**LAPACK REFERENCE MANUAL**

**参考手册**

**版本管理**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本历史** | | | | |
| 序号 | 修改日期 | 版本 | 修订人 | 修订内容 |
| 1 | 2016-05-16 | 1.0 |  | 文档拟制 |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **发行控制** | | | | |
| 编制 |  | 签字日期 |  | |
| 审核 |  | 签字日期 |  | |
| 批准 |  | 签字日期 |  | |
| 发布 |  | 签字日期 |  | |
| 密级 | 绝密□ | 机密□ | 一般■ | 公开□ |

目录

[引言 1](#_Toc6998434)

[1. 编写目的 1](#_Toc6998435)

[2. 编写背景 1](#_Toc6998436)

[3. 概念定义 1](#_Toc6998437)

[4. 参考资料 1](#_Toc6998438)

[1 热力学 2](#_Toc6998439)

[2 空气动力学 2](#_Toc6998440)

# 引言

## 1. 编写目的

说明编写这份数据要求说明书的目的，指出预期的读者。

## 2. 编写背景

a.待开发软件系统的名称；

b.列出本项目的任务提出者、开发者、用户以及将运行该项软件的计算站或计算机网络系统。

## 3. 概念定义

列出本文件中用到的专门术语的定义和外文首字母组词的原词组。

## 4. 参考资料

列出有关的参考资料。

<http://www.netlib.org/lapack/explore-html/>

# 1 介绍

# 2 LAPACK Routine List

## Simple Driver and Divide and Conquer Driver Subprograms

| Name† | Prefixes | Description |
| --- | --- | --- |
| ?GESV | S D C Z | Solves a general system of linear equations AX=B. |
| ?GBSV | S D C Z | Solves a general banded system of linear equations AX=B. |
| ?GTSV | S D C Z | Solves a general tridiagonal system of linear equations AX=B. |
| ?POSV | S D C Z | Solves a symmetric/Hermitian positive definite system of linear equations AX=B. |
| ?PPSV | S D C Z | Solves a symmetric/Hermitian positive definite system of linear equations AX=B, where A is held in packed storage. |
| ?PBSV | S D C Z | Solves a symmetric/Hermitian positive definite banded system of linear equations AX=B. |
| ?PTSV | S D C Z | Solves a symmetric/Hermitian positive definite tridiagonal system of linear equations AX=B. |
| ?SYSV | S D C Z | Solves a symmetric indefinite system of linear equations AX=B. |
| ?SPSV | S D C Z | Solves a symmetric indefinite system of linear equations AX=B, where A is held in packed storage. |
| ?GELS | S D C Z | Computes the least squares solution to an over-determined system of linear equations, AX=B or AHX=B, or the minimum norm solution of an under-determined system, where A is a general rectangular matrix of full rank, using a QR or LQ factorization of A. |
| ?GELSD | S D C Z | Computes the least squares solution to an over-determined system of linear equations, AX=B or AHX=B, or the minimum norm solution of an under-determined system, using a divide and conquer method, where A is a general rectangular matrix of full rank, using a QR or LQ factorization of A. |
| ?GGLSE | S D C Z | Solves the LSE (Constrained Linear Least Squares Problem) using the GRQ (Generalized RQ) factorization. |
| ?GGGLM | S D C Z | Solves the GLM (Generalized Linear Regression Model) using the GQR (Generalized QR) factorization. |
| ?SYEV | S D | Computes all eigenvalues, and optionally, eigenvectors of a real symmetric matrix. |
| ?SYEVD | S D | Computes all eigenvalues, and optionally, eigenvectors of a real symmetric matrix. If eigenvectors are desired, it uses a divide and conquer algorithm. |
| ?SPEV | S D | Computes all eigenvalues, and optionally, eigenvectors of a real symmetric matrix in packed storage. |
| ?SPEVD | S D | Computes all eigenvalues, and optionally, eigenvectors of a real symmetric matrix in packed storage. If eigenvectors are desired, it uses a divide and conquer algorithm. |
| ?SBEV | S D | Computes all eigenvalues, and optionally, eigenvectors of a real symmetric band matrix. |
| ?SBEVD | S D | Computes all eigenvalues, and optionally, eigenvectors of a real symmetric band matrix. If eigenvectors are desired, it uses a divide and conquer algorithm. |
| ?STEV | S D | Computes all eigenvalues, and optionally, eigenvectors of a real symmetric tridiagonal matrix. |
| ?STEVD | S D | Computes all eigenvalues, and optionally, eigenvectors of a real symmetric tridiagonal matrix. If eigenvectors are desired, it uses a divide and conquer algorithm. |
| ?GEES | S D C Z | Computes the eigenvalues and Schur factorization of a general matrix, and orders the factorization so that selected eigenvalues are at the top left of the Schur form. |
| ?GEEV | S D C Z | Computes the eigenvalues and left and right eigenvectors of a general matrix. |
| ?GESVD | S D C Z | Computes the singular value decomposition (SVD) of a general rectangular matrix. |
| ?GESDD | S D C Z | Computes the singular value decomposition (SVD) of a general rectangular matrix using divide-and-conquer. |
| ?SYGV | S D | Computes all eigenvalues and the eigenvectors of a generalized symmetric-definite generalized eigenproblem, Ax = Î»Bx, ABx = Î»x, or BAx = Î»x. |
| ?SYGVD | S D | Computes all eigenvalues and the eigenvectors of a generalized symmetric-definite generalized eigenproblem, Ax=Î»Bx, ABx=Î»x, or BAx=Î»x. If eigenvectors are desired, it uses a divide and conquer algorithm. |
| ?SPGV | S D | Computes all eigenvalues and eigenvectors of a generalized symmetric-definite generalized eigenproblem, Ax=Î»Bx, ABx=Î»x, or BAx=Î»x, where A and B are in packed storage. |
| ?SPGVD | S D | Computes all eigenvalues and eigenvectors of a generalized symmetric-definite generalized eigenproblem, Ax=Î»Bx, ABx=Î»x, or BAx=Î»x, where A and B are in packed storage. If eigenvectors are desired, it uses a divide and conquer algorithm. |
| ?SBGV | S D | Computes all the eigenvalues, and optionally, the eigenvectors of a real generalized symmetric-definite banded eigenproblem, of the form Ax=Î»Bx. A and B are assumed to be symmetric and banded, and B is also positive definite. |
| ?SBGVD | S D | Computes all the eigenvalues and optionally, the eigenvectors of a real generalized symmetric-definite banded eigenproblem, of the form Ax=Î»Bx. A and B are assumed to be symmetric and banded, and B is also positive definite. If eigenvectors are desired, it uses a divide and conquer algorithm. |
| ?GEGS | S D C Z | Computes the generalized eigenvalues, Schur form, and left and/or right Schur vectors for a pair of non-symmetric matrices. |
| ?GGES | S D C Z | Computes the generalized eigenvalues, Schur form, and left and/or right Schur vectors for a pair of non-symmetric matrices. |
| ?GEGV | S D C Z | Computes the generalized eigenvalues, and left and/or right generalized eigenvectors for a pair of non-symmetric matrices. |
| ?GGEV | S D C Z | Computes the generalized eigenvalues, and left and/or right generalized eigenvectors for a pair of non-symmetric matrices. |
| ?GGSVD | S D C Z | Computes the Generalized Singular Value Decomposition. |
| ?HESV | C Z | Solves a Hermitian indefinite system of linear equations AX=B. |
| ?HPSV | C Z | Solves a Hermitian indefinite system of linear equations AX=B, where A is held in packed storage. |
| ?HEEV | C Z | Computes all eigenvalues and, optionally, eigenvectors of a Hermitian matrix. |
| ?HEEVD | C Z | Computes all eigenvalues and, optionally, eigenvectors of a Hermitian matrix. If eigenvectors are desired, it uses a divide and conquer algorithm. |
| ?HEEVR | C Z | Computes selected eigenvalues, and optionally, eigenvectors of a Hermitian matrix. Eigenvalues are computed by the dqds algorithm, and eigenvectors are computed from various "good" LDLT representations (also known as Relatively Robust Representations). |
| ?HPEV | C Z | Computes all eigenvalues and, optionally, eigenvectors of a Hermitian matrix in packed storage. |
| ?HPEVD | C Z | Computes all eigenvalues and, optionally, eigenvectors of a Hermitian matrix in packed storage. If eigenvectors are desired, it uses a divide and conquer algorithm. |
| ?HBEV | C Z | Computes all eigenvalues and, optionally, eigenvectors of a Hermitian band matrix. |
| ?HBEVD | C Z | Computes all eigenvalues and, optionally, eigenvectors of a Hermitian band matrix. If eigenvectors are desired, it uses a divide and conquer algorithm. |
| ?HEGV | C Z | Computes all eigenvalues and the eigenvectors of a generalized Hermitian-definite generalized eigenproblem, Ax=Î»Bx, ABx=Î»x, or BAx=Î»x. |
| ?HEGVD | C Z | Computes all eigenvalues and the eigenvectors of a generalized Hermitian-definite generalized eigenproblem, Ax=Î»Bx, ABx=Î»x, or BAx=Î»x. If eigenvectors are desired, it uses a divide and conquer algorithm. |
| ?HPGV | C Z | Computes all eigenvalues and eigenvectors of a generalized Hermitian-definite generalized eigenproblem, Ax=Î»Bx, ABx=Î»x, or BAx=Î»x, where A and B are in packed storage. |
| ?HPGVD | C Z | Computes all eigenvalues and eigenvectors of a generalized Hermitian-definite generalized eigenproblem, Ax=Î»Bx, ABx=Î»x, or BAx=Î»x, where A and B are in packed storage. If eigenvectors are desired, it uses a divide and conquer algorithm. |
| ?HBGV | C Z | Computes all the eigenvalues, and optionally, the eigenvectors of a complex generalized Hermitian-definite banded eigenproblem, of the form Ax=Î»Bx. A and B are assumed to be Hermitian and banded, and B is also positive definite. |
| ?HBGVD | C Z | Computes all the eigenvalues, and optionally, the eigenvectors of a complex generalized Hermitian-definite banded eigenproblem, of the form Ax=Î»Bx. A and B are assumed to be Hermitian and banded, and B is also positive definite. If eigenvectors are desired, it uses a divide and conquer algorithm. |
| † ?  indicates prefix which must be filled with a combination of: S = REAL(kind=4), D = REAL(kind=8), C = COMPLEX(kind=4), Z = COMPLEX(kind=8) | | |

## Expert Driver and RRR Driver Subprograms

| Name† | Prefixes | Description |
| --- | --- | --- |
| ?GESVX | S D C Z | Solves a general system of linear equations AX=B, ATX=B or AHX=B, and provides an estimate of the condition number and error bounds on the solution. |
| ?GBSVX | S D C Z | Solves a general banded system of linear equations AX=B, ATX=B or AHX=B, and provides an estimate of the condition number and error bounds on the solution. |
| ?GTSVX | S D C Z | Solves a general tridiagonal system of linear equations AX=B, ATX=B or AHX=B, and provides an estimate of the condition number and error bounds on the solution. |
| ?POSVX | S D C Z | Solves a symmetric/Hermitian positive definite system of linear equations AX=B, and provides an estimate of the condition number and error bounds on the solution. |
| ?PPSVX | S D C Z | Solves a symmetric/Hermitian positive definite system of linear equations AX=B, where A is held in packed storage, and provides an estimate of the condition number and error bounds on the solution. |
| ?PBSVX | S D C Z | Solves a symmetric/Hermitian positive definite banded system of linear equations AX=B, and provides an estimate of the condition number and error bounds on the solution. |
| ?PTSVX | S D C Z | Solves a symmetric/Hermitian positive definite tridiagonal system of linear equations AX=B, and provides an estimate of the condition number and error bounds on the solution. |
| ?SYSVX | S D C Z | Solves a symmetric indefinite system of linear equations AX=B, and provides an estimate of the condition number and error bounds on the solution. |
| ?SPSVX | S D C Z | Solves a symmetric indefinite system of linear equations AX=B, where A is held in packed storage, and provides an estimate of the condition number and error bounds on the solution. |
| ?GELSX | S D C Z | Computes the minimum norm least squares solution to an over- or under-determined system of linear equations AX=B, using a complete orthogonal factorization of A. |
| ?GELSY | S D C Z | Computes the minimum norm least squares solution to an over- or under-determined system of linear equations AX=B, using a complete orthogonal factorization of A. |
| ?GELSS | S D C Z | Computes the minimum norm least squares solution to an over- or under-determined system of linear equations AX=B, using the singular value decomposition of A. |
| ?SYEVX | S D | Computes selected eigenvalues and eigenvectors of a symmetric matrix. |
| ?SYEVR | S D | Computes selected eigenvalues, and optionally, eigenvectors of a real symmetric matrix. Eigenvalues are computed by the dqds algorithm, and eigenvectors are computed from various "good" LDLT, representations (also known as Relatively Robust Representations). |
| ?SYGVX | S D | Computes selected eigenvalues, and optionally, the eigenvectors of a generalized symmetric-definite generalized eigenproblem, Ax=Î»Bx, ABx=Î»x, or BAx=Î»x. |
| ?SPEVX | S D | Computes selected eigenvalues and eigenvectors of a symmetric matrix in packed storage. |
| ?SPGVX | S D | Computes selected eigenvalues, and optionally, eigenvectors of a generalized symmetric-definite generalized eigenproblem, Ax=Î»Bx, ABx=Î»x, or BAx=Î»x, where A and B are in packed storage. |
| ?SBEVX | S D | Computes selected eigenvalues and eigenvectors of a symmetric band matrix. |
| ?SBGVX | S D | Computes selected eigenvalues, and optionally, the eigenvectors of a real generalized symmetric-definite banded eigenproblem, of, the form Ax=Î»Bx. A and B are assumed to be symmetric and banded, and B is also positive definite. |
| ?STEVX | S D | Computes selected eigenvalues and eigenvectors of a real symmetric tridiagonal matrix. |
| ?STEVR | S D | Computes selected eigenvalues, and optionally, eigenvectors of a real symmetric tridiagonal matrix. Eigenvalues are computed by the dqds algorithm, and eigenvectors are computed from various "good" LDLT representations (also known as Relatively Robust Representations). |
| ?GEESX | S D C Z | Computes the eigenvalues and Schur factorization of a general matrix, orders the factorization so that selected eigenvalues are at the top left of the Schur form, and computes reciprocal condition numbers for the average of the selected eigenvalues, and for the associated right invariant subspace. |
| ?GGESX | S D C Z | Computes the generalized eigenvalues, the real Schur form, and, optionally, the left and/or right matrices of Schur vectors. |
| ?GEEVX | S D C Z | Computes the eigenvalues and left and right eigenvectors of a general matrix, with preliminary balancing of the matrix, and computes reciprocal condition numbers for the eigenvalues and right eigenvectors. |
| ?GGEVX | S D C Z | Computes the generalized eigenvalues, and optionally, the left and/or right generalized eigenvectors. |
| ?HESVX | C Z | Solves a Hermitian indefinite system of linear equations AX=B, and provides an estimate of the condition number and error bounds on the solution. |
| ?HPSVX | C Z | Solves a Hermitian indefinite system of linear equations AX=B, where A is held in packed storage, and provides an estimate of the condition number and error bounds on the solution. |
| ?HEEVX | C Z | Computes selected eigenvalues and eigenvectors of a Hermitian matrix. |
| ?HEGVX | C Z | Computes selected eigenvalues, and optionally, the eigenvectors of a generalized Hermitian-definite generalized eigenproblem, Ax=Î»Bx, ABx=Î»x, or BAx=Î»x. |
| ?HPEVX | C Z | Computes selected eigenvalues and eigenvectors of a Hermitian matrix in packed storage. |
| ?HPGVX | C Z | Computes selected eigenvalues, and optionally, the eigenvectors of a generalized Hermitian-definite generalized eigenproblem, Ax=Î»Bx, ABx=Î»x, or BAx=Î»x, where A and B are in packed storage. |
| ?HBEVX | C Z | Computes selected eigenvalues and eigenvectors of a Hermitian band matrix. |
| ?HBGVX | C Z | Computes selected eigenvalues, and optionally, the eigenvectors of a complex generalized Hermitian-definite banded eigenproblem, of the form Ax=Î»Bx. A and B are assumed to be Hermitian and banded, and B is also positive definite. |
| † ?  indicates prefix which must be filled with a combination of: S = REAL(kind=4), D = REAL(kind=8), C = COMPLEX(kind=4), Z = COMPLEX(kind=8) | | |

