**BLAS**

**技术手册**

**版本管理**

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

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

目录

[引言 1](#_Toc6998434)

[1. 编写目的 1](#_Toc6998435)

[2. 编写背景 1](#_Toc6998436)

[3. 概念定义 1](#_Toc6998437)

[4. 参考资料 1](#_Toc6998438)

[1 热力学 2](#_Toc6998439)

[2 空气动力学 2](#_Toc6998440)

# 引言

## 1. 编写目的

说明编写这份数据要求说明书的目的，指出预期的读者。

## 2. 编写背景

a.待开发软件系统的名称；

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

## 3. 概念定义

列出本文件中用到的专门术语的定义和外文首字母组词的原词组。

## 4. 参考资料

列出有关的参考资料。

<https://www.netlib.org/blas/>

<https://www.netlib.org/lapack/index.html>

<http://www.netlib.org/lapack/lawns/downloads/index.html>

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

https://www.mathkeisan.com/UsersGuide

https://www.mathkeisan.com/UsersGuide/cblas.cfm

https://www.mathkeisan.com/UsersGuide/lapack.cfm

# 1 介绍

####################################################################

#  This is the makefile to create a library for the BLAS.

#  The files are grouped as follows:

#

#       SBLAS1 -- Single precision real BLAS routines

#       CBLAS1 -- Single precision complex BLAS routines

#       DBLAS1 -- Double precision real BLAS routines

#       ZBLAS1 -- Double precision complex BLAS routines

#

#       CB1AUX -- Real BLAS routines called by complex routines

#       ZB1AUX -- D.P. real BLAS routines called by d.p. complex

#                 routines

#

#      ALLBLAS -- Auxiliary routines for Level 2 and 3 BLAS

#

#       SBLAS2 -- Single precision real BLAS2 routines

#       CBLAS2 -- Single precision complex BLAS2 routines

#       DBLAS2 -- Double precision real BLAS2 routines

#       ZBLAS2 -- Double precision complex BLAS2 routines

#

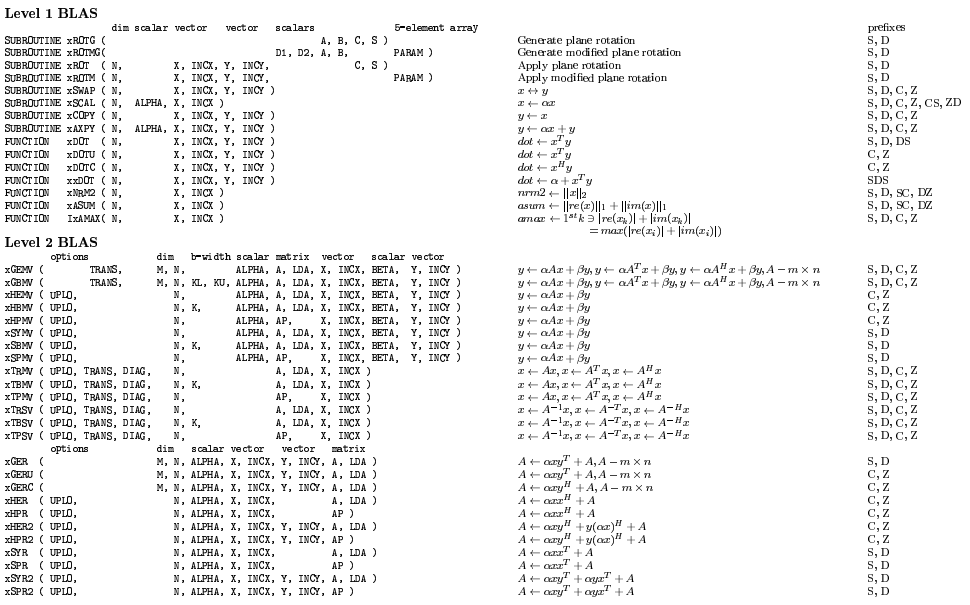
#       SBLAS3 -- Single precision real BLAS3 routines

#       CBLAS3 -- Single precision complex BLAS3 routines

#       DBLAS3 -- Double precision real BLAS3 routines

#       ZBLAS3 -- Double precision complex BLAS3 routines

#



# 2 BLAS Routine List

|  | Name† | Prefixes | Description |
| --- | --- | --- | --- |
| Level 1 BLAS | ?ROTG | S D C Z | Generate plane rotation |
| ?ROTMG | S D | Generate modified plane rotation |
| ?ROT | S D C Z | Apply plane rotation |
| ?ROTM | S D | Apply modified plane rotation |
| ?SWAP | S D C Z | Swap vectors |
| ?SCAL | S D C Z CS ZD | Scale vector |
| ?COPY | S D C Z | Copy vector |
| ?AXPY | S D C Z | Vector scale and add |
| ?DOT | S D SDS DS | Dot product, real |
| ?DOTU | C Z | Dot product, complex |
| ?DOTC | C Z | Dot product, complex, conjugate first vector |
| ?NRM2 | S D SC DZ | Euclidean norm |
| ?ASUM | S D SC DZ | Sum absolute values |
| I?AMAX | S D C Z | Index of maximum absolute value |
| Level 2 BLAS | ?GEMV | S D C Z | General matrix-vector multiplication |
| ?GBMV | S D C Z | General banded matrix-vector multiplication |
| ?HEMV | C Z | Hermitian matrix-vector multiplication |
| ?HBMV | C Z | Hermitian banded matrix-vector multiplication |
| ?HPMV | C Z | Hermitian packed matrix-vector multiplication |
| ?SYMV | S D | Symmetric matrix-vector multiplication |
| ?SBNV | S D | Symmetric banded matrix-vector multiplication |
| ?SPMV | S D | Symmetric packed matrix-vector multiplication |
| ?TRMV | S D C Z | Triangular matrix-vector multiplication |
| ?TBMV | S D C Z | Triangular banded matrix-vector multiplication |
| ?TPMV | S D C Z | Triangular packed matrix-vector multiplication |
| ?TRSV | S D C Z | Triangular solve |
| ?TBSV | S D C Z | Triangular banded solve |
| ?TPSV | S D C Z | Triangular packed solve |
| ?GER | S D | General rank-1 update, real |
| ?GERU | C Z | General rank-1 update, complex |
| ?GERC | C Z | General rank-1 update, complex, second vector conjugate |
| ?HER | C Z | Hermitian rank-1 update |
| ?HPR | C Z | Hermitian packed rank-1 update |
| ?HER2 | C Z | Hermitian rank-2 update |
| ?HPR2 | C Z | Hermitian packed rank-2 update |
| ?SYR | S D | Symmetric rank-1 update |
| ?SPR | S D | Symmetric packed rank-1 update |
| ?SYR2 | S D | Symmetric rank-2 update |
| ?SPR2 | S D | Symmetric packed rank-2 update |
| Level 3 BLAS | ?GEMM | S D C Z | General matrix-matrix multiplication |
| ?GEMM3M | C Z | General complex matrix-matrix multiplication (Performance Priority Version) |
| ?SYMM | S D C Z | Symmetric matrix-matrix multiplication |
| ?HEMM | C Z | Hermitian matrix-matrix multiplication |
| ?SYRK | S D C Z | Symmetric rank-k update |
| ?HERK | C Z | Hermitian rank-k update |
| ?SYR2K | S D C Z | Symmetric rank-2k update |
| ?HER2K | C Z | Hermitian rank-2k update |
| ?TRMM | S D C Z | Triangular matrix-matrix multiply |
| ?TRSM | S D C Z | Triangular solve |
| Auxiliary Subprograms | XERBLA | | Print argument error messages |
| MKVERSION | | Print library version information |
| MK\_GET\_VERSION | | Return library version number as an integer |
| MK\_GET\_VERSION\_DATE | | Return library version date as an integer |
| † ?  indicates prefix which must be filled with a combination of: S = REAL(kind=4), D = REAL(kind=8), C = COMPLEX(kind=4), Z = COMPLEX(kind=8) | | | |

# 3 CBLAS Routine List

Only routine names are listed. For descriptions, look in the corresponding [BLAS Routine List](https://www.mathkeisan.com/UsersGuide/blas.cfm#RoutList).

|  | Routines | | | |
| --- | --- | --- | --- | --- |
|  | float | double | float complex | double complex |
| Level 1 BLAS | cblas\_sdot | cblas\_ddot |  |  |
| cblas\_sdsdot | cblas\_dsdot |  |  |
|  |  | cblas\_cdotu\_sub | cblas\_zdotu\_sub |
|  |  | cblas\_cdotc\_sub | cblas\_zdotc\_sub |
| cblas\_snrm2 | cblas\_dnrm2 | cblas\_scnrm2 | cblas\_dznrm2 |
| cblas\_sasum | cblas\_dasum | cblas\_scasum | cblas\_dzasum |
| cblas\_isamax | cblas\_idamax | cblas\_icamax | cblas\_izamax |
| cblas\_sswap | cblas\_dswap | cblas\_cswap | cblas\_zswap |
| cblas\_saxpy | cblas\_daxpy | cblas\_caxpy | cblas\_zaxpy |
| cblas\_scopy | cblas\_dcopy | cblas\_ccopy | cblas\_zcopy |
| cblas\_srot | cblas\_drot |  |  |
| cblas\_srotg | cblas\_drotg |  |  |
| cblas\_srotm | cblas\_drotm |  |  |
| cblas\_srotmg | cblas\_drotmg |  |  |
| cblas\_sscal | cblas\_dscal | cblas\_cscal | cblas\_zscal |
|  |  | cblas\_csscal | cblas\_zdscal |
| Level 2 BLAS | cblas\_sgemv | cblas\_dgemv | cblas\_cgemv | cblas\_zgemv |
| cblas\_sgbmv | cblas\_dgbmv | cblas\_cgbmv | cblas\_zgbmv |
| cblas\_strmv | cblas\_dtrmv | cblas\_ctrmv | cblas\_ztrmv |
| cblas\_stbmv | cblas\_dtbmv | cblas\_ctbmv | cblas\_ztbmv |
| cblas\_stpmv | cblas\_dtpmv | cblas\_ctpmv | cblas\_ztpmv |
| cblas\_strsv | cblas\_dtrsv | cblas\_ctrsv | cblas\_ztrsv |
| cblas\_stbsv | cblas\_dtbsv | cblas\_ctbsv | cblas\_ztbsv |
| cblas\_stpsv | cblas\_dtpsv | cblas\_ctpsv | cblas\_ztpsv |
| cblas\_ssymv | cblas\_dsymv | cblas\_chemv | cblas\_zhemv |
| cblas\_ssbmv | cblas\_dsbmv | cblas\_chbmv | cblas\_zhbmv |
| cblas\_sspmv | cblas\_dspmv | cblas\_chpmv | cblas\_zhpmv |
| cblas\_sger | cblas\_dger |  |  |
| cblas\_cgeru | cblas\_zgeru |  |  |
| cblas\_cgerc | cblas\_zgerc |  |  |
| cblas\_ssyr | cblas\_dsyr |  |  |
| cblas\_sspr | cblas\_dspr |  |  |
| cblas\_ssyr2 | cblas\_dsyr2 |  |  |
| cblas\_sspr2 | cblas\_dspr2 |  |  |
|  |  | cblas\_cher | cblas\_zher |
|  |  | cblas\_chpr | cblas\_zhpr |
|  |  | cblas\_cher2 | cblas\_zher2 |
|  |  | cblas\_chpr2 | cblas\_zhpr2 |
| Level 3 BLAS | cblas\_sgemm | cblas\_dgemm | cblas\_cgemm | cblas\_zgemm |
| cblas\_ssymm | cblas\_dsymm | cblas\_csymm | cblas\_zsymm |
| cblas\_ssyrk | cblas\_dsyrk | cblas\_csyrk | cblas\_zsyrk |
| cblas\_ssyr2k | cblas\_dsyr2k | cblas\_csyr2k | cblas\_zsyr2k |
| cblas\_strmm | cblas\_dtrmm | cblas\_ctrmm | cblas\_ztrmm |
| cblas\_strsm | cblas\_dtrsm | cblas\_ctrsm | cblas\_ztrsm |
|  |  | cblas\_chemm | cblas\_zhemm |
|  |  | cblas\_cherk | cblas\_zherk |
|  |  | cblas\_cher2k | cblas\_zher2k |
| Auxiliary Functions | cblas\_xerbla | | | |
| mkversion\_ | | | |
| mk\_get\_version\_ | | | |
| mk\_get\_version\_date\_ | | | |

# 2 函数 - DOUBLE

## DASUM

\brief \b DASUM

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       DOUBLE PRECISION FUNCTION DASUM(N,DX,INCX)

       .. Scalar Arguments ..

       INTEGER INCX,N

       ..

       .. Array Arguments ..

       DOUBLE PRECISION DX()

       ..

Purpose:

=============

    DASUM takes the sum of the absolute values.

Arguments:

==========

[in] N

     N is INTEGER,

     Number of elements in input vector(s)

[in] DX

      DX is DOUBLE PRECISION array,

dimension ( 1 + ( N - 1 )abs( INCX ) )

[in] INCX

          INCX is INTEGER,

         storage spacing between elements of DX

Group double\_blas\_level1

## DAXPY

\brief \b DAXPY

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DAXPY(N,DA,DX,INCX,DY,INCY)

       .. Scalar Arguments ..

       DOUBLE PRECISION DA

       INTEGER INCX,INCY,N

       ..

       .. Array Arguments ..

      DOUBLE PRECISION DX(),DY()

       ..

Purpose:

=============

常数乘以一个向量加上一个向量。

    uses unrolled loops for increments equal to one.

Arguments:

==========

[in] N

     N is INTEGER

     number of elements in input vector(s)

[in] DA

     DA is DOUBLE PRECISION

     On entry, DA specifies the scalar alpha.

[in] DX

     DX is DOUBLE PRECISION array, dimension ( 1 + ( N - 1 )abs( INCX ) )

[in] INCX

     INCX is INTEGER

     storage spacing between elements of DX

[in,out] DY

         DY is DOUBLE PRECISION array,

dimension ( 1 + ( N - 1 )abs( INCY ) )

[in] INCY

          INCY is INTEGER

         storage spacing between elements of DY

Group: double\_blas\_level1

## DCABS1

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       DOUBLE PRECISION FUNCTION DCABS1(Z)

       .. Scalar Arguments ..

       COMPLEX16 Z

       ..

Purpose:

=============

DCABS1 computes |Re(.)| + |Im(.)| of a double complex number

计算复数 实部绝对值 + 虚部实系数绝对值

Arguments:

==========

[in] Z

          Z is COMPLEX16

Group: double\_blas\_level1

## DCOPY

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DCOPY(N,DX,INCX,DY,INCY)

       .. Scalar Arguments ..

       INTEGER INCX,INCY,N

       ..

       .. Array Arguments ..

       DOUBLE PRECISION DX(),DY()

       ..

Purpose:

=============

    DCOPY copies a vector, x, to a vector, y.

    uses unrolled loops for increments equal to 1.

Arguments:

==========

[in] N

          N is INTEGER

         number of elements in input vector(s)

[in] DX

          DX is DOUBLE PRECISION array, dimension ( 1 + ( N - 1 )abs( INCX ) )

[in] INCX

          INCX is INTEGER

         storage spacing between elements of DX

[out] DY

          DY is DOUBLE PRECISION array, dimension ( 1 + ( N - 1 )abs( INCY ) )

[in] INCY

          INCY is INTEGER

         storage spacing between elements of DY

Group: double\_blas\_level1

## DDOT

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       DOUBLE PRECISION FUNCTION DDOT(N,DX,INCX,DY,INCY)

       .. Scalar Arguments ..

       INTEGER INCX,INCY,N

       ..

       .. Array Arguments ..

       DOUBLE PRECISION DX(),DY()

       ..

Purpose:

=============

    DDOT forms the dot product of two vectors.

    uses unrolled loops for increments equal to one.

Arguments:

==========

[in] N

          N is INTEGER

         number of elements in input vector(s)

[in] DX

          DX is DOUBLE PRECISION array, dimension ( 1 + ( N - 1 )abs( INCX ) )

[in] INCX

          INCX is INTEGER

         storage spacing between elements of DX

[in] DY

          DY is DOUBLE PRECISION array, dimension ( 1 + ( N - 1 )abs( INCY ) )

[in] INCY

          INCY is INTEGER

         storage spacing between elements of DY

Group: double\_blas\_level1

## DGBMV

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DGBMV(TRANS,M,N,KL,KU,ALPHA,A,LDA,X,INCX,BETA,Y,INCY)

       .. Scalar Arguments ..

       DOUBLE PRECISION ALPHA,BETA

       INTEGER INCX,INCY,KL,KU,LDA,M,N

       CHARACTER TRANS

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),X(),Y()

       ..

Purpose:

=============

 DGBMV  performs one of the matrix-vector operations

    y := alphaAx + betay,   or   y := alphaATx + betay,

 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.

Arguments:

==========

[in] TRANS

          TRANS is CHARACTER1

           On entry, TRANS specifies the operation to be performed as

           follows:

              TRANS = 'N' or 'n'   y := alphaAx + betay.

              TRANS = 'T' or 't'   y := alphaATx + betay.

              TRANS = 'C' or 'c'   y := alphaATx + betay.

[in] M

          M is INTEGER

           On entry, M specifies the number of rows of the matrix A.

           M must be at least zero.

[in] N

          N is INTEGER

           On entry, N specifies the number of columns of the matrix A.

           N must be at least zero.

[in] KL

          KL is INTEGER

           On entry, KL specifies the number of sub-diagonals of the

           matrix A. KL must satisfy  0 .le. KL.

[in] KU

          KU is INTEGER

           On entry, KU specifies the number of super-diagonals of the

           matrix A. KU must satisfy  0 .le. KU.

[in] ALPHA

          ALPHA is DOUBLE PRECISION.

           On entry, ALPHA specifies the scalar alpha.

[in] A

          A is DOUBLE PRECISION array, 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.

           The following program segment will transfer a band matrix

           from conventional full matrix storage to band storage:

                 DO 20, J = 1, N

                    K = KU + 1 - J

                    DO 10, I = MAX( 1, J - KU ), MIN( M, J + KL )

                       A( K + I, J ) = matrix( I, J )

              10    CONTINUE

              20 CONTINUE

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in the calling (sub) program. LDA must be at least

           ( kl + ku + 1 ).

[in] X

          X is DOUBLE PRECISION array, 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.

[in] INCX

          INCX is INTEGER

           On entry, INCX specifies the increment for the elements of

           X. INCX must not be zero.

[in] BETA

          BETA is DOUBLE PRECISION.

           On entry, BETA specifies the scalar beta. When BETA is

           supplied as zero then Y need not be set on input.

[in,out] Y

          Y is DOUBLE PRECISION array, dimension at least

           ( 1 + ( m - 1 )abs( INCY ) ) when TRANS = 'N' or 'n'

           and at least

           ( 1 + ( n - 1 )abs( INCY ) ) otherwise.

           Before entry, the incremented array Y must contain the

           vector y. On exit, Y is overwritten by the updated vector y.

[in] INCY

          INCY is INTEGER

           On entry, INCY specifies the increment for the elements of

           Y. INCY must not be zero.

Group: double\_blas\_level2

## DGEMM

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DGEMM(TRANSA,TRANSB,M,N,K,ALPHA,A,LDA,B,LDB,BETA,C,LDC)

       .. Scalar Arguments ..

       DOUBLE PRECISION ALPHA,BETA

       INTEGER K,LDA,LDB,LDC,M,N

       CHARACTER TRANSA,TRANSB

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),B(LDB,),C(LDC,)

       ..

Purpose:

=============

 DGEMM  performs one of the matrix-matrix operations

    C := alphaop( A )op( B ) + betaC,

 where  op( X ) is one of

    op( X ) = X   or   op( X ) = XT,

 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.

Arguments:

==========

[in] TRANSA

          TRANSA is CHARACTER1

           On entry, 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 ) = AT.

              TRANSA = 'C' or 'c',  op( A ) = AT.

[in] TRANSB

          TRANSB is CHARACTER1

           On entry, 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 ) = BT.

              TRANSB = 'C' or 'c',  op( B ) = BT.

[in] M

          M is INTEGER

           On entry,  M  specifies  the number  of rows  of the  matrix

           op( A )  and of the  matrix  C.  M  must  be at least  zero.

[in] N

          N is INTEGER

           On entry,  N  specifies the number  of columns of the matrix

           op( B ) and the number of columns of the matrix C. N must be

           at least zero.

[in] K

          K is INTEGER

           On entry,  K  specifies  the number of columns of the matrix

           op( A ) and the number of rows of the matrix op( B ). K must

           be at least  zero.

[in] ALPHA

          ALPHA is DOUBLE PRECISION.

           On entry, ALPHA specifies the scalar alpha.

[in] A

          A is DOUBLE PRECISION array, 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.

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in the calling (sub) program. When  TRANSA = 'N' or 'n' then

           LDA must be at least  max( 1, m ), otherwise  LDA must be at

           least  max( 1, k ).

[in] B

          B is DOUBLE PRECISION array, dimension ( LDB, 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 array  B  must contain  the

           matrix B.

[in] LDB

          LDB is INTEGER

           On entry, LDB specifies the first dimension of B as declared

           in the calling (sub) program. When  TRANSB = 'N' or 'n' then

           LDB must be at least  max( 1, k ), otherwise  LDB must be at

           least  max( 1, n ).

[in] BETA

          BETA is DOUBLE PRECISION.

           On entry,  BETA  specifies the scalar  beta.  When  BETA  is

           supplied as zero then C need not be set on input.

[in,out] C

          C is DOUBLE PRECISION array, 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  matrix

           ( alphaop( A )op( B ) + betaC ).

[in] LDC

          LDC is INTEGER

           On entry, LDC specifies the first dimension of C as declared

           in  the  calling  (sub)  program.   LDC  must  be  at  least

           max( 1, m ).

Group: double\_blas\_level3

## DGEMV

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DGEMV(TRANS,M,N,ALPHA,A,LDA,X,INCX,BETA,Y,INCY)

       .. Scalar Arguments ..

       DOUBLE PRECISION ALPHA,BETA

       INTEGER INCX,INCY,LDA,M,N

       CHARACTER TRANS

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),X(),Y()

       ..

Purpose:

=============

 DGEMV  performs one of the matrix-vector operations

    y := alphaAx + betay,   or   y := alphaATx + betay,

 where alpha and beta are scalars, x and y are vectors and A is an

 m by n matrix.

Arguments:

==========

[in] TRANS

          TRANS is CHARACTER1

           On entry, TRANS specifies the operation to be performed as

           follows:

              TRANS = 'N' or 'n'   y := alphaAx + betay.

              TRANS = 'T' or 't'   y := alphaATx + betay.

              TRANS = 'C' or 'c'   y := alphaATx + betay.

[in] M

          M is INTEGER

           On entry, M specifies the number of rows of the matrix A.

           M must be at least zero.

[in] N

          N is INTEGER

           On entry, N specifies the number of columns of the matrix A.

           N must be at least zero.

[in] ALPHA

          ALPHA is DOUBLE PRECISION.

           On entry, ALPHA specifies the scalar alpha.

[in] A

          A is DOUBLE PRECISION array, dimension ( LDA, N )

           Before entry, the leading m by n part of the array A must

           contain the matrix of coefficients.

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in the calling (sub) program. LDA must be at least

           max( 1, m ).

[in] X

          X is DOUBLE PRECISION array, 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.

[in] INCX

          INCX is INTEGER

           On entry, INCX specifies the increment for the elements of

           X. INCX must not be zero.

[in] BETA

          BETA is DOUBLE PRECISION.

           On entry, BETA specifies the scalar beta. When BETA is

           supplied as zero then Y need not be set on input.

[in,out] Y

          Y is DOUBLE PRECISION array, 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 non-zero, the incremented array Y

           must contain the vector y. On exit, Y is overwritten by the

           updated vector y.

[in] INCY

          INCY is INTEGER

           On entry, INCY specifies the increment for the elements of

           Y. INCY must not be zero.

Group: double\_blas\_level2

## DGER

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DGER(M,N,ALPHA,X,INCX,Y,INCY,A,LDA)

       .. Scalar Arguments ..

       DOUBLE PRECISION ALPHA

       INTEGER INCX,INCY,LDA,M,N

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),X(),Y()

       ..

Purpose:

=============

 DGER   performs the rank 1 operation

    A := alphaxyT + 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.

Arguments:

==========

[in] M

          M is INTEGER

           On entry, M specifies the number of rows of the matrix A.

           M must be at least zero.

[in] N

          N is INTEGER

           On entry, N specifies the number of columns of the matrix A.

           N must be at least zero.

[in] ALPHA

          ALPHA is DOUBLE PRECISION.

           On entry, ALPHA specifies the scalar alpha.

[in] X

          X is DOUBLE PRECISION array, dimension at least

           ( 1 + ( m - 1 )abs( INCX ) ).

           Before entry, the incremented array X must contain the m

           element vector x.

[in] INCX

          INCX is INTEGER

           On entry, INCX specifies the increment for the elements of

           X. INCX must not be zero.

[in] Y

          Y is DOUBLE PRECISION array, dimension at least

           ( 1 + ( n - 1 )abs( INCY ) ).

           Before entry, the incremented array Y must contain the n

           element vector y.

[in] INCY

          INCY is INTEGER

           On entry, INCY specifies the increment for the elements of

           Y. INCY must not be zero.

[in,out] A

          A is DOUBLE PRECISION array, 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.

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in the calling (sub) program. LDA must be at least

           max( 1, m ).

Group: double\_blas\_level2

## DNRM2

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       DOUBLE PRECISION FUNCTION DNRM2(N,X,INCX)

       .. Scalar Arguments ..

       INTEGER INCX,N

       ..

       .. Array Arguments ..

       DOUBLE PRECISION X()

       ..

Purpose:

=============

 DNRM2 returns the euclidean norm of a vector via the function

 name, so that

    DNRM2 := sqrt( x'x )

Arguments:

==========

[in] N

          N is INTEGER

         number of elements in input vector(s)

[in] X

          X is DOUBLE PRECISION array, dimension ( 1 + ( N - 1 )abs( INCX ) )

[in] INCX

          INCX is INTEGER

         storage spacing between elements of DX

Group: double\_blas\_level1

## DROT

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DROT(N,DX,INCX,DY,INCY,C,S)

       .. Scalar Arguments ..

       DOUBLE PRECISION C,S

       INTEGER INCX,INCY,N

       ..

       .. Array Arguments ..

       DOUBLE PRECISION DX(),DY()

       ..

Purpose:

=============

    DROT applies a plane rotation.

Arguments:

==========

[in] N

          N is INTEGER

         number of elements in input vector(s)

[in,out] DX

          DX is DOUBLE PRECISION array, dimension ( 1 + ( N - 1 )abs( INCX ) )

[in] INCX

          INCX is INTEGER

         storage spacing between elements of DX

[in,out] DY

          DY is DOUBLE PRECISION array, dimension ( 1 + ( N - 1 )abs( INCY ) )

[in] INCY

          INCY is INTEGER

         storage spacing between elements of DY

[in] C

          C is DOUBLE PRECISION

[in] S

          S is DOUBLE PRECISION

Group: double\_blas\_level1

## DROTG

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DROTG(DA,DB,C,S)

       .. Scalar Arguments ..

       DOUBLE PRECISION C,DA,DB,S

       ..

Purpose:

=============

    DROTG construct givens plane rotation.

Arguments:

==========

[in] DA

          DA is DOUBLE PRECISION

[in] DB

          DB is DOUBLE PRECISION

[out] C

          C is DOUBLE PRECISION

[out] S

          S is DOUBLE PRECISION

Group: double\_blas\_level1

## DROTM

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DROTM(N,DX,INCX,DY,INCY,DPARAM)

       .. Scalar Arguments ..

       INTEGER INCX,INCY,N

       ..

       .. Array Arguments ..

       DOUBLE PRECISION DPARAM(5),DX(),DY()

       ..

Purpose:

=============

    APPLY THE MODIFIED GIVENS TRANSFORMATION, H, TO THE 2 BY N MATRIX

    (DXT) , WHERE T INDICATES TRANSPOSE. THE ELEMENTS OF DX ARE IN

    (DYT)

    DX(LX+IINCX), I = 0 TO N-1, WHERE LX = 1 IF INCX .GE. 0, ELSE

    LX = (-INCX)N, AND SIMILARLY FOR SY USING LY AND INCY.

    WITH DPARAM(1)=DFLAG, H HAS ONE OF THE FOLLOWING FORMS..

    DFLAG=-1.D0     DFLAG=0.D0        DFLAG=1.D0     DFLAG=-2.D0

      (DH11  DH12)    (1.D0  DH12)    (DH11  1.D0)    (1.D0  0.D0)

    H=(          )    (          )    (          )    (          )

      (DH21  DH22),   (DH21  1.D0),   (-1.D0 DH22),   (0.D0  1.D0).

    SEE DROTMG FOR A DESCRIPTION OF DATA STORAGE IN DPARAM.

Arguments:

==========

[in] N

          N is INTEGER

         number of elements in input vector(s)

[in,out] DX

          DX is DOUBLE PRECISION array, dimension ( 1 + ( N - 1 )abs( INCX ) )

[in] INCX

          INCX is INTEGER

         storage spacing between elements of DX

[in,out] DY

          DY is DOUBLE PRECISION array, dimension ( 1 + ( N - 1 )abs( INCY ) )

[in] INCY

          INCY is INTEGER

         storage spacing between elements of DY

[in] DPARAM

          DPARAM is DOUBLE PRECISION array, dimension (5)

     DPARAM(1)=DFLAG

     DPARAM(2)=DH11

     DPARAM(3)=DH21

     DPARAM(4)=DH12

     DPARAM(5)=DH22

Group: double\_blas\_level1

## DROTMG

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DROTMG(DD1,DD2,DX1,DY1,DPARAM)

       .. Scalar Arguments ..

       DOUBLE PRECISION DD1,DD2,DX1,DY1

       ..

       .. Array Arguments ..

       DOUBLE PRECISION DPARAM(5)

       ..

Purpose:

=============

    CONSTRUCT THE MODIFIED GIVENS TRANSFORMATION MATRIX H WHICH ZEROS

    THE SECOND COMPONENT OF THE 2-VECTOR  (DSQRT(DD1)DX1,DSQRT(DD2)    DY2)T.

    WITH DPARAM(1)=DFLAG, H HAS ONE OF THE FOLLOWING FORMS..

    DFLAG=-1.D0     DFLAG=0.D0        DFLAG=1.D0     DFLAG=-2.D0

      (DH11  DH12)    (1.D0  DH12)    (DH11  1.D0)    (1.D0  0.D0)

    H=(          )    (          )    (          )    (          )

      (DH21  DH22),   (DH21  1.D0),   (-1.D0 DH22),   (0.D0  1.D0).

    LOCATIONS 2-4 OF DPARAM CONTAIN DH11, DH21, DH12, AND DH22

    RESPECTIVELY. (VALUES OF 1.D0, -1.D0, OR 0.D0 IMPLIED BY THE

    VALUE OF DPARAM(1) ARE NOT STORED IN DPARAM.)

    THE VALUES OF GAMSQ AND RGAMSQ SET IN THE DATA STATEMENT MAY BE

    INEXACT.  THIS IS OK AS THEY ARE ONLY USED FOR TESTING THE SIZE

    OF DD1 AND DD2.  ALL ACTUAL SCALING OF DATA IS DONE USING GAM.

Arguments:

==========

[in,out] DD1

          DD1 is DOUBLE PRECISION

[in,out] DD2

          DD2 is DOUBLE PRECISION

[in,out] DX1

          DX1 is DOUBLE PRECISION

[in] DY1

          DY1 is DOUBLE PRECISION

[out] DPARAM

          DPARAM is DOUBLE PRECISION array, dimension (5)

     DPARAM(1)=DFLAG

     DPARAM(2)=DH11

     DPARAM(3)=DH21

     DPARAM(4)=DH12

     DPARAM(5)=DH22

Group: double\_blas\_level1

## DSBMV

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DSBMV(UPLO,N,K,ALPHA,A,LDA,X,INCX,BETA,Y,INCY)

       .. Scalar Arguments ..

       DOUBLE PRECISION ALPHA,BETA

       INTEGER INCX,INCY,K,LDA,N

       CHARACTER UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),X(),Y()

       ..

Purpose:

=============

 DSBMV  performs the matrix-vector  operation

    y := alphaAx + betay,

 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.

Arguments:

==========

[in] UPLO

          UPLO is CHARACTER1

           On entry, UPLO specifies whether the upper or lower

           triangular part of the band matrix A is being supplied as

           follows:

              UPLO = 'U' or 'u'   The upper triangular part of A is

                                  being supplied.

              UPLO = 'L' or 'l'   The lower triangular part of A is

                                  being supplied.

[in] N

          N is INTEGER

           On entry, N specifies the order of the matrix A.

           N must be at least zero.

[in] K

          K is INTEGER

           On entry, K specifies the number of super-diagonals of the

           matrix A. K must satisfy  0 .le. K.

[in] ALPHA

          ALPHA is DOUBLE PRECISION.

           On entry, ALPHA specifies the scalar alpha.

[in] A

          A is DOUBLE PRECISION array, 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 row k, and so on. The top left k by k triangle

           of the array A is not referenced.

           The following program segment will transfer the upper

           triangular part of a symmetric band matrix from conventional

           full matrix storage to band storage:

                 DO 20, J = 1, N

                    M = K + 1 - J

                    DO 10, I = MAX( 1, J - K ), J

                       A( M + I, J ) = matrix( I, J )

              10    CONTINUE

              20 CONTINUE

           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.

           The following program segment will transfer the lower

           triangular part of a symmetric band matrix from conventional

           full matrix storage to band storage:

                 DO 20, J = 1, N

                    M = 1 - J

                    DO 10, I = J, MIN( N, J + K )

                       A( M + I, J ) = matrix( I, J )

              10    CONTINUE

              20 CONTINUE

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in the calling (sub) program. LDA must be at least

           ( k + 1 ).

[in] X

          X is DOUBLE PRECISION array, dimension at least

           ( 1 + ( n - 1 )abs( INCX ) ).

           Before entry, the incremented array X must contain the

           vector x.

[in] INCX

          INCX is INTEGER

           On entry, INCX specifies the increment for the elements of

           X. INCX must not be zero.

[in] BETA

          BETA is DOUBLE PRECISION.

           On entry, BETA specifies the scalar beta.

[in,out] Y

          Y is DOUBLE PRECISION array, dimension at least

           ( 1 + ( n - 1 )abs( INCY ) ).

           Before entry, the incremented array Y must contain the

           vector y. On exit, Y is overwritten by the updated vector y.

[in] INCY

          INCY is INTEGER

           On entry, INCY specifies the increment for the elements of

           Y. INCY must not be zero.

Group: double\_blas\_level2

## DSCAL

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DSCAL(N,DA,DX,INCX)

       .. Scalar Arguments ..

       DOUBLE PRECISION DA

       INTEGER INCX,N

       ..

       .. Array Arguments ..

       DOUBLE PRECISION DX()

       ..

Purpose:

=============

    DSCAL scales a vector by a constant.

    uses unrolled loops for increment equal to 1.

Arguments:

==========

[in] N

          N is INTEGER

         number of elements in input vector(s)

[in] DA

          DA is DOUBLE PRECISION

           On entry, DA specifies the scalar alpha.

[in,out] DX

          DX is DOUBLE PRECISION array, dimension ( 1 + ( N - 1 )abs( INCX ) )

[in] INCX

          INCX is INTEGER

         storage spacing between elements of DX

Group: double\_blas\_level1

## DSDOT

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       DOUBLE PRECISION FUNCTION DSDOT(N,SX,INCX,SY,INCY)

       .. Scalar Arguments ..

       INTEGER INCX,INCY,N

       ..

       .. Array Arguments ..

       REAL SX(),SY()

       ..

    AUTHORS

    =======

    Lawson, C. L., (JPL), Hanson, R. J., (SNLA),

    Kincaid, D. R., (U. of Texas), Krogh, F. T., (JPL)

Purpose:

=============

 Compute the inner product of two vectors with extended

 precision accumulation and result.

 Returns D.P. dot product accumulated in D.P., for S.P. SX and SY

 DSDOT = sum for I = 0 to N-1 of  SX(LX+IINCX)  SY(LY+IINCY),

 where LX = 1 if INCX .GE. 0, else LX = 1+(1-N)INCX, and LY is

 defined in a similar way using INCY.

Arguments:

==========

[in] N

          N is INTEGER

         number of elements in input vector(s)

[in] SX

          SX is REAL array, dimension(N)

         single precision vector with N elements

[in] INCX

          INCX is INTEGER

          storage spacing between elements of SX

[in] SY

          SY is REAL array, dimension(N)

         single precision vector with N elements

[in] INCY

          INCY is INTEGER

         storage spacing between elements of SY

Result DSDOT

          DSDOT is DOUBLE PRECISION

         DSDOT  double precision dot product (zero if N.LE.0)

Group: double\_blas\_level1

References:

================

  C. L. Lawson, R. J. Hanson, D. R. Kincaid and F. T.

  Krogh, Basic linear algebra subprograms for Fortran

  usage, Algorithm No. 539, Transactions on Mathematical

  Software 5, 3 (September 1979), pp. 308-323.

  REVISION HISTORY  (YYMMDD)

  791001  DATE WRITTEN

  890831  Modified array declarations.  (WRB)

  890831  REVISION DATE from Version 3.2

  891214  Prologue converted to Version 4.0 format.  (BAB)

  920310  Corrected definition of LX in DESCRIPTION.  (WRB)

  920501  Reformatted the REFERENCES section.  (WRB)

  070118  Reformat to LAPACK style (JL)

## DSPMV

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DSPMV(UPLO,N,ALPHA,AP,X,INCX,BETA,Y,INCY)

       .. Scalar Arguments ..

       DOUBLE PRECISION ALPHA,BETA

       INTEGER INCX,INCY,N

       CHARACTER UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION AP(),X(),Y()

       ..

Purpose:

=============

 DSPMV  performs the matrix-vector operation

    y := alphaAx + betay,

 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.

Arguments:

==========

[in] UPLO

          UPLO is CHARACTER1

           On entry, UPLO specifies whether the upper or lower

           triangular part of the matrix A is supplied in the packed

           array AP as follows:

              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.

[in] N

          N is INTEGER

           On entry, N specifies the order of the matrix A.

           N must be at least zero.

[in] ALPHA

          ALPHA is DOUBLE PRECISION.

           On entry, ALPHA specifies the scalar alpha.

[in] AP

          AP is DOUBLE PRECISION array, 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 a( 1, 1 ), AP( 2 ) and AP( 3 ) contain a( 2, 1 )

           and a( 3, 1 ) respectively, and so on.

[in] X

          X is DOUBLE PRECISION array, dimension at least

           ( 1 + ( n - 1 )abs( INCX ) ).

           Before entry, the incremented array X must contain the n

           element vector x.

[in] INCX

          INCX is INTEGER

           On entry, INCX specifies the increment for the elements of

           X. INCX must not be zero.

[in] BETA

          BETA is DOUBLE PRECISION.

           On entry, BETA specifies the scalar beta. When BETA is

           supplied as zero then Y need not be set on input.

[in,out] Y

          Y is DOUBLE PRECISION array, 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.

[in] INCY

          INCY is INTEGER

           On entry, INCY specifies the increment for the elements of

           Y. INCY must not be zero.

Group: double\_blas\_level2

Further Details:

=====================

  Level 2 Blas routine.

  The vector and matrix arguments are not referenced when N = 0, or M = 0

  -- Written on 22-October-1986.

     Jack Dongarra, Argonne National Lab.

     Jeremy Du Croz, Nag Central Office.

     Sven Hammarling, Nag Central Office.

     Richard Hanson, Sandia National Labs.

## DSPR

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DSPR(UPLO,N,ALPHA,X,INCX,AP)

       .. Scalar Arguments ..

       DOUBLE PRECISION ALPHA

       INTEGER INCX,N

       CHARACTER UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION AP(),X()

       ..

Purpose:

=============

 DSPR    performs the symmetric rank 1 operation

    A := alphaxxT + 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.

Arguments:

==========

[in] UPLO

          UPLO is CHARACTER1

           On entry, UPLO specifies whether the upper or lower

           triangular part of the matrix A is supplied in the packed

           array AP as follows:

              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.

[in] N

          N is INTEGER

           On entry, N specifies the order of the matrix A.

           N must be at least zero.

[in] ALPHA

          ALPHA is DOUBLE PRECISION.

           On entry, ALPHA specifies the scalar alpha.

[in] X

          X is DOUBLE PRECISION array, dimension at least

           ( 1 + ( n - 1 )abs( INCX ) ).

           Before entry, the incremented array X must contain the n

           element vector x.

[in] INCX

          INCX is INTEGER

           On entry, INCX specifies the increment for the elements of

           X. INCX must not be zero.

[in,out] AP

          AP is DOUBLE PRECISION array, 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.

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date December 2016

Group: double\_blas\_level2

Further Details:

=====================

  Level 2 Blas routine.

  -- Written on 22-October-1986.

     Jack Dongarra, Argonne National Lab.

     Jeremy Du Croz, Nag Central Office.

     Sven Hammarling, Nag Central Office.

     Richard Hanson, Sandia National Labs.

## DSPR2

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DSPR2(UPLO,N,ALPHA,X,INCX,Y,INCY,AP)

       .. Scalar Arguments ..

       DOUBLE PRECISION ALPHA

       INTEGER INCX,INCY,N

       CHARACTER UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION AP(),X(),Y()

       ..

Purpose:

=============

 DSPR2  performs the symmetric rank 2 operation

    A := alphaxyT + alphayxT + 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.

Arguments:

==========

[in] UPLO

          UPLO is CHARACTER1

           On entry, UPLO specifies whether the upper or lower

           triangular part of the matrix A is supplied in the packed

           array AP as follows:

              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.

[in] N

          N is INTEGER

           On entry, N specifies the order of the matrix A.

           N must be at least zero.

[in] ALPHA

          ALPHA is DOUBLE PRECISION.

           On entry, ALPHA specifies the scalar alpha.

[in] X

          X is DOUBLE PRECISION array, dimension at least

           ( 1 + ( n - 1 )abs( INCX ) ).

           Before entry, the incremented array X must contain the n

           element vector x.

[in] INCX

          INCX is INTEGER

           On entry, INCX specifies the increment for the elements of

           X. INCX must not be zero.

[in] Y

          Y is DOUBLE PRECISION array, dimension at least

           ( 1 + ( n - 1 )abs( INCY ) ).

           Before entry, the incremented array Y must contain the n

           element vector y.

[in] INCY

          INCY is INTEGER

           On entry, INCY specifies the increment for the elements of

           Y. INCY must not be zero.

[in,out] AP

          AP is DOUBLE PRECISION array, 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.

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date December 2016

Group: double\_blas\_level2

Further Details:

=====================

  Level 2 Blas routine.

  -- Written on 22-October-1986.

     Jack Dongarra, Argonne National Lab.

     Jeremy Du Croz, Nag Central Office.

     Sven Hammarling, Nag Central Office.

     Richard Hanson, Sandia National Labs.

## DSWAP

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DSWAP(N,DX,INCX,DY,INCY)

       .. Scalar Arguments ..

       INTEGER INCX,INCY,N

       ..

       .. Array Arguments ..

       DOUBLE PRECISION DX(),DY()

       ..

Purpose:

=============

    DSWAP interchanges two vectors.

    uses unrolled loops for increments equal to 1.

Arguments:

==========

[in] N

          N is INTEGER

         number of elements in input vector(s)

[in,out] DX

          DX is DOUBLE PRECISION array, dimension ( 1 + ( N - 1 )abs( INCX ) )

[in] INCX

          INCX is INTEGER

         storage spacing between elements of DX

[in,out] DY

          DY is DOUBLE PRECISION array, dimension ( 1 + ( N - 1 )abs( INCY ) )

[in] INCY

          INCY is INTEGER

         storage spacing between elements of DY

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date November 2017

Group: double\_blas\_level1

Further Details:

=====================

     jack dongarra, linpack, 3/11/78.

     modified 12/3/93, array(1) declarations changed to array()

## DSYMM

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DSYMM(SIDE,UPLO,M,N,ALPHA,A,LDA,B,LDB,BETA,C,LDC)

       .. Scalar Arguments ..

       DOUBLE PRECISION ALPHA,BETA

       INTEGER LDA,LDB,LDC,M,N

       CHARACTER SIDE,UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),B(LDB,),C(LDC,)

       ..

Purpose:

=============

 DSYMM  performs one of the matrix-matrix operations

    C := alphaAB + betaC,

 or

    C := alphaBA + betaC,

 where alpha and beta are scalars,  A is a symmetric matrix and  B and

 C are  m by n matrices.

Arguments:

==========

[in] SIDE

          SIDE is CHARACTER1

           On entry,  SIDE  specifies whether  the  symmetric matrix  A

           appears on the  left or right  in the  operation as follows:

              SIDE = 'L' or 'l'   C := alphaAB + betaC,

              SIDE = 'R' or 'r'   C := alphaBA + betaC,

[in] UPLO

          UPLO is CHARACTER1

           On  entry,   UPLO  specifies  whether  the  upper  or  lower

           triangular  part  of  the  symmetric  matrix   A  is  to  be

           referenced as follows:

              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.

[in] M

          M is INTEGER

           On entry,  M  specifies the number of rows of the matrix  C.

           M  must be at least zero.

[in] N

          N is INTEGER

           On entry, N specifies the number of columns of the matrix C.

           N  must be at least zero.

[in] ALPHA

          ALPHA is DOUBLE PRECISION.

           On entry, ALPHA specifies the scalar alpha.

[in] A

          A is DOUBLE PRECISION array, dimension ( LDA, ka ), where ka is

           m  when  SIDE = 'L' or 'l'  and is  n otherwise.

           Before entry  with  SIDE = 'L' or 'l',  the  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.

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in the calling (sub) program.  When  SIDE = 'L' or 'l'  then

           LDA must be at least  max( 1, m ), otherwise  LDA must be at

           least  max( 1, n ).

[in] B

          B is DOUBLE PRECISION array, dimension ( LDB, N )

           Before entry, the leading  m by n part of the array  B  must

           contain the matrix B.

[in] LDB

          LDB is INTEGER

           On entry, LDB specifies the first dimension of B as declared

           in  the  calling  (sub)  program.   LDB  must  be  at  least

           max( 1, m ).

[in] BETA

          BETA is DOUBLE PRECISION.

           On entry,  BETA  specifies the scalar  beta.  When  BETA  is

           supplied as zero then C need not be set on input.

[in,out] C

          C is DOUBLE PRECISION array, 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.

[in] LDC

          LDC is INTEGER

           On entry, LDC specifies the first dimension of C as declared

           in  the  calling  (sub)  program.   LDC  must  be  at  least

           max( 1, m ).

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date December 2016

Group: double\_blas\_level3

Further Details:

=====================

  Level 3 Blas routine.

  -- Written on 8-February-1989.

     Jack Dongarra, Argonne National Laboratory.

     Iain Duff, AERE Harwell.

     Jeremy Du Croz, Numerical AlgorithmsGroup: Ltd.

     Sven Hammarling, Numerical AlgorithmsGroup: Ltd.

## DSYMV

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DSYMV(UPLO,N,ALPHA,A,LDA,X,INCX,BETA,Y,INCY)

       .. Scalar Arguments ..

       DOUBLE PRECISION ALPHA,BETA

       INTEGER INCX,INCY,LDA,N

       CHARACTER UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),X(),Y()

       ..

Purpose:

=============

 DSYMV  performs the matrix-vector  operation

    y := alphaAx + betay,

 where alpha and beta are scalars, x and y are n element vectors and

 A is an n by n symmetric matrix.

Arguments:

==========

[in] UPLO

          UPLO is CHARACTER1

           On entry, UPLO specifies whether the upper or lower

           triangular part of the array A is to be referenced as

           follows:

              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.

[in] N

          N is INTEGER

           On entry, N specifies the order of the matrix A.

           N must be at least zero.

[in] ALPHA

          ALPHA is DOUBLE PRECISION.

           On entry, ALPHA specifies the scalar alpha.

[in] A

          A is DOUBLE PRECISION array, 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.

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in the calling (sub) program. LDA must be at least

           max( 1, n ).

[in] X

          X is DOUBLE PRECISION array, dimension at least

           ( 1 + ( n - 1 )abs( INCX ) ).

           Before entry, the incremented array X must contain the n

           element vector x.

[in] INCX

          INCX is INTEGER

           On entry, INCX specifies the increment for the elements of

           X. INCX must not be zero.

[in] BETA

          BETA is DOUBLE PRECISION.

           On entry, BETA specifies the scalar beta. When BETA is

           supplied as zero then Y need not be set on input.

[in,out] Y

          Y is DOUBLE PRECISION array, 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.

[in] INCY

          INCY is INTEGER

           On entry, INCY specifies the increment for the elements of

           Y. INCY must not be zero.

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date December 2016

Group: double\_blas\_level2

Further Details:

=====================

  Level 2 Blas routine.

  The vector and matrix arguments are not referenced when N = 0, or M = 0

  -- Written on 22-October-1986.

     Jack Dongarra, Argonne National Lab.

     Jeremy Du Croz, Nag Central Office.

     Sven Hammarling, Nag Central Office.

     Richard Hanson, Sandia National Labs.

## DSYR

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DSYR(UPLO,N,ALPHA,X,INCX,A,LDA)

       .. Scalar Arguments ..

       DOUBLE PRECISION ALPHA

       INTEGER INCX,LDA,N

       CHARACTER UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),X()

       ..

Purpose:

=============

 DSYR   performs the symmetric rank 1 operation

    A := alphaxxT + A,

 where alpha is a real scalar, x is an n element vector and A is an

 n by n symmetric matrix.

Arguments:

==========

[in] UPLO

          UPLO is CHARACTER1

           On entry, UPLO specifies whether the upper or lower

           triangular part of the array A is to be referenced as

           follows:

              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.

[in] N

          N is INTEGER

           On entry, N specifies the order of the matrix A.

           N must be at least zero.

[in] ALPHA

          ALPHA is DOUBLE PRECISION.

           On entry, ALPHA specifies the scalar alpha.

[in] X

          X is DOUBLE PRECISION array, dimension at least

           ( 1 + ( n - 1 )abs( INCX ) ).

           Before entry, the incremented array X must contain the n

           element vector x.

[in] INCX

          INCX is INTEGER

           On entry, INCX specifies the increment for the elements of

           X. INCX must not be zero.

[in,out] A

          A is DOUBLE PRECISION array, 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.

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in the calling (sub) program. LDA must be at least

           max( 1, n ).

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date December 2016

Group: double\_blas\_level2

Further Details:

=====================

  Level 2 Blas routine.

  -- Written on 22-October-1986.

     Jack Dongarra, Argonne National Lab.

     Jeremy Du Croz, Nag Central Office.

     Sven Hammarling, Nag Central Office.

     Richard Hanson, Sandia National Labs.

## DSYR2

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DSYR2(UPLO,N,ALPHA,X,INCX,Y,INCY,A,LDA)

       .. Scalar Arguments ..

       DOUBLE PRECISION ALPHA

       INTEGER INCX,INCY,LDA,N

       CHARACTER UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),X(),Y()

       ..

Purpose:

=============

 DSYR2  performs the symmetric rank 2 operation

    A := alphaxyT + alphayxT + A,

 where alpha is a scalar, x and y are n element vectors and A is an n

 by n symmetric matrix.

Arguments:

==========

[in] UPLO

          UPLO is CHARACTER1

           On entry, UPLO specifies whether the upper or lower

           triangular part of the array A is to be referenced as

           follows:

              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.

[in] N

          N is INTEGER

           On entry, N specifies the order of the matrix A.

           N must be at least zero.

[in] ALPHA

          ALPHA is DOUBLE PRECISION.

           On entry, ALPHA specifies the scalar alpha.

[in] X

          X is DOUBLE PRECISION array, dimension at least

           ( 1 + ( n - 1 )abs( INCX ) ).

           Before entry, the incremented array X must contain the n

           element vector x.

[in] INCX

          INCX is INTEGER

           On entry, INCX specifies the increment for the elements of

           X. INCX must not be zero.

[in] Y

          Y is DOUBLE PRECISION array, dimension at least

           ( 1 + ( n - 1 )abs( INCY ) ).

           Before entry, the incremented array Y must contain the n

           element vector y.

[in] INCY

          INCY is INTEGER

           On entry, INCY specifies the increment for the elements of

           Y. INCY must not be zero.

[in,out] A

          A is DOUBLE PRECISION array, 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.

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in the calling (sub) program. LDA must be at least

           max( 1, n ).

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date December 2016

Group: double\_blas\_level2

Further Details:

=====================

  Level 2 Blas routine.

  -- Written on 22-October-1986.

     Jack Dongarra, Argonne National Lab.

     Jeremy Du Croz, Nag Central Office.

     Sven Hammarling, Nag Central Office.

     Richard Hanson, Sandia National Labs.

## DSYR2K

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DSYR2K(UPLO,TRANS,N,K,ALPHA,A,LDA,B,LDB,BETA,C,LDC)

       .. Scalar Arguments ..

       DOUBLE PRECISION ALPHA,BETA

       INTEGER K,LDA,LDB,LDC,N

       CHARACTER TRANS,UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),B(LDB,),C(LDC,)

       ..

Purpose:

=============

 DSYR2K  performs one of the symmetric rank 2k operations

    C := alphaABT + alphaBAT + betaC,

 or

    C := alphaATB + alphaBTA + betaC,

 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.

Arguments:

==========

[in] UPLO

          UPLO is CHARACTER1

           On  entry,   UPLO  specifies  whether  the  upper  or  lower

           triangular  part  of the  array  C  is to be  referenced  as

           follows:

              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.

[in] TRANS

          TRANS is CHARACTER1

           On entry,  TRANS  specifies the operation to be performed as

           follows:

              TRANS = 'N' or 'n'   C := alphaABT + alphaBAT +

                                        betaC.

              TRANS = 'T' or 't'   C := alphaATB + alphaBTA +

                                        betaC.

              TRANS = 'C' or 'c'   C := alphaATB + alphaBTA +

                                        betaC.

[in] N

          N is INTEGER

           On entry,  N specifies the order of the matrix C.  N must be

           at least zero.

[in] K

          K is INTEGER

           On entry with  TRANS = 'N' or 'n',  K  specifies  the number

           of  columns  of the  matrices  A and B,  and on  entry  with

           TRANS = 'T' or 't' or 'C' or 'c',  K  specifies  the  number

           of rows of the matrices  A and B.  K must be at least  zero.

[in] ALPHA

          ALPHA is DOUBLE PRECISION.

           On entry, ALPHA specifies the scalar alpha.

[in] A

          A is DOUBLE PRECISION array, 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 by k

           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.

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in  the  calling  (sub)  program.   When  TRANS = 'N' or 'n'

           then  LDA must be at least  max( 1, n ), otherwise  LDA must

           be at least  max( 1, k ).

[in] B

          B is DOUBLE PRECISION array, dimension ( LDB, kb ), where kb is

           k  when  TRANS = 'N' or 'n',  and is  n  otherwise.

           Before entry with  TRANS = 'N' or 'n',  the  leading  n by k

           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.

[in] LDB

          LDB is INTEGER

           On entry, LDB specifies the first dimension of B as declared

           in  the  calling  (sub)  program.   When  TRANS = 'N' or 'n'

           then  LDB must be at least  max( 1, n ), otherwise  LDB must

           be at least  max( 1, k ).

[in] BETA

          BETA is DOUBLE PRECISION.

           On entry, BETA specifies the scalar beta.

[in,out] C

          C is DOUBLE PRECISION array, 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.

[in] LDC

          LDC is INTEGER

           On entry, LDC specifies the first dimension of C as declared

           in  the  calling  (sub)  program.   LDC  must  be  at  least

           max( 1, n ).

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date December 2016

Group: double\_blas\_level3

Further Details:

=====================

  Level 3 Blas routine.

  -- Written on 8-February-1989.

     Jack Dongarra, Argonne National Laboratory.

     Iain Duff, AERE Harwell.

     Jeremy Du Croz, Numerical AlgorithmsGroup: Ltd.

     Sven Hammarling, Numerical AlgorithmsGroup: Ltd.

## DSYRK

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DSYRK(UPLO,TRANS,N,K,ALPHA,A,LDA,BETA,C,LDC)

       .. Scalar Arguments ..

       DOUBLE PRECISION ALPHA,BETA

       INTEGER K,LDA,LDC,N

       CHARACTER TRANS,UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),C(LDC,)

       ..

Purpose:

=============

 DSYRK  performs one of the symmetric rank k operations

    C := alphaAAT + betaC,

 or

    C := alphaATA + betaC,

 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.

Arguments:

==========

[in] UPLO

          UPLO is CHARACTER1

           On  entry,   UPLO  specifies  whether  the  upper  or  lower

           triangular  part  of the  array  C  is to be  referenced  as

           follows:

              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.

[in] TRANS

          TRANS is CHARACTER1

           On entry,  TRANS  specifies the operation to be performed as

           follows:

              TRANS = 'N' or 'n'   C := alphaAAT + betaC.

              TRANS = 'T' or 't'   C := alphaATA + betaC.

              TRANS = 'C' or 'c'   C := alphaATA + betaC.

[in] N

          N is INTEGER

           On entry,  N specifies the order of the matrix C.  N must be

           at least zero.

[in] K

          K is INTEGER

           On entry with  TRANS = 'N' or 'n',  K  specifies  the number

           of  columns   of  the   matrix   A,   and  on   entry   with

           TRANS = 'T' or 't' or 'C' or 'c',  K  specifies  the  number

           of rows of the matrix  A.  K must be at least zero.

[in] ALPHA

          ALPHA is DOUBLE PRECISION.

           On entry, ALPHA specifies the scalar alpha.

[in] A

          A is DOUBLE PRECISION array, 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 by k

           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.

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in  the  calling  (sub)  program.   When  TRANS = 'N' or 'n'

           then  LDA must be at least  max( 1, n ), otherwise  LDA must

           be at least  max( 1, k ).

[in] BETA

          BETA is DOUBLE PRECISION.

           On entry, BETA specifies the scalar beta.

[in,out] C

          C is DOUBLE PRECISION array, 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.

[in] LDC

          LDC is INTEGER

           On entry, LDC specifies the first dimension of C as declared

           in  the  calling  (sub)  program.   LDC  must  be  at  least

           max( 1, n ).

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date December 2016

Group: double\_blas\_level3

Further Details:

=====================

  Level 3 Blas routine.

  -- Written on 8-February-1989.

     Jack Dongarra, Argonne National Laboratory.

     Iain Duff, AERE Harwell.

     Jeremy Du Croz, Numerical AlgorithmsGroup: Ltd.

     Sven Hammarling, Numerical AlgorithmsGroup: Ltd.

## DTBMV

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DTBMV(UPLO,TRANS,DIAG,N,K,A,LDA,X,INCX)

       .. Scalar Arguments ..

       INTEGER INCX,K,LDA,N

       CHARACTER DIAG,TRANS,UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),X()

       ..

Purpose:

=============

 DTBMV  performs one of the matrix-vector operations

    x := Ax,   or   x := ATx,

 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.

Arguments:

==========

[in] UPLO

          UPLO is CHARACTER1

           On entry, UPLO specifies whether the matrix is an upper or

           lower triangular matrix as follows:

              UPLO = 'U' or 'u'   A is an upper triangular matrix.

              UPLO = 'L' or 'l'   A is a lower triangular matrix.

[in] TRANS

          TRANS is CHARACTER1

           On entry, TRANS specifies the operation to be performed as

           follows:

              TRANS = 'N' or 'n'   x := Ax.

              TRANS = 'T' or 't'   x := ATx.

              TRANