAU,W ORK
COM PLEX (8),D IM ENSION (:,:) :: C
\(\mathbb{N}\) TEGER ::M,N,LDC, \(\mathbb{N} F O\)
SU BROUTINE UPM TR_64 (SDEE,UPLO, [TRANS], \(\mathbb{M}], \mathbb{N}], A P, T A U, C,[L D C]\), [ W ORK], [ \(\mathbb{N} F \mathrm{O}\) ])

CHARACTER (LEN=1)::SIDE,UPLO,TRANS

\section*{C INTERFACE}
\#include <sunperfh>
void zupm tr (char side, charuplo, chartrans, intm, int n, doublecom plex *ap, doublecom plex *tau, doublecom plex *c, intldc, int*info);
void zupm tr_64 (charside, charuplo, char trans, long m, long n , doublecom plex *ap, doublecom plex *tau, doublecom plex *c, long ldc, long *info);

\section*{PURPOSE}
zupm troverw rites the general com plex M -by -N m atrix C w th TRANS = C : \(\quad\) Q** \({ }^{*} \mathrm{C} \quad \mathrm{C} * \mathrm{Q} * \mathrm{H}^{2}\)
\(w\) here \(Q\) is a com plex unitary \(m\) atrix of ordernq, \(w\) th \(n q=m\) if \(S \mathbb{D E}=\mathrm{L}\) 'and \(\mathrm{nq}=\mathrm{n}\) if \(S \mathbb{D} \mathrm{E}=\mathrm{R}\) '. Q is defined as the productofnq-1 elem entary reflectors, as retumed by CH PTRD using packed storage:
if \(U P L O=U ', Q=H(n q-1) \ldots\) (2) \(\mathrm{H}(1)\);
if UPLO = L', \(\mathrm{Q}=\mathrm{H}(1) \mathrm{H}(2) \ldots \mathrm{H}(\mathrm{nq}-1)\).

\section*{ARGUMENTS}

STDE (input)
\(=\mathrm{L}\) : \(:\) apply Q orQ \({ }^{* *} \mathrm{H}\) from the Left;
= R ': apply Q orQ \({ }^{* * H}\) from the Right .
UPLO (input)
= U ::U ppertriangular packed storage used in previous call to CH PTRD ; = L ': Low er triangular packed storage used in previous call to CH PTRD .

TRANS (input)
\(=N^{\prime}\) : N o transpose, apply Q ;
\(=\mathrm{C}\) ': C onjugate transpose, apply Q **H .
TRANS is defaulted to \(N\) 'forF \(95 \mathbb{N}\) TERFACE .

M (input) The num ber of row sof the \(m\) atrix \(C . M>=0\).

N (input) The num ber of colum ns of the m atrix \(\mathrm{C} . \mathrm{N}>=0\).
AP (input)
\((M *(M+1) / 2)\) if \(S D E=L(\mathbb{N} *(N+1) / 2)\) if \(S D E=\)
\(R\) ' The vectors which define the elem entary
reflectors, as retumed by CHPTRD. AP ism odified
by the routine but restored on exit.
TAU (input)
or \((\mathbb{N}-1)\) if \(S \mathbb{D} E=R^{\prime} T A U(i)\) must contain the scalar factor of the elem entary reflectorH (i), as retumed by CHPTRD.
C (input/output)
On entry, the M by -N matrix C. On exit, C is overw rilten by Q * C or \(\mathrm{Q}{ }^{* *} \mathrm{H}\) * C or C \(\mathrm{Q} * * \mathrm{H}\) orC \({ }^{*} \mathrm{Q}\).

LD C (input)
The leading dim ension of the array C. LD C >= \(\max (1, \mathrm{M})\).

W ORK (w orkspace)
\(\mathbb{N})\) if \(S \mathbb{D} E=L^{\prime}(M)\) if \(S \mathbb{D} E=R^{\prime}\)
\(\mathbb{N} F O\) (output)
= 0 : successfulexit
\(<0:\) if \(\mathbb{N}\) FO \(=-\) i, the \(i\)-th argum enthad an ille-
galvalue

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
zvbrm m -variable block sparse row form atm atrix-m atrix m ultiply

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZVBRMM (TRANSA,MB,N,KB,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,BINDX,RPNTR,CPNTR,BPNTRB,BPNTRE,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,MB,N,KB,DESCRA (5),LDB,LDC,LW ORK}
INTEGER INDX(*),B\mathbb{NDX (*),RPNTR M B+1),CPNTR(KB+1),}
* BPNTRB MB),BPNTRE MB)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (*),B (LDB,*),C (LDC,*),W ORK (LW ORK)
SUBROUT\mathbb{NE ZVBRMM_64(TRANSA,M B,N,KB,ALPHA,DESCRA,}
* VAL,\mathbb{NDX,B}\mathbb{N}DX,RPNTR,CPNTR,BPNTRB,BPNTRE,
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
INTEGER*8 TRANSA,M B,N,KB,DESCRA (5),LDB,LDC,LW ORK

```

```

* BPNTRB MB),BPNTREMB)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX VAL (*),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NE VBRMM (TRANSA,MB, N ],KB,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,BINDX,RPNTR,CPNTR,BPNTRB,BPNTRE,}
* B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])
INTEGER TRANSA,MB,KB
\mathbb{NTEGER,D IM ENSION (:) :: DESCRA, INDX,B INDX}
\mathbb{NTEGER,D IM ENSION (:) :: RPNTR,CPNTR,BPNTRB,BPNTRE}
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D IM ENSION (:) ::VAL

```

SUBROUTINEVBRMM_64 (TRANSA,MB, \(\mathbb{N}], K B, A L P H A, D E S C R A\), * VAL, \(\mathbb{N} D \mathrm{X}, \mathrm{B} \mathbb{N} D \mathrm{X}, \mathrm{RPNTR}, \mathrm{CPN} T \mathrm{R}, \mathrm{BPNTRB}, \mathrm{BPNTRE}\),
* \(\quad B,[L D B], B E T A, C,[L D C],[\mathbb{W} O R K],[L W O R K])\)
\(\mathbb{N} T E G E R * 8\) TRANSA, M B , KB
\(\mathbb{N} T E G E R * 8, D \mathbb{M}\) ENSION (:) :: DESCRA, \(\mathbb{N} D X, B \mathbb{N} D X\)
\(\mathbb{N} T E G E R * 8, D \mathbb{M}\) ENSION (:) :: RPNTR,CPNTR,BPNTRB,BPNTRE
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX,D \(\mathbb{I M}\) ENSION (:) ::VAL
DOUBLE COM PLEX,D \(\mathbb{I}\) ENSION (: : : : : B, C

\section*{DESCRIPTION}
C <-alpha op (A) B + beta C
where A LPHA and BETA are scalar, \(C\) and \(B\) are \(m\) atrices, A is a m atrix represented in variableblock sparse row form at and op (A ) is one of
\(o p(A)=A\) or \(o p(A)=A^{\prime}\) or \(o p(A)=\operatorname{con} \dot{g}\left(A^{\prime}\right)\).
( 'indicatesm atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) ith the sparse \(m\) atrix
\[
0 \text { : operate } w \text { ith } m \text { atrix }
\]

1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) ith the conjugate transpose ofm atrix.
2 is equivalent to 1 if the \(m\) atrix is real.

M B \(\quad\) um ber ofblock row \(s\) in \(m\) atrix A
N \(\quad\) Num berof \(C o l u m n s\) in \(m\) atrix \(C\)

K B \(\quad N\) umber ofblock colum ns in m atrix A

A LPH A Scalarparam eter
DESCRA () D escriptor argum ent. Five elem ent integer aray
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
\(2:\) Herm Aian ( \(\mathrm{A}=\mathrm{CONJG}\) (A))
3 :Triangular
4 : Skew (Anti)-Symm etric ( \(A=-A\) )
5 : D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CONJ}\) ( A ) )

D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) m ain diagonal type
0 :non-unit
1 : unit
DESCRA (4) A ray base \(N\) OT \(\mathbb{M}\) PLEM ENTED)
0 :C C++ com patible
1 :Fortran com patible
DESCRA (5) repeated indices? NOT IM PLEM ENTED )
0 : unknown
1 : no repeated indices
VAL ( scalar array of length NN Z consisting of the block entries of A w here each block entry is a dense rectangularm atrix stored colum \(n\) by colum \(n\).
NN Z is the total num berof pointentries in all nonzero block entries of am atrix A.
\(\mathbb{N} D \mathrm{X}\) () integer anray of length BNN Z +1 where BNNZ is the num berof block entries of a m atrix A such that the I-th elem entof \(\mathbb{N}\) D X [] points to the location in VAL of the \((1,1)\) elem ent of the I-th block entry.

B IND X () integer array of length BNN Z consisting of the block colum \(n\) indiges of the block entries of \(A\) where BNNZ is the num berblock entries of a m atrix A.

RPN TR 0) integer amay of length M B+1 such thatRPN TR (I) RPN TR (1) +1 is the row index of the firstpoint row in the \(I\)-th block
row .
RPN TR \(M B+1\) ) is set to \(M+\operatorname{RPN} \operatorname{TR}(1)\) where \(M\) is the num ber of row \(s\) in \(m\) atrix \(A\).
Thus, the num berof point row sin the I-th block row is RPNTR (I+1)RPNTR (I).

CPN TR 0 integer array of length \(K B+1\) such that CPN TR (J)-CPN TR (1)+1 is the colum \(n\) index of the firstpoint colum \(n\) in the \(J\) th block colum n. CPN TR ( \(K B+1\) ) is set to \(K+C P N T R(1)\) where \(K\) is the num ber of \(c o l u m n s\) in \(m\) atrix \(A\). Thus, the num ber of point \(\infty 0\) lum ns in the \(J\) th block colum n is CPNTR ( \(\mathrm{J}+1\) )-CPNTR ( \(J\) ).

BPNTRB () integer array of length \(M B\) such thatBPNTRB (I) BPNTRB (1) +1 points to location in B IND X of the first.block entry of the I-th block row of A.

BPNTRE 0 integer anay of length \(M B\) such that BPN TRE (I) BPNTRB (1) points to location in B \(\mathbb{N}\) D X of the lastblock entry of the I-th block row of A.

B 0 rectangular array w ith firstdin ension LD B .

LD B leading dim ension ofB

BETA Scalarparam eter
C 0 rectangular anray w ith firstdim ension LD C .

LD C leading dim ension ofC
W ORK () scratch array of length LW ORK.W ORK is not referenced in the cumentversion.

LW ORK length ofW ORK array. LW ORK is not referenced in the cumentversion.

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U ser's G uide available at: http://m ath nistgov/m cso/Staff/k Rem ington/tspblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1.For a generalm atrix (DESCRA (1)=0), array CPN TR can be different from RPNTR. Forallotherm atrix types, RPNTR \(m\) ustequalCPN TR and a single array can be passed forboth argum ents.
2. It is know \(n\) that there exists another representation of the variable block sparse row form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of six anay instead of the seven used in the current im plem entation. Them ain difference is that only one array, \(\mathbb{I A}\), containing the pointers to the beginning of each block row in the array \(B \mathbb{N} D X\) is used instead of two arrays BPN TRB and BPN TRE.To use the routine w th this kind of variable block sparse row form at the follow ing calling sequence should be used

SUBROUTINE ZVBRMM (TRANSA,MB,N,KB,ALPHA,DESCRA,
* \(\quad V A L, \mathbb{N} D X, B \mathbb{N} D X, R P N T R, C P N T R, \mathbb{A}, \mathbb{A}(2)\),
* B,LDB,BETA, C,LDC,W ORK,LW ORK )

\section*{Contents}
- NAME
- SYNOPSIS

\section*{- F95 INTERFACE}
- DESCRIPTION
- ARGUMENTS
- SEE ALSO

\section*{NAME}
```

zvbrsm -variable block sparse row form attriangular solve

```

\section*{SYNOPSIS}
```

SUBROUT\mathbb{NE ZVBRSM (TRANSA,M B,N,UNITD,DV,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,B\mathbb{NDX,RPNTR,CPNTR,BPNTRB,BPNTRE,}}\mathbf{}\mathrm{ ,}
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\mathbb{NTEGER TRANSA,MB,N,UNITD,DESCRA (5),LDB,LDC,LW ORK}
\mathbb{NTEGER }\mathbb{NDX(*),B\mathbb{NDX (*),RPNTR M B+1),CPNTR M B+1),}}\mathbf{(}),
* BPNTRBMB),BPNTREMB)
DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX DV (*),VAL (*),B (LDB,*),C (LDC,*),W ORK (LW ORK)

```
SU BROUTINE ZVBRSM_64(TRANSA, MB,N,UNITD,DV,ALPHA,DESCRA,
* VAL, \(\mathbb{N} D \mathrm{X}, \mathrm{B} \mathbb{N} D \mathrm{X}, \mathrm{RPNTR}, \mathrm{CPNTR}, \mathrm{BPNTRB}, \mathrm{BPNTRE}\),
* B,LDB,BETA,C,LDC,W ORK,LW ORK)
\(\mathbb{N} T E G E R * 8\) TRANSA, M B , N, UN ITD, DESCRA (5), LD B, LD C ,LW ORK
\(\mathbb{N} T E G E R * 8 \mathbb{N} D X(*), B \mathbb{N} D X(*), R P N T R(M+1), C P N T R(M+1)\),
* BPNTRB MB), BPNTRE MB)
DOUBLE COM PLEX ALPHA, BETA
D OUBLE COM PLEX DV (*),VAL (*), B (LDB,*), C (LD C , *), W ORK (LW ORK)

\section*{F95 INTERFACE}
```

SUBROUT\mathbb{NEVBRSM (TRANSA,M B, N ],UNITD,DV,ALPHA,DESCRA,}

* VAL,\mathbb{NDX,B}\mathbb{N}DX,RPNTR,CPNTR,BPNTRB,BPNTRE,
* B,[LDB],BETA,C,[LDC],[W ORK],[LW ORK])
INTEGER TRANSA,MB,UNITD

```

```

\mathbb{NTEGER,D IM ENSION (:) :: RPNTR,CPNTR,BPNTRB,BPNTRE}
DOUBLE COMPLEX ALPHA,BETA
DOUBLE COM PLEX,D IM ENSION (:) ::VAL,DV
DOUBLE COM PLEX,D IM ENSION (:,:):: B,C

```

SUBROUTINEVBRSM_64 (TRANSA, MB, \(\mathbb{N}], U N I T D, D V, A L P H A, D E S C R A\),
* \(B,[\) LDB], BETA, \(C,[L D C],[W O R K],[L W O R K])\)
\(\mathbb{N T E G E R *} 8\) TRANSA, MB,UNITD
\(\mathbb{N} T E G E R * 8, D \mathbb{M} \operatorname{ENS} \mathbb{O} N(:):: \operatorname{DESCRA}, \mathbb{N} D X, B \mathbb{N} D X\)
\(\mathbb{N} T E G E R * 8, D \mathbb{M} E N S \mathbb{O N}(:):: R P N T R, C P N T R, B P N T R B, B P N T R E\)
DOUBLE COMPLEX ALPHA,BETA
DOUBLE COM PLEX ,D \(\mathbb{M} E N S I O N(:):: V A L, D V\)
DOUBLE COM PLEX ,D \(\mathbb{M}\) ENSION (: : : : : B , C

\section*{DESCRIPTION}
\[
\begin{aligned}
& C<-A L P H A \text { Op (A) B + BETA C C <-ALPHA D op (A) B + BETA C } \\
& C<-A L P H A \text { op (A) D B + BETA C } \\
& \text { where A LPH A and BETA are scalar, C and B arem by n densem atrices, } \\
& D \text { is a block diagonalm atrix, A is a unit, ornon-unit, upper or } \\
& \text { low ertriangularm atrix represented in variable block sparse row } \\
& \text { form atand op (A ) is one of }
\end{aligned}
\]
op (A) \()=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}(A)\) or op (A) \(=\operatorname{inv}\left(c o n g\left(A^{\prime}\right)\right)\) (inv denotesm atrix inverse, 'indicates \(m\) atrix transpose)

\section*{ARGUMENTS}

TRANSA Indicates how to operate \(w\) th the sparse \(m\) atrix
0 : operate \(w\) ith \(m\) atrix
1 : operate \(w\) ith transpose \(m\) atrix
2 : operate \(w\) th the conjugate transpose ofm atrix.
2 is equivalent to 1 ifm atrix is real.

M B \(\quad\) Num ber ofblock row \(\sin m\) atrix \(A\)

N \(\quad\) Num berof colum ns in matrix C

UN ITD Type of scaling:
1 : Identity m atrix (argum entD V [] is ignored)
2 : Scale on left (row block scaling)
3 : Scale on right (colum \(n\) block scaling)

DV () A rray containing the block entries of the block diagonalm atrix D. The size of the Jth block is RPN TR ( \(\mathrm{J}+1\) )-RPN TR (J) and each block containsm atrix entries stored colum n-m ajor. The total length of aray DV is given by the form ula:
sum over J from 1 to M B:

A LPH A Scalarparam eter

DESCRA () D escriptor argum ent. Five elem ent integer array
DESCRA (1) m atrix structure
0 :general
1 : symm etric ( \(\mathrm{A}=\mathrm{A}\) )
2 : Herm itian ( \(\mathrm{A}=\mathrm{CONJG}(\mathrm{A})\) )
3 :Triangular
4 : Skew (Anti)-Symm etric ( \(A=-A\) )
5 :D iagonal
6 : Skew Herm itian ( \(\mathrm{A}=-\mathrm{CON}\) J ( A ) )
N ote: For the routine, DESCRA (1)=3 is only supported.
D ESCRA (2) upper/low er triangular indicator
1 : low er
2 :upper
DESCRA (3) main diagonal type
0 : non-identily blocks on the \(m\) ain diagonal
1 : identity diagonalblocks
2 : diagonalblocks are densem atrices
DESCRA (4) A ray base \(\mathbb{N} O T \mathbb{I}\) PLEM ENTED )
0 :C C++ com patible
1 :Fortran com patible
D ESCRA (5) repeated indices? NOT IM PLEM EN TED )
0 : unknown
1 :no repeated indices
VAL 0 scalar array of length NN Z consisting of the block entries ofA where each block entry is a dense rectangularm atrix stored colum \(n\) by colum \(n\).
NN Z is the total num berof pointentries in allnonzero block entries of a m atrix A.
\(\mathbb{I N}\) D X 0 integer array of length BNN Z +1 where BNN \(Z\) is the num ber block entries of a \(m\) atrix A such that the I-th elem ent of \(\mathbb{N} D \mathrm{X}[]\) points to the location in VAL of the \((1,1)\) elem ent of the I-th block entry .
\(B \mathbb{N} D\) X () integer array of length BNNZ consisting of the block colum \(n\) indiges of the block entries of \(A\) where BNN \(Z\) is the num berblock entries of a m atrix A. B lock colum n indices M U ST be sorted in increasing order foreach block row .

RPN TR 0) integer aray of length M B +1 such thatRPN TR (I) RPN TR (1) +1 is the row index of the firstpoint row in the \(I\)-th block
row.
RPN TR \(M B+1\) ) is set to \(M+R P N T R(1)\) where \(M\) is the num ber of row \(s\) in square triangularm atrix \(A\).

Thus, the num berof point row s in the I-th block row is RPNTR (I+1)RPNTR (I).

NOTE: For the cumentversion CPN TR m ustequalRPN TR and a single array can be passed forboth argum ents

CPNTR 0 integeramay of length \(M B+1\) such thatCPN TR (J)-CPN TR (1) +1 is the colum \(n\) index of the firstpoint colum \(n\) in the \(J\) th block colum n. CPN TR M B+1) is set to M +CPN TR (1). Thus, the num ber of pointcolum ns in the J-th block colum n is CPNTR ( \(\mathrm{J}+1\) )-CPNTR (J).

NO TE: For the current version CPN TR m ustequal RPN TR and a single aray can be passed forboth argum ents
BPN TRB 0 integer aray of length \(M B\) such thatBPN TRB (I)-BPNTRB (1)+1 points to location in B IND X of the firstblock entry of the I-th block row of A.

BPN TRE () integer array of length \(M B\) such thatBPN TRE (I) BPN TRB (1) points to location in B \(\mathbb{N} D \mathrm{X}\) of the last.block entry of the I-th block row of A.

B 0 rectangular aray w th first dim ension LD B.

LD B leading dim ension ofB
BETA Scalarparam eter
C 0 rectangular array w ith first dim ension LD C .

LD C leading dim ension ofC
W ORK 0 scratch array of length LW ORK. On exit, ifLW ORK = -1,W ORK (1) retums the optim um size of LW ORK.

LW ORK length of ORK array. LW ORK should be at least \(\mathrm{M}=\mathrm{RPNTR} \mathrm{M} \mathrm{B}+1)\) RPNTR (1).

Forgood perform ance, LW ORK should generally be larger. Foroptim um perform ance on \(m\) ultiple processors, LW ORK \(>=\mathrm{M} * \mathrm{~N}\) _CPUS where N _CPUS is the \(m\) axim um num berof processors available to the program .

IfLW ORK \(=0\), the routine is to allocate \(w\) orkspace needed.
IfLW ORK = -1 , then a w orkspace query is assum ed; the routine only calculates the optim um size of the W ORK array, retums this value as the firstentry of the W ORK amray, and no enrorm essage related to LW ORK is issued

\section*{SEE ALSO}

N IST FO RTRA N Sparse B las U sers G uide available at:
http://m ath nist.gov/n csd/Staff/k Rem ington/Espblas/
"D ocum ent for the B asic LinearA lgebra Subprogram s (BLA S) Standard", U niversity of Tennessee, K noxville, Tennessee, 1996:
http://w w w netlib .org/utk/papers/sparse.ps

\section*{NOTES/BUGS}
1. N o test for singularity ornear-singularity is included in this routine. Such testsm ust.be perform ed before calling this routine.
2. If \(D E S C R A\) (3)=0, the low erorupper triangular part of each diagonalblock is used by the routine depending on DESCRA (2).
3. If \(D E S C R A(3)=1\), the unit diagonalblocksm ightorm ight not.be referenced in the VBR representation of a sparse \(m\) atrix. They are notused anyw ay .
4. If \(D E S C R A\) (3)=2, diagonalblocks are considered as dense \(m\) atrices and the LU factorization \(w\) ith partialpivoting is used by the routine. WORK (1)=0 on retum if the factorization for alldiagonalblocks has been com pleted successfully, otherw ise WORK (1) = -iw here is the block num ber forw hich the LU factorization could notbe com puted.
5.The routine can be applied for solving triangular system s w hen the upper or low er triangle of the general sparse m atrix A is used. H ow enver DESCRA (1) m ustbe equal to 3 .
6. It is know \(n\) that there exists another representation of the variable block sparse row form at (see forexam ple Y Saad, "Iterative M ethods forSparse LinearSystem s", W PS, 1996). Its data structure consists of six anay instead of the seven used in the current im plem entation. Them ain difference is that only one array, IA , containing the pointers to the beginning ofeach block row in the amay \(B \mathbb{N} D X\) is used instead of two arrays BPN TRB and BPN TRE.To use the routine w ith this kind of variable block sparse row form at the follow ing calling sequence should be used

SUBROUT \(\mathbb{N} E Z V B R S M\) (TRANSA, MB,N, UNITD, DV,ALPHA,DESCRA,
\(V A L, \mathbb{N} D X, B \mathbb{N} D X, R P N T R, C P N T R, \mathbb{A}, \mathbb{A}(2)\), B, LD B, BETA, C, LDC,WORK,LW ORK )

\section*{Contents}
- NAME
- SYNOPSIS
- F95 INTERFACE
- C INTERFACE
- PURPOSE
- ARGUMENTS

\section*{NAME}
```

zvm ul-com pute the scaled product of com plex vectors

```

\section*{SYNOPSIS}

```

DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX X (*),Y (*),Z (*)
\mathbb{NTEGERN,INCX,}\mathbb{N}CY,\mathbb{NCZ}

```

```

DOUBLE COM PLEX ALPHA,BETA
DOUBLE COM PLEX X (*),Y (*),Z (*)
\mathbb{NTEGER*8N,}\mathbb{NCX,\mathbb{NCY, INCZ}}\mathbf{N}=\mp@code{N}
F95 INTERFACE

```

```

    COMPLEX (8) ::ALPHA,BETA
    COM PLEX (8),D IM ENSION (:) ::X,Y,Z
    \mathbb{NTEGER::N,\mathbb{NCX,INCY,}\mathbb{NCZ}}\mathbf{~}=\mp@code{N}
    ```

```

    COM PLEX (8) ::ALPHA,BETA
    COM PLEX (8),D IM ENSION (:) ::X,Y,Z
    ```


\section*{C INTERFACE}
```

\#include <sunperfh>

```
void zvm ul(intn, doublecom plex *alpha, doublecom plex *x, int incx, doublecom plex *y, int incy, doublecom plex *beta, doublecom plex *z, int incz);
void zvm ul 64 (long \(n\), doublecom plex *alpha, doublecom plex
*x, long incx, doublecom plex *y, long incy, doublecom plex *beta, doublecom plex *z, long incz);

\section*{PURPOSE}
zvm ulcom putes the scaled product of com plex vectors:
\(z(i)=A L P H A * x(i) * y(i)+B E T A * z(i)\)
forl \(<=i<=N\).

\section*{ARGUMENTS}

N (input)
Length of the vectors. \(\mathrm{N}>=0\). ZVM U L w ill retum im \(m\) ediately if \(N=0\).

A LPHA (input)
Scale factor on the m ultiplicand vectors.
X (input) dim ension (*)
M ultiplicand vector.
\(\mathbb{N} C X\) (input)
Stride betw een elem ents of the \(m\) ultiplicand vector \(\mathrm{X} . \mathbb{N} C X>0\).

Y (input) dim ension (*)
M ultiplicand vector.
\(\mathbb{N C Y}\) (input)
Stride betw een elem ents of the \(m\) ultiplicand vector
Y. \(\mathbb{N} C Y>0\).

BETA (input)
Scale factor on the product vector.
Z (input/output)
dim ension (*)
Product vector. On exit, \(z(i)=A L P H A * x(i) *\)
\(y(i)+B E T A * z(i)\).
\(\mathbb{N C Z}\) (input)
Stride betw een elem ents of \(Z . \mathbb{N C Z}>0\).```