## Computational Subprograms

| Name† | Prefixes | Description |
| --- | --- | --- |
| ?BDSDC | S D C Z | Computes the singular value decomposition (SVD) of a real bidiagonal matrix, using a divide and conquer method. |
| ?BDSQR | S D C Z | Computes the singular value decomposition (SVD) of a real bidiagonal matrix, using the bidiagonal QR algorithm. |
| ?DISNA | S D | Computes the reciprocal condition numbers for the eigenvectors of a real symmetric or Hermitian matrix or for the left or right singular vectors of a general matrix. |
| ?GBBRD | S D C Z | Reduces a general band matrix to real upper bidiagonal form by an orthogonal/unitary transformation. |
| ?GBCON | S D C Z | Estimates the reciprocal of the condition number of a general band matrix, in either the 1-norm or the Infinity-norm, using the LU factorization computed by ?GBTRF. |
| ?GBEQU | S D C Z | Computes row and column scalings to equilibrate a general band matrix and reduce its condition number. |
| ?GBRFS | S D C Z | Improves the computed solution to a general banded system of linear equations AX=B, ATX=B or AHX=B, and provides forward and backward error bounds for the solution. |
| ?GBTRF | S D C Z | Computes an LU factorization of a general band matrix, using partial pivoting with row interchanges. |
| ?GBTRS | S D C Z | Solves a general banded system of linear equations AX=B, ATX=B or AHX=B, using the LU factorization computed by ?GBTRF. |
| ?GEBAK | S D C Z | Transforms eigenvectors of a balanced matrix to those of the original matrix supplied to ?GEBAL. |
| ?GEBAL | S D C Z | Balances a general matrix in order to improve the accuracy of computed eigenvalues. |
| ?GEBRD | S D C Z | Reduces a general rectangular matrix to real bidiagonal form by an orthogonal/unitary transformation. |
| ?GECON | S D C Z | Estimates the reciprocal of the condition number of a general matrix, in either the 1-norm or the Infinity-norm, using the LU factorization computed by ?GETRF. |
| ?GEEQU | S D C Z | Computes row and column scalings to equilibrate a general rectangular matrix and reduce its condition number. |
| ?GEHRD | S D C Z | Reduces a general matrix to upper Hessenberg form by an orthogonal/unitary similarity transformation. |
| ?GELQF | S D C Z | Computes an LQ factorization of a general rectangular matrix. |
| ?GEQLF | S D C Z | Computes a QL factorization of a general rectangular matrix. |
| ?GEQP3 | S D C Z | Computes a QR factorization with column pivoting of a general rectangular matrix using Level 3 BLAS. |
| ?GEQPF | S D C Z | Computes a QR factorization with column pivoting of a general rectangular matrix. |
| ?GEQRF | S D C Z | Computes a QR factorization of a general rectangular matrix. |
| ?GERFS | S D C Z | Improves the computed solution to a general system of linear equations AX=B, ATX=B or AHX=B, and provides forward and backward error bounds for the solution. |
| ?GERQF | S D C Z | Computes an RQ factorization of a general rectangular matrix. |
| ?GETRF | S D C Z | Computes an LU factorization of a general matrix, using partial pivoting with row interchanges. |
| ?GETRI | S D C Z | Computes the inverse of a general matrix, using the LU factorization computed by ?GETRF. |
| ?GETRS | S D C Z | Solves a general system of linear equations AX=B, ATX=B or AHX=B, using the LU factorization computed by ?GETRF. |
| ?GGBAK | S D C Z | Forms the right or left eigenvectors of the generalized eigenvalue problem by backward transformation on the computed eigenvectors of the balanced pair of matrices output by ?GGBAL. |
| ?GGBAL | S D C Z | Balances a pair of general matrices for the generalized eigenvalue problem Ax=Î»Bx. |
| ?GGHRD | S D C Z | Reduces a pair of matrices to generalized upper Hessenberg form using orthogonal/unitary similarity transformations. |
| ?GGQRF | S D C Z | Computes a generalized QR factorization of a pair of matrices. |
| ?GGRQF | S D C Z | Computes a generalized RQ factorization of a pair of matrices. |
| ?GGSVP | S D C Z | Computes orthogonal/unitary matrices as a preprocessing step for computing the generalized singular value decomposition. |
| ?GTCON | S D C Z | Estimates the reciprocal of the condition number of a general tridiagonal matrix, in either the 1-norm or the Infinity-norm, using the LU factorization computed by ?GTTRF. |
| ?GTRFS | S D C Z | Improves the computed solution to a general tridiagonal system of linear equations AX=B, ATX=B or AHX=B, and provides forward and backward error bounds for the solution. |
| ?GTTRF | S D C Z | Computes an LU factorization of a general tridiagonal matrix, using partial pivoting with row interchanges. |
| ?GTTRS | S D C Z | Solves a general tridiagonal system of linear equations AX=B, ATX=B or AHX=B, using the LU factorization computed by ?GTTRF. |
| ?HGEQZ | S D C Z | Implements a single/double-shift version of the QZ method for finding the generalized eigenvalues of the equation det(A - w(i) B) = 0. |
| ?HSEIN | S D C Z | Computes specified right and/or left eigenvectors of an upper Hessenberg matrix by inverse iteration. |
| ?HSEQR | S D C Z | Computes the eigenvalues and Schur factorization of an upper Hessenberg matrix, using the multishift QR algorithm. |
| ?TTQRE | S D C Z | Computes the eigenvalues and Schur factorization of an upper Hessenberg matrix, using the multishift QR algorithm with aggressive early deflation. **(Only available on SX.)** |
| ?OPGTR | S D | Generates the orthogonal transformation matrix from a reduction to tridiagonal form determined by ?SPTRD. |
| ?OPMTR | S D | Multiplies a general matrix by the orthogonal transformation matrix from a reduction to tridiagonal form, determined by ?SPTRD. |
| ?ORGBR | S D | Generates the orthogonal transformation matrices from a reduction to bidiagonal form determined by ?GEBRD. |
| ?ORGHR | S D | Generates the orthogonal transformation matrix from a reduction to Hessenberg form determined by ?GEHRD. |
| ?ORGLQ | S D | Generates all or part of the orthogonal matrix Q from an LQ factorization determined by ?GELQF. |
| ?ORGQL | S D | Generates all or part of the orthogonal matrix Q from a QL factorization determined by ?GEQLF. |
| ?ORGQR | S D | Generates all or part of the orthogonal matrix Q from a QR factorization determined by ?GEQRF. |
| ?ORGRQ | S D | Generates all or part of the orthogonal matrix Q from an RQ factorization determined by ?GERQF. |
| ?ORGTR | S D | Generates the orthogonal transformation matrix from a reduction to tridiagonal form determined by ?SYTRD. |
| ?ORMBR | S D | Multiplies a general matrix by one of the orthogonal transformation matrices from a reduction to bidiagonal form determined by ?GEBRD. |
| ?ORMHR | S D | Multiplies a general matrix by the orthogonal transformation matrix from a reduction to Hessenberg form determined by ?GEHRD. |
| ?ORMLQ | S D | Multiplies a general matrix by the orthogonal matrix from an LQ factorization determined by ?GELQF. |
| ?ORMQL | S D | Multiplies a general matrix by the orthogonal matrix from a QL factorization determined by ?GEQLF. |
| ?ORMQR | S D | Multiplies a general matrix by the orthogonal matrix from a QR factorization determined by ?GEQRF. |
| ?ORMR3 | S D | Multiples a general matrix by the orthogonal matrix from an RZ factorization determined by ?TZRZF. |
| ?ORMRQ | S D | Multiplies a general matrix by the orthogonal matrix from an RQ factorization determined by ?GERQF. |
| ?ORMRZ | S D | Multiples a general matrix by the orthogonal matrix from an RZ factorization determined by ?TZRZF. |
| ?ORMTR | S D | Multiplies a general matrix by the orthogonal transformation matrix from a reduction to tridiagonal form determined by ?SYTRD. |
| ?PBCON | S D C Z | Estimates the reciprocal of the condition number of a symmetric/Hermitian positive definite band matrix, using the Cholesky factorization computed by ?PBTRF. |
| ?PBEQU | S D C Z | Computes row and column scalings to equilibrate a symmetric/Hermitian positive definite band matrix and reduce its condition number. |
| ?PBRFS | S D C Z | Improves the computed solution to a symmetric/Hermitian positive definite banded system of linear equations AX=B, and provides forward and backward error bounds for the solution. |
| ?PBSTF | S D C Z | Computes a split Cholesky factorization of a symmetric/Hermitian positive definite band matrix. |
| ?PBTRF | S D C Z | Computes the Cholesky factorization of a symmetric/Hermitian positive definite band matrix. |
| ?PBTRS | S D C Z | Solves a symmetric/Hermitian positive definite banded system of linear equations AX=B, using the Cholesky factorization computed by ?PBTRF. |
| ?POCON | S D C Z | Estimates the reciprocal of the condition number of a symmetric/Hermitian positive definite matrix, using the Cholesky factorization computed by ?POTRF. |
| ?POEQU | S D C Z | Computes row and column scalings to equilibrate a symmetric/Hermitian positive definite matrix and reduce its condition number. |
| ?PORFS | S D C Z | Improves the computed solution to a symmetric/Hermitian positive definite system of linear equations AX=B, and provides forward and backward error bounds for the solution. |
| ?POTRF | S D C Z | Computes the Cholesky factorization of a symmetric/Hermitian positive definite matrix. |
| ?POTRI | S D C Z | Computes the inverse of a symmetric/Hermitian positive definite matrix, using the Cholesky factorization computed by ?POTRF. |
| ?POTRS | S D C Z | Solves a symmetric/Hermitian positive definite system of linear equations AX=B, using the Cholesky factorization computed by ?POTRF. |
| ?PPCON | S D C Z | Estimates the reciprocal of the condition number of a symmetric/Hermitian positive definite matrix in packed storage, using the Cholesky factorization computed by ?PPTRF. |
| ?PPEQU | S D C Z | Computes row and column scalings to equilibrate a symmetric/Hermitian positive definite matrix in packed storage and reduce its condition number. |
| ?PPRFS | S D C Z | Improves the computed solution to a symmetric/Hermitian positive definite system of linear equations AX=B, where A is held in packed storage, and provides forward and backward error bounds for the solution. |
| ?PPTRF | S D C Z | Computes the Cholesky factorization of a symmetric/Hermitian positive definite matrix in packed storage. |
| ?PPTRI | S D C Z | Computes the inverse of a symmetric/Hermitian positive definite matrix in packed storage, using the Cholesky factorization computed by ?PPTRF. |
| ?PPTRS | S D C Z | Solves a symmetric/Hermitian positive definite system of linear equations AX=B, where A is held in packed storage, using the Cholesky factorization computed by ?PPTRF. |
| ?PTCON | S D C Z | Computes the reciprocal of the condition number of a symmetric/Hermitian positive definite tridiagonal matrix, using the LDLH factorization computed by ?PTTRF. |
| ?PTEQR | S D C Z | Computes all eigenvalues and eigenvectors of a symmetric positive definite tridiagonal matrix, by computing the SVD of its bidiagonal Cholesky factor. |
| ?PTRFS | S D C Z | Improves the computed solution to a symmetric/Hermitian positive definite tridiagonal system of linear equations AX=B, and provides forward and backward error bounds for the solution. |
| ?PTTRF | S D C Z | Computes the LDLH factorization of a symmetric/Hermitian positive definite tridiagonal matrix. |
| ?PTTRS | S D C Z | Solves a symmetric/Hermitian positive definite tridiagonal system of linear equations, using the LDLH factorization computed by ?PTTRF. |
| ?SBGST | S D | Reduces a real symmetric-definite banded generalized eigenproblem Ax=Î»Bx to standard form, where B has been factorized by ?PBSTF (Crawford's algorithm). |
| ?SBTRD | S D | Reduces a symmetric band matrix to real symmetric tridiagonal form by an orthogonal similarity transformation. |
| ?SPCON | S D C Z | Estimates the reciprocal of the condition number of a symmetric indefinite matrix in packed storage, using the factorization computed by ?SPTRF. |
| ?SPGST | S D | Reduces a symmetric-definite generalized eigenproblem Ax=Î»Bx, ABx=Î»x, or BAx=Î»x, to standard form, where A and B are held in packed storage, and B has been factorized by ?PPTRF. |
| ?SPRFS | S D C Z | Improves the computed solution to a symmetric indefinite system of linear equations AX=B, where A is held in packed storage, and provides forward and backward error bounds for the solution. |
| ?SPTRD | S D | Reduces a symmetric matrix in packed storage to real symmetric tridiagonal form by an orthogonal similarity transformation. |
| ?SPTRF | S D C Z | Computes the factorization of a symmetric-indefinite matrix in packed storage, using the diagonal pivoting method. |
| ?SPTRI | S D C Z | Computes the inverse of a symmetric indefinite matrix in packed storage, using the factorization computed by ?SPTRF. |
| ?SPTRS | S D C Z | Solves a symmetric indefinite system of linear equations AX=B, where A is held in packed storage, using the factorization computed by ?SPTRF. |
| ?STEBZ | S D | Computes selected eigenvalues of a real symmetric tridiagonal matrix by bisection. |
| ?STEDC | S D C Z | Computes all eigenvalues and, optionally, eigenvectors of a symmetric tridiagonal matrix using the divide and conquer algorithm. |
| ?STEGR | S D C Z | Computes selected eigenvalues and, optionally, eigenvectors of a symmetric tridiagonal matrix. The eigenvalues are computed by the dqds algorithm, while eigenvectors are computed from various "good" LDLT representations (also known as Relatively Robust Representations). |
| ?STEIN | S D C Z | Computes selected eigenvectors of a real symmetric tridiagonal matrix by inverse iteration. |
| ?STEQR | S D C Z | Computes all eigenvalues and eigenvectors of a real symmetric tridiagonal matrix, using the implicit QL or QR algorithm. |
| ?STERF | S D | Computes all eigenvalues of a real symmetric tridiagonal matrix, using a root-free variant of the QL or QR algorithm. |
| ?SYCON | S D C Z | Estimates the reciprocal of the condition number of a symmetric indefinite matrix, using the factorization computed by ?SYTRF. |
| ?SYGST | S D | Reduces a symmetric-definite generalized eigenproblem Ax=Î»Bx, ABx=Î»x, or BAx=Î»x, to standard form, where B has been factorized by ?POTRF. |
| ?SYRFS | S D C Z | Improves the computed solution to a symmetric indefinite system of linear equations AX=B, and provides forward and backward error bounds for the solution. |
| ?SYTRD | S D | Reduces a symmetric matrix to real symmetric tridiagonal form by an orthogonal similarity transformation. |
| ?SYTRF | S D C Z | Computes the factorization of a symmetric-indefinite matrix, using the diagonal pivoting method. |
| ?SYTRI | S D C Z | Computes the inverse of a symmetric indefinite matrix, using the factorization computed by ?SYTRF. |
| ?SYTRS | S D C Z | Solves a symmetric indefinite system of linear equations AX=B, using the factorization computed by ?SPTRF. |
| ?TBCON | S D C Z | Estimates the reciprocal of the condition number of a triangular band matrix, in either the 1-norm or the Infinity-norm. |
| ?TBRFS | S D C Z | Provides forward and backward error bounds for the solution of a triangular banded system of linear equations AX=B, ATX=B or AHX=B. |
| ?TBTRS | S D C Z | Solves a triangular banded system of linear equations AX=B, ATX=B or AHX=B. |
| ?TGEVC | S D C Z | Computes some or all of the right and/or left generalized eigenvectors of a pair of upper triangular matrices. |
| ?TGEXC | S D C Z | Reorders the generalized Schur decomposition of a matrix pair (A,B) using an orthogonal/unitary equivalence transformation so that the diagonal block of (A,B) with row index IFST is moved to row ILST. |
| ?TGSEN | S D C Z | Reorders the generalized Schur decomposition of a matrix pair (A,B) so that a selected cluster of eigenvalues appears in the leading diagonal blocks of the pair (A,B). |
| ?TGSJA | S D C Z | Computes the generalized singular value decomposition of two upper triangular (or trapezoidal) matrices as output by ?GGSVP. |
| ?TGSNA | S D C Z | Estimates reciprocal condition numbers for specified eigenvalues and/or eigenvectors of a matrix pair (A,B) in generalized Schur canonical form, as returned by ?GGES. |
| ?TGSYL | S D C Z | Solves the generalized Sylvester equation. |
| ?TPCON | S D C Z | Estimates the reciprocal of the condition number of a triangular matrix in packed storage, in either the 1-norm or the Infinity-norm. |
| ?TPRFS | S D C Z | Provides forward and backward error bounds for the solution of a triangular system of linear equations AX=B, ATX=B or AHX=B, where A is held in packed storage. |
| ?TPTRI | S D C Z | Computes the inverse of a triangular matrix in packed storage. |
| ?TPTRS | S D C Z | Solves a triangular system of linear equations AX=B, ATX=B or AHX=B, where A is held in packed storage. |
| ?TRCON | S D C Z | Estimates the reciprocal of the condition number of a triangular matrix, in either the 1-norm or the Infinity-norm. |
| ?TREVC | S D C Z | Computes left and right eigenvectors of an upper quasi-triangular/triangular matrix. |
| ?TREXC | S D C Z | Reorders the Schur factorization of a matrix by an orthogonal/unitary similarity transformation. |
| ?TRRFS | S D C Z | Provides forward and backward error bounds for the solution of a triangular system of linear equations AX=B, ATX=B or AHX=B. |
| ?TRSEN | S D C Z | Reorders the Schur factorization of a matrix in order to find an orthonormal basis of a right invariant subspace corresponding to selected eigenvalues, and returns reciprocal condition numbers (sensitivities) of the average of the cluster of eigenvalues and of the invariant subspace. |
| ?TRSNA | S D C Z | Estimates the reciprocal condition numbers (sensitivities) of selected eigenvalues and eigenvectors of an upper quasi-triangular/triangular matrix. |
| ?TRSYL | S D C Z | Solves the Sylvester matrix equation AX ± XB=C, where A and B are upper quasi-triangular/triangular, and may be transposed. |
| ?TRTRI | S D C Z | Computes the inverse of a triangular matrix. |
| ?TRTRS | S D C Z | Solves a triangular system of linear equations AX=B, ATX=B or AHX=B. |
| ?TZRQF | S D C Z | Computes an RQ factorization of an upper trapezoidal matrix. |
| ?TZRZF | S D C Z | Computes an RZ factorization of an upper trapezoidal matrix (blocked version of ?TZRQF). |
| ?UPGTR | C Z | Generates the unitary transformation matrix from a reduction to tridiagonal form determined by ?HPTRD. |
| ?UPMTR | C Z | Multiplies a general matrix by the unitary transformation matrix from a reduction to tridiagonal form determined by ?HPTRD. |
| ?UNGBR | C Z | Generates the unitary transformation matrices from a reduction to bidiagonal form determined by ?GEBRD. |
| ?UNGHR | C Z | Generates the unitary transformation matrix from a reduction to Hessenberg form determined by ?GEHRD. |
| ?UNGLQ | C Z | Generates all or part of the unitary matrix Q from an LQ factorization determined by ?GELQF. |
| ?UNGQL | C Z | Generates all or part of the unitary matrix Q from a QL factorization determined by ?GEQLF. |
| ?UNGQR | C Z | Generates all or part of the unitary matrix Q from a QR factorization determined by ?GEQRF. |
| ?UNGRQ | C Z | Generates all or part of the unitary matrix Q from an RQ factorization determined by ?GERQF. |
| ?UNGTR | C Z | Generates the unitary transformation matrix from a reduction to tridiagonal form determined by ?HETRD. |
| ?UNMBR | C Z | Multiplies a general matrix by one of the unitary transformation matrices from a reduction to bidiagonal form determined by ?GEBRD. |
| ?UNMHR | C Z | Multiplies a general matrix by the unitary transformation matrix from a reduction to Hessenberg form determined by ?GEHRD. |
| ?UNMLQ | C Z | Multiplies a general matrix by the unitary matrix from an LQ factorization determined by ?GELQF. |
| ?UNMQL | C Z | Multiplies a general matrix by the unitary matrix from a QL factorization determined by ?GEQLF. |
| ?UNMQR | C Z | Multiplies a general matrix by the unitary matrix from a QR factorization determined by ?GEQRF. |
| ?UNMR3 | C Z | Multiples a general matrix by the unitary matrix from an RZ factorization determined by ?TZRZF. |
| ?UNMRQ | C Z | Multiplies a general matrix by the unitary matrix from an RQ factorization determined by ?GERQF. |
| ?UNMRZ | C Z | Multiples a general matrix by the unitary matrix from an RZ factorization determined by ?TZRZF. |
| ?UNMTR | C Z | Multiplies a general matrix by the unitary transformation matrix from a reduction to tridiagonal form determined by ?HETRD. |
| ?HBGST | C Z | Reduces a Hermitian-definite banded generalized eigenproblem Ax=Î»Bx to standard form, where B has been factorized by ?PBSTF (Crawford's algorithm). |
| ?HBTRD | C Z | Reduces a Hermitian band matrix to real symmetric tridiagonal form by a unitary similarity transformation. |
| ?HPCON | C Z | Estimates the reciprocal of the condition number of a Hermitian indefinite matrix in packed storage, using the factorization computed by ?HPTRF. |
| ?HPGST | C Z | Reduces a Hermitian-definite generalized eigenproblem Ax=Î»Bx, ABx=Î»x, or BAx=Î»x, to standard form, where A and B are held in packed storage, and B has been factorized by ?PPTRF. |
| ?HPRFS | C Z | Improves the computed solution to a Hermitian indefinite system of linear equations AX=B, where A is held in packed storage, and provides forward and backward error bounds for the solution. |
| ?HPTRD | C Z | Reduces a Hermitian matrix in packed storage to real symmetric tridiagonal form by a unitary similarity transformation. |
| ?HPTRF | C Z | Computes the factorization of a Hermitian-indefinite matrix in packed storage, using the diagonal pivoting method. |
| ?HPTRI | C Z | Computes the inverse of a Hermitian indefinite matrix in packed storage, using the factorization computed by ?HPTRF. |
| ?HPTRS | C Z | Solves a Hermitian indefinite system of linear equations AX=B, where A is held in packed storage, using the factorization computed by ?HPTRF. |
| ?HECON | C Z | Estimates the reciprocal of the condition number of a Hermitian indefinite matrix, using the factorization computed by ?HETRF. |
| ?HEGST | C Z | Reduces a Hermitian-definite generalized eigenproblem Ax=Î»Bx, ABx=Î»x, or BAx=Î»x, to standard form, where B has been factorized by ?POTRF. |
| ?HERFS | C Z | Improves the computed solution to a Hermitian indefinite system of linear equations AX=B, and provides forward and backward error bounds for the solution. |
| ?HETRD | C Z | Reduces a Hermitian matrix to real symmetric tridiagonal form by an orthogonal/unitary similarity transformation. |
| ?HETRF | C Z | Computes the factorization of a Hermitian-indefinite matrix, using the diagonal pivoting method. |
| ?HETRI | C Z | Computes the inverse of a Hermitian indefinite matrix, using the factorization computed by ?HETRF. |
| ?HETRS | C Z | Solves a Hermitian indefinite system of linear equations AX=B, using the factorization computed by ?HPTRF. |
| † ?  indicates prefix which must be filled with a combination of: S = REAL(kind=4), D = REAL(kind=8), C = COMPLEX(kind=4), Z = COMPLEX(kind=8) | | |

## Auxiliary Subprograms

| Prefixes | Routine† |
| --- | --- |
| S D C Z | ?GBTF2  ?GEBD2  ?GEHD2  ?GELQ2  ?GEQL2  ?GEQR2  ?GERQ2  ?GESC2  ?GETC2  ?GETF2  ?GTTS2  ?LABRD  ?LACON  ?LACPY  ?LADIV  ?LAED0  ?LAED7  ?LAED8  ?LAEIN  ?LAEV2  ?LAGTM  ?LAHQR  ?LAHRD  ?LAIC1  ?LALS0  ?LALSA  ?LALSD  ?LANGB  ?LANGE  ?LANGT  ?LANHS  ?LANSB  ?LANSP  ?LANSY  ?LANTB  ?LANTP  ?LANTR  ?LAPLL  ?LAPMT  ?LAQGB  ?LAQGE  ?LAQP2  ?LAQPS  ?LAQSB  ?LAQSP  ?LAQSY  ?LAR1V  ?LAR2V  ?LARF  ?LARFB  ?LARFG  ?LARFT  ?LARFX  ?LARGV  ?LARNV  ?LARRV  ?LARTG  ?LARTV  ?LARZ  ?LARZB  ?LARZT  ?LASCL  ?LASET  ?LASR  ?LASSQ  ?LASWP  ?LASYF  ?LATBS  ?LATDF  ?LATPS  ?LATRD  ?LATRS  ?LATRZ  ?LAUU2  ?LAUUM  ?PBTF2  ?POTF2  ?PTTS2  ?SYTF2  ?TGEX2  ?TGSY2  ?TRTI2 |
| S D | ?LABAD  ?LAE2  ?LAEBZ  ?LAED1  ?LAED2  ?LAED3  ?LAED4  ?LAED5  ?LAED6  ?LAED9  ?LAEDA  ?LAEXC  ?LAG2  ?LAGS2  ?LAGTF  ?LAGTS  ?LAGV2  ?LALN2  ?LAMCH  ?LAMRG  ?LANST  ?LANV2  ?LAPY2  ?LAPY3  ?LAQTR  ?LARRB  ?LARRE  ?LARRF  ?LARUV  ?LAS2  ?LASD0  ?LASD1  ?LASD2  ?LASD3  ?LASD4  ?LASD5  ?LASD6  ?LASD7  ?LASD8  ?LASD9  ?LASDA  ?LASDQ  ?LASDT  ?LASQ1  ?LASQ2  ?LASQ3  ?LASQ4  ?LASQ5  ?LASQ6  ?LASRT  ?LASV2  ?LASY2  ?ORG2L  ?ORG2R  ?ORGL2  ?ORGR2  ?ORM2L  ?ORM2R  ?ORML2  ?ORMR2  ?ORMR3  ?RSCL  ?SYGS2  ?SYTD2 |
| C Z | ?HEGS2  ?HETD2  ?HETF2  ?LACGV  ?LACRM  ?LACRT  ?LAESY  ?LAHEF  ?LANHB  ?LANHE  ?LANHP  ?LANHT  ?SRSCL  ?UNG2L  ?UNG2R  ?UNGL2  ?UNGR2  ?UNM2L  ?UNM2R  ?UNML2  ?UNMR2  ?UNMR3 |
| **n/a** | ILAENV  LSAME  LSAMEN  XERBLA |
| † ?  indicates prefix which must be filled with a combination of: S = REAL(kind=4), D = REAL(kind=8), C = COMPLEX(kind=4), Z = COMPLEX(kind=8) | |

# 2 函数 - DOUBLE

## DBBCSD

\*> \brief \b DBBCSD

\*

\*  =========== DOCUMENTATION ===========

\*

\* Online html documentation available at

\*            http://www.netlib.org/lapack/explore-html/

\*

\*> \htmlonly

\*> Download DBBCSD + dependencies

\*> <a href="http://www.netlib.org/cgi-bin/netlibfiles.tgz?format=tgz&filename=/lapack/lapack\_routine/dbbcsd.f">

\*> [TGZ]</a>

\*> <a href="http://www.netlib.org/cgi-bin/netlibfiles.zip?format=zip&filename=/lapack/lapack\_routine/dbbcsd.f">

\*> [ZIP]</a>

\*> <a href="http://www.netlib.org/cgi-bin/netlibfiles.txt?format=txt&filename=/lapack/lapack\_routine/dbbcsd.f">

\*> [TXT]</a>

\*> \endhtmlonly

\*

\*  Definition:

\*  ===========

\*

\*       SUBROUTINE DBBCSD( JOBU1, JOBU2, JOBV1T, JOBV2T, TRANS, M, P, Q,

\*                          THETA, PHI, U1, LDU1, U2, LDU2, V1T, LDV1T,

\*                          V2T, LDV2T, B11D, B11E, B12D, B12E, B21D, B21E,

\*                          B22D, B22E, WORK, LWORK, INFO )

\*

\*       .. Scalar Arguments ..

\*       CHARACTER          JOBU1, JOBU2, JOBV1T, JOBV2T, TRANS

\*       INTEGER            INFO, LDU1, LDU2, LDV1T, LDV2T, LWORK, M, P, Q

\*       ..

\*       .. Array Arguments ..

\*       DOUBLE PRECISION   B11D( \* ), B11E( \* ), B12D( \* ), B12E( \* ),

\*      $                   B21D( \* ), B21E( \* ), B22D( \* ), B22E( \* ),

\*      $                   PHI( \* ), THETA( \* ), WORK( \* )

\*       DOUBLE PRECISION   U1( LDU1, \* ), U2( LDU2, \* ), V1T( LDV1T, \* ),

\*      $                   V2T( LDV2T, \* )

\*       ..

\*

\*

\*> \par Purpose:

\*  =============

\*>

\*> \verbatim

\*>

\*> DBBCSD computes the CS decomposition of an orthogonal matrix in

\*> bidiagonal-block form,

\*>

\*>

\*>     [ B11 | B12 0  0 ]

\*>     [  0  |  0 -I  0 ]

\*> X = [----------------]

\*>     [ B21 | B22 0  0 ]

\*>     [  0  |  0  0  I ]

\*>

\*>                               [  C | -S  0  0 ]

\*>                   [ U1 |    ] [  0 |  0 -I  0 ] [ V1 |    ]\*\*T

\*>                 = [---------] [---------------] [---------]   .

\*>                   [    | U2 ] [  S |  C  0  0 ] [    | V2 ]

\*>                               [  0 |  0  0  I ]

\*>

\*> X is M-by-M, its top-left block is P-by-Q, and Q must be no larger

\*> than P, M-P, or M-Q. (If Q is not the smallest index, then X must be

\*> transposed and/or permuted. This can be done in constant time using

\*> the TRANS and SIGNS options. See DORCSD for details.)

\*>

\*> The bidiagonal matrices B11, B12, B21, and B22 are represented

\*> implicitly by angles THETA(1:Q) and PHI(1:Q-1).

\*>

\*> The orthogonal matrices U1, U2, V1T, and V2T are input/output.

\*> The input matrices are pre- or post-multiplied by the appropriate

\*> singular vector matrices.

\*> \endverbatim

\*

\*  Arguments:

\*  ==========

\*

\*> \param[in] JOBU1

\*> \verbatim

\*>          JOBU1 is CHARACTER

\*>          = 'Y':      U1 is updated;

\*>          otherwise:  U1 is not updated.

\*> \endverbatim

\*>

\*> \param[in] JOBU2

\*> \verbatim

\*>          JOBU2 is CHARACTER

\*>          = 'Y':      U2 is updated;

\*>          otherwise:  U2 is not updated.

\*> \endverbatim

\*>

\*> \param[in] JOBV1T

\*> \verbatim

\*>          JOBV1T is CHARACTER

\*>          = 'Y':      V1T is updated;

\*>          otherwise:  V1T is not updated.

\*> \endverbatim

\*>

\*> \param[in] JOBV2T

\*> \verbatim

\*>          JOBV2T is CHARACTER

\*>          = 'Y':      V2T is updated;

\*>          otherwise:  V2T is not updated.

\*> \endverbatim

\*>

\*> \param[in] TRANS

\*> \verbatim

\*>          TRANS is CHARACTER

\*>          = 'T':      X, U1, U2, V1T, and V2T are stored in row-major

\*>                      order;

\*>          otherwise:  X, U1, U2, V1T, and V2T are stored in column-

\*>                      major order.

\*> \endverbatim

\*>

\*> \param[in] M

\*> \verbatim

\*>          M is INTEGER

\*>          The number of rows and columns in X, the orthogonal matrix in

\*>          bidiagonal-block form.

\*> \endverbatim

\*>

\*> \param[in] P

\*> \verbatim

\*>          P is INTEGER

\*>          The number of rows in the top-left block of X. 0 <= P <= M.

\*> \endverbatim

\*>

\*> \param[in] Q

\*> \verbatim

\*>          Q is INTEGER

\*>          The number of columns in the top-left block of X.

\*>          0 <= Q <= MIN(P,M-P,M-Q).

\*> \endverbatim

\*>

\*> \param[in,out] THETA

\*> \verbatim

\*>          THETA is DOUBLE PRECISION array, dimension (Q)

\*>          On entry, the angles THETA(1),...,THETA(Q) that, along with

\*>          PHI(1), ...,PHI(Q-1), define the matrix in bidiagonal-block

\*>          form. On exit, the angles whose cosines and sines define the

\*>          diagonal blocks in the CS decomposition.

\*> \endverbatim

\*>

\*> \param[in,out] PHI

\*> \verbatim

\*>          PHI is DOUBLE PRECISION array, dimension (Q-1)

\*>          The angles PHI(1),...,PHI(Q-1) that, along with THETA(1),...,

\*>          THETA(Q), define the matrix in bidiagonal-block form.

\*> \endverbatim

\*>

\*> \param[in,out] U1

\*> \verbatim

\*>          U1 is DOUBLE PRECISION array, dimension (LDU1,P)

\*>          On entry, a P-by-P matrix. On exit, U1 is postmultiplied

\*>          by the left singular vector matrix common to [ B11 ; 0 ] and

\*>          [ B12 0 0 ; 0 -I 0 0 ].

\*> \endverbatim

\*>

\*> \param[in] LDU1

\*> \verbatim

\*>          LDU1 is INTEGER

\*>          The leading dimension of the array U1, LDU1 >= MAX(1,P).

\*> \endverbatim

\*>

\*> \param[in,out] U2

\*> \verbatim

\*>          U2 is DOUBLE PRECISION array, dimension (LDU2,M-P)

\*>          On entry, an (M-P)-by-(M-P) matrix. On exit, U2 is

\*>          postmultiplied by the left singular vector matrix common to

\*>          [ B21 ; 0 ] and [ B22 0 0 ; 0 0 I ].

\*> \endverbatim

\*>

\*> \param[in] LDU2

\*> \verbatim

\*>          LDU2 is INTEGER

\*>          The leading dimension of the array U2, LDU2 >= MAX(1,M-P).

\*> \endverbatim

\*>

\*> \param[in,out] V1T

\*> \verbatim

\*>          V1T is DOUBLE PRECISION array, dimension (LDV1T,Q)

\*>          On entry, a Q-by-Q matrix. On exit, V1T is premultiplied

\*>          by the transpose of the right singular vector

\*>          matrix common to [ B11 ; 0 ] and [ B21 ; 0 ].

\*> \endverbatim

\*>

\*> \param[in] LDV1T

\*> \verbatim

\*>          LDV1T is INTEGER

\*>          The leading dimension of the array V1T, LDV1T >= MAX(1,Q).

\*> \endverbatim

\*>

\*> \param[in,out] V2T

\*> \verbatim

\*>          V2T is DOUBLE PRECISION array, dimension (LDV2T,M-Q)

\*>          On entry, an (M-Q)-by-(M-Q) matrix. On exit, V2T is

\*>          premultiplied by the transpose of the right

\*>          singular vector matrix common to [ B12 0 0 ; 0 -I 0 ] and

\*>          [ B22 0 0 ; 0 0 I ].

\*> \endverbatim

\*>

\*> \param[in] LDV2T

\*> \verbatim

\*>          LDV2T is INTEGER

\*>          The leading dimension of the array V2T, LDV2T >= MAX(1,M-Q).

\*> \endverbatim

\*>

\*> \param[out] B11D

\*> \verbatim

\*>          B11D is DOUBLE PRECISION array, dimension (Q)

\*>          When DBBCSD converges, B11D contains the cosines of THETA(1),

\*>          ..., THETA(Q). If DBBCSD fails to converge, then B11D

\*>          contains the diagonal of the partially reduced top-left

\*>          block.

\*> \endverbatim

\*>

\*> \param[out] B11E

\*> \verbatim

\*>          B11E is DOUBLE PRECISION array, dimension (Q-1)

\*>          When DBBCSD converges, B11E contains zeros. If DBBCSD fails

\*>          to converge, then B11E contains the superdiagonal of the

\*>          partially reduced top-left block.

\*> \endverbatim

\*>

\*> \param[out] B12D

\*> \verbatim

\*>          B12D is DOUBLE PRECISION array, dimension (Q)

\*>          When DBBCSD converges, B12D contains the negative sines of

\*>          THETA(1), ..., THETA(Q). If DBBCSD fails to converge, then

\*>          B12D contains the diagonal of the partially reduced top-right

\*>          block.

\*> \endverbatim

\*>

\*> \param[out] B12E

\*> \verbatim

\*>          B12E is DOUBLE PRECISION array, dimension (Q-1)

\*>          When DBBCSD converges, B12E contains zeros. If DBBCSD fails

\*>          to converge, then B12E contains the subdiagonal of the

\*>          partially reduced top-right block.

\*> \endverbatim

\*>

\*> \param[out] B21D

\*> \verbatim

\*>          B21D is DOUBLE PRECISION  array, dimension (Q)

\*>          When DBBCSD converges, B21D contains the negative sines of

\*>          THETA(1), ..., THETA(Q). If DBBCSD fails to converge, then

\*>          B21D contains the diagonal of the partially reduced bottom-left

\*>          block.

\*> \endverbatim

\*>

\*> \param[out] B21E

\*> \verbatim

\*>          B21E is DOUBLE PRECISION  array, dimension (Q-1)

\*>          When DBBCSD converges, B21E contains zeros. If DBBCSD fails

\*>          to converge, then B21E contains the subdiagonal of the

\*>          partially reduced bottom-left block.

\*> \endverbatim

\*>

\*> \param[out] B22D

\*> \verbatim

\*>          B22D is DOUBLE PRECISION  array, dimension (Q)

\*>          When DBBCSD converges, B22D contains the negative sines of

\*>          THETA(1), ..., THETA(Q). If DBBCSD fails to converge, then

\*>          B22D contains the diagonal of the partially reduced bottom-right

\*>          block.

\*> \endverbatim

\*>

\*> \param[out] B22E

\*> \verbatim

\*>          B22E is DOUBLE PRECISION  array, dimension (Q-1)

\*>          When DBBCSD converges, B22E contains zeros. If DBBCSD fails

\*>          to converge, then B22E contains the subdiagonal of the

\*>          partially reduced bottom-right block.

\*> \endverbatim

\*>

\*> \param[out] WORK

\*> \verbatim

\*>          WORK is DOUBLE PRECISION array, dimension (MAX(1,LWORK))

\*>          On exit, if INFO = 0, WORK(1) returns the optimal LWORK.

\*> \endverbatim

\*>

\*> \param[in] LWORK

\*> \verbatim

\*>          LWORK is INTEGER

\*>          The dimension of the array WORK. LWORK >= MAX(1,8\*Q).

\*>

\*>          If LWORK = -1, then a workspace query is assumed; the

\*>          routine only calculates the optimal size of the WORK array,

\*>          returns this value as the first entry of the work array, and

\*>          no error message related to LWORK is issued by XERBLA.

\*> \endverbatim

\*>

\*> \param[out] INFO

\*> \verbatim

\*>          INFO is INTEGER

\*>          = 0:  successful exit.

\*>          < 0:  if INFO = -i, the i-th argument had an illegal value.

\*>          > 0:  if DBBCSD did not converge, INFO specifies the number

\*>                of nonzero entries in PHI, and B11D, B11E, etc.,

\*>                contain the partially reduced matrix.

\*> \endverbatim

\*

\*> \par Internal Parameters:

\*  =========================

\*>

\*> \verbatim

\*>  TOLMUL  DOUBLE PRECISION, default = MAX(10,MIN(100,EPS\*\*(-1/8)))

\*>          TOLMUL controls the convergence criterion of the QR loop.

\*>          Angles THETA(i), PHI(i) are rounded to 0 or PI/2 when they

\*>          are within TOLMUL\*EPS of either bound.

\*> \endverbatim

\*

\*> \par References:

\*  ================

\*>

\*>  [1] Brian D. Sutton. Computing the complete CS decomposition. Numer.

\*>      Algorithms, 50(1):33-65, 2009.

\*

\*  Authors:

\*  ========

\*

\*> \author Univ. of Tennessee

\*> \author Univ. of California Berkeley

\*> \author Univ. of Colorado Denver

\*> \author NAG Ltd.

\*

\*> \date June 2016

\*

\*> \ingroup doubleOTHERcomputational

\*

\*  =====================================================================

## DBDSDC

## DBDSQR

## DBDSVDX

## DCOMBSSQ

## DDISNA

## DGBBRD

## DGBCON

## DGBEQU

## DGBEQUB

## DGBRFS

## DGBRFSX

## DGBSV

## DGBSVX

## DGBSVXX

## DGBTF2

## DGBTRF

## DGBTRS

## DGEBAK

## DGEBAL

## DGEBD2

## DGEBRD

## DGECON

## DGEEQU

## DGEEQUB

## DGEES

## DGEESX

## DGEEV

## DGEEVX

## DGEHD2

## DGEHRD

## DGEJSV

## DGELQ

## DGELQ2

## DGELQF

## DGELQT

## DGELQT3

## DGELS

## DGELSD

## DGELSS

## DGELSY

## DGEMLQ

## DGEMLQT

## DGEMQR

## DGEMQRT

## DGEQL2

## DGEQLF

## DGEQP3

## DGEQR

## DGEQR2

## DGEQR2P

## DGEQRF

## DGEQRFP

## DGEQRT

## DGEQRT2

## DGEQRT3

## DGERFS

## DGERFSX

## DGERQ2

## DGERQF

## DGESC2

## DGESDD

## DGESV

## DGESVD

## DGESVDQ

## DGESVDX

## DGESVJ

## DGESVX

## DGESVXX

## DGETC2

## DGETF2

## DGETRF

## DGETRF2

## DGETRI

## DGETRS

## DGETSLS

## DGGBAK

## DGGBAL

## DGGES

## DGGES3

## DGGESX

## DGGEV

## DGGEV3

## DGGEVX

## DGGGLM

## DGGHD3

## DGGHRD

## DGGLSE

## DGGQRF

## DGGRQF

## DGGSVD3

## DGGSVP3

## DGSVJ0

## DGSVJ1

## DGTCON

## DGTRFS

## DGTSV

## DGTSVX

## DGTTRF

## DGTTRS

## DGTTS2

## DHGEQZ

## DHSEIN

## DHSEQR

## DISNAN

# 3 函数 - SINGLE

# 4 函数 - COMPLAX

# 5