297

UTL_NLA

The `UTL_NLA` package exposes a subset of the BLAS and LAPACK (Version 3.0) operations on vectors and matrices represented as `VARRAY`s.

This chapter contains the following topics:

UTL_NLA Overview

The `UTL_NLA` package exposes a subset of the BLAS (Basic Linear Algebra Subprograms) and LAPACK (Linear Algebra PACKage) (Version 3.0) operations on vectors and matrices represented as `VARRAY`s.

Standards

For more information on the BLAS and LAPACK standards see

```
http://www.netlib.org/blas/
http://www.netlib.org/lapack/
```

Required Expertise

Users of this package are expected to have a sound grasp of linear algebra in general and of the BLAS and LAPACK libraries in particular.

Implementation

The mapping between BLAS and LAPACK procedures and their corresponding PL/SQL calls is one-to-one.

- All BLAS functions have the `BLAS_` prefix (for example, the BLAS_ASUM Functions). The subroutines and functions in BLAS are mapped to PL/SQL procedures and functions, respectively.
- All LAPACK functions have the `LAPACK_` prefix (for example, the LAPACK_GBSV Procedures). The subroutines in LAPACK are mapped to PL/SQL procedures. Procedures

that perform the same operation but differ only on the datatype of the arguments have the same overloaded names.

The mapping between BLAS and LAPACK procedure parameters and those of their corresponding PL/SQL subprograms is almost one-to-one.

- Also in the PL/SQL interface for LAPACK, all `/work/` arguments have been removed. The `UTL_NLA` package manages the allocation and de-allocation of all work areas required by the libraries.
- A new optional parameter, `pack`, has been added to the end of each LAPACK procedure that specifies if the matrix has been linearized in the row-major or column-major (default) format.

UTL_NLA Rules and Limits

Vectors and matrices are stored in `VARRAY`s with a maximum size of one million entries. Given this restriction, `UTL_NLA` vectors can be up to one million entries but matrices need to be of size RxC <= 1,000,000.

UTL_NLA Security Model

The `UTL_NLA` package is owned by user `SYS` and is installed as part of database installation. Execution privilege on the package is granted to public. The routines in the package are run with invokers' rights (run with the privileges of the current user).

Subprogram Groups

The `UTL_NLA` package contains subprogram groups for BLAS and LAPACK operations.

- BLAS Level 1 (Vector-Vector Operations) Subprograms
- BLAS Level 2 (Matrix-Vector Operations) Subprograms
- BLAS Level 3 (Matrix-Matrix Operations) Subprograms
- LAPACK Driver Routines (Linear Equations) Subprograms
- LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms

UTL_NLA BLAS Level 1 (Vector-Vector Operations) Subprograms

This table lists and briefly describes the UTL_NLA BLAS Level 1 Vector-Vector Operations subprograms.

Table 297-1 BLAS Level 1 (Vector-Vector Operations) Subprograms

Subprogram	Description
BLAS_ASUM Functions	Computes the sum of the absolute values of the vector components
BLAS_AXPY Procedures	Copies `alpha*X + Y` into vector `Y`
BLAS_COPY Procedures	Copies the contents of vector `X` to vector `Y`
BLAS_DOT Functions	Returns the dot (scalar) product of two vectors `X` and `Y`

Table 297-1 (Cont.) BLAS Level 1 (Vector-Vector Operations) Subprograms

Subprogram	Description
BLAS_IAMAX Functions	Computes the index of the first element of a vector that has the largest absolute value
BLAS_NRM2 Functions	Computes the vector 2-norm (Euclidean norm)
BLAS_ROT Procedures	Returns the plane rotation of points
BLAS_ROTG Procedures	Returns the Givens rotation of points
BLAS_SCAL Procedures	Scales a vector by a constant
BLAS_SWAP Procedures	Swaps the contents of two vectors each of size n

UTL_NLA BLAS Level 2 (Matrix-Vector Operations) Subprograms

This table lists and briefly describes the UTL_NLA BLAS Level 2 Matrix-Vector Operations subprograms.

Table 297-2 BLAS Level 2 (Matrix-Vector Operations) Subprograms

Subprogram	Description
BLAS_GBMV Procedures	Performs the matrix-vector operation `y := alpha*A*x + beta*y` or `y := alpha*A'*x + beta*y` where `alpha` and `beta` are scalars, `x` and `y` are vectors and `A` is an `m` by `n` band matrix, with `kl` sub-diagonals and `ku` super-diagonals
BLAS_GEMV Procedures	Performs the matrix-vector operations `y := alpha*A*x + beta*y` or `y := alpha*A'*x + beta*y` where `alpha` and `beta` are scalars, `x` and `y` are vectors and `A` is an `m` by `n` matrix
BLAS_GER Procedures	Performs a rank 1 operation `A := alpha*x*y' + A` where `alpha` is a scalar, `x` is an `m` element vector, `y` is an n element vector and `A` is an `m` by `n` matrix
BLAS_SBMV Procedures	Performs a matrix-vector operation `y := alpha*A*x + beta*y` where `alpha` and `beta` are scalars, `x` and `y` are n element vectors and `A` is an `n` by `n` symmetric band matrix, with `k` super-diagonals
BLAS_SPMV Procedures	Performs a matrix-vector operation `y := alpha*A*x + beta*y` where `alpha` and `beta` are scalars, `x` and `y` are `n` element vectors and `A` is an `n` by `n` symmetric matrix, supplied in packed form
BLAS_SPR Procedures	Performs a symmetric rank 1 operation `A := alpha*x*x' + A` where `alpha` is a real scalar, `x` is an `n` element vector, and `A` is an `n` by `n` symmetric matrix, supplied in packed form
BLAS_SPR2 Procedures	Performs a symmetric rank 2 operation `A := alpha*x*y' + alpha*y*x' +A` where `alpha` is a scalar, `x` and `y` are `n` element vectors, and `A` is an `n` by `n` symmetric matrix, supplied in packed form
BLAS_SBMV Procedures	Performs a matrix-vector operation `y := alpha*A*x + beta*y` where `alpha` and `beta` are scalars, `x` and `y` are n element vectors and `A` is an `n` by `n` symmetric band matrix, with `k` super-diagonals
BLAS_SYMV Procedures	Performs a matrix-vector operation `y := alpha*A*x + beta*y` where `alpha` and `beta` are scalars, `x` and `y` are `n` element vectors and `A` is an `n` by `n` symmetric matrix

Table 297-2 (Cont.) BLAS Level 2 (Matrix-Vector Operations) Subprograms

Subprogram	Description
BLAS_SYR Procedures	Performs a symmetric rank 1 operation `A := alpha*x*x' + A` where `alpha` is a real scalar, `x` is an `n` element vector, and `A` is an `n` by `n` symmetric matrix
BLAS_SYR2 Procedures	Performs a symmetric rank 2 operation `A := alpha*x*y' + alpha*y*x' + A` where `alpha` is a scalar, `x` and `y` are `n` element vectors, and `A` is an `n` by `n` symmetric matrix
BLAS_TBMV Procedures	Performs a matrix-vector operation `x := A*x` or `A'*x = b` where `x` is an `n` element vector and `A` is an `n` by `n` unit, or non-unit, upper or lower triangular band matrix, with `(k + 1)` diagonals
BLAS_TBSV Procedures	Solves one of the systems of equation `A*x = b` or `A'*x = b` where `b` and `x` are `n` element vectors and `A` is an `n` by n unit, or non-unit, upper or lower triangular band matrix, with `(k + 1)` diagonals
BLAS_TPMV Procedures	Performs a matrix-vector operation `x := A*x` or `x := A'*x` where `x` is an `n` element vector and `A` is an `n` by `n` unit, or non-unit, upper or lower triangular matrix, supplied in packed form
BLAS_TPSV Procedures	Solves one of the systems of equation `A*x = b` or `A'*x = b` where `b` and `x` are `n` element vectors and `A` is an `n` by `n` unit, or non-unit, upper or lower triangular matrix, supplied in packed form
BLAS_TRMV Procedures	Performs a matrix-vector operation `x := A*x` or `x := A'*x` where `x` is an `n` element vector and `A` is an `n` by `n` unit, or non-unit, upper or lower triangular matrix
BLAS_TRSV Procedures	Solves one of the systems of equation `A*x = b` or `A'*x = b` where `b` and `x` are `n` element vectors and `A` is an n by `n` unit, or non-unit, upper or lower triangular matrix

UTL_NLA BLAS Level 3 (Matrix-Matrix Operations) Subprograms

This table lists and briefly describes the UTL_NLA BLAS Level 3 Matrix-Matrix Operations subprograms.

Table 297-3 BLAS Level 3 (Matrix-Matrix Operations) Subprograms

Subprogram	Description
BLAS_GEMM Procedures	Performs one of the matrix-vector operations `C := alpha*op(A)*op(B) + beta*C` where `op(X) is one of op(X) = X or op(X) = X'` where `alpha` and `beta` are scalars, and `A`, `B` and `C` are matrices, with `op(A)` an `m` by `k` matrix, `op(B)` a `k` by `n` matrix and `C` an `m` by `n` matrix
BLAS_SYMM Procedures	Performs one of the matrix-vector operations `C := alpha*A*B + beta*C` or `C := alpha*B*A + beta*C` where `alpha` and `beta` are scalars, `A` is a symmetric matrix, and `B` and `C` are `m` by `n` matrices

Table 297-3 (Cont.) BLAS Level 3 (Matrix-Matrix Operations) Subprograms

Subprogram	Description
BLAS_SYR2K Procedures	Performs one of the symmetric rank2 k operations `C := alpha*A*B' + alpha*B*A' + beta*C` or `C := alpha*A'*B + alpha*B'*A + beta*C` where `alpha` and `beta` are scalars, `C` is an `n` by `n` symmetric matrix and `A` and `B` are `n` by `k` matrices in the first case and `k` by `n` matrices in the second case
BLAS_SYRK Procedures	Performs one of the symmetric rank `k` operations `C := alpha*A*A' + beta*C` or `C := alpha*A'*A + beta*C` where `alpha` and `beta` are scalars, `C` is an `n` by `n` symmetric matrix and `A` is an `n` by `k` matrix in the first case and a `k` by `n` matrix in the second case
BLAS_TRMM Procedures	Performs one of the matrix-vector operations `B := alpha*op(A)*B` or `B := alpha*B*op(A)` where `alpha` is a scalar, `B` is an `m` by `n` matrix, `A` is a unit, or non-unit, upper or lower triangular matrix and `op(A)` is one of two alternatives
BLAS_TRSM Procedures	Performs one of the matrix-vector operations `op(A)*X = alpha*B` or `X*op(A) = alpha*B` where `alpha` is a scalar, `X` and `B` are `m` by `n` matrices, `A` is a unit, or non-unit, upper or lower triangular matrix, `op(A)` is one of two alternatives. The matrix `X` is overwritten on `B`

UTL_NLA LAPACK Driver Routines (Linear Equations) Subprograms

This table lists and briefly describes the LAPACK Driver Routines (Linear Equations) subprograms.

Table 297-4 LAPACK Driver Routines (Linear Equations) Subprograms

Subprogram	Description
LAPACK_GBSV Procedures	This procedure computes the solution to a real system of linear equations `a * x = b` where `a` is an `n` by `n` matrix and `x` and `b` are `n` by `nrhs` matrices. The `LU` decomposition with partial pivoting and row interchanges is used to factor `A`.
LAPACK_GESV Procedures	This procedure computes the solution to a real system of linear equations `a * x = b` where `a` is an `n` by `n` matrix and `x` and `b` are `n` by `nrhs` matrices. The `LU` decomposition with partial pivoting and row interchanges is used to factor `A`.
LAPACK_GTSV Procedures	This procedure solves the equation `a * x = b` where `a` is an `n` by `n` tridiagonal matrix, by Gaussian elimination with partial pivoting.
LAPACK_PBSV Procedures	This procedure computes the solution to a real system of linear equations `a * x = b` where `a` is an n by n symmetric positive definite band matrix and `x` and `b` are `n` by `nrhs` matrices. The Cholesky decomposition is used to factor `A`.
LAPACK_POSV Procedures	This procedure computes the solution to a real system of linear equations `a * x = b` where `a` is an n by n symmetric positive definite matrix and `x` and `b` are `n` by `nrhs` matrices. The Cholesky decomposition is used to factor `A`.

Table 297-4 (Cont.) LAPACK Driver Routines (Linear Equations) Subprograms

Subprogram	Description
LAPACK_PPSV Procedures	This procedure computes the solution to a real system of linear equations `a * x = b` where `a` is an n by n symmetric positive definite matrix stored in packed format and `x` and `b` are `n` by `nrhs` matrices. The Cholesky decomposition is used to factor `A`.
LAPACK_PTSV Procedures	This procedure computes the solution to a real system of linear equations `a * x = b` where `a` is an n by n symmetric positive definite tridiagonal matrix, and `x` and `b` are `n` by `nrhs` matrices.
LAPACK_SPSV Procedures	This procedure computes the solution to a real system of linear equations `a * x = b` where `a` is an n by n symmetric matrix stored in packed format, and `x` and `b` are `n` by `nrhs` matrices. The diagonal pivoting method is used to factor `A`.
LAPACK_SYSV Procedures	This procedure computes the solution to a real system of linear equations `a * x = b` where `a` is an n by n symmetric matrix, and `x` and `b` are `n` by `nrhs` matrices. The diagonal pivoting method is used to factor `A`.

UTL_NLA LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms

This table lists and briefly describes the LAPACK Driver Routines (LLS and Eigenvalue) subprograms.

Table 297-5 LAPACK Driver Routines (LLS and Eigenvalue Problems)

Subprogram	Description
LAPACK_GEES Procedures	Computes for an `n` by `n` real nonsymmetric matrix `A`, the eigenvalues, the real Schur form `T`, and, optionally, the matrix of Schur vectors `Z`. This gives the Schur factorization `A = Z*T*(Z**T)`.
LAPACK_GEEV Procedures	Computes for an `n` by `n` real nonsymmetric matrix `A`, the eigenvalues and, optionally, the left and/or right eigenvectors.
LAPACK_GELS Procedures	Solves overdetermined or underdetermined real linear systems involving an `m` by `n` matrix `A`, or its transpose, using a `QR` or `LQ` factorization of `A`. It is assumed that `A` has full rank.
LAPACK_GESDD Procedures	Computes the singular value decomposition (SVD) of a real `m` by `n` matrix `A`, optionally computing the left and right singular vectors. If singular vectors are desired, it uses a divide-and-conquer algorithm that makes mild assumptions about floating point arithmetic.
LAPACK_GESVD Procedures	Computes the singular value decomposition (SVD) of a real `m` by `n` matrix `A`, optionally computing the left and/or right singular vectors. The SVD is written `A = U * SIGMA * transpose(V)`.
LAPACK_SBEV Procedures	Computes all the eigenvalues and, optionally, eigenvectors of a real symmetric band matrix `A`
LAPACK_SBEVD Procedures	Ccomputes all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix `A`. If eigenvectors are desired, it uses a divide and conquer algorithm that makes mild assumptions about floating point arithmetic.

Table 297-5 (Cont.) LAPACK Driver Routines (LLS and Eigenvalue Problems)

Subprogram	Description
LAPACK_SPEV Procedures	Computes all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix `A` in packed storage
LAPACK_SPEVD Procedures	Computes all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix `A` in packed storage. If eigenvectors are desired, it uses a divide and conquer algorithm that makes mild assumptions about floating point arithmetic.
LAPACK_STEV Procedures	Computes all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix `A`
LAPACK_STEVD Procedures	Computes all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix `A`. If eigenvectors are desired, it uses a divide and conquer algorithm that makes mild assumptions about floating point arithmetic.
LAPACK_SYEV Procedures	Computes all eigenvalues and, optionally, eigenvectors of a real symmetric matrix `A`
LAPACK_SYEVD Procedures	Computes all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix `A`. If eigenvectors are desired, it uses a divide and conquer algorithm that makes mild assumptions about floating point arithmetic.

Summary of UTL_NLA Subprograms

This table lists the `UTL_NLA` subprograms and briefly describes them.

Table 297-6 UTL_NLA Package Subprograms

Subprogram	Description	Group
BLAS_ASUM Functions	Computes the sum of the absolute values of the vector components	BLAS Level 1 (Vector-Vector Operations) Subprograms
BLAS_AXPY Procedures	Copies `alpha*X + Y` into vector `Y`	BLAS Level 1 (Vector-Vector Operations) Subprograms
BLAS_COPY Procedures	Copies the contents of vector `X` to vector `Y`	BLAS Level 1 (Vector-Vector Operations) Subprograms
BLAS_DOT Functions	Returns the dot (scalar) product of two vectors `X` and `Y`	BLAS Level 1 (Vector-Vector Operations) Subprograms
BLAS_GBMV Procedures	Performs the matrix-vector operation `y := alpha*A*x + beta*y` or `y := alpha*A'*x + beta*y` where `alpha` and `beta` are scalars, `x` and `y` are vectors and `A` is an `m` by `n` band matrix, with `kl` sub-diagonals and `ku` super-diagonals	BLAS Level 2 (Matrix-Vector Operations) Subprograms

Table 297-6 (Cont.) UTL_NLA Package Subprograms

Subprogram	Description	Group
BLAS_GEMM Procedures	Performs one of the matrix-vector operations where `alpha` and `beta` are scalars, and `A`, `B` and `C` are matrices, with `op(A)` an `m` by `k` matrix, `op(B)` a `k` by `n` matrix and `C` an `m` by `n` matrix	BLAS Level 3 (Matrix-Matrix Operations) Subprograms
BLAS_GEMV Procedures	Performs the matrix-vector operations `y := alpha*A*x + beta*y` or `y := alpha*A'*x + beta*y` where `alpha` and `beta` are scalars, `x` and `y` are vectors and `A` is an `m` by `n` matrix	BLAS Level 2 (Matrix-Vector Operations) Subprograms
BLAS_GER Procedures	Performs a rank 1 operation `A := alpha*x*y' + A` where `alpha` is a scalar, `x` is an `m` element vector, `y` is an n element vector and `A` is an `m` by `n` matrix	BLAS Level 2 (Matrix-Vector Operations) Subprograms
BLAS_IAMAX Functions	Computes the index of the first element of a vector that has the largest absolute value	BLAS Level 1 (Vector-Vector Operations) Subprograms
BLAS_NRM2 Functions	Computes the vector 2-norm (Euclidean norm)	BLAS Level 1 (Vector-Vector Operations) Subprograms
BLAS_ROT Procedures	Returns the plane rotation of points	BLAS Level 1 (Vector-Vector Operations) Subprograms
BLAS_ROTG Procedures	Returns the Givens rotation of points	BLAS Level 1 (Vector-Vector Operations) Subprograms
BLAS_SBMV Procedures	Performs a matrix-vector operation `y := alpha*A*x + beta*y` where `alpha` and `beta` are scalars, `x` and `y` are n element vectors and `A` is an `n` by `n` symmetric band matrix, with `k` super-diagonals	BLAS Level 2 (Matrix-Vector Operations) Subprograms
BLAS_SCAL Procedures	Scales a vector by a constant	BLAS Level 1 (Vector-Vector Operations) Subprograms
BLAS_SPMV Procedures	Performs a matrix-vector operation `y := alpha*A*x + beta*y` where `alpha` and `beta` are scalars, `x` and `y` are `n` element vectors and `A` is an `n` by `n` symmetric matrix, supplied in packed form	BLAS Level 2 (Matrix-Vector Operations) Subprograms
BLAS_SPR Procedures	Performs a symmetric rank 1 operation `A := alpha*x*x' + A` where `alpha` is a real scalar, `x` is an `n` element vector, and `A` is an `n` by `n` symmetric matrix, supplied in packed form	BLAS Level 2 (Matrix-Vector Operations) Subprograms

Table 297-6 (Cont.) UTL_NLA Package Subprograms

Subprogram	Description	Group
BLAS_SPR2 Procedures	Performs a symmetric rank 2 operation where `alpha` is a scalar, `x` and `y` are `n` element vectors, and `A` is an `n` by `n` symmetric matrix, supplied in packed form	BLAS Level 2 (Matrix-Vector Operations) Subprograms
BLAS_SWAP Procedures	Swaps the contents of two vectors each of size n	BLAS Level 1 (Vector-Vector Operations) Subprograms
BLAS_SYMM Procedures	Performs one of the matrix-vector operations where `alpha` and `beta` are scalars, `A` is a symmetric matrix, and `B` and `C` are `m` by `n` matrices	BLAS Level 3 (Matrix-Matrix Operations) Subprograms
BLAS_SYMV Procedures	Performs a matrix-vector operation where `alpha` and `beta` are scalars, `x` and `y` are `n` element vectors and `A` is an `n` by `n` symmetric matrix	BLAS Level 2 (Matrix-Vector Operations) Subprograms
BLAS_SYR Procedures	Performs a symmetric rank 1 operation where `alpha` is a real scalar, `x` is an `n` element vector, and `A` is an `n` by `n` symmetric matrix	BLAS Level 2 (Matrix-Vector Operations) Subprograms
BLAS_SYR2 Procedures	Performs a symmetric rank 2 operation where `alpha` is a scalar, `x` and `y` are `n` element vectors, and `A` is an `n` by `n` symmetric matrix	BLAS Level 2 (Matrix-Vector Operations) Subprograms
BLAS_SYR2K Procedures	Performs one of the symmetric rank2 k operations where `alpha` and `beta` are scalars, `C` is an `n` by `n` symmetric matrix and `A` and `B` are `n` by `k` matrices in the first case and `k` by `n` matrices in the second case	BLAS Level 3 (Matrix-Matrix Operations) Subprograms
BLAS_SYRK Procedures	Performs one of the symmetric rank `k` operations where `alpha` and `beta` are scalars, `C` is an `n` by `n` symmetric matrix and `A` is an `n` by `k` matrix in the first case and a `k` by `n` matrix in the second case	BLAS Level 3 (Matrix-Matrix Operations) Subprograms
BLAS_TBMV Procedures	Performs a matrix-vector operation where `x` is an `n` element vector and `A` is an `n` by `n` unit, or non-unit, upper or lower triangular band matrix, with `(k + 1)` diagonals	BLAS Level 2 (Matrix-Vector Operations) Subprograms
BLAS_TBSV Procedures	Solves one of the systems of equation where `b` and `x` are `n` element vectors and `A` is an `n` by n unit, or non-unit, upper or lower triangular band matrix, with `(k + 1)` diagonals	BLAS Level 2 (Matrix-Vector Operations) Subprograms

Table 297-6 (Cont.) UTL_NLA Package Subprograms

Subprogram	Description	Group
BLAS_TPMV Procedures	Performs a matrix-vector operation where `x` is an `n` element vector and `A` is an `n` by `n` unit, or non-unit, upper or lower triangular matrix, supplied in packed form	BLAS Level 2 (Matrix-Vector Operations) Subprograms
BLAS_TPSV Procedures	Solves one of the systems of equation where `b` and `x` are `n` element vectors and `A` is an `n` by n unit, or non-unit, upper or lower triangular matrix, supplied in packed form	BLAS Level 2 (Matrix-Vector Operations) Subprograms
BLAS_TRMM Procedures	Performs one of the matrix-vector operations where`alpha` is a scalar, `B` is an `m` by `n` matrix, `A` is a unit, or non-unit, upper or lower triangular matrix and `op(A)` is one of two alternatives	BLAS Level 2 (Matrix-Vector Operations) Subprograms
BLAS_TRMV Procedures	Performs a matrix-vector operation where `x` is an `n` element vector and `A` is an `n` by `n` unit, or non-unit, upper or lower triangular matrix	BLAS Level 2 (Matrix-Vector Operations) Subprograms
BLAS_TRSM Procedures	Performs one of the matrix-vector operations `op( A )*X = alpha*B` or `X*op(A) = alpha*B` where `alpha` is a scalar, `X` and `B` are `m` by `n` matrices, `A` is a unit, or non-unit, upper or lower triangular matrix, `op(A)` is one of two alternatives. The matrix `X` is overwritten on `B`	BLAS Level 3 (Matrix-Matrix Operations) Subprograms
BLAS_TRSV Procedures	Solves one of the systems of equation where `b` and `x` are `n` element vectors and `A` is an n by `n` unit, or non-unit, upper or lower triangular matrix	BLAS Level 2 (Matrix-Vector Operations) Subprograms
LAPACK_GBSV Procedures	This procedure computes the solution to a real system of linear equations `a * x = b` where `a` is an `n` by `n` matrix and `x` and `b` are `n` by `nrhs` matrices. The `LU` decomposition with partial pivoting and row interchanges is used to factor `A`.	LAPACK Driver Routines (Linear Equations) Subprograms
LAPACK_GEES Procedures	Computes for an `n` by `n` real nonsymmetric matrix `A`, the eigenvalues, the real Schur form `T`, and, optionally, the matrix of Schur vectors `Z`. This gives the Schur factorization `A = Z*T*(Z**T)`.	LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms

Table 297-6 (Cont.) UTL_NLA Package Subprograms

Subprogram	Description	Group
LAPACK_GEEV Procedures	Computes for an `n` by `n` real nonsymmetric matrix `A`, the eigenvalues and, optionally, the left and/or right eigenvectors.	LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms
LAPACK_GELS Procedures	Solves overdetermined or underdetermined real linear systems involving an `m` by `n` matrix `A`, or its transpose, using a `QR` or `LQ` factorization of `A`. It is assumed that `A` has full rank.	LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms
LAPACK_GESDD Procedures	Computes the singular value decomposition (SVD) of a real `m` by `n` matrix `A`, optionally computing the left and right singular vectors. If singular vectors are desired, it uses a divide-and-conquer algorithm that makes mild assumptions about floating point arithmetic.	LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms
LAPACK_GESV Procedures	This procedure computes the solution to a real system of linear equations `a * x = b` where `a` is an `n` by `n` matrix and `x` and `b` are `n` by `nrhs` matrices. The `LU` decomposition with partial pivoting and row interchanges is used to factor `A`.	LAPACK Driver Routines (Linear Equations) Subprograms
LAPACK_GESVD Procedures	Computes the singular value decomposition (SVD) of a real `m` by `n` matrix `A`, optionally computing the left and/or right singular vectors. The SVD is written `A = U * SIGMA * transpose(V)`.	LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms
LAPACK_GTSV Procedures	This procedure solves the equation `a * x = b` where `a` is an `n` by `n` tridiagonal matrix, by Gaussian elimination with partial pivoting.	LAPACK Driver Routines (Linear Equations) Subprograms
LAPACK_PBSV Procedures	This procedure computes the solution to a real system of linear equations `a * x = b` where `a` is an n by n symmetric positive definite band matrix and `x` and `b` are `n` by `nrhs` matrices. The Cholesky decomposition is used to factor `A`.	LAPACK Driver Routines (Linear Equations) Subprograms
LAPACK_POSV Procedures	This procedure computes the solution to a real system of linear equations `a * x = b` where `a` is an n by n symmetric positive definite matrix and `x` and `b` are `n` by `nrhs` matrices. The Cholesky decomposition is used to factor `A`.	LAPACK Driver Routines (Linear Equations) Subprograms

Table 297-6 (Cont.) UTL_NLA Package Subprograms

Subprogram	Description	Group
LAPACK_PPSV Procedures	This procedure computes the solution to a real system of linear equations `a * x = b` where `a` is an n by n symmetric positive definite matrix stored in packed format and `x` and `b` are `n` by `nrhs` matrices. The Cholesky decomposition is used to factor `A`.	LAPACK Driver Routines (Linear Equations) Subprograms
LAPACK_PTSV Procedures	This procedure computes the solution to a real system of linear equations `a * x = b` where `a` is an n by n symmetric positive definite tridiagonal matrix, and `x` and `b` are `n` by `nrhs` matrices.	LAPACK Driver Routines (Linear Equations) Subprograms
LAPACK_SBEV Procedures	Computes all the eigenvalues and, optionally, eigenvectors of a real symmetric band matrix `A`	LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms
LAPACK_SBEVD Procedures	Computes all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix `A`. If eigenvectors are desired, it uses a divide and conquer algorithm that makes mild assumptions about floating point arithmetic.	LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms
LAPACK_SPEV Procedures	Computes all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix `A` in packed storage	LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms
LAPACK_SPEVD Procedures	Computes all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix `A` in packed storage. If eigenvectors are desired, it uses a divide and conquer algorithm that makes mild assumptions about floating point arithmetic.	LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms
LAPACK_SPSV Procedures	This procedure computes the solution to a real system of linear equations `a * x = b` where `a` is an n by n symmetric matrix stored in packed format, and `x` and `b` are `n` by `nrhs` matrices. The diagonal pivoting method is used to factor `A`.	LAPACK Driver Routines (Linear Equations) Subprograms
LAPACK_STEV Procedures	Computes all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix `A`	LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms

Table 297-6 (Cont.) UTL_NLA Package Subprograms

Subprogram	Description	Group
LAPACK_STEVD Procedures	Computes all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix `A`. If eigenvectors are desired, it uses a divide and conquer algorithm that makes mild assumptions about floating point arithmetic.	LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms
LAPACK_SYEVD Procedures	Computes all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix `A`. If eigenvectors are desired, it uses a divide and conquer algorithm that makes mild assumptions about floating point arithmetic.	LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms
LAPACK_SYSV Procedures	This procedure computes the solution to a real system of linear equations `a * x = b` where `a` is an n by n symmetric matrix, and `x` and `b` are `n` by `nrhs` matrices. The diagonal pivoting method is used to factor `A`.	LAPACK Driver Routines (Linear Equations) Subprograms

BLAS_ASUM Functions

This procedure computes the sum of the absolute values of the vector components.

> **See Also:**
>
> BLAS Level 1 (Vector-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_ASUM (
   n      IN       POSITIVEN,
   x      IN       UTL_NLA_ARRAY_DBL,
   incx   IN       POSITIVEN)
 RETURN BINARY_DOUBLE;

UTL_NLA.BLAS_ASUM (
   n      IN       POSITIVEN,
   alpha  IN       SCALAR_DOUBLE,
   x      IN       UTL_NLA_ARRAY_FLT)
 RETURN BINARY_FLOAT
```

Parameters

Table 297-7 BLAS_ASUM Function Parameters

Parameter	Description
n	Specifies the number of elements of the vectors x and y. n must be at least zero.
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (n - 1)*abs(incx))
incx	Specifies the increment for the elements of x. incx must not be zero.

BLAS_AXPY Procedures

This procedure copies alpha*X + Y into vector Y.

> **See Also:**
>
> BLAS Level 1 (Vector-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_AXPY (
   n      IN       POSITIVEN,
   alpha  IN       SCALAR_DOUBLE,
   x      IN       UTL_NLA_ARRAY_DBL,
   incx   IN       POSITIVEN,
   y      IN OUT   UTL_NLA_ARRAY_DBL,
   incy   IN       POSITIVEN);

UTL_NLA.BLAS_AXPY (
   n      IN       POSITIVEN,
   alpha  IN       SCALAR_DOUBLE,
   x      IN       UTL_NLA_ARRAY_FLT,
   incx   IN       POSITIVEN,
   y      IN OUT   UTL_NLA_ARRAY_FLT,
   incy   IN       POSITIVEN);
```

Parameters

Table 297-8 BLAS_AXPY Procedure Parameters

Parameter	Description
n	Specifies the number of elements of the vectors x and y. n must be at least zero.
alpha	Specifies the scalar alpha.

Table 297-8 (Cont.) BLAS_AXPY Procedure Parameters

Parameter	Description
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (n - 1)*abs(incx))
incx	Specifies the increment for the elements of x. incx must not be zero.
y	UTL_NLA_ARRAY_FLT/DBL of DIMENSION at least (1 + (n - 1)*abs(incy))
incy	Specifies the increment for the elements of y. incy must not be zero.

BLAS_COPY Procedures

This procedure copies the contents of vector X to vector Y.

> **See Also:**
>
> BLAS Level 1 (Vector-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_COPY (
   n     IN     POSITIVEN,
   x     IN     UTL_NLA_ARRAY_DBL,
   incx  IN     POSITIVEN,
   y     IN OUT UTL_NLA_ARRAY_DBL,
   incy  IN     POSITIVEN);

UTL_NLA.BLAS_COPY (
   n     IN     POSITIVEN,
   x     IN     UTL_NLA_ARRAY_FLT,
   incx  IN     POSITIVEN,
   y     IN OUT UTL_NLA_ARRAY_FLT,
   incy  IN     POSITIVEN);
```

Parameters

Table 297-9 BLAS_COPY Procedure Parameters

Parameter	Description
n	Specifies the number of elements of the vectors x and y. n must be at least zero.
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (n - 1)*abs(incx))
incx	Specifies the increment for the elements of x. incx must not be zero.

Table 297-9 (Cont.) BLAS_COPY Procedure Parameters

Parameter	Description
y	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (n - 1)*abs(incy))
incy	Specifies the increment for the elements of y. incy must not be zero.

BLAS_DOT Functions

This function returns the dot (scalar) product of two vectors X and Y.

> **See Also:**
>
> BLAS Level 1 (Vector-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_DOT (
   n     IN   POSITIVEN,
   x     IN   UTL_NLA_ARRAY_DBL,
   incx  IN   POSITIVEN,
   y     IN   UTL_NLA_ARRAY_DBL,
   incy  IN   POSITIVEN)
  RETURN BINARY_DOUBLE;

UTL_NLA.BLAS_DOT (
   n     IN   POSITIVEN,
   x     IN   UTL_NLA_ARRAY_FLT,
   incx  IN   POSITIVEN,
   y     IN   UTL_NLA_ARRAY_FLT,
   incy  IN   POSITIVEN)
  RETURN BINARY_FLOAT;
```

Parameters

Table 297-10 BLAS_DOT Function Parameters

Parameter	Description
n	Specifies the number of elements of the vectors x and y. n must be at least zero.
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (n - 1)*abs(incx))
incx	Specifies the increment for the elements of x. incx must not be zero.
y	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (n - 1)*abs(incy))

Table 297-10 (Cont.) BLAS_DOT Function Parameters

Parameter	Description
incy	Specifies the increment for the elements of y. incy must not be zero.

BLAS_GBMV Procedures

This procedure performs one of the matrix-vector operations `y := alpha*A*x + beta*y` or `y := alpha*A'*x + beta*y`, where `alpha` and `beta` are scalars, `x` and `y` are vectors and `A` is an `m` by `n` band matrix, with `kl` sub-diagonals and `ku` super-diagonals.

> **See Also:**
>
> BLAS Level 2 (Matrix-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_GBMV (
   trans  IN      flag,
   m      IN      POSITIVEN,    n      IN      POSITIVEN,
   kl     IN      NATURALN,
   ku     IN      NATURALN,
   alpha  IN      SCALAR_DOUBLE,
   a      IN      UTL_NLA_ARRAY_DBL,
   lda    IN      POSITIVEN,
   x      IN      UTL_NLA_ARRAY_DBL,
   incx   IN      POSITIVEN,
   beta   IN      SCALAR_DOUBLE,
   y      IN OUT  UTL_NLA_ARRAY_DBL,
   incy   IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_GBMV (
   trans  IN      flag,
   m      IN      POSITIVEN,
   n      IN      POSITIVEN,
   kl     IN      NATURALN,
   ku     IN      NATURALN,
   alpha  IN      SCALAR_FLOAT,
   a      IN      UTL_NLA_ARRAY_FLT,
   lda    IN      POSITIVEN,
   x      IN      UTL_NLA_ARRAY_FLT,
   incx   IN      POSITIVEN,
   beta   IN      SCALAR_FLOAT,
   y      IN OUT  UTL_NLA_ARRAY_FLT,
   incy   IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-11 BLAS_GBMV Procedure Parameters

Parameter	Description
trans	Specifies the operation to be performed: • trans = 'N' or 'n'y := alpha*A*x + beta*y • trans = 'T' or 't'y := alpha*A'*x + beta*y • trans = 'C' or 'c'y := alpha*A'*x + beta*y
m	Specifies the number of rows of the matrix A. m must be at least zero.
n	Specifies the number of columns of the matrix A. n must be at least zero.
kl	Specifies the number of sub-diagonals of the matrix A. kl must satisfy 0. le. kl.
ku	Specifies the number of super-diagonals of the matrix A. ku must satisfy 0 .le. ku.
alpha	SCALAR_FLOAT/DOUBLE. Specifies the scalar alpha.
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda,n). Before entry, the leading (kl + ku + 1) by n part of the array A must contain the matrix of coefficients, supplied column by column, with the leading diagonal of the matrix in row (ku+1) of the array, the first super-diagonal starting at position 2 in row ku, the first sub-diagonal starting at position 1 in row (ku+2), and so on. Elements in the array A that do not correspond to elements in the band matrix (such as the top left ku by ku triangle) are not referenced.
lda	Specifies the first dimension of a as declared in the calling (sub) program. lda must be at least (kl+ku+1).
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (n - 1)*abs(incx)) when trans = ''N' or 'n' and at least (1 + (m - 1)*abs(incx)) otherwise. Before entry, the incremented array X must contain the vector x.
incx	Specifies the increment for the elements of x. Must not be zero.
beta	SCALAR_FLOAT/DOUBLE. Specifies the scalar beta. When beta is supplied as zero then y need not be set on input.
y	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (m - 1)*abs(incy)) when trans = 'N' or 'n' and at least (1+(n-1)*abs(incy)) otherwise. Before entry with beta nonzero, the incremented array Y must contain the vector y. On exit, Y is overwritten by the updated vector y.

Table 297-11 (Cont.) BLAS_GBMV Procedure Parameters

Parameter	Description
incy	Specifies the increment for the elements of y. Must not be zero.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_GEMM Procedures

This procedure performs one of the matrix-matrix operations.

```
C := alpha*op( A )*op( B ) + beta*C
```

where op(X) is one of

```
op(X) = X
```

or

```
op(X) = X'
```

where alpha and beta are scalars, and A, B and C are matrices, with op(A) an m by k matrix, op(B) a k by n matrix and C an m by n matrix.

> **See Also:**
>
> BLAS Level 3 (Matrix-Matrix Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_GEMM (
   transa IN      flag,
   transb IN      flag,
   m      IN      POSITIVEN,
   n      IN      POSITIVEN,
   k      IN      POSITIVEN,
   alpha  IN      SCALAR_DOUBLE,
   a      IN      UTL_NLA_ARRAY_DBL,
   lda    IN      POSITIVEN,
   b      IN      UTL_NLA_ARRAY_DBL,
   ldb    IN      POSITIVEN,
   beta   IN      SCALAR_DOUBLE,
   c      IN OUT  UTL_NLA_ARRAY_DBL,
   ldc    IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_GEMM (
   transa IN      flag,
   transb IN      flag,
   m      IN      POSITIVEN,
   n      IN      POSITIVEN,
   k      IN      POSITIVEN,
```

```
  alpha  IN      SCALAR_FLOAT,
  a      IN      UTL_NLA_ARRAY_FLT,
  lda    IN      POSITIVEN,
  b      IN      UTL_NLA_ARRAY_FLT,
  ldb    IN      POSITIVEN,
  beta   IN      SCALAR_FLOAT,
  c      IN OUT  UTL_NLA_ARRAY_FLT,
  ldc    IN      POSITIVEN,
  pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-12 BLAS_GEMM Procedure Parameters

Parameter	Description
transa	Specifies the form of op(A) to be used in the matrix multiplication as follows: • transa = 'N' or 'n' : op(A) = 'A' • transa = 'T' or 't' : op(A) = 'A' • transa = 'C' or 'c' : op(A) = 'A'
transb	Specifies the form of op(B) to be used in the matrix multiplication as follows: • transb = 'N' or 'n' : op(B) = B • transb ='T' or 't' : op(B) = B' • transb = 'C' or 'c': op(B) = B'
m	Specifies the number of rows of the matrix op(A) and of the matrix C. m must be at least zero.
n	Specifies the number of columns of the matrix op(B) and of the matrix C. n must be at least zero.
k	Specifies the rows of the matrix op(A) and the number of columns of the matrix op(B). k must be at least zero.
alpha	SCALAR_FLOAT/DOUBLE. Specifies the scalar alpha.
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda, ka) where ka is k when transa = 'N' or 'n', and is m otherwise. Before entry with transa = 'N' or 'n', the leading m by k part of the array A must contain the matrix A, otherwise the leading k by m part of the array A must contain the matrix A.
lda	Specifies the first dimension of a as declared in the calling (sub) program. When transa = 'N' or 'n', lda must be at least max (1,k).
b	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda, kb) where kb is n when transb = ''N' or 'n', and is k otherwise. Before entry with transb = 'N' or 'n', the leading k by n part of the array b must contain the matrix B, otherwise the leading n by k part of the arrayb must contain the matrix B.
ldb	Specifies the first dimension of b as declared in the calling (sub) program. When transb = 'N' or 'n', ldb must be at least max (1, n).
beta	SCALAR_FLOAT/DOUBLE. Specifies the scalar beta. When beta is supplied as zero then c need not be set on input.

Table 297-12 (Cont.) BLAS_GEMM Procedure Parameters

Parameter	Description
c	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (ldc, n). Before entry, the leading m by n part of the array C must contain the matrix C, except when beta is zero, in which case C need not be set on entry. On exit, the arrayC is overwritten by the m by n matrix (alpha*op(A)*op(B) + beta*C).
ldc	Specifies the first dimension of C as declared in the calling (sub) program. ldc must be at least max(1, m).
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_GEMV Procedures

This procedure performs one of the matrix-vector operations: `y := alpha*A*x + beta*y` or `y := alpha*A'*x + beta*y` where `alpha` and `beta` are scalars, `x` and `y` are vectors and `A` is an `m` by `n` matrix.

> **See Also:**
>
> BLAS Level 2 (Matrix-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_GEMV (
   trans  IN      flag,
   m      IN      POSITIVEN,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_DOUBLE,
   a      IN      UTL_NLA_ARRAY_DBL,
   lda    IN      POSITIVEN,
   x      IN      UTL_NLA_ARRAY_DBL,
   incx   IN      POSITIVEN,
   beta   IN      SCALAR_DOUBLE,
   y      IN OUT  UTL_NLA_ARRAY_DBL,
   incy   IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_GEMV (
   trans  IN      flag,
   m      IN      POSITIVEN,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_FLOAT,
   a      IN      UTL_NLA_ARRAY_FLT,
   lda    IN      POSITIVEN,
   x      IN      UTL_NLA_ARRAY_FLT,
   incx   IN      POSITIVEN,
   beta   IN      SCALAR_FLOAT,
   y      IN OUT  UTL_NLA_ARRAY_FLT,
```

```
  incy   IN      POSITIVEN,
  pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-13 BLAS_GEMV Procedure Parameters

Parameter	Description
trans	Specifies the operation to be performed: • trans = 'N' or 'n',y := alpha*A*x + beta*y • trans = 'T' or 't'y := alpha*A'*x + beta*y • trans = 'C' or 'c'y := alpha*A'*x + beta*y
m	Specifies the number of rows of the matrix A. m must be at least zero.
n	Specifies the number of columns of the matrix A. n must be at least zero.
alpha	SCALAR_FLOAT/DOUBLE. Specifies the scalar alpha.
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda, n). Before entry, the leading m by n part of the array a must contain the matrix of coefficients.
lda	Specifies the first dimension of a as declared in the calling (sub) program. lda must be at least max(1, m).
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (n - 1)*abs(incx)) when trans = ''N' or 'n' and at least (1+(m-1)*abs(incx)) otherwise. Before entry, the incremented array X must contain the vector x.
incx	Specifies the increment for the elements of x. Must not be zero.
beta	SCALAR_FLOAT/DOUBLE. Specifies the scalar beta. When beta is supplied as zero then y need not be set on input.
y	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (m - 1)*abs(incy)) when trans = 'N' or 'n' and at least (1 + (n - 1)*abs(incy)) otherwise. Before entry with beta nonzero, the incremented array Y must contain the vector y. On exit, Y is overwritten by the updated vector y.
incy	Specifies the increment for the elements of y. Must not be zero.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_GER Procedures

This procedure performs the rank 1 operation: `A := alpha*x*y' + A` where `alpha` is a scalar, `x` is an `m` element vector, `y` is an n element vector and `A` is an `m` by `n` matrix.

> **See Also:**
>
> BLAS Level 2 (Matrix-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_GER (
   m      IN      POSITIVEN,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_DBL,
   x      IN OUT  UTL_NLA_ARRAY_DBL,
   incx   IN      POSITIVEN,
   y      IN      UTL_NLA_ARRAY_DBL,
   incy   IN      POSITIVEN,
   a      IN OUT  UTL_NLA_ARRAY_DBL,
   lda    IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_GER (
   m      IN      POSITIVEN,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_FLT,
   x      IN OUT  UTL_NLA_ARRAY_FLT,
   incx   IN      POSITIVEN,
   y      IN      UTL_NLA_ARRAY_FLT,
   incy   IN      POSITIVEN,
   a      IN OUT  UTL_NLA_ARRAY_FLT,
   lda    IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-14 BLAS_GER Procedure Parameters

Parameter	Description
`m`	Specifies the number of rows of the matrix `A`. `m` must be at least zero.
`n`	Specifies the number of columns of the matrix `A`. `n` must be at least zero.
`alpha`	Specifies the scalar alpha.
`x`	`UTL_NLA_ARRAY_FLT/DBL` of dimension at least `( 1 + ( m - 1 )*abs( incx ) )` Before entry, the incremented array `X` must contain the m element vector `x`.
`incx`	Specifies the increment for the elements of `x`. incx must not be zero.

Table 297-14 (Cont.) BLAS_GER Procedure Parameters

Parameter	Description
y	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (n - 1)*abs(incy)) Before entry, the incremented array Y must contain the m element vector y.
incy	Specifies the increment for the elements of y. incx must not be zero.
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda, n). Before entry, the leading m by n part of the array a must contain the matrix of coefficients. On exit, a is overwritten by the updated matrix.
lda	Specifies the first dimension of a as declared in the calling (sub) program. lda must be at least max(1, m)
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_IAMAX Functions

This function computes the index of first element of a vector that has the largest absolute value.

> **See Also:**
>
> BLAS Level 1 (Vector-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_IAMAX (
   n     IN   POSITIVEN,
   x     IN   UTL_NLA_ARRAY_DBL,
   incx  IN   POSITIVEN,
  RETURN POSITIVEN;

UTL_NLA.BLAS_IAMAX (
   n     IN   POSITIVEN,
   x     IN   UTL_NLA_ARRAY_FLT,
   incx  IN   POSITIVEN,
  RETURN POSITIVEN;
```

Parameters

Table 297-15 BLAS_IAMAX Function Parameters

Parameter	Description
n	Specifies the number of elements of the vectors x and y. n must be at least zero.
x	UTL_NLA_ARRAY_FLT/DBL of DIMENSION at least (1 + (n - 1)*abs(incx))
incx	Specifies the increment for the elements of x. incx must not be zero.

BLAS_NRM2 Functions

This function computes the vector 2-norm (Euclidean norm).

> **See Also:**
>
> BLAS Level 1 (Vector-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_NRM2 (
   n     IN   POSITIVEN,
   x     IN   UTL_NLA_ARRAY_DBL,
   incx  IN   POSITIVEN)
  RETURN BINARY_DOUBLE;

UTL_NLA.BLAS_NRM2 (
   n     IN   POSITIVEN,
   x     IN   UTL_NLA_ARRAY_FLT,
   incx  IN   POSITIVEN)
  RETURN BINARY_FLOAT;
```

Parameters

Table 297-16 BLAS_NRM2 Function Parameters

Parameter	Description
n	Specifies the number of elements of the vectors x and y. n must be at least zero.
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (n - 1)*abs(incx))
incx	Specifies the increment for the elements of x. incx must not be zero.

BLAS_ROT Procedures

This procedure returns the plane rotation of points.

> **See Also:**
>
> BLAS Level 1 (Vector-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_ROT (
   n      IN      POSITIVEN,
   x      IN OUT  UTL_NLA_ARRAY_DBL,
   incx   IN      POSITIVEN,
   y      IN OUT  UTL_NLA_ARRAY_DBL,
   incy   IN      POSITIVEN,
   c      IN      SCALAR_DOUBLE,
   s      IN      SCALAR_DOUBLE);

UTL_NLA.BLAS_ROT (
   n      IN      POSITIVEN,
   x      IN OUT  UTL_NLA_ARRAY_FLT,
   incx   IN      POSITIVEN,
   y      IN OUT  UTL_NLA_ARRAY_FLT,
   incy   IN      POSITIVEN,
   c      IN      SCALAR_DOUBLE,
   s      IN      SCALAR_DOUBLE);
```

Parameters

Table 297-17 BLAS_ROT Procedure Parameters

Parameter	Description
n	Specifies the number of elements of the vectors x and y. n must be at least zero.
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1+(n-1)* abs(incx))
incx	Specifies the increment for the elements of x. incx must not be zero.
y	UTL_NLA_ARRAY_FLT/DBL of DIMENSION at least (1+(n-1)*abs(incy))
incy	Specifies the increment for the elements of y. incy must not be zero.
c	SCALAR_FLOAT/DOUBLE.Specifies the scalar C.
s	SCALAR_FLOAT/DOUBLE.Specifies the scalar S.

BLAS_ROTG Procedures

This procedure returns the Givens rotation of points.

> **See Also:**
>
> BLAS Level 1 (Vector-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_ROTG (
   a    IN OUT   SCALAR_DOUBLE,
   b    IN OUT   SCALAR_DOUBLE,
   c    IN OUT   SCALAR_DOUBLE,
   s    IN OUT   SCALAR_DOUBLE);

UTL_NLA.BLAS_ROTG (
   a    IN OUT   SCALAR_FLOAT,
   b    IN OUT   SCALAR_FLOAT,
   c    IN OUT   SCALAR_FLOAT,
   s    IN OUT   SCALAR_FLOAT);
```

Parameters

Table 297-18 BLAS_ROTG Procedure Parameters

Parameter	Description
a	SCALAR_FLOAT/DOUBLE. Specifies the scalar A.
b	SCALAR_FLOAT/DOUBLE. Specifies the scalar B.
c	SCALAR_FLOAT/DOUBLE. Specifies the scalar C.
s	SCALAR_FLOAT/DOUBLE. Specifies the scalar S.

BLAS_SCAL Procedures

This procedure scales a vector by a constant.

> **See Also:**
>
> BLAS Level 1 (Vector-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_SCAL (
   n      IN  POSITIVEN,
   alpha  IN  SCALAR_DOUBLE,
   x      IN  OUT UTL_NLA_ARRAY_DBL,
   incx   IN  POSITIVEN);
```

```
UTL_NLA.BLAS_SCAL (
   n      IN  POSITIVEN,
   alpha  IN  SCALAR_FLOAT,
   x      IN  OUT UTL_NLA_ARRAY_FLT,
   incx   IN  POSITIVEN);
```

Parameters

Table 297-19 BLAS_SCAL Procedure Parameters

Parameter	Description
n	Specifies the number of elements of the vectors x and y. n must be at least zero.
alpha	Specifies the scalar alpha.
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1+(n-1)*abs(incx))
incx	Specifies the increment for the elements of x. incx must not be zero.

BLAS_SPMV Procedures

This procedure performs the matrix-vector operation `y := alpha*A*x + beta*y`, where `alpha` and `beta` are scalars, `x` and `y` are n element vectors and `A` is an `n` by `n` symmetric matrix, supplied in packed form.

> **See Also:**
>
> BLAS Level 2 (Matrix-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_SPMV (
   uplo   IN      flag,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_DOUBLE,
   ap     IN      UTL_NLA_ARRAY_DBL,
   x      IN      UTL_NLA_ARRAY_DBL,
   incx   IN      POSITIVEN,
   beta   IN      SCALAR_DOUBLE,
   y      IN OUT  UTL_NLA_ARRAY_DBL,
   incy   IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_SPMV (
   uplo   IN      flag,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_FLOAT,
   ap      IN     UTL_NLA_ARRAY_FLT,
   x      IN      UTL_NLA_ARRAY_FLT,
   incx   IN      POSITIVEN,
   beta   IN      SCALAR_FLOAT,
   y      IN OUT  UTL_NLA_ARRAY_FLT,
```

```
  incy   IN      POSITIVEN,
  pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-20 BLAS_SPMV Procedure Parameters

Parameter	Description
uplo	Specifies the upper or lower triangular part of the matrix A is supplied in the packed array AP: • uplo = 'U' or 'u'. The upper triangular part of A is supplied in AP. • uplo = 'L' or 'l'. The lower triangular part of A is supplied in AP.
n	Specifies the order of the matrix A. n must be at least zero.
alpha	SCALAR_FLOAT/DOUBLE. Specifies the scalar alpha.
ap	UTL_NLA_ARRAY_FLT/DBL of dimension at least ((n*(n+1))/2) Before entry with uplo = 'U' or 'u', the array ap must contain the upper triangular part of the symmetric matrix packed sequentially, column by column, so that ap(1) contains a(1,1), ap(2) and ap(3) contain a(1,2) and a(2,2) respectively, and so on. Before entry with uplo = 'L' or 'l', the array ap must contain the lower triangular part of the symmetric matrix packed sequentially, column by column, so that ap(1) contains, ap(2) and ap(3) contain a(2,1) and a(3,1) respectively, and so on.
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1+(n-1)*abs(incx)) Before entry, the incremented array X must contain the n element vector x.
incx	Specifies the increment for the elements of x. Must not be zero.
beta	SCALAR_FLOAT/DOUBLE. Specifies the scalar beta. When beta is supplied as zero then Y need not be set on input.
y	UTL_NLA_ARRAY_FLT/DBL of dimension at leasT (1+(n-1)*abs(incy)) Before entry, the incremented array Y must contain the n element vector y. On exit, Y is overwritten by the updated vector y.
incy	Specifies the increment for the elements of y. Must not be zero.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_SPR Procedures

This procedure performs the rank 1 operation `A := alpha*x*x' + A`, where `alpha` is a real scalar, `x` is an `n` element vector, and `A` is an `n` by `n` symmetric matrix, supplied in packed form.

> **See Also:**
>
> BLAS Level 2 (Matrix-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_SPR (
   uplo   IN      flag,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_DBL,
   x      IN OUT  UTL_NLA_ARRAY_DBL,
   incx   IN      POSITIVEN,
   ap     IN OUT  UTL_NLA_ARRAY_DBL,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_SPR (
   uplo   IN      flag,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_FLT,
   x      IN OUT  UTL_NLA_ARRAY_FLT,
   incx   IN      POSITIVEN,
   ap     IN OUT  UTL_NLA_ARRAY_FLT,
   pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-21 BLAS_SPR Procedure Parameters

Parameter	Description
`uplo`	Specifies whether the upper or lower triangular part of the matrix `A` is supplied in the packed array `ap`: • `uplo` = `'U'` or `'u'`: The upper triangular part of `A` is supplied in `ap`. • `uplo` = `'L'` or `'l'` : The lower triangular part of `A` is supplied in `ap`.
`n`	Specifies the order of the matrix `A`. `n` must be at least zero.
`alpha`	Specifies the scalar alpha.
`x`	`UTL_NLA_ARRAY_FLT/DBL` of dimension at least `(1+(n-1)*abs(incx))` Before entry, the incremented array `X` must contain the m element vector `x`.
`incx`	Specifies the increment for the elements of `x`. `incx` must not be zero.

Table 297-21 (Cont.) BLAS_SPR Procedure Parameters

Parameter	Description
ap	UTL_NLA_ARRAY_FLT/DBL of dimension at least ((n*(n +1))/2) Before entry with uplo = 'U' or 'u', the array ap must contain the upper triangular part of the symmetric matrix packed sequentially, column by column, so that ap(1) contains a(1,1), ap(2) and ap(3) contain a(1,2) and a(2,2) respectively, and so on. On exit, the array ap is overwritten by the upper triangular part of the updated matrix. Before entry with uplo = 'L' or 'l', the array ap must contain the lower triangular part of the symmetric matrix packed sequentially, column by column, so that ap(1) contains a(1,1), ap(2) and ap(3) contain a(2,1) and a(3,1) respectively, and so on. On exit, the array ap is overwritten by the lower triangular part of the updated matrix
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_SPR2 Procedures

This procedure performs the rank 2 operation A := alpha*x*y' + alpha*y*x' +A, where alpha is a scalar, x and y are n element vectors, and A is an n by n symmetric matrix, supplied in packed form.

> **See Also:**
>
> BLAS Level 2 (Matrix-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_SPR2 (
   uplo   IN      flag,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_DBL,
   x      IN      UTL_NLA_ARRAY_DBL,
   incx   IN      POSITIVEN,
   y      IN      UTL_NLA_ARRAY_DBL,
   incy   IN      POSITIVEN,
   a      IN OUT  UTL_NLA_ARRAY_DBL,
   lda    IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_SPR2 (
   uplo   IN      flag,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_FLT,
   x      IN      UTL_NLA_ARRAY_FLT,
   incx   IN      POSITIVEN,
   y      IN      UTL_NLA_ARRAY_FLT,
```

```
  incy   IN      POSITIVEN,
  a      IN OUT  UTL_NLA_ARRAY_FLT,
  lda    IN      POSITIVEN,
  pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-22 BLAS_SPR2 Procedure Parameters

Parameter	Description
uplo	Specifies whether the upper or lower triangular part of the matrix A is supplied in the packed array ap : • uplo = 'U' or 'u' : The upper triangular part of A is supplied in ap. • uplo = 'L' or 'l' : The lower triangular part of A is supplied in ap.
n	Specifies the order of the matrix A. n must be at least zero.
alpha	Specifies the scalar alpha.
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1+(n-1)*abs(incx)) Before entry, the incremented array X must contain the m element vector x.
incx	Specifies the increment for the elements of x. incx must not be zero.
y	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1+(n-1)*abs(incy)) Before entry, the incremented array X must contain the m element vector y.
incy	Specifies the increment for the elements of y. incy must not be zero.
ap	UTL_NLA_ARRAY_FLT/DBL of dimension at least ((n*(n+1))/2) Before entry with uplo = 'U' or 'u', the array ap must contain the upper triangular part of the symmetric matrix packed sequentially, column by column, so that ap(1) contains ap(1) contains a(1,1), ap(2) and ap(3) contain a(1,2) and a(2,2) respectively, and so on. On exit, the array ap is overwritten by the upper triangular part of the updated matrix. Before entry with uplo = 'L' or 'l', the array ap must contain the lower triangular part of the symmetric matrix packed sequentially, column by column, so that ap(1) contains a(1,1), ap(2) and ap(3) contain a(2,1) and a(3,1) respectively, and so on. On exit, the array ap is overwritten by the lower triangular part of the updated matrix
lda	Specifies the first dimension of a as declared in the calling (sub) program. lda must be at least (k + 1).
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_SBMV Procedures

This procedure performs the matrix-vector operation `y := alpha*A*x + beta*y`, where `alpha` and `beta` are scalars, `x` and `y` are n element vectors and `A` is an `n` by `n` symmetric band matrix, with `k` super-diagonals.

> **See Also:**
>
> BLAS Level 2 (Matrix-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_SBMV (
   uplo    IN      flag,
   n       IN      POSITIVEN,
   k       IN      NATURALN,
   alpha   IN      SCALAR_DOUBLE,
   a       IN      UTL_NLA_ARRAY_DBL,
   lda     IN      POSITIVEN,
   x       IN      UTL_NLA_ARRAY_DBL,
   incx    IN      POSITIVEN,
   beta    IN      SCALAR_DOUBLE,
   y       IN OUT  UTL_NLA_ARRAY_DBL,
   incy    IN      POSITIVEN,
   pack    IN      flag DEFAULT 'C');

UTL_NLA.BLAS_SBMV (
   uplo    IN      flag,
   n       IN      POSITIVEN,
   k       IN      NATURALN,
   alpha   IN      SCALAR_FLOAT,
   a       IN      UTL_NLA_ARRAY_FLT,
   lda     IN      POSITIVEN,
   x       IN      UTL_NLA_ARRAY_FLT,
   incx    IN      POSITIVEN,
   beta    IN      SCALAR_FLOAT,
   y       IN OUT  UTL_NLA_ARRAY_FLT,
   incy    IN      POSITIVEN,
   pack    IN      flag DEFAULT 'C');
```

Parameters

Table 297-23 BLAS_SBMV Procedure Parameters

Parameter	Description
`uplo`	Specifies whether the upper or lower triangular part of the band matrix `A` is being supplied: • `uplo` = `'U'` or `'u'`. The upper triangular part of `A` is supplied. • `uplo` = `'L'` or `'l'`. The lower triangular part of `A` is supplied.
`n`	Specifies the order of the matrix `A`. `n` must be at least zero.
`k`	Specifies the number of super-diagonals of the matrix `A`. `k` must satisfy `0 .le. k`.

Table 297-23 (Cont.) BLAS_SBMV Procedure Parameters

Parameter	Description
alpha	SCALAR_FLOAT/DOUBLE. Specifies the scalar alpha.
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda,n). Before entry with uplo = 'U' or 'u', the leading (k+1) by n part of the array A must contain the upper triangular band part of the symmetric matrix, supplied column by column, with the leading diagonal of the matrix in row (k+1) of the array, the first super-diagonal starting at position 2 in rowk, and so on. The top left k by k triangle of the array A is not referenced. Before entry with uplo = 'L' or 'l', the leading (k+1) by n part of the array A must contain the lower triangular band part of the symmetric matrix, supplied column by column, with the leading diagonal of the matrix in row 1 of the array, the first sub-diagonal starting at position 1 in row 2, and so on. The bottom right k by k triangle of the array A is not referenced. Unchanged on exit
lda	Specifies the first dimension of a as declared in the calling (sub) program. lda must be at least (k + 1).
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1+(n-1)*abs(incx)) Before entry, the incremented array X must contain the n element vector x.
incx	Specifies the increment for the elements of x. Must not be zero.
beta	SCALAR_FLOAT/DOUBLE. Specifies the scalar beta.
y	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1+(n-1)*abs(incy)) Before entry, the incremented array Y must contain the n element vector y. On exit, Y is overwritten by the updated vector y.
incy	Specifies the increment for the elements of y. Must not be zero.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_SWAP Procedures

This procedure swaps the contents of two vectors each of size n.

Syntax

```
UTL_NLA.BLAS_SWAP (
   n     IN     POSITIVEN,
   x     IN OUT UTL_NLA_ARRAY_DBL,
   incx  IN     POSITIVEN,
   y     IN OUT UTL_NLA_ARRAY_DBL,
   incy  IN     POSITIVEN);
```

```
UTL_NLA.BLAS_SWAP (
   n     IN     POSITIVEN,
   x     IN OUT UTL_NLA_ARRAY_FLT,
   incx  IN     POSITIVEN,
   y     IN OUT UTL_NLA_ARRAY_FLT,
   incy  IN     POSITIVEN);
```

Parameters

Table 297-24 BLAS_SWAP Procedure Parameters

Parameter	Description
n	Specifies the number of elements of the vectors x and y. n must be at least zero.
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1+(n-1)*abs(incx))
incx	Specifies the increment for the elements of x. incx must not be zero.
y	UTL_NLA_ARRAY_FLT/DBL of DIMENSION at least (1+(n-1)*abs(incy))
incy	Specifies the increment for the elements of y. incy must not be zero.

BLAS_SYMM Procedures

This procedure performs one of the matrix-matrix operations `C := alpha*A*B + beta*C` or `C := alpha*B*A + beta*C`, where `alpha` and `beta` are scalars, `A` is a symmetric matrix, and `B` and `C` are `m` by `n` matrices.

> **See Also:**
>
> BLAS Level 3 (Matrix-Matrix Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_SYMM (
   side   IN     flag,
   uplo   IN     flag,
   m      IN     POSITIVEN,
   n      IN     POSITIVEN,
   alpha  IN     SCALAR_DOUBLE,
   a      IN     UTL_NLA_ARRAY_DBL,
   lda    IN     POSITIVEN,
   b      IN     UTL_NLA_ARRAY_DBL,
   ldb    IN     POSITIVEN,
   beta   IN     SCALAR_DOUBLE,
   c      IN OUT UTL_NLA_ARRAY_DBL,
   ldc    IN     POSITIVEN,
   pack   IN     flag DEFAULT 'C');
```

```
UTL_NLA.BLAS_SYMM (
   side   IN      flag,
   uplo   IN      flag,
   m      IN      POSITIVEN,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_FLOAT,
   a      IN      UTL_NLA_ARRAY_FLT,
   lda    IN      POSITIVEN,
   b      IN      UTL_NLA_ARRAY_FLT,
   ldb    IN      POSITIVEN,
   beta   IN      SCALAR_FLOAT,
   c      IN OUT  UTL_NLA_ARRAY_FLT,
   ldc    IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-25 BLAS_SYMM Procedure Parameters

Parameter	Description
side	Specifies whether the symmetric matrix A appears on the left or right in the operation: • side = 'L' or 'l' : C := alpha*A*B + beta*C • side = 'R' or 'r' : C := alpha*B*A + beta*C
uplo	Specifies whether the upper or lower triangular part of the array A is to be referenced: • uplo = 'U' or 'u' : Only the upper triangular part of the symmetric matrix is to be referenced. • uplo = 'L' or 'l' : Only the lower triangular part of the symmetric matrix is to be referenced.
m	Specifies the number of rows of the matrix C. m must be at least zero.
n	Specifies the number of columns of the matrix C. n must be at least zero.
alpha	SCALAR_FLOAT/DOUBLE. Specifies the scalar alpha.
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda,ka) where ka is m when side = 'L' or 'l', and is n otherwise. Before entry with side = 'L' or 'l', the leading m by m part of the array A must contain the symmetric matrix, such that when uplo = 'U' or 'u', the leading m by m upper triangular part of the array A must contain the upper triangular part of the symmetric matrix and the strictly lower triangular part of A is not referenced, and when uplo = 'L' or 'l', the leading m by m lower triangular part of the array A must contain the lower triangular part of the symmetric matrix and the strictly upper triangular part of A is not referenced. Before entry with side = 'R' or 'r', the n by n part of the array A must contain the symmetric matrix, such that when uplo = 'U' or 'u', the leading n by n upper triangular part of the array A must contain the upper triangular part of the symmetric matrix and the strictly lower triangular part of A is not referenced, and when uplo = 'L' or 'l', the leading n by n lower triangular part of the array A must contain the lower triangular part of the symmetric matrix and the strictly upper triangular part of A is not referenced.

Table 297-25 (Cont.) BLAS_SYMM Procedure Parameters

Parameter	Description
lda	Specifies the first dimension of a as declared in the calling (sub) program. When side = 'L' or 'l', lda must be at least max(1,m), otherwise lda must be at least max(1,n).
b	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (ldb,n). Before entry, the leading m by n part of the array B must contain the matrix B.
ldb	Specifies the first dimension of b as declared in the calling (sub) program. ldb must be at least max(1,m).
beta	SCALAR_FLOAT/DOUBLE. Specifies the scalar beta. When beta is supplied as zero then c need not be set on input.
c	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (ldc,n). Before entry, the leading m by n part of the array C must contain the matrix C, except when beta is zero, in which case C need not be set on entry. On exit, the array C is overwritten by the m by n updated matrix.
ldc	Specifies the first dimension of C as declared in the calling (sub) program. ldc must be at least max (1,m).
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_SYMV Procedures

This procedure performs the matrix-vector operation y := alpha*A*x + beta*y , where alpha and beta are scalars, x and y are n element vectors and A is an n by n symmetric matrix.

> **See Also:**
>
> BLAS Level 2 (Matrix-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_SYMV (
   uplo   IN      flag,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_DOUBLE,
   a      IN      UTL_NLA_ARRAY_DBL,
   lda    IN      POSITIVEN,
   x      IN      UTL_NLA_ARRAY_DBL,
   incx   IN      POSITIVEN,
   beta   IN      SCALAR_DOUBLE,
   y      IN OUT  UTL_NLA_ARRAY_DBL,
   incy   IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_SYMV (
   uplo   IN      flag,
```

```
  n      IN      POSITIVEN,
  alpha  IN      SCALAR_FLOAT,
  a      IN      UTL_NLA_ARRAY_FLT,
  lda    IN      POSITIVEN,
  x      IN      UTL_NLA_ARRAY_FLT,
  incx   IN      POSITIVEN,
  beta   IN      SCALAR_FLOAT,
  y      IN OUT  UTL_NLA_ARRAY_FLT,
  incy   IN      POSITIVEN,
  pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-26 BLAS_SYMV Procedure Parameters

Parameter	Description
uplo	Specifies whether the upper or lower triangular part of the array A is to be referenced: • uplo = 'U'or 'u'. Only the upper triangular part ofA is to be referenced. • uplo = 'L' or 'l'. Only the lower triangular part of A is to be referenced.
n	Specifies the order of the matrix A. n must be at least zero.
alpha	SCALAR_FLOAT/DOUBLE. Specifies the scalar alpha.
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda,n). Before entry with uplo = 'U' or 'u', the leading n by n upper triangular part of the array A must contain the upper triangular part of the symmetric matrix and the strictly lower triangular part of A is not referenced. Before entry with uplo = 'L' or 'l', the leading n by n lower triangular part of the array A must contain the lower triangular part of the symmetric matrix and the strictly upper triangular part of A is not referenced.
lda	Specifies the first dimension of a as declared in the calling (sub) program. lda must be at least max(1,n).
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1+(n-1)*abs(incx)) Before entry, the incremented array X must contain the n element vector x.
incx	Specifies the increment for the elements of x. Must not be zero.
beta	SCALAR_FLOAT/DOUBLE. Specifies the scalar beta. When beta is supplied as zero then y need not be set on input.
y	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1+(n-1)*abs(incy)) Before entry, the incremented array Y must contain the n element vector y. On exit, Y is overwritten by the updated vector y.
incy	Specifies the increment for the elements of y. Must not be zero.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_SYR Procedures

This procedure performs the rank 1 operation `A := alpha*x*x' + A` , where `alpha` is a real scalar, `x` is an `n` element vector, and `A` is an `n` by `n` symmetric matrix.

> **See Also:**
>
> BLAS Level 2 (Matrix-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_SYR (
   uplo   IN      flag,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_DBL,
   x      IN OUT  UTL_NLA_ARRAY_DBL,
   incx   IN      POSITIVEN,
   a      IN OUT  UTL_NLA_ARRAY_DBL,
   lda    IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_SYR (
   uplo   IN      flag,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_FLT,
   x      IN OUT  UTL_NLA_ARRAY_FLT,
   incx   IN      POSITIVEN,
   a      IN OUT  UTL_NLA_ARRAY_FLT,
   lda    IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-27 BLAS_SYR Procedure Parameters

Parameter	Description
`uplo`	Specifies whether the upper or lower triangular part of the array `A` is to be referenced: • `uplo` = `'U'` or `'u'` : Only the upper triangular part of `A` is to be referenced. • `uplo` = `'L'` or `'l'` : Only the lower triangular part of A is to be referenced.
`n`	Specifies the order of the matrix `A`. `n` must be at least zero.
`alpha`	Specifies the scalar alpha.
`x`	`UTL_NLA_ARRAY_FLT/DBL` of dimension at least `(1+(n-1)*abs(incx))` Before entry, the incremented array `X` must contain the m element vector `x`.
`incx`	Specifies the increment for the elements of `x`. `incx` must not be zero.

Table 297-27 (Cont.) BLAS_SYR Procedure Parameters

Parameter	Description
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda, n) Before entry with uplo = 'U' or 'u', the leading n by n upper triangular part of the array A must contain the upper triangular part of the symmetric matrix and the strictly lower triangular part of A is not referenced. On exit, the upper triangular part of the array A is overwritten by the upper triangular part of the updated matrix. Before entry with uplo = 'L' or 'l', the leading n by n lower triangular part of the array A must contain the lower triangular part of the symmetric matrix and the strictly upper triangular part of A is not referenced. On exit, the lower triangular part of the array A is overwritten by the lower triangular part of the updated matrix.
lda	Specifies the first dimension of a as declared in the calling (sub) program. lda must be at least max(1, n)
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_SYR2 Procedures

This procedure performs the rank 2 operation A := alpha*x*y' + alpha*y*x' + A , where alpha is a scalar, x and y are n element vectors, and A is an n by n symmetric matrix.

> **See Also:**
>
> BLAS Level 2 (Matrix-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_SYR2 (
   uplo   IN      flag,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_DBL,
   x      IN      UTL_NLA_ARRAY_DBL,
   incx   IN      POSITIVEN,
   y      IN      UTL_NLA_ARRAY_DBL,
   incy   IN      POSITIVEN,
   a      IN OUT  UTL_NLA_ARRAY_DBL,
   lda    IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_SYR2 (
   uplo   IN      flag,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_FLT,
   x      IN      UTL_NLA_ARRAY_FLT,
   incx   IN      POSITIVEN,
```

```
  y      IN      UTL_NLA_ARRAY_FLT,
  incy   IN      POSITIVEN,
  a      IN OUT  UTL_NLA_ARRAY_FLT,
  lda    IN      POSITIVEN,
  pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-28 BLAS_SYR2 Procedure Parameters

Parameter	Description
uplo	Specifies whether the upper or lower triangular part of the array A is to be referenced: • uplo = 'U' or 'u' : Only the upper triangular part of A is to be referenced. • uplo = 'L' or 'l' : Only the lower triangular part of A is to be referenced.
n	Specifies the order of the matrix A. n must be at least zero.
alpha	Specifies the scalar alpha.
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (n - 1)*abs(incx)) Before entry, the incremented array X must contain the m element vector x.
incx	Specifies the increment for the elements of x. incx must not be zero.
y	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (n - 1)*abs(incy)) Before entry, the incremented array Y must contain the m element vector y.
incy	Specifies the increment for the elements of y. incy must not be zero.
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda, n) With uplo = 'U' or 'u', the leading n by n upper triangular part of the array A must contain the upper triangular part of the symmetric matrix and the strictly lower triangular part of A is not referenced. On exit, the upper triangular part of the array A is overwritten by the upper triangular part of the updated matrix. With uplo = 'L' or 'l', the leading n by n lower triangular part of the array A must contain the lower triangular part of the symmetric matrix and the strictly upper triangular part of A is not referenced. On exit, the lower triangular part of the array A is overwritten by the lower triangular part of the updated matrix.
lda	Specifies the first dimension of a as declared in the calling (sub) program. lda must be at least max(1, n)
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_SYR2K Procedures

It performs one of the symmetric rank2 k operations `C := alpha*A*B' + alpha*B*A' + beta*C` or `C := alpha*A'*B + alpha*B'*A + beta*C`, where `alpha` and `beta` are scalars, `C` is an `n` by `n` symmetric matrix and `A` and `B` are `n` by `k` matrices in the first case and `k` by `n` matrices in the second case.

> **See Also:**
>
> BLAS Level 3 (Matrix-Matrix Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_SYR2K (
   uplo   IN      flag,
   trans  IN      flag,
   n      IN      POSITIVEN,
   k      IN      POSITIVEN,
   alpha  IN      SCALAR_DOUBLE,
   a      IN      UTL_NLA_ARRAY_DBL,
   lda    IN      POSITIVEN,
   b      IN      UTL_NLA_ARRAY_DBL,
   ldb    IN      POSITIVEN,
   beta   IN      SCALAR_DOUBLE,
   c      IN OUT  UTL_NLA_ARRAY_DBL,
   ldc    IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_SYR2K (
   uplo   IN      flag,
   trans  IN      flag,
   n      IN      POSITIVEN,
   k      IN      POSITIVEN,
   alpha  IN      SCALAR_FLOAT,
   a      IN      UTL_NLA_ARRAY_FLT,
   lda    IN      POSITIVEN,
   b      IN OUT  UTL_NLA_ARRAY_FLT,
   ldb    IN      POSITIVEN,
   beta   IN      SCALAR_FLOAT,
   c      IN OUT  UTL_NLA_ARRAY_FLT,
   ldc    IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-29 BLAS_SYR2K Procedure Parameters

Parameter	Description
uplo	Specifies whether the upper or lower triangular part of the array C is to be referenced: • uplo = 'U' or 'u' : Only the upper triangular part of C is to be referenced. • uplo = 'L' or 'l' : Only the lower triangular part of C is to be referenced.
trans	Specifies the operations to be performed: • trans = 'N' or 'n'C : C := alpha*A*B' + alpha*B*A' + beta*C • trans = 'T' or 't'C : C := alpha*A'*B + alpha*B'*A + beta*C • trans = 'C' or 'c'C : C := alpha*A'*B + alpha*B'*A + beta*C
n	Specifies the order of matrix C. n must be at least zero.
k	On entry with trans = 'N' or 'n', k specifies the number of columns of the matrices A and B. On entry with trans = 'T' or 't' or trans = 'C' or 'c', k specifies the number of rows of the matrices A and B. k must be at least zero.
alpha	SCALAR_FLOAT/DOUBLE. Specifies the scalar alpha.
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda,ka) where kb is k when trans = 'N' or 'n', and is n otherwise. Before entry with trans = 'N' or 'n', the leading n byk part of the array A must contain the matrix A, otherwise the leading k by n part of the array A must contain the matrix A.
lda	Specifies the first dimension of a as declared in the calling (sub) program. When trans = 'N' or 'n', lda must be at leastmax(1,n), otherwise lda must be at least max(1,k).
b	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda,kb) where kb is k when trans = 'N' or 'n', and is n otherwise. Before entry with trans = 'N' or 'n', the leading n byk part of the array B must contain the matrix B, otherwise the leading k by n part of the array B must contain the matrix B.
ldb	Specifies the first dimension of b as declared in the calling (sub) program. When trans = 'N' or 'n', ldb must be at least max (1,n), otherwise ldb must be at least max (1,k).
beta	SCALAR_FLOAT/DOUBLE. Specifies the scalar beta.

Table 297-29 (Cont.) BLAS_SYR2K Procedure Parameters

Parameter	Description
c	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (ldc,n). Before entry with uplo = 'U' or 'u', the leading n by n upper triangular part of the array C must contain the upper triangular part of the symmetric matrix and the strictly lower triangular part of C is not referenced. On exit, the upper triangular part of the array C is overwritten by the upper triangular part of the updated matrix. Before entry with uplo = 'L' or 'l', the leading n by n lower triangular part of the array C must contain the lower triangular part of the symmetric matrix and the strictly upper triangular part of C is not referenced. On exit, the lower triangular part of the array C is overwritten by the lower triangular part of the updated matrix.
ldc	Specifies the first dimension of C as declared in the calling (sub) program. ldc must be at least max(1,n).
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_SYRK Procedures

This procedure performs one of the symmetric rank k operations C := alpha*A*A' + beta*C or C := alpha*A'*A + beta*C, where alpha and beta are scalars, C is an n by n symmetric matrix andA is an n by k matrix in the first case and a k byn matrix in the second case.

> **See Also:**
>
> BLAS Level 3 (Matrix-Matrix Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_SYRK (
   uplo   IN      flag,
   trans  IN      flag,
   n      IN      POSITIVEN,
   k      IN      POSITIVEN,
   alpha  IN      SCALAR_DOUBLE,
   a      IN      UTL_NLA_ARRAY_DBL,
   lda    IN      POSITIVEN,
   beta   IN      SCALAR_DOUBLE,
   c      IN OUT  UTL_NLA_ARRAY_DBL,
   ldc    IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_SYRK (
   uplo   IN      flag,
   trans  IN      flag,
   n      IN      POSITIVEN,
   k      IN      POSITIVEN,
   alpha  IN      SCALAR_FLOAT,
```

```
  a      IN      UTL_NLA_ARRAY_FLT,
  lda    IN      POSITIVEN,
  beta   IN      SCALAR_FLOAT,
  c      IN OUT  UTL_NLA_ARRAY_DBL,
  ldc    IN      POSITIVEN,
  pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-30 BLAS_SYRK Procedure Parameters

Parameter	Description
uplo	Specifies whether the upper or lower triangular part of the array C is to be referenced: • uplo = 'U' or 'u' : Only the upper triangular part of C is to be referenced. • uplo = 'L' or 'l' : Only the lower triangular part of C is to be referenced.
trans	Specifies the operations to be performed: • trans = 'N' or 'n' : C := alpha*A*A' + beta*C • trans = 'T' or 't' : C := alpha*A'*A + beta*C • trans ='C' or 'c' : C := alpha*A'*A + beta*C
n	Specifies the order of matrix C. n must be at least zero.
k	On entry with trans = 'N' or 'n', k specifies the number of columns of the matrix A. On entry with trans = 'T' or 't' or trans = 'C' or 'c', k specifies the number of rows of the matrix A. k must be at least zero.
alpha	SCALAR_FLOAT/DOUBLE. Specifies the scalar alpha.
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda,ka) where ka is k when trans = 'N' or 'n', and is n otherwise. Before entry with trans = 'N' or 'n', the leading n byk part of the array A must contain the matrix A, otherwise the leading k by n part of the array A must contain the matrix A.
lda	Specifies the first dimension of a as declared in the calling (sub) program. When trans = 'N' or 'n', lda must be at least max(1,n), otherwiselda must be at least max(1,k).
beta	SCALAR_FLOAT/DOUBLE. Specifies the scalar beta.
c	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (ldc,n). Before entry with uplo = 'U' or 'u', the leading n by n upper triangular part of the array C must contain the upper triangular part of the symmetric matrix and the strictly lower triangular part of C is not referenced. On exit, the upper triangular part of the array C is overwritten by the upper triangular part of the updated matrix. Before entry with uplo = 'L' or 'l', the leading n by n lower triangular part of the array C must contain the lower triangular part of the symmetric matrix and the strictly upper triangular part of C is not referenced. On exit, the lower triangular part of the array C is overwritten by the lower triangular part of the updated matrix.
ldc	Specifies the first dimension of C as declared in the calling (sub) program. ldc must be at least max(1,n).

Table 297-30 (Cont.) BLAS_SYRK Procedure Parameters

Parameter	Description
`pack`	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_TBMV Procedures

This procedure performs the matrix-vector operations `x := A*x` or `x := A'*x`, where `x` is an `n` element vector and `A` is an `n` by `n` unit, or non-unit, upper or lower triangular band matrix, with `(k+1)` diagonals.

> **See Also:**
>
> BLAS Level 2 (Matrix-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_TBMV (
   uplo   IN      flag,
   trans  IN      flag,
   diag   IN      flag,
   n      IN      POSITIVEN,
   k      IN      NATURALN,
   a      IN      UTL_NLA_ARRAY_DBL,
   lda    IN      POSITIVEN,
   x      IN OUT  UTL_NLA_ARRAY_DBL,
   incx   IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_TBMV (
   uplo   IN      flag,
   trans  IN      flag,
   diag   IN      flag,
   n      IN      POSITIVEN,
   k      IN      NATURALN,
   a      IN      UTL_NLA_ARRAY_FLT,
   lda    IN      POSITIVEN,
   x      IN OUT  UTL_NLA_ARRAY_FLT,
   incx   IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-31 BLAS_TBMV Procedure Parameters

Parameter	Description
uplo	Specifies whether the matrix is an upper or lower triangular matrix: • uplo = 'U' or 'u'. A is an upper triangular matrix. • uplo = 'L' or 'l'. A is a lower triangular matrix.
trans	Specifies the operation to be performed: • trans = 'N' or 'n'x := A*x • trans = 'T' or 't'x := A'*x • trans = 'C' or 'c'x := A'*x
diag	Specifies whether or not A is unit triangular: • diag = 'U' or 'u'. A is assumed to be unit triangular. • diag = 'N' or 'n'. A is not assumed to be unit triangular.
n	Specifies the order of the matrix A. n must be at least zero.
k	Specifies whether or not A is unit triangular: • with uplo = 'U' or 'u', K specifies the number of super-diagonals of the matrix A. • with uplo = 'L' or 'l', K specifies the number of sub-diagonals of the matrix A. K must satisfy 0 .le. k.
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda, n). Before entry with uplo = 'U' or 'u', the leading (k+1) by n part of the array A must contain the upper triangular band part of the matrix of coefficients, supplied column by column, with the leading diagonal of the matrix in row (k+1) of the array, the first super-diagonal starting at position 2 in row k, and so on. The top left k by k triangle of the array A is not referenced. Before entry with uplo = 'L' or 'l', the leading (k+1) by n part of the array A must contain the lower triangular band part of the matrix of coefficients, supplied column by column, with the leading diagonal of the matrix in row 1 of the array, the first sub-diagonal starting at position 1 in row 2, and so on. The bottom right k by k triangle of the array A is not referenced. Note that when diag = ''U' or 'u', the elements of the array A corresponding to the diagonal elements of the matrix are not referenced, but are assumed to be unity.
lda	Specifies the first dimension of a as declared in the calling (sub) program. lda must be at least (k+1).
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1+(n-1)*abs(incx)). Before entry, the incremented array X must contain the n element vector x. On exit, X is overwritten with the transformed vector x.
incx	Specifies the increment for the elements of x. Must not be zero.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_TBSV Procedures

This procedure solves one of the systems of equations `A*x = b` or `A'*x = b`, where `b` and `x` are `n` element vectors and `A` is an `n` by n unit, or non-unit, upper or lower triangular band matrix, with `(k+1)` diagonals.

> **See Also:**
>
> BLAS Level 2 (Matrix-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_TBSV (
   uplo    IN      flag,
   trans   IN      flag,
   diag    IN      flag,
   n       IN      POSITIVEN,
   k       IN      NATURALN,
   a       IN      UTL_NLA_ARRAY_DBL,
   lda     IN      POSITIVEN,
   x       IN OUT  UTL_NLA_ARRAY_DBL,
   incx    IN      POSITIVEN,
   pack    IN      flag DEFAULT 'C');

UTL_NLA.BLAS_STBSV (
   uplo    IN      flag,
   trans   IN      flag,
   diag    IN      flag,
   n       IN      POSITIVEN,
   k       IN      NATURALN,
   a       IN      UTL_NLA_ARRAY_FLT,
   lda     IN      POSITIVEN,
   x       IN OUT  UTL_NLA_ARRAY_FLT,
   incx    IN      POSITIVEN,
   pack    IN      flag DEFAULT 'C');
```

Parameters

Table 297-32 BLAS_TBSV Procedure Parameters

Parameter	Description
`uplo`	Specifies whether the matrix is an upper or lower triangular matrix: • `uplo` = `'U'` or `'u'`. `A` is an upper triangular matrix. • `uplo` = `'L'` or `'l'`. `A` is a lower triangular matrix.
`trans`	Specifies the equations to be solved: • `trans` = `'N'` or `'n : 'A*x = b` • `trans` = `'T'` or `'t': A'*x = b` • `trans` = `'C'` or `'c': A'*x = b`
`diag`	Specifies whether or not `A` is unit triangular: • `diag` = `'U'` or `'u'` : A is assumed to be unit triangular. • `diag` = `'N'` or `'n'` : `A` is not assumed to be unit triangular.

Table 297-32 (Cont.) BLAS_TBSV Procedure Parameters

Parameter	Description
n	Specifies the order of the matrix A. n must be at least zero.
k	Specifies whether or not A is unit triangular: • with uplo = 'U' or 'u', K specifies the number of super-diagonals of the matrix A. • with uplo = 'L' or 'l', K specifies the number of sub-diagonals of the matrix A. K must satisfy 0 .le. k.
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda,n). Before entry with uplo = 'U' or 'u', the leading (k+1) by n part of the array A must contain the upper triangular band part of the matrix of coefficients, supplied column by column, with the leading diagonal of the matrix in row (k+1) of the array, the first super-diagonal starting at position 2 in row k, and so on. The top left k by k triangle of the array A is not referenced. Before entry with uplo = 'L' or 'l', the leading (k+1) by n part of the array A must contain the lower triangular band part of the matrix of coefficients, supplied column by column, with the leading diagonal of the matrix in row 1 of the array, the first sub-diagonal starting at position 1 in row 2, and so on. The bottom right k by k triangle of the array A is not referenced. Note that when diag = 'U' or 'u', the elements of the array A corresponding to the diagonal elements of the matrix are not referenced, but are assumed to be unity.
lda	On entry, lda specifies the first dimension of A as declared in the calling (sub) program. lda must be at least (k+1).
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (n - 1) *abs(incx)) Before entry, the incremented array X must contain the n element right-hand side vector b. On exit, X is overwritten with the solution vector x.
incx	Specifies the increment for the elements of x. incx must not be zero.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

Usage Notes

No test for singularity or near-singularity is included in this routine. Such tests must be performed before calling this routine.

BLAS_TPMV Procedures

This procedure performs the matrix-vector operations `x := A*x` or `x := A'*x`, where `x` is an `n` element vector and `A` is an `n` by `n` unit, or non-unit, upper or lower triangular matrix, supplied in packed form.

> **See Also:**
>
> BLAS Level 2 (Matrix-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_TPMV (
   uplo   IN      flag,
   trans  IN      flag,
   diag   IN      flag,
   n      IN      POSITIVEN,
   ap     IN      UTL_NLA_ARRAY_DBL,
   x      IN OUT  UTL_NLA_ARRAY_DBL,
   incx   IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_TBMV (
   uplo   IN      flag,
   trans  IN      flag,
   diag   IN      flag,
   n      IN      POSITIVEN,
   ap     IN      UTL_NLA_ARRAY_FLT,
   x      IN OUT  UTL_NLA_ARRAY_FLT,
   incx   IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-33 BLAS_TPMV Procedure Parameters

Parameter	Description
`uplo`	Specifies whether the matrix is an upper or lower triangular matrix: • `uplo` = `'U'` or `'u'`. `A` is an upper triangular matrix. • `uplo` = `'L'` or `'l'`. `A` is a lower triangular matrix.
`trans`	Specifies the operation to e performed: • `trans` = `'N'` or `'n'` `x := A*x` • `trans` = `'T'` or `'t'` `x := A'*x` • `trans` = `'C'` or `'c'` `x := A'*x`
`diag`	Specifies whether or not `A` is unit triangular: • `diag` = `'U'` or `'u'`. A is assumed to be unit triangular. • `diag` = `'N'` or `'n'`. `A` is not assumed to be unit triangular.
`n`	Specifies the order of the matrix `A`. `n` must be at least zero.

Table 297-33 (Cont.) BLAS_TPMV Procedure Parameters

Parameter	Description
ap	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda,n). Before entry with uplo = 'U' or 'u', the leading (k+1) by n part of the array A must contain the upper triangular band part of the matrix of coefficients, supplied column by column, with the leading diagonal of the matrix in row (k+1) of the array, the first super-diagonal starting at position 2 in row k, and so on. The top left k by k triangle of the array A is not referenced. Before entry with uplo = 'L' or 'l', the leading (k+1) by n part of the array A must contain the lower triangular band part of the matrix of coefficients, supplied column by column, with the leading diagonal of the matrix in row 1 of the array, the first sub-diagonal starting at position 1 in row 2, and so on. The bottom right k by k triangle of the array A is not referenced. Note that when diag = 'U' or 'u', the elements of the array A corresponding to the diagonal elements of the matrix are not referenced, but are assumed to be unity.
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1+(n-1)*abs(incx)). Before entry, the incremented array X must contain the n element vector x. On exit, X is overwritten with the transformed vector x.
incx	Specifies the increment for the elements of x. Must not be zero.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_TPSV Procedures

This procedure solves one of the systems of equations A*x = b or A'*x = b, where b and x are n element vectors and A is an n by n unit, or non-unit, upper or lower triangular matrix, supplied in packed form.

> **See Also:**
>
> BLAS Level 2 (Matrix-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_TPSV (
   uplo   IN      flag,
   trans  IN      flag,
   diag   IN      flag,
   n      IN      POSITIVEN,
   ap     IN      UTL_NLA_ARRAY_DBL,
   x      IN OUT  UTL_NLA_ARRAY_DBL,
   incx   IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');
```

```
UTL_NLA.BLAS_TPSV (
   uplo   IN      flag,
   trans  IN      flag,
   diag   IN      flag,
   n      IN      POSITIVEN,
   ap     IN      UTL_NLA_ARRAY_FLT,
   x      IN OUT  UTL_NLA_ARRAY_FLT,
   incx   IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-34 BLAS_TPSV Procedure Parameters

Parameter	Description
uplo	Specifies whether the matrix is an upper or lower triangular matrix: • uplo = 'U' or 'u' : A is an upper triangular matrix. • uplo = 'L' or 'l' : A is a lower triangular matrix.
trans	Specifies the operation to be performed: • trans = 'N' or 'n' : A*x = b • trans = 'T' or 't' : A'*x = b • trans = 'C' or 'c' : A'*x = b
diag	Specifies whether or not A is unit triangular: • diag = 'U' or 'u' : A is assumed to be unit triangular. • diag = 'N' or 'n' : 'A is not assumed to be unit triangular.
n	Specifies the order of the matrix A. n must be at least zero.
ap	UTL_NLA_ARRAY_FLT/DBL of dimension at least ((n*(n+1))/2) Before entry with uplo = 'U' or 'u', the array ap must contain the upper triangular matrix packed sequentially, column by column, so that ap(1) contains a(1,1), ap(2) and ap(3) contain a(1,2) and a(2,2) respectively, and so on. Before entry with uplo = 'L' or 'l', the array ap must contain the lower triangular matrix packed sequentially, column by column, so that ap(1) contains a(1,1), ap(2) and ap(3) contain a(2,1) and a(3,1) respectively, and so on. Note that when diag = 'U' or 'u', the diagonal elements of A are not referenced, but are assumed to be unity.
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (n - 1) *abs(incx)) Before entry, the incremented array X must contain the n element right-hand side vector b. On exit, X is overwritten with the solution vector x.
incx	Specifies the increment for the elements of x. incx must not be zero.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

Usage Notes

No test for singularity or near-singularity is included in this routine. Such tests must be performed before calling this routine.

BLAS_TRMM Procedures

This procedure performs a matrix-matrix operation.

It performs one of the following matrix-matrix operations:

```
B := alpha*op( A )*B
```

or

```
B := alpha*B*op( A )
```

where `alpha` is a scalar, `B` is an `m` by `n` matrix, `A` is a unit, or non-unit, upper or lower triangular matrix and `op(A)` is one of

```
op( A ) = A
```

or

```
op( A ) = A'
```

> **See Also:**
>
> BLAS Level 3 (Matrix-Matrix Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_TRMM (
   side   IN      flag,
   uplo   IN      flag,
   transa IN      flag,
   diag   IN      flag,
   m      IN      POSITIVEN,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_DOUBLE,
   a      IN      UTL_NLA_ARRAY_DBL,
   lda    IN      POSITIVEN,
   b      IN OUT  UTL_NLA_ARRAY_DBL,
   ldb    IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_TRMM (
   side   IN      flag,
   uplo   IN      flag,
   transa IN      flag,
   diag   IN      flag,
   m      IN      POSITIVEN,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_FLOAT,
   a      IN      UTL_NLA_ARRAY_FLT,
   lda    IN      POSITIVEN,
```

```
  b      IN OUT  UTL_NLA_ARRAY_FLT,
  ldb    IN      POSITIVEN,
  pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-35 BLAS_TRMM Procedure Parameters

Parameter	Description
side	Specifies whether the symmetric matrix A appears on the left or right in the operation: • side = 'L' or 'l' : B := alpha*op(A)*B • side = 'R' or 'r' : B := alpha*B*op(A)
uplo	Specifies whether the upper or lower triangular part of the array A is to be referenced: • uplo = 'U' or 'u' : A is an upper triangular matrix. • uplo = 'L' or 'l'' : A is a lower triangular matrix.
transa	Specifies the form of op(A) to be used in the matrix multiplication as follows: • transa = 'N' or 'n' : op(A) = A • transa = 'T' or 't' : op(A) = A' • transa ='C' or 'c' : op(A) = A'
diag	Specifies whether or not A is unit triangular: • diag = 'U' or 'u'. A is assumed to be unit triangular. • diag = 'N' or 'n'. A is not assumed to be unit triangular.
m	Specifies the number of rows of the B. m must be at least zero.
n	Specifies the number of columns of B. n must be at least zero.
alpha	SCALAR_FLOAT/DOUBLE. Specifies the scalar alpha. When alpha is zero then A is not referenced and B need not be set before entry.
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda,k) where k is m when side = 'L' or 'l', and is n when side = ''R' or 'r'. Before entry with uplo = 'U' or 'u' , the leading k by k upper triangular part of the array A must contain the upper triangular matrix, and the strictly lower triangular part of A is not referenced. Before entry with uplo = 'L' or 'l', the leading k byk lower triangular part of the array A must contain the lower triangular matrix and the strictly upper triangular part of A is not referenced. Note that when diag = ''U' or 'u', the diagonal elements of A are not referenced either, but are assumed to be unity.
lda	Specifies the first dimension of a as declared in the calling (sub) program. When side = 'L' or 'l', lda must be at least max(1,m), otherwise lda must be at least max(1,n).
b	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (ldb,n). Before entry, the leading m by n part of the array B must contain the matrix B, and on exit is overwritten by the transformed matrix.
ldb	Specifies the first dimension of b as declared in the calling (sub) program. ldb must be at least max(1,m).
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_TRMV Procedures

This procedure performs the matrix-vector operations `x := A*x` or `x := A'*x`, where `x` is an `n` element vector and `A` is an `n` by `n` unit, or non-unit, upper or lower triangular matrix.

> **See Also:**
>
> BLAS Level 2 (Matrix-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_TRMV (
   uplo   IN      flag,
   trans  IN      flag,
   diag   IN      flag,
   n      IN      POSITIVEN,
   a      IN      UTL_NLA_ARRAY_DBL,
   lda    IN      POSITIVEN,
   x      IN OUT  UTL_NLA_ARRAY_DBL,
   incx   IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_TRMV (
   uplo   IN      flag,
   trans  IN      flag,
   diag   IN      flag,
   n      IN      POSITIVEN,
   a      IN      UTL_NLA_ARRAY_FLT,
   lda    IN      POSITIVEN,
   x      IN OUT  UTL_NLA_ARRAY_FLT,
   incx   IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-36 BLAS_TRMV Procedure Parameters

Parameter	Description
`uplo`	Specifies whether the matrix is an upper or lower triangular matrix: • `uplo` = `'U'` or `'u'`. `A` is an upper triangular matrix. • `uplo` = `'L'` or `'l'`. `A` is a lower triangular matrix.
`trans`	Specifies the operation to be performed: • `trans` = `'N'` or `'n'` `x := A*x` • `trans`= `'T'` or `'t'` `x := A'*x` • `trans` = `'C'` or `'c'` `x := A'*x`
`diag`	Specifies whether or not `A` is unit triangular: • `diag` = `'U'` or `'u'`. A is assumed to be unit triangular. • `diag` = `'N'` or `'n'`. `A` is not assumed to be unit triangular.
`n`	Specifies the order of the matrix `A`. `n` must be at least zero.

Table 297-36 (Cont.) BLAS_TRMV Procedure Parameters

Parameter	Description
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda, n). Before entry with uplo = 'U' or 'u', the leading n by n upper triangular part of the array A must contain the upper triangular matrix and the strictly lower triangular part of A is not referenced. Before entry with uplo = 'L' or 'l', the leading n by n lower triangular part of the array A must contain the lower triangular matrix and the strictly upper triangular part of A is not referenced. Note that when diag = 'U' or 'u', the diagonal elements of A are not referenced either, but are assumed to be unity
lda	Specifies the first dimension of a as declared in the calling (sub) program. lda must be at least max(1,n).
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1+(n-1)*as(incx)). Before entry, the incremented array X must contain the n element vector x.
incx	Specifies the increment for the elements of x. Must not be zero.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_TRSM Procedures

This procedure performs a matrix-matrix operation.

It performs one of the matrix-matrix operations:

```
op( A )*X = alpha*B
```

or

```
X*op( A ) = alpha*B
```

where alpha is a scalar, X and B are m by n matrices, A is a unit, or non-unit, upper or lower triangular matrix and op(A) is one of

```
op( A ) = A
```

or

```
op( A ) = A'
```

The matrix X is overwritten on B.

> **See Also:**
>
> BLAS Level 3 (Matrix-Matrix Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_TRSM (
   side   IN      flag,
   uplo   IN      flag,
   transa IN      flag,
   diag   IN      flag,
   m      IN      POSITIVEN,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_DOUBLE,
   a      IN      UTL_NLA_ARRAY_DBL,
   lda    IN      POSITIVEN,
   b      IN OUT  UTL_NLA_ARRAY_DBL,
   ldb    IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.BLAS_TRSM (
   side   IN      flag,
   uplo   IN      flag,
   transa IN      flag,
   diag   IN      flag,
   m      IN      POSITIVEN,
   n      IN      POSITIVEN,
   alpha  IN      SCALAR_FLOAT,
   a      IN      UTL_NLA_ARRAY_FLT,
   lda    IN      POSITIVEN,
   b      IN OUT  UTL_NLA_ARRAY_FLT,
   ldb    IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-37 BLAS_TRSM Procedure Parameters

Parameter	Description
side	Specifies whether the symmetric matrix A appears on the left or right in the operation: • side = 'L' or 'l' : op(A)*X = alpha*B • side = 'R' or 'r' : X*op(A) = alpha*B
uplo	Specifies whether the upper or lower triangular part of the array A is to be referenced: • uplo = 'U' or 'u' : A is an upper triangular matrix. • uplo = 'L' or 'l' : A is a lower triangular matrix.
transa	Specifies the form of op(A) to be used in the matrix multiplication as follows: • transa = 'N' or 'n' : op(A) = A • transa = 'T' or 't' : op(A) = A' • transa = 'C' or 'c' : op(A) = A'
diag	Specifies whether or not A is unit triangular: • diag = 'U' or 'u'. A is assumed to be unit triangular. • diag = 'N' or 'n'. A is not assumed to be unit triangular.
m	Specifies the number of rows of the B. m must be at least zero.
n	Specifies the number of columns of B. n must be at least zero.

Table 297-37 (Cont.) BLAS_TRSM Procedure Parameters

Parameter	Description
alpha	SCALAR_FLOAT/DOUBLE. Specifies the scalar alpha. When alpha is zero then A is not referenced and B need not be set before entry.
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda, k) where k is m when side = 'L' or 'l', and is n when side = 'R' or 'r'. Before entry with uplo = 'U' or 'u', the leading k by k upper triangular part of the array A must contain the upper triangular matrix, and the strictly lower triangular part of A is not referenced. Before entry with uplo = 'L' or 'l', the leading k by k lower triangular part of the array A must contain the lower triangular matrix and the strictly upper triangular part of A is not referenced. Note that when diag = 'U' or 'u', the diagonal elements of A are not referenced either, but are assumed to be unity.
lda	Specifies the first dimension of a as declared in the calling (sub) program. When side = 'L' or 'l', lda must be at least max(1, m), otherwise lda must be at least max (1, n).
b	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (ldb, n). Before entry, the leading m by n part of the array B must contain the matrix B, and on exit is overwritten by the solution matrix X.
ldb	Specifies the first dimension of b as declared in the calling (sub) program. ldb must be at least max(1, m).
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

BLAS_TRSV Procedures

This procedure solves one of the systems of equations A*x = b or A'*x = b, where b and x are n element vectors and A is an n by n unit, or non-unit, upper or lower triangular matrix.

> **See Also:**
>
> BLAS Level 2 (Matrix-Vector Operations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.BLAS_TRSV (
   uplo   IN      flag,
   trans  IN      flag,
   diag   IN      flag,
   n      IN      POSITIVEN,
   a      IN      UTL_NLA_ARRAY_DBL,
   lda    IN      POSITIVEN,
   x      IN OUT  UTL_NLA_ARRAY_DBL,
   incx   IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');
```

```
UTL_NLA.BLAS_TRSV (
   uplo   IN      flag,
   trans  IN      flag,
   diag   IN      flag,
   n      IN      POSITIVEN,
   a      IN      UTL_NLA_ARRAY_FLT,
   lda    IN      POSITIVEN,
   x      IN OUT  UTL_NLA_ARRAY_FLT,
   incx   IN      POSITIVEN,
   pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-38 BLAS_TRSV Procedure Parameters

Parameter	Description
uplo	Specifies whether the matrix is an upper or lower triangular matrix: • uplo = 'U' or 'u'. A is an upper triangular matrix. • uplo = 'L' or 'l'. A is a lower triangular matrix.
trans	Specifies the operation to be performed: • trans = 'N' or 'n'A*x = b • trans = 'T' or 't'A'*x = b • trans = 'C' or 'c'A'*x = b
diag	Specifies whether or not A is unit triangular: • diag = 'U' or 'u'. A is assumed to be unit triangular. • diag = 'N' or 'n'. A is not assumed to be unit triangular.
n	Specifies the order of the matrix A. n must be at least zero.
a	UTL_NLA_ARRAY_FLT/DBL of DIMENSION (lda, n). Before entry with uplo = 'U' or 'u', the leading n by n upper triangular part of the array A must contain the upper triangular matrix and the strictly lower triangular part of A is not referenced. Before entry with uplo = 'L' or 'l', the leading n by n lower triangular part of the array A must contain the lower triangular matrix and the strictly upper triangular part of A is not referenced. Note that when diag = 'U' or 'u', the diagonal elements of A are not referenced either, but are assumed to be unity.
lda	Specifies the first dimension of A as declared in the calling (sub) program. lda must be at least max(1, n).
x	UTL_NLA_ARRAY_FLT/DBL of dimension at least (1 + (n - 1) * abs (incx)) Before entry, the incremented array X must contain the n element right-hand side vector b. On exit, X is overwritten with the solution vector x.
incx	Specifies the increment for the elements of x. Must not be zero.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

Usage Notes

No test for singularity or near-singularity is included in this routine. Such tests must be performed before calling this routine.

LAPACK_GBSV Procedures

This procedure computes the solution to a real system of linear equations `a * x = b`, where `a` is a band matrix of order `n` with `kl` sub diagonals and `ku` superdiagonals, and `x` and `b` are `n` by `nrhs` matrices.

The `LU` decomposition with partial pivoting and row interchanges is used to factor `A` as

```
a = L * U
```

where L is a product of permutation and unit lower triangular matrices with kl sub diagonals, and `U` is upper triangular with `kl+ku` superdiagonals. The factored form of `a` is then used to solve the system of equations

```
a * x = b
```

> **See Also:**
>
> LAPACK Driver Routines (Linear Equations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_GBSV (
   n      IN      POSITIVEN,
   kl     IN      NATURALN,
   ku     IN      NATURALN,
   nrhs   IN      POSITIVEN,
   ab     IN OUT  UTL_NLA_ARRAY_DBL,
   ldab   IN      POSITIVEN,
   ipiv   IN OUT  UTL_NLA_ARRAY_INT,
   b      IN OUT  UTL_NLA_ARRAY_DBL,
   ldb    IN      POSITIVEN,
   info   OUT  INTEGER,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.LAPACK_GBSV (
   n      IN      POSITIVEN,
   kl     IN      NATURALN,
   ku     IN      NATURALN,
   nrhs   IN      POSITIVEN,
   ab     IN OUT  UTL_NLA_ARRAY_FLT,
   ldab   IN      POSITIVEN,
   ipiv   IN OUT  UTL_NLA_ARRAY_INT,
   b      IN OUT  UTL_NLA_ARRAY_FLT,
   ldb    IN      POSITIVEN,
   info   OUT  INTEGER,
   pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-39 LAPACK_GBSV Procedure Parameters

Parameter	Description
n	The number of linear equations, equivalent to the order of the matrixa .n >= 0
kl	The number of sub diagonals within the band of a. kl >= 0.
ku	The number of superdiagonals within the band of a . ku >= 0.
nrhs	The number of right-hand sides, which is the number of columns of the matrix b. nrhs >= 0.
ab	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldab, n). On entry, the matrix a in band storage, in rows kl+1 to 2*kl+ku+1; rows 1 to kl of the array need not be set. The j-th column of A is stored in the j-th column of the array ab: ab(kl+ku+1+i-j,j) = a(i,j) for max(1,j-ku)<=i<=min(n,j+kl) On exit, details of the factorization: U is stored as an upper triangular band matrix with kl+ku superdiagonals in rows 1 to KL+KU+1, and the multipliers used during the factorization are stored in rows: kl+ku+2 to 2*kl+ku+1
ldab	The leading dimension of the array ab. ldab >= 2*kl+ku+1
ipiv	INTEGER array, DIMENSION (n). The pivot indices that define the permutation matrix P; row iof the matrix was interchanged with row ipiv(i).
b	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldb, nrhs). On entry, the n by nrhs matrix of right hand side matrix b. On exit, if info = 0 , the n by nrhs solution matrix X.
ldb	The leading dimension of the array b. ldb >= max(1,n)
info	• = 0 : successful exit • < 0 : if info = -i , the i-th argument had an illegal value • > 0 : if info = i, U(i,i) is exactly zero. The factorization has been completed, but the factor U is exactly singular, and the solution has not been computed
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_GEES Procedures

This procedure computes for an n by n real nonsymmetric matrix A, the eigenvalues, the real Schur form T, and, optionally, the matrix of Schur vectors Z.

This gives the Schur factorization A = Z*T*(Z**T).

A matrix is in real Schur form if it is upper quasi-triangular with `1` by `1` and `2` by `2` blocks. `2` by `2` blocks will be standardized in the form

```
[ a b ]

[ c a ]
```

where `b*c < 0`. The eigenvalues of such a block are `a +- sqrt(bc)`.

> **See Also:**
>
> LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_GEES (
   jobvs    IN       flag,
   n        IN       POSITIVEN,
   a        IN  OUT  UTL_NLA_ARRAY_DBL,
   lda      IN       POSITIVEN,
   wr       IN OUT   UTL_NLA_ARRAY_DBL,
   wi       IN OUT   UTL_NLA_ARRAY_DBL,
   vs       IN OUT   UTL_NLA_ARRAY_DBL,
   ldvs     IN       POSITIVEN,
   info     OUT      INTEGER,
   pack     IN       flag DEFAULT 'C');

UTL_NLA.LAPACK_GEES (
   jobvs    IN       flag,
   n        IN       POSITIVEN,
   a        IN OUT   UTL_NLA_ARRAY_FLT,
   lda      IN       POSITIVEN,
   wr       IN       OUT UTL_NLA_ARRAY_FLT,
   wi       IN       OUT UTL_NLA_ARRAY_FLT,
   vs       IN OUT   UTL_NLA_ARRAY_FLT,
   ldvs     IN       POSITIVEN,
   info     OUT      integer,
   pack     IN       flag DEFAULT 'C');
```

Parameters

Table 297-40 LAPACK_GEES Procedure Parameters

Parameter	Description
`jobz`	• `'N'`: Schur vectors are not computed. • `'V'`: Schur vectors are computed.
`n`	The order of the matrix `a. N >= 0`.
a	`UTL_NLA_ARRAY_FLT/DBL, DIMENSION (lda, n).` • On entry, the `n` by `n` matrix `A`. • On exit, `A` has been overwritten by its real Schur form `T`.
`lda`	The leading dimension of the array `a. lda >= max(1,n).`

Table 297-40 (Cont.) LAPACK_GEES Procedure Parameters

Parameter	Description
wr	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n). wr and wi contain the real and imaginary parts respectively of the computed eigenvalues in the same order that they appear on the diagonal of the output Schur form T. Complex conjugate pairs of eigenvalues will appear consecutively with the eigenvalue having the positive imaginary part first.
wi	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldz, n). wr and wi contain the real and imaginary parts respectively of the computed eigenvalues in the same order that they appear on the diagonal of the output Schur form T. Complex conjugate pairs of eigenvalues will appear consecutively with the eigenvalue having the positive imaginary part first.
vs	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n). • If jobvs = 'V', vs contains the orthogonal matrix Z of Schur vectors. • If jobvs = 'N', vs is not referenced.
ldvs	The leading dimension of the array vs. VS. ldvs >= 1. If jobvs = 'V', ldvs >= N
info	• = 0 : successful exit • < 0 : if info = -i, the i-th argument had an illegal value • > 0 : if info = i, and i is <= N: the QR algorithm failed to compute all the eigenvalues. Elements 1:ILO-1 and i+1:N of wr and wi contain those eigenvalues which have converged. If jobvs = 'V', vs contains the matrix which reduces A to its partially converged Schur form.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_GELS Procedures

This procedure solves overdetermined or underdetermined real linear systems involving an m by n matrix A, or its transpose, using a QR or LQ factorization of A. It is assumed that A has full rank.

The following options are provided:

1. If TRANS = 'N' and m >= n: find the least squares solution of an overdetermined system, that is, solve the least squares problem.

   ```
   minimize || B - A*X ||
   ```

2. If TRANS = 'N' and m < n: find the minimum norm solution of an underdetermined system A * X = B.
3. If TRANS = 'T' and m >= n: find the minimum norm solution of an undetermined system A**T * X = B.
4. If TRANS = 'T' and m < n: find the least squares solution of an overdetermined system, that is, solve the least squares problem minimize || B - A**T * X ||.

> **See Also:**
>
> LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_GELS (
   trans    IN       flag,
   m        IN       POSITIVEN,
   n        IN       POSITIVEN,
   nrhs     IN       POSITIVEN,
   a        IN OUT   UTL_NLA_ARRAY_DBL,
   lda      IN       POSITIVEN,
   b        IN OUT   UTL_NLA_ARRAY_DBL,
   ldb      IN       POSITIVEN,
   info     OUT      INTEGER,
   pack     IN       flag DEFAULT 'C');

UTL_NLA.LAPACK_GELS (
  trans     IN       flag,
  m         IN       POSITIVEN,
  n         IN       POSITIVEN,
  nrhs      IN       POSITIVEN,
  a         IN OUT   UTL_NLA_ARRAY_FLT,
  lda       IN       POSITIVEN,
  b         IN OUT   UTL_NLA_ARRAY_FLT,
  ldb       IN       POSITIVEN,
  info      OUT      INTEGER,
  pack      IN       flag DEFAULT 'C');
```

Parameters

Table 297-41 LAPACK_GELS Procedure Parameters

Parameter	Description
trans	• CHARACTER = 'N': The linear system involves A. • CHARACTER = 'T': The linear system involves A**T .
m	The number of rows of the matrix a. M >= 0.
n	The number of columns of the matrix a. N >= 0.
nrhs	The number of right-hand sides, which is the number of columns of the matrix band x.nrhs >= 0.
a	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (lda, n). On entry, the matrix b of right hand side vectors, stored columnwise; b is m by nrhs if TRANS = 'N', or n by nrhsif trans = 'T'. On exit, if m >= n, a is overwritten by details of its QR factorization as returned by SGEQRF. If m < n, A is overwritten by details of its LQ factorization as returned by SGELQF.
lda	The leading dimension of the array A. lda >= max(1,m).

Table 297-41 (Cont.) LAPACK_GELS Procedure Parameters

Parameter	Description
b	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldb, nrhs). On entry, the matrix b of right hand side vectors, stored columnwise. b is m bynrhs if trans = 'n', or n by nrhs if trans = 'T'. On exit, b is overwritten by the solution vectors, stored columnwise: • If trans = 'n' and m >= n, rows 1 to n of b contain the least squares solution vectors; the residual sum of squares for the solution in each column is given by the sum of squares of elements n+1 to m in that column. • If trans = 'n' and m < n, rows 1 to n of b contain the minimum norm solution vectors. • If trans = 'T' and m >= n, rows 1 to m of b contain the minimum norm solution vectors. • If trans = 'T' and m < n, rows 1 to m of b contain the least squares solution vectors; the residual sum of squares for the solution in each column is given by the sum of squares of elements m+1 to n in that column.
ldb	The leading dimension of the array b. ldb >= max(1,m,n)
info	• = 0 : successful exit • < 0 : if info = -i, the i-th argument had an illegal value
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_GESDD Procedures

This procedures computes the singular value decomposition (SVD) of a real m by n matrix A, optionally computing the left and right singular vectors. If singular vectors are desired, it uses a divide-and-conquer algorithm that makes mild assumptions about floating point arithmetic.

The SVD is written

```
A = U * SIGMA * transpose(V)
```

where SIGMA is an m by n matrix which is zero except for its min(m,n) diagonal elements, U is anm by m orthogonal matrix, and V is ann by n orthogonal matrix. The diagonal elements of SIGMA are the singular values of A, they are real and non-negative, and are returned in descending order. The first min(m,n) columns of U and V are the left and right singular vectors of A.

Note that the routine returns V**T, not V.

> **See Also:**
>
> LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_GESDD (
   jobz   IN       flag,
   m      IN       POSITIVEN,
   n      IN       POSITIVEN,
   a      IN OUT   UTL_NLA_ARRAY_DBL,
   lda    IN       POSITIVEN,
   s      IN OUT   UTL_NLA_ARRAY_DBL,
   u      IN OUT   UTL_NLA_ARRAY_DBL,
   ldu    IN       POSITIVEN,
   vt     IN OUT   UTL_NLA_ARRAY_DBL,
   ldvt   IN       POSITIVEN,
   info   OUT      INTEGER,
   pack   IN       flag DEFAULT 'C');

UTL_NLA.LAPACK_GESDD (
   jobz   IN       flag,
   m      IN       POSITIVEN,
   n      IN       POSITIVEN,
   a      IN OUT   UTL_NLA_ARRAY_FLT,
   lda    IN       POSITIVEN,
   s      IN OUT   UTL_NLA_ARRAY_FLT,
   u      IN OUT   UTL_NLA_ARRAY_FLT,
   ldu    IN       POSITIVEN,
   vt     IN OUT   UTL_NLA_ARRAY_FLT,
   ldvt   IN       POSITIVEN,
   info   OUT      INTEGER,
   pack   IN       flag DEFAULT 'C');
```

Parameters

Table 297-42 LAPACK_GESDD Procedure Parameters

Parameter	Description
jobz	Specifies options for computing all or part of the matrix U: • 'A': All m columns of u and all n rows of V**T are returned in arrays u and vt. • 'S': The first min(m,n) columns of u and the first min(m,n) rows of V**T are returned in the arrays u and vt. • 'O': The first min(m,n) columns of u (the left singular vectors) are overwritten on the array a. jobu and jobvt cannot both be 'O • 'N': No columns of u (no left singular vectors) are computed.
m	The order of the matrix a. m >= 0.
n	The order of the matrix a. n >= 0.
a	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (lda, n). On entry, the n by n matrix A. On exit: • If jobz = 'O', a is overwritten with the first min(m,n) columns of u (the left singular vectors, stored columnwise). • If m >= n, a is overwritten with the first m rows of V**T (the right singular vectors, stored rowwise). • If jobz .ne. 'O', the contents of a are destroyed.
lda	The leading dimension of the array a. lda >= max(1,m).

Table 297-42 (Cont.) LAPACK_GESDD Procedure Parameters

Parameter	Description
s	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (min(m,n)). The singular values of a, sorted so that S(i) >= S(i+1).
u	UTL_NLA_ARRAY_FLT/DBL. ucol = m if jobz = 'A' or jobz = 'O' and m < n; ucol = min(m,n) if jobz = 'S'. • If jobz = 'A' or jobz = 'O' and m < n, u contains the m by m orthogonal matrix u. • If jobz = 'S', u contains the first min(m,n) columns of u (the left singular vectors, stored columnwise). • If jobz = 'O' and m >= n, or jobz = 'n', u is not referenced.
ldu	The leading dimension of the array U. ldu >= 1. If jobz = 'S' or 'A',or jobz = 'O' and m < n, ldu >= m.
vt	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldvt, n). • If jobz = 'A' or jobz = 'O' and m >= n, vt contains the n by n orthogonal matrix V**T. • If jobz = 'S', vt contains the first min(m,n) rows of V**T (the right singular vectors, stored rowwise). • If jobz = 'O' and m < n, or jobz = 'N', vt is not referenced.
ldvt	The leading dimension of the array vt. ldvt >= 1. • If jobz = 'A', or jobz = 'O' and m >= n, ldvt >= n. • If jobz = 'S', ldvt >= min(m,n).
info	• = 0 : successful exit • < 0 : If info = -i, the i-th argument had an illegal value • > 0 : SBDSDC did not converge, updating process failed.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_GESV Procedure

This procedure computes the solution to a real system of linear equations a * x = b, where a is an n by n matrix and x and b are n by nrhs matrices.

The LU decomposition with partial pivoting and row interchanges is used to factor A as

```
a = P * L * U
```

where P is a permutation matrix, L is unit lower triangular, and U is upper triangular. The factored form of a is then used to solve the system of equations

```
a * x = b
```

> **See Also:**
>
> LAPACK Driver Routines (Linear Equations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_GESV (
   n      IN      POSITIVEN,
   nrhs   IN      POSITIVEN,
   a      IN OUT  UTL_NLA_ARRAY_DBL,
   lda    IN      POSITIVEN,
   ipiv   IN OUT  UTL_NLA_ARRAY_INT,
   b      IN OUT  UTL_NLA_ARRAY_DBL,
   ldb    IN      POSITIVEN,
   info      OUT  INTEGER,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.LAPACK_GESV (
   n      IN      POSITIVEN,
   nrhs   IN      POSITIVEN,
   a      IN OUT  UTL_NLA_ARRAY_FLT,
   lda    IN      POSITIVEN,
   ipiv   IN OUT  UTL_NLA_ARRAY_INT,
   b      IN OUT  UTL_NLA_ARRAY_FLT,
   ldb    IN      POSITIVEN,
   info      OUT  INTEGER,
   pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-43 LAPACK_GESV Procedure Parameters

Parameter	Description
n	The number of linear equations, equivalent to the order of the matrix a. n >= 0
nrhs	The number of right-hand sides, which is the number of columns of the matrix b. nrhs >= 0.
a	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (lda, n). On entry, the n by n coefficient matrix a. On exit, the factors L and U from the factorization a = P*L*U; the unit diagonal elements of L are not stored.
lda	The leading dimension of the array a. lda >= max(1,n)
ipiv	INTEGER array, DIMENSION (n). The pivot indices that define the permutation matrix P; row iof the matrix was interchanged with row ipiv(i).
b	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldb, nrhs). On entry, the n by nrhs matrix of right hand side matrix b. On exit, if info = 0 , the n by nrhs solution matrix X.
ldb	The leading dimension of the array b. ldb >= max(1,n)
info	• = 0 : successful exit • < 0 : if info = -i , the i-th argument had an illegal value • > 0 : if info = i, U(i,i) is exactly zero. The factorization has been completed, but the factor U is exactly singular, so the solution could not be computed.

Table 297-43 (Cont.) LAPACK_GESV Procedure Parameters

Parameter	Description
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_GESVD Procedures

This procedures computes the singular value decomposition (SVD) of a real m by n matrix A, optionally computing the left and/or right singular vectors.

The SVD is written

```
A = U * SIGMA * transpose(V)
```

where SIGMA is an m by n matrix which is zero except for its min(m,n) diagonal elements, U is anm by m orthogonal matrix, and V is ann by n orthogonal matrix. The diagonal elements of SIGMA are the singular values of A, they are real and non-negative, and are returned in descending order. The first min(m,n) columns of U and V are the left and right singular vectors of A.

Note that the routine returns V**T, not V.

> **See Also:**
>
> LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_GESVD (
   jobu   IN       flag,
   jobvt  IN       flag,
   m      IN       POSITIVEN,
   n      IN       POSITIVEN,
   a      IN OUT   UTL_NLA_ARRAY_DBL,
   lda    IN       POSITIVEN,
   s      IN OUT   UTL_NLA_ARRAY_DBL,
   u      IN OUT   UTL_NLA_ARRAY_DBL,
   ldu    IN       POSITIVEN,
   vt     IN OUT   UTL_NLA_ARRAY_DBL,
   ldvt   IN       POSITIVEN,
   info   OUT      INTEGER,
   pack   IN       flag DEFAULT 'C');

UTL_NLA.LAPACK_GESVD (
   jobu   IN       flag,
   jobvt  IN       flag,
   m      IN       POSITIVEN,
   n      IN       POSITIVEN,
   a      IN OUT   UTL_NLA_ARRAY_FLT,
   lda    IN       POSITIVEN,
   s      IN OUT   UTL_NLA_ARRAY_FLT,
```

```
u      IN OUT   UTL_NLA_ARRAY_FLT,
ldu    IN       POSITIVEN,
vt     IN OUT   UTL_NLA_ARRAY_FLT,
ldvt   IN       POSITIVEN,
info   OUT      INTEGER,
pack   IN       flag DEFAULT 'C');
```

Parameters

Table 297-44 LAPACK_GESVD Procedure Parameters

Parameter	Description
jobu	Specifies options for computing all or part of the matrix U: • 'A': All m columns of U are returned in array U. • 'S': The first min(m,n) columns of U (the left singular vectors) are returned in the array U. • 'O': The first min(m,n) columns of U (the left singular vectors) are overwritten on the array a. jobu and jobvt cannot both be 'O' • 'N': No columns of U (no left singular vectors) are computed.
jobvt	Specifies options for computing all or part of the matrix V**T: • 'A': All n rows of V**T are returned in the array vt. • 'S': The first min(m,n) rows of V**T (the right singular vectors) are returned in the array vt. • 'O': The first min(m,n) rows of V**T (the right singular vectors) are overwritten on the array a. jobvt and jobu cannot both be 'O'. • 'N': No rows of V**T (no right singular vectors) are computed.
m	The order of the matrix a. M >= 0.
n	The order of the matrix a. N >= 0.
a	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (lda, n). On entry, the n by n matrix A. On exit: • If jobu = 'O', A is overwritten with the first min(m,n) columns of U (the left singular vectors, stored columnwise); • If jobvt = 'O', A is overwritten with the first min(m,n) rows of V**T (the right singular vectors, stored rowwise); • If jobu.ne.'O' and jobvt.ne.'O', the contents of A are destroyed.
lda	The leading dimension of the array a. lda >= max(1,n).
s	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (min(m,n)). The singular values of A, sorted so that S(i) >= S(i+1).
u	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldu,ucol).(ldu,m)if jobu = 'A' or (ldu,min(m,n)) if jobu = 'S'. • If jobu = 'A', U contains the m by m orthogonal matrix U. • If jobu = 'S', U contains the first min(m,n) columns of U (the left singular vectors, stored columnwise). • If jobu = 'N' or 'O', U is not referenced.
ldu	The leading dimension of the array U. ldu >= 1. If jobu = 'S' or 'a', ldu >= m.

Table 297-44 (Cont.) LAPACK_GESVD Procedure Parameters

Parameter	Description
vt	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldvt, n). • If jobvt = 'A', vt contains the n by n orthogonal matrix V**T. • If jobvt = 'S', vt contains the first min(m,n) rows of V**T (the right singular vectors, stored rowwise). • If jobvt = 'N' or 'O', vt is not referenced.
ldvt	The leading dimension of the array vt. ldvt >= 1. • If jobvt = 'A', ldvt >= n. • If jobvt = 'S', ldvt >= min(m,n).
info	• = 0 : successful exit • < 0 : If info = -i, the i-th argument had an illegal value • > 0 : If SBDSQR did not converge, info specifies how many superdiagonals of an intermediate bidiagonal form B did not converge to zero.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_GEEV Procedures

This procedures computes for an n by n real nonsymmetric matrix A, the eigenvalues and, optionally, the left and/or right eigenvectors.

- The right eigenvector v(j) of A satisfies A * v(j) = lambda(j) * v(j) where lambda(j) is its eigenvalue.
- The left eigenvector u(j) of A satisfies u(j)**H * A = lambda(j) * u(j)**H where u(j)**H denotes the conjugate transpose of u(j).

The computed eigenvectors are normalized to have Euclidean norm equal to 1 and largest component real.

> **See Also:**
>
> LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_GEEV (
   jobvl    IN      flag,
   jobvr    IN      flag,
   n        IN      POSITIVEN,
   a        IN OUT  UTL_NLA_ARRAY_DBL,
   lda      IN      POSITIVEN,
   wr       IN OUT  UTL_NLA_ARRAY_DBL,
   wi       IN OUT  UTL_NLA_ARRAY_DBL,
   vl       IN OUT  UTL_NLA_ARRAY_DBL,
   ldvl     IN      POSITIVEN,
   vr       IN OUT  UTL_NLA_ARRAY_DBL,
```

```
   ldvr    IN      POSITIVEN,
   info    OUT     INTEGER,
   pack    IN      flag DEFAULT 'C');

UTL_NLA.LAPACK_GEEV (
   jobvl   IN      flag,
   jobvr   IN      flag,
   n       IN      POSITIVEN,
   a       IN OUT  UTL_NLA_ARRAY_FLT,
   lda     IN      POSITIVEN,
   wr      IN OUT  UTL_NLA_ARRAY_FLT,
   wi      IN OUT  UTL_NLA_ARRAY_FLT,
   vl      IN OUT  UTL_NLA_ARRAY_FLT,
   ldvl    IN      POSITIVEN,
   vr      IN OUT  UTL_NLA_ARRAY_FLT,
   ldvr    IN      POSITIVEN,
   info    OUT     INTEGER,
   pack    IN      flag DEFAULT 'C');
```

Parameters

Table 297-45 LAPACK_GEEV Procedure Parameters

Parameter	Description
jobvl	• 'N': Left eigenvectors of A are not computed. • 'V': Left eigenvectors of A are computed.
jobvr	• 'N': Right eigenvectors of A are not computed. • 'V': Right eigenvectors of A are computed.
n	The order of the matrix a. N >= 0.
a	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (lda, n). • On entry, the n by n matrix A. • On exit, A has been overwritten.
lda	The leading dimension of the array a. lda >= max(1,n).
wr	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n). wr and wi contain the real and imaginary parts respectively of the computed eigenvalues. Complex conjugate pairs of eigenvalues will appear consecutively with the eigenvalue having the positive imaginary part first.
wi	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldz, n). wr and wi contain the real and imaginary parts respectively of the computed eigenvalues. Complex conjugate pairs of eigenvalues will appear consecutively with the eigenvalue having the positive imaginary part first.
vl	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n). • If jobvl = 'V', the left eigenvectors u(j) are stored one after another in the columns of vl, in the same order as their eigenvalues. • If jobvs = 'N', vl is not referenced. • If the j-th eigenvalue is real, then u(j) = VL(:,j), the j-th column of vl. • If the j-th and (j+1)-st eigenvalues form a complex conjugate pair, then u(j) = VL(:,j) + i*VL(:,j+1) and u(j+1) = VL(:,j) - i*VL(:,j+1).

Table 297-45 (Cont.) LAPACK_GEEV Procedure Parameters

Parameter	Description
ldv1	The leading dimension of the array vl. ldvl >= 1. If jobvl = 'v', ldvl >= n.
vr	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldvr, n). • If jobvr = 'V', the right eigenvectors v(j) are stored one after another in the columns of vr, in the same order as their eigenvalues.. • If jobvr = 'N', vr is not referenced. • If the j-th eigenvalue is real, then v(j) = VR(:,j), the j-th column of vr. • If the j-th and (j+1)-st eigenvalues form a complex conjugate pair, then v(j) = VR(:,j) + i*VR(:,j+1) and v(j+1) = VR(:,j) - i*VR(:,j+1).
ldvr	The leading dimension of the array vr. vr.ldvr >= 1. If jobvr = 'V', ldvr >= N
info	• = 0 : successful exit • < 0 : if info = -i, the i-th argument had an illegal value • > 0 : if info = i, and i is <= N: the QR algorithm failed to compute all the eigenvalues, and no eigenvectors have been computed. Elements i+1:N of wr and wi contain eigenvalues which have converged..
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_GTSV Procedure

This procedure solves the equation a * x = b, where a is an n by n tridiagonal matrix, by Gaussian elimination with partial pivoting.

Note that the equation a'*x = b may be solved by interchanging the order of the arguments du and dl.

> **See Also:**
>
> LAPACK Driver Routines (Linear Equations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_GTSV (
   n      IN      POSITIVEN,
   nrhs   IN      POSITIVEN,
   dl     IN OUT  UTL_NLA_ARRAY_DBL,
   d      IN OUT  UTL_NLA_ARRAY_DBL,
   du     IN OUT  UTL_NLA_ARRAY_DBL,
   b      IN OUT  UTL_NLA_ARRAY_DBL,
   ldb    IN      POSITIVEN,
```

```
   info      OUT  INTEGER,
   pack   IN      flag DEFAULT 'C');

UTL_NLA.LAPACK_GTSV (
   n      IN      POSITIVEN,
   nrhs   IN      POSITIVEN,
   dl     IN OUT  UTL_NLA_ARRAY_FLT,
   d      IN OUT  UTL_NLA_ARRAY_FLT,
   du     IN OUT  UTL_NLA_ARRAY_FLT,
   b      IN OUT  UTL_NLA_ARRAY_FLT,
   ldb    IN      POSITIVEN,
   info      OUT  INTEGER,
   pack   IN      flag DEFAULT 'C');
```

Parameters

Table 297-46 LAPACK_GTSV Procedure Parameters

Parameter	Description
n	The order of the matrix a .n >= 0
nrhs	The number of right-hand sides, which is the number of columns of the matrix b. nrhs >= 0.
dl	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n-1). On entry, dl must contain the (n-1) sub-diagonal elements of a. On exit, dl is overwritten by the (n-2) elements of the second super-diagonal of the upper triangular matrix U from the LU factorization of a, in dl(1), ..., dl(n-2).
d	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n). On entry, d must contain the diagonal elements of a. On exit, d is overwritten by the n diagonal elements of U.
du	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n-1). On entry, du must contain the (n-1) super-diagonal elements of a. On exit, du is overwritten by the (n-1) elements of the first super-diagonal of U.
b	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (LDB, nrhs). On entry, the n by nrhs matrix of right hand side matrix b. On exit, if info = 0, the n by nrhs solution matrix X.
ldb	The leading dimension of the array b. ldb >= max (1, n)
info	• = 0 : successful exit • < 0 : if info = -i , the i-th argument had an illegal value • > 0 : if info = i, U(i,i) is exactly zero, and the solution has not been computed. The factorization has not been completed unless i = n.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_PBSV Procedures

This procedure computes the solution to a real system of linear equations `a * x = b`, where `a` is an `n` by `n` symmetric positive definite band matrix and `x` and `b` are `n` by `nrhs` matrices.

The Cholesky decomposition is used to factor `A` as

`A = U**T*U` if `UPLO ='U'`

or

`A = L * L**T` if `UPLO = 'L'`

where `U` is an upper triangular matrix and `L` is a lower triangular matrix. The factored form of `A` is then used to solve the system of equations `A * X = B`.

> **See Also:**
>
> LAPACK Driver Routines (Linear Equations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_PBSV (
uplo        IN      flag,
n           IN      POSITIVEN,
kd          IN      NATURALN,
nrhs        IN      POSITIVEN,
ab          IN OUT  UTL_NLA_ARRAY_DBL,
ldab        IN      POSITIVEN,
b           IN OUT  UTL_NLA_ARRAY_DBL,
ldb         IN      POSITIVEN,
info        OUT     INTEGER,
pack        IN      flag DEFAULT 'C');


UTL_NLA.LAPACK_PBSV (
uplo       IN      flag,
n          IN      POSITIVEN,
kd         IN      NATURALN,
nrhs       IN      POSITIVEN,
ab         IN OUT  UTL_NLA_ARRAY_FLT,
ldab       IN      POSITIVEN,
b          IN OUT  UTL_NLA_ARRAY_FLT,
ldb        IN      POSITIVEN,
info       OUT     INTEGER,
pack       IN      flag DEFAULT 'C');
```

Parameters

Table 297-47 LAPACK_PBSV Procedure Parameters

Parameter	Description
`uplo`	• `uplo = 'U'`. Upper triangular of `A` is stored. • `uplo = 'L'`. Lower triangular of `A` is stored.

Table 297-47 (Cont.) LAPACK_PBSV Procedure Parameters

Parameter	Description
n	The number of linear equations, that is, the order of the matrix a .n >= 0
kd	The number of superdiagonals of the matrix A if uplo = 'U', or the number of subdiagonals if UPLO = 'L'. KD >= 0.
nrhs	The number of right-hand sides, which is the number of columns of the matrix b. nrhs >= 0.
ab	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldab, n). On entry, the upper or lower triangle of the symmetric band matrix a, stored in the first kd+1 rows of the array. The j-th column of a is stored in the j-th column of the array ab is as follows: • if uplo = 'U', AB(KD+1+i-j,j) = A(i,j) for max(1,j-KD)<=i<=j; • if uplo = 'L', AB(1+i-j,j) = A(i,j) for j<=i<=min(N,j+KD) .See below for further details.On exit, if info = 0, the triangular factor U or L from the Cholesky factorization A = U**T*U or A = L*L**T of the bandmatrix A, in the same storage format as a.
ldab	The leading dimension of the array ab. ldb >= kd+1
b	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldb, nrhs). On entry, the n by nrhs matrix of right hand side matrix b. On exit, if info = 0 , the n by nrhs solution matrix X.
ldb	The leading dimension of the array b. ldb >= max(1,n)
info	• = 0 : successful exit • < 0 : if info = -i , the i-th argument had an illegal value • > 0 : if info = i, the leading minor of order i of a is not positive definite, so the factorization could not be completed, and the solution has not been computed.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_POSV Procedures

This procedure computes the solution to a real system of linear equations a * x = b, where a is an n by n symmetric positive definite matrix and x and b are n by nrhs matrices.

The Cholesky decomposition is used to factor A as

A = U**T* U if uplo = 'U'

or

A = L * L**T if UPLO = 'L'

where U is an upper triangular matrix and L is a lower triangular matrix. The factored form of A is then used to solve the system of equations A * X = B.

> **See Also:**
>
> LAPACK Driver Routines (Linear Equations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_POSV (
   uplo      IN      flag,
   n         IN      POSITIVEN,
   nrhs      IN      POSITIVEN,
   a         IN OUT  UTL_NLA_ARRAY_DBL,
   lda       IN      POSITIVEN,
   b         IN OUT  UTL_NLA_ARRAY_DBL,
   ldb       IN      POSITIVEN,
   info      OUT     INTEGER,
   pack      IN      flag DEFAULT 'C');

UTL_NLA.LAPACK_POSV (
   uplo      IN      flag,
   n         IN      POSITIVEN,
   nrhs      IN      POSITIVEN,
   a         IN OUT  UTL_NLA_ARRAY_FLT,
   lda       IN      POSITIVEN,
   b         IN OUT  UTL_NLA_ARRAY_FLT,
   ldb       IN      POSITIVEN,
   info      OUT     INTEGER,
   pack      IN      flag DEFAULT 'C');
```

Parameters

Table 297-48 LAPACK_POSV Procedure Parameters

Parameter	Description
uplo	• uplo = 'U'. Upper triangular of A is stored. • uplo = 'L'. Lower triangular of A is stored.
n	The number of linear equations, that is, the order of the matrix a .n >= 0
nrhs	The number of right-hand sides, which is the number of columns of the matrix b. nrhs >= 0.
a	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (lda, n). If uplo = 'U', the leading NRHS n by n upper triangular part of a contains the upper NRHS triangular part of the matrix A, and the strictly lower NRHS triangular part of A is not referenced. If uplo = ''L', then rhs leading n by n lower triangular part of a contains the lower nrhs triangular part of the matrix a, and the strictly upper nrhs triangular part of a is not referenced. On exit, if info = 0, the factor U or L from the Cholesky factorization A = U**T*U or A = L*L**T.

Table 297-48 (Cont.) LAPACK_POSV Procedure Parameters

Parameter	Description
lda	The leading dimension of the array a. lda >= max (1, n)
b	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldb, nrhs). On entry, the n by nrhs matrix of right hand side matrix b. On exit, if info = 0 , the n by nrhs solution matrix X.
ldb	The leading dimension of the array b. ldb >= max(1,n)
info	• = 0 : successful exit • < 0 : if info = -i , the i-th argument had an illegal value • > 0 : if info = i, the leading minor of order i of a is not positive definite, so the factorization could not be completed, and the solution has not been computed.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_PPSV Procedures

This procedure computes the solution to a real system of linear equations a * x = b where a is an n by n symmetric positive definite matrix stored in packed format and x and b are n by nrhs matrices.

The Cholesky decomposition is used to factor A as

A = U**T* U if UPLO = 'U'

or

A = L * L**T if UPLO = 'L'

where **U** is an upper triangular matrix and L is a lower triangular matrix. The factored form of A is then used to solve the system of equations A * X = B.

> **See Also:**
>
> LAPACK Driver Routines (Linear Equations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_PPSV (
   uplo      IN      flag,
   n         IN      POSITIVEN,
   nrhs      IN      POSITIVEN,
   ap        IN OUT  UTL_NLA_ARRAY_DBL,
   b         IN OUT  UTL_NLA_ARRAY_DBL,
   ldb       IN      POSITIVEN,
   info      OUT     INTEGER,
```

```
   pack      IN      flag DEFAULT 'C');


UTL_NLA.LAPACK_PPSV (
   uplo      IN      flag,
   n         IN      POSITIVEN,
   nrhs      IN      POSITIVEN,
   ap        IN OUT  UTL_NLA_ARRAY_FLT,
   b         IN OUT  UTL_NLA_ARRAY_FLT,
   ldb       IN      POSITIVEN,
   info      OUT     INTEGER,
   pack      IN      flag DEFAULT 'C');
```

Parameters

Table 297-49 LAPACK_PPSV Procedure Parameters

Parameter	Description
uplo	• uplo = 'U' . Upper triangular of A is stored. • uplo = 'L'. Lower triangular of A is stored.
n	The number of linear equations, that is, the order of the matrix a .n >= 0
nrhs	The number of right-hand sides, which is the number of columns of the matrix b. nrhs >= 0.
ap	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n*(n+1)/2). On entry, the upper or lower triangle of the symmetric matrix a, packed columnwise in a linear array. The j-th column of a is stored in the array ap as follows: If uplo = 'U', AP(i + (j-1)*j/2) = A(i,j) for 1<=i<=j; If uplo = 'L', AP(i + (j-1)*(2n-j)/2) = A(i,j) for j<=i<=n; On exit, if info = 0, the factor U or 'L' from the Cholesky factorization A = U**T*U or A = L*L**T in the same storage format as A.
b	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldb, nrhs). On entry, the n by nrhs matrix of right hand side matrix b. On exit, if info = 0 , the n by nrhs solution matrix X.
ldb	The leading dimension of the array b. ldb >= max(1,n)
info	• = 0 : successful exit • < 0 : if info = -i , the i-th argument had an illegal value • > 0 : if info = i, the leading minor of order i of a is not positive definite, so the factorization could not be completed, and the solution has not been computed.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_PTSV Procedures

This procedure computes the solution to a real system of linear equations `a * x = b`, where `a` is an `n` by `n` symmetric positive definite tridiagonal matrix, and `x` and `b` are `n` by `nrhs` matrices.

`a` is factored as `A = L*D*L**T`, and the factored form of a is then used to solve the system of equations.

> **See Also:**
>
> LAPACK Driver Routines (Linear Equations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_PTSV (
   n      IN       POSITIVEN,
   nrhs   IN       POSITIVEN,
   d      IN OUT   UTL_NLA_ARRAY_DBL,
   e      IN OUT   UTL_NLA_ARRAY_DBL,
   b      IN OUT   UTL_NLA_ARRAY_DBL,
   ldb    IN       POSITIVEN,
   info   OUT      INTEGER,
   pack   IN       flag  DEFAULT 'C');


UTL_NLA.LAPACK_PTSV (
   n      IN       POSITIVEN,
   nrhs   IN       POSITIVEN,
   d      IN OUT   UTL_NLA_ARRAY_FLT,
   e      IN OUT   UTL_NLA_ARRAY_FLT,
   b      IN OUT   UTL_NLA_ARRAY_FLT,
   ldb    IN       POSITIVEN,
   info   OUT      INTEGER,
   pack   IN       flag DEFAULT 'C');
```

Parameters

Table 297-50 LAPACK_PTSV Procedure Parameters

Parameter	Description
`n`	The order of the matrix `a`. `N >= 0`.
`nrhs`	The number of right-hand sides, which is the number of columns of the matrix `b`. `nrhs >= 0`.
`d`	`UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n)`. On entry, the `n` diagonal elements of the tridiagonal matrix `a`. On exit, the `n` diagonal elements of the diagonal matrix `d` from the factorization `A = L*D*L**T`.

Table 297-50 (Cont.) LAPACK_PTSV Procedure Parameters

Parameter	Description
e	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n-1). On entry, the (n-1) subdiagonal elements of the tridiagonal matrix a. On exit, the (n-1) diagonal elements of the unit bidiagonal factor L from the factorization A = L*D*L**T of a.(e can also be regarded as the superdiagonal of the unit bidiagonal factor U from the U**T*D*U factorization of a)
b	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldb, nrhs). On entry, the n by nrhs matrix of right hand side matrix b. On exit, if info = 0 , the n by nrhs solution matrix X.
ldb	The leading dimension of the array b. ldb >= max(1,n)
info	• = 0 : successful exit • < 0 : if info = -i , the i-th argument had an illegal value • > 0 : if info = i, the leading minor of order i of a is not positive definite, so the factorization could not be completed, and the solution has not been computed.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_SBEV Procedures

This procedure computes all the eigenvalues and, optionally, eigenvectors of a real symmetric band matrix A.

> **See Also:**
>
> LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_SBEV (
   jobz     IN      flag,
   uplo     IN      flag,
   n        IN      POSITIVEN,
   kd       IN      NATURALN,
   ab       IN OUT  UTL_NLA_ARRAY_DBL,
   ldab     IN      POSITIVEN,
   w        IN OUT  UTL_NLA_ARRAY_DBL,
   z        IN OUT  UTL_NLA_ARRAY_DBL,
   ldz       IN      POSITIVEN,
   info     OUT     INTEGER,
   pack     IN      flag DEFAULT 'C');

UTL_NLA.LAPACK_SBEV (
   jobz     IN      flag,
```

```
  uplo     IN      flag,
  n        IN      POSITIVEN,
  kd       IN      NATURALN,
  ab       IN OUT  UTL_NLA_ARRAY_FLT,
  ldab     IN      POSITIVEN,
  w        IN OUT  UTL_NLA_ARRAY_FLT,
  z        IN OUT  UTL_NLA_ARRAY_FLT,
  ldz      IN      POSITIVEN,
  info     OUT     INTEGER,
  pack     IN      flag DEFAULT 'C');
```

Parameters

Table 297-51 LAPACK_SBEV Procedure Parameters

Parameter	Description
jobz	• 'N': Compute eigenvalues only. • 'V': Compute eigenvalues and eigenvectors.
uplo	• 'U': Upper triangle of A is stored. • 'L': Lower triangle of A is stored.
n	The order of the matrix a. N >= 0.
kd	The number of superdiagonals of the matrix A if uplo = 'U', or the number of subdiagonals if uplo = 'L'. kd >= 0.
ab	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldab, n). On entry, the upper or lower triangle of the symmetric band matrix A stored in the first kd+1 rows of the array. The j-th column of A is stored in the j-th column of the array ab: • If uplo = 'U', ab(kd+1+i-j,j) = a(i,j) for max(1,j-kd)<=i<=j. • If uplo = 'L', AB(1+i-j,j) = A(i,j) for j<=i<=min(n,j+kd). On exit, ab is overwritten by values generated during the reduction to tridiagonal form: • If uplo = 'U', the diagonal and first superdiagonal of the tridiagonal matrix T are returned in rows kd and kd+1 of ab. • If uplo = 'L', the diagonal and first subdiagonal of T are returned in the first two rows of ab.
ldab	The leading dimension of the array ab. ldab >= kd + 1.
w	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n). If info = 0, the eigenvalues in ascending order.
z	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n). • If jobz = 'V', then if info = 0, z contains the orthonormal eigenvectors of the matrix A, with the i-th column of z holding the eigenvector associated with w(i). • If jobz = 'N', then z is not referenced.
ldz	The leading dimension of the array z. ldz >= 1, and if jobz = 'v', ldz >= max(1,n).
info	• = 0 : successful exit • < 0 : if info = -i, the i-th argument had an illegal value • > 0 : if info = i, the algorithm failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero

Table 297-51 (Cont.) LAPACK_SBEV Procedure Parameters

Parameter	Description
`pack`	(Optional) Flags the packing of the matrices: • `'C'`: column-major (default) • `'R'`: row-major

LAPACK_SBEVD Procedures

This procedure computes all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix `A`. If eigenvectors are desired, it uses a divide and conquer algorithm that makes mild assumptions about floating point arithmetic.

> **See Also:**
>
> LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_SBEVD (
   jobz     IN      flag,
   uplo     IN      flag,
   n        IN      POSITIVEN,
   kd       IN      NATURALN,
   ab       IN OUT  UTL_NLA_ARRAY_DBL,
   ldab     IN      POSITIVEN,
   w        IN OUT  UTL_NLA_ARRAY_DBL,
   z        IN OUT  UTL_NLA_ARRAY_DBL,
   ldz       IN      POSITIVEN,
   info     OUT     INTEGER,
   pack     IN      flag DEFAULT 'C');

UTL_NLA.LAPACK_SBEVD (
   jobz     IN      flag,
   uplo     IN      flag,
   n        IN      POSITIVEN,
   kd       IN      NATURALN,
   ab       IN OUT  UTL_NLA_ARRAY_FLT,
   ldab     IN      POSITIVEN,
   w        IN OUT  UTL_NLA_ARRAY_FLT,
   z        IN OUT  UTL_NLA_ARRAY_FLT,
   ldz      IN      POSITIVEN,
   info     OUT     INTEGER,
   pack     IN      flag DEFAULT 'C');
```

Parameters

Table 297-52 LAPACK_SBEVD Procedure Parameters

Parameter	Description
jobz	• 'N': Compute eigenvalues only. • 'V': Compute eigenvalues and eigenvectors.
uplo	• 'U': Upper triangle of A is stored. • 'L': Lower triangle of A is stored.
n	The order of the matrix a. N >= 0.
kd	The number of superdiagonals of the matrix A if uplo = 'U', or the number of subdiagonals if uplo = 'L'. kd >= 0.
ab	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldab, n). On entry, the upper or lower triangle of the symmetric band matrix A stored in the first kd+1 rows of the array. The j-th column of A is stored in the j-th column of the array ab: • If uplo = 'U', ab(kd+1+i-j,j) = a(i,j) for max(1,j-kd)<=i<=j. • If uplo = 'L', AB(1+i-j,j) = A(i,j) for j<=i<=min(n,j+kd). On exit, ab is overwritten by values generated during the reduction to tridiagonal form: • If uplo = 'U', the diagonal and first superdiagonal of the tridiagonal matrix T are returned in rows kd and kd+1 of ab. • If uplo = 'L', the diagonal and first subdiagonal of T are returned in the first two rows of ab.
ldab	The leading dimension of the array ab. ldab >= kd + 1.
w	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldz,n). If info = 0, the eigenvalues in ascending order.
z	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n). • If jobz = 'V', then if info = 0, z contains the orthonormal eigenvectors of the matrix A, with the i-th column of z holding the eigenvector associated with w(i). • If jobz = 'N', then z is not referenced.
ldz	The leading dimension of the array z. ldz >= 1, and if jobz = 'v', ldz >= max(1,n).
info	• = 0 : successful exit • < 0 : if info = -i, the i-th argument had an illegal value • > 0 : if info = i, the algorithm failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_SPEV Procedures

This procedure computes all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix `A` in packed storage.

> **See Also:**
>
> LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_SPEV (
   jobz     IN      flag,
   uplo     IN      flag,
   n        IN      POSITIVEN,
   ap       IN OUT  UTL_NLA_ARRAY_DBL,
   w        IN OUT  UTL_NLA_ARRAY_DBL,
   z        IN OUT  UTL_NLA_ARRAY_DBL,
   ldz      IN      POSITIVEN,
   info     OUT     INTEGER,
   pack     IN      flag DEFAULT 'C');

UTL_NLA.LAPACK_SPEV (
   jobz     IN      flag,
   uplo     IN      flag,
   n        IN      POSITIVEN,
   ap       IN OUT  UTL_NLA_ARRAY_FLT,
   w        IN OUT  UTL_NLA_ARRAY_FLT,
   z        IN OUT  UTL_NLA_ARRAY_FLT,
   ldz      IN      POSITIVEN,
   info     OUT     INTEGER,
   pack     IN      flag DEFAULT 'C');
```

Parameters

Table 297-53 LAPACK_SPEV Procedure Parameters

Parameter	Description
`jobz`	• `'N'`: Compute eigenvalues only. • `'V'`: Compute eigenvalues and eigenvectors.
`uplo`	• `'U'`: Upper triangle of `A` is stored. • `'L'`: Lower triangle of A is stored.
`n`	The order of the matrix `a. N >= 0`.

Table 297-53 (Cont.) LAPACK_SPEV Procedure Parameters

Parameter	Description
ap	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n*(n+1)/2). On entry, the upper or lower triangle of the symmetric matrix a packed columnwise in a linear array. The j-th column of a is stored in the array ap: • If uplo = 'U', ap(i + (j-1)*j/2) = a(i,j) for 1<=i<=j. • If uplo = 'L', ap(i + (j-1)*(2*n-j)/2) = a(i,j) for j<=i<=n. On exit, ap is overwritten by values generated during the reduction to tridiagonal form: • If uplo = 'U', the diagonal and first superdiagonal of the tridiagonal matrix T overwrite the corresponding elements of A. • If uplo = 'L', the diagonal and first subdiagonal of T overwrite the corresponding elements of A.
w	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n). If info = 0, the eigenvalues in ascending order.
z	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldz,n). • If jobz = 'V', then if info = 0, z contains the orthonormal eigenvectors of the matrix A, with the i-th column of z holding the eigenvector associated with w(i). • If jobz = 'N', then z is not referenced.
ldz	The leading dimension of the array z. ldz >= 1, and if jobz = 'v', ldz >= max(1,n).
info	• = 0 : successful exit • < 0 : if info = -i, the i-th argument had an illegal value • > 0 : if info = i, the algorithm failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_SPEVD Procedures

This procedure computes all the eigenvalues and, optionally, eigenvectors of a real symmetric matrix A in packed storage. If eigenvectors are desired, it uses a divide and conquer algorithm. The divide and conquer algorithm makes very mild assumptions about floating point arithmetic.

> **See Also:**
>
> LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_SPEVD (
   jobz     IN      flag,
   uplo     IN      flag,
   n        IN      POSITIVEN,
   ap       IN OUT  UTL_NLA_ARRAY_DBL,
   w        IN OUT  UTL_NLA_ARRAY_DBL,
   z        IN OUT  UTL_NLA_ARRAY_DBL,
   ldz       IN      POSITIVEN,
   info     OUT     INTEGER,
   pack     IN      flag DEFAULT 'C');

UTL_NLA.LAPACK_SPEVD (
   jobz     IN      flag,
   uplo     IN      flag,
   n        IN      POSITIVEN,
   ap       IN OUT  UTL_NLA_ARRAY_FLT,
   w        IN OUT  UTL_NLA_ARRAY_FLT,
   z        IN OUT  UTL_NLA_ARRAY_FLT,
   ldz      IN      POSITIVEN,
   info     OUT     INTEGER,
   pack     IN      flag DEFAULT 'C');
```

Parameters

Table 297-54 LAPACK_SPEVD Procedure Parameters

Parameter	Description
jobz	• 'N': Compute eigenvalues only. • 'V': Compute eigenvalues and eigenvectors.
uplo	• 'U': Upper triangle of A is stored. • 'L': Lower triangle of A is stored.
n	The order of the matrix a. N >= 0.
ap	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n*(n+1)/2). On entry, the upper or lower triangle of the symmetric matrix a packed columnwise in a linear array. The j-th column of a is stored in the array ap: • If uplo = 'U', ap(i + (j-1)*j/2) = a(i,j) for 1<=i<=j. • If uplo = 'L', ap(i + (j-1)*(2*n-j)/2) = a(i,j) for j<=i<=n. On exit, ap is overwritten by values generated during the reduction to tridiagonal form: • If uplo = 'U', the diagonal and first superdiagonal of the tridiagonal matrix T overwrite the corresponding elements of A. • If uplo = 'L', the diagonal and first subdiagonal of T overwrite the corresponding elements of A.
w	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n). If info = 0, the eigenvalues in ascending order.

Table 297-54 (Cont.) LAPACK_SPEVD Procedure Parameters

Parameter	Description
z	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldz,n). • If jobz = 'V', then if info = 0, z contains the orthonormal eigenvectors of the matrix A, with the i-th column of z holding the eigenvector associated with w(i). • If jobz = 'N', then z is not referenced.
ldz	The leading dimension of the array z. ldz >= 1, and if jobz = 'v', ldz >= max(1,n).
info	• = 0 : successful exit • < 0 : if info = -i, the i-th argument had an illegal value • > 0 : if info = i, the algorithm failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_SPSV Procedures

This procedure computes the solution to a real system of linear equations a * x = b, where a is an n by n symmetric matrix stored in packed format, and x and b are n by nrhs matrices.

The diagonal pivoting method is used to factor A as

A = U * D * U**T, if UPLO = 'U'

or

A = L * D * L**T, if UPLO = 'L'

where U (or L) is a product of permutation and unit upper (lower) triangular matrices, and D is symmetric and block diagonal with 1 by 1 and 2 by 2 diagonal blocks. The factored form of A is then used to solve the system of equations A * X = B.

> **See Also:**
>
> LAPACK Driver Routines (Linear Equations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_SPSV (
   uplo    IN      flag,
   n       IN      POSITIVEN,
   nrhs    IN      POSITIVEN,
   ap      IN OUT  UTL_NLA_ARRAY_DBL,
   ipiv    IN OUT  UTL_NLA_ARRAY_INT,
   b       IN OUT  UTL_NLA_ARRAY_DBL,
   ldb     IN      POSITIVEN,
```

```
   info    OUT     INTEGER,
   pack    IN      flag DEFAULT 'C');

UTL_NLA.LAPACK_SPSV (
   uplo    IN      flag,
   n       IN      POSITIVEN,
   nrhs    IN      POSITIVEN,
   ap      IN OUT  UTL_NLA_ARRAY_FLT,
   ipiv    IN OUT  UTL_NLA_ARRAY_INT,
   b       IN OUT  UTL_NLA_ARRAY_FLT,
   ldb     IN      POSITIVEN,
   info    OUT     INTEGER,
   pack    IN      flag DEFAULT 'C');
```

Parameters

Table 297-55 LAPACK_SPSV Procedure Parameters

Parameter	Description
uplo	• uplo = 'U'. Upper triangular of A is stored. • uplo = 'L' . Lower triangular of A is stored.
n	The number of linear equations, which is the order of the matrix a. N >= 0.
nrhs	The number of right-hand sides, which is the number of columns of the matrix b. nrhs >= 0.
ap	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n*(n+1)/2). On entry, the upper or lower triangle of the symmetric matrix A, packed columnwise in a linear array. The j-th column of A is stored in the array ap as follows: • uplo = 'U': AP(i + (j-1)*j/2) = A(i,j) for 1<=i<=j • uplo = 'L' : AP(i + (j-1)*(2n-j)/2) = A(i,j) for j<=i<=n See below for further details. On exit, the block diagonal matrix D and the multipliers used to obtain the factor U or L from the factorization A = U*D*U**T or A = L*D*L**T as computed by SSPTRF, stored as a packed triangular matrix in the same storage format as A.
ipiv	INTEGER array, DIMENSION (n). Details of the interchanges and the block structure of d, as determined by SSPTRF. • If ipiv(k) > 0, then rows and columns k and ipiv(k) were interchanged, and d(k,k) is a 1 by1 diagonal block. • If uplo = 'U' and ipiv(k) = ipiv(k-1) < 0, then rows and columns k-1 and -ipiv(k) were interchanged and d(k-1:k,k-1:k) is a 2 by 2 diagonal block. • If uplo = 'L' and ipiv(k) = ipiv(k+1) < 0, then rows and columns k+1 and -ipiv(k) were interchanged and d(k:k+1,k:k+1) is a 2 by 2 diagonal block.
b	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldb, nrhs). On entry, the n by nrhs right hand side matrix b. On exit, if info = 0 , the n by nrhs solution matrix X.
ldb	The leading dimension of the array b. ldb >= max(1,n)

Table 297-55 (Cont.) LAPACK_SPSV Procedure Parameters

Parameter	Description
info	• = 0 : successful exit • < 0 : if info = -i, the i-th argument had an illegal value • > 0 : if info = i, d(i,i) is exactly zero. The factorization has been completed, but the block diagonal matrix d is exactly singular, so the solution could not be computed.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_STEV Procedures

This procedure computes all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix A.

> **See Also:**
>
> LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_STEV (
   jobz     IN      flag,
   n        IN      POSITIVEN,
   d        IN OUT  UTL_NLA_ARRAY_DBL,
   e        IN OUT  UTL_NLA_ARRAY_DBL,
   z        IN OUT  UTL_NLA_ARRAY_DBL,
   ldz      IN      POSITIVEN,
   info     OUT     INTEGER,
   pack     IN      flag DEFAULT 'C');

UTL_NLA.LAPACK_STEV (
   jobz     IN      flag,
   n        IN      POSITIVEN,
   d        IN OUT  UTL_NLA_ARRAY_FLT,
   e        IN OUT  UTL_NLA_ARRAY_FLT,
   z        IN OUT  UTL_NLA_ARRAY_FLT,
   ldz      IN      POSITIVEN,
   info     OUT     INTEGER,
   pack     IN      flag DEFAULT 'C');
```

Parameters

Table 297-56 LAPACK_STEV Procedure Parameters

Parameter	Description
jobz	• 'N': Compute eigenvalues only. • 'V': Compute eigenvalues and eigenvectors.

Table 297-56 (Cont.) LAPACK_STEV Procedure Parameters

Parameter	Description
n	The order of the matrix a. N >= 0.
d	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n). • On entry, the n diagonal elements of the tridiagonal matrix A. • On exit, if info = 0, the eigenvalues in ascending order.
e	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n). • On entry, the (n-1) subdiagonal elements of the tridiagonal matrix A, stored in elements 1 to n-1 of e. e(n) need not be set, but is used by the subprogram. • On exit, the contents of e are destroyed.
z	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldz, n). • If jobz = 'V', then if info = 0, z contains the orthonormal eigenvectors of the matrix A, with the i-th column of z holding the eigenvector associated with d(i). • If jobz = 'N', then z is not referenced.
ldz	The leading dimension of the array z. ldz >= 1, and if jobz = 'v', ldz >= max(1,n).
info	• = 0 : successful exit • < 0 : if info = -i, the i-th argument had an illegal value • > 0 : if info = i, the algorithm failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_STEVD Procedures

This procedure computes all eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix. If eigenvectors are desired, it uses a divide and conquer algorithm that makes mild assumptions about floating point arithmetic.

> **See Also:**
>
> LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_STEVD (
   jobz     IN      flag,
   n        IN      POSITIVEN,
   d        IN OUT  UTL_NLA_ARRAY_DBL,
   e        IN OUT  UTL_NLA_ARRAY_DBL,
   z        IN OUT  UTL_NLA_ARRAY_DBL,
   ldz      IN      POSITIVEN,
```

```
   info     OUT     INTEGER,
   pack     IN      flag DEFAULT 'C');

UTL_NLA.LAPACK_STEVD(
   jobz     IN      flag,
   n        IN      POSITIVEN,
   d        IN OUT  UTL_NLA_ARRAY_FLT,
   e        IN OUT  UTL_NLA_ARRAY_FLT,
   z        IN OUT  UTL_NLA_ARRAY_FLT,
   ldz      IN      POSITIVEN,
   info     OUT     INTEGER,
   pack     IN      flag DEFAULT 'C');
```

Parameters

Table 297-57 LAPACK_STEVD Procedure Parameters

Parameter	Description
jobz	• 'N': Compute eigenvalues only. • 'V': Compute eigenvalues and eigenvectors.
n	The order of the matrix a. N >= 0.
d	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n). • On entry, the n diagonal elements of the tridiagonal matrix A. • On exit, if info = 0, the eigenvalues in ascending order.
e	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n). • On entry, the (n-1) subdiagonal elements of the tridiagonal matrix A, stored in elements 1 to n-1 of e. e(n) need not be set, but is used by the subprogram. • On exit, the contents of e are destroyed.
z	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldz, n). • If jobz = 'V', then if info = 0, z contains the orthonormal eigenvectors of the matrix A, with the i-th column of z holding the eigenvector associated with d(i). • If jobz = 'N', then z is not referenced.
ldz	The leading dimension of the array z. ldz >= 1, and if jobz = 'v', ldz >= max(1,n).
info	• = 0 : successful exit • < 0 : if info = -i, the i-th argument had an illegal value • > 0 : if info = i, the algorithm failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_SYEV Procedures

This procedure computes all eigenvalues and, optionally, eigenvectors of a real symmetric matrix `A`.

> **See Also:**
>
> LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_SYEV (
   jobz     IN      flag,
   uplo     IN      flag,
   n        IN      POSITIVEN,
   a        IN OUT  UTL_NLA_ARRAY_DBL,
   lda      IN      POSITIVEN,
   w        IN OUT  UTL_NLA_ARRAY_DBL,
   info     OUT     INTEGER,
   pack     IN      flag DEFAULT 'C');

UTL_NLA.LAPACK_SYEV (
   jobz     IN      flag,
   uplo     IN      flag,
   n        IN      POSITIVEN,
   a        IN OUT  UTL_NLA_ARRAY_FLT,
   lda      IN      POSITIVEN,
   w        IN OUT  UTL_NLA_ARRAY_FLT,
   info     OUT     INTEGER,
   pack     IN      flag DEFAULT 'C');
```

Parameters

Table 297-58 LAPACK_SYEV Procedure Parameters

Paramete	Description
`jobz`	• `'N'`: Compute eigenvalues only. • `'V'`: Compute eigenvalues and eigenvectors.
`uplo`	• `'U'`: Upper triangle of A is stored. • `'L'`: Upper triangle of A is stored.
`n`	The order of the matrix `a. N >= 0`.

Table 297-58 (Cont.) LAPACK_SYEV Procedure Parameters

Paramete	Description
a	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (lda, n). On entry, the symmetric matrix a: • If uplo = 'U', the leading n by n upper triangular part of a contains the upper triangular part of the matrix a. • If uplo = 'L', the leading n byn lower triangular part of a contains the lower triangular part of the matrix a. On exit: • If jobz = 'V', then if info = 0, a contains the orthonormal eigenvectors of the matrix a. • If jobz = 'N', then on exit the lower triangle (if uplo = 'L') or the upper triangle (if uplo='U') of a, including the diagonal, is destroyed.
lda	The leading dimension of the array a. lda >= max(1,n).
w	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n). If info = 0, the eigenvalues in ascending order.
info	• = 0 : successful exit • < 0 : if info = -i, the i-th argument had an illegal value • > 0 : if info = i, the algorithm failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_SYEVD Procedures

This procedure computes all eigenvalues and, optionally, eigenvectors of a real symmetric matrix A. If eigenvectors are desired, it uses a divide and conquer algorithm that makes mild assumptions about floating point arithmetic.

> **See Also:**
>
> LAPACK Driver Routines (LLS and Eigenvalue Problems) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_SYEVD (

   jobz     IN      flag,
   uplo     IN      flag,
   n        IN      POSITIVEN,
   a        IN OUT  UTL_NLA_ARRAY_DBL,
   lda      IN      POSITIVEN,
   w        IN OUT  UTL_NLA_ARRAY_DBL,
   info     OUT     INTEGER,
   pack     IN      flag DEFAULT 'C');
```

```
UTL_NLA.LAPACK_SYEVD (
   jobz     IN      flag,
   uplo     IN      flag,
   n        IN      POSITIVEN,
   a        IN OUT  UTL_NLA_ARRAY_FLT,
   lda      IN      POSITIVEN,
   w        IN OUT  UTL_NLA_ARRAY_FLT,
   info     OUT     INTEGER,
   pack     IN      flag DEFAULT 'C');
```

Parameters

Table 297-59 LAPACK_SYEVD Procedure Parameters

Parameter	Description
jobz	• 'N': Compute eigenvalues only. • 'V': Compute eigenvalues and eigenvectors.
uplo	• 'U' : Upper triangle of A is stored. • 'L': Upper triangle of A is stored.
n	The order of the matrix a. N >= 0.
a	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (lda, n). On entry, the symmetric matrix a: • If uplo = 'U', the leading n by n upper triangular part of a contains the upper triangular part of the matrix a. • If uplo = 'L', the leading n byn lower triangular part of a contains the lower triangular part of the matrix a. On exit: • If jobz = 'V', then if info = 0, a contains the orthonormal eigenvectors of the matrix a. • If jobz = 'N', then on exit the lower triangle (if uplo = 'L') or the upper triangle (if uplo = 'U') of a, including the diagonal, is destroyed.
lda	The leading dimension of the array a. lda >= max(1,n).
w	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n). If info = 0, the eigenvalues in ascending order.
info	• = 0 : successful exit • < 0 : if info = -i, the i-th argument had an illegal value • > 0 : if info = i, the algorithm failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major

LAPACK_SYSV Procedures

This procedure computes the solution to a real system of linear equations a * x = b, where a is an n by n symmetric matrix, and x and b are n by nrhs matrices.

The diagonal pivoting method is used to factor A as

`A = U * D * U**T`, if `UPLO = 'U'`

or

`A = L * D * L**T`, if `UPLO = 'L'`

where `U` (or `L`) is a product of permutation and unit upper (lower) triangular matrices, and `D` is symmetric and block diagonal with 1 by 1 and 2 by 2 diagonal blocks. The factored form of `A` is then used to solve the system of equations `A * X = B`.

> **See Also:**
>
> LAPACK Driver Routines (Linear Equations) Subprograms for other subprograms in this group

Syntax

```
UTL_NLA.LAPACK_SYSV (
   uplo     IN      flag,
   n        IN      POSITIVEN,
   nrhs     IN      POSITIVEN,
   a        IN OUT  UTL_NLA_ARRAY_DBL,
   lda      IN      POSITIVEN,
   ipiv     IN OUT  UTL_NLA_ARRAY_INT,
   b        IN OUT  UTL_NLA_ARRAY_DBL,
   ldb      IN      POSITIVEN,
   info     OUT     INTEGER,
   pack     IN      flag DEFAULT 'C');

UTL_NLA.LAPACK_SYSV (
   uplo    IN       flag,
   n       IN       POSITIVEN,
   nrhs    IN       POSITIVEN,
   a       IN OUT   UTL_NLA_ARRAY_FLT,
   lda     IN       POSITIVEN,
   ipiv    IN OUT   UTL_NLA_ARRAY_INT,
   b       IN OUT   UTL_NLA_ARRAY_FLT,
   ldb     IN       POSITIVEN,
   info    OUT      INTEGER,
   pack    IN       flag DEFAULT 'C');
```

Parameters

Table 297-60 LAPACK_SYSV Procedure Parameters

Parameter	Description
`uplo`	• `uplo = 'U'`. Upper triangular of `A` is stored. • `uplo = 'L'` . Lower triangular of `A` is stored.
`n`	The number of linear equations, which is the order of the matrix `a`. `N >= 0`.
`nrhs`	The number of right-hand sides, which is the number of columns of the matrix `b`. `nrhs >= 0`.

Table 297-60 (Cont.) LAPACK_SYSV Procedure Parameters

Parameter	Description
a	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (n-1). On entry, the symmetric matrix a. If UPLO = 'U', the leading n by n upper triangular part of a contains the upper triangular part of the matrix a, and the strictly lower triangular part of a is not referenced. If uplo = 'L', the leading n by n lower triangular part of a contains the lower triangular part of the matrix a, and the strictly upper triangular part of a is not referenced. On exit, if info = 0, the block diagonal matrix d and the multipliers used to obtain the factor U or L from the factorization A = U*D*U**T or A = L*D*L**T as computed by SSYTRF.
lda	The leading dimension of the array a. lda >= max(1,n)
ipiv	INTEGER array, DIMENSION (ldb, nrhs). Details of the interchanges and the block structure of d, as determined by SSYTRF. • If ipiv(k) > 0, then rows and columns k and ipiv(k) were interchanged, and d(k,k) is a 1 by1 diagonal block. • If uplo = 'U' and ipiv(k) = ipiv(k-1) < 0, then rows and columns k-1 and -ipiv(k) were interchanged and d(k-1:k,k-1:k) is a 2 by 2 diagonal block. • If uplo = 'L' and ipiv(k) = ipiv(k+1) < 0, then rows and columns k+1 and -ipiv(k) were interchanged and d(k:k+1,k:k+1) is a 2 by 2 diagonal block.
b	UTL_NLA_ARRAY_FLT/DBL, DIMENSION (ldb, nrhs). On entry, the n by nrhs matrix of right hand side matrix b. On exit, if info = 0 , the n by nrhs solution matrix X.
ldb	The leading dimension of the array b. ldb >= max(1,n)
info	• = 0 : successful exit • < 0 : if info = -i , the i-th argument had an illegal value • > 0 : if info = i, d(i,i) is exactly zero. The factorization has been completed, but the block diagonal matrix d is exactly singular, so the solution could not be computed.
pack	(Optional) Flags the packing of the matrices: • 'C': column-major (default) • 'R': row-major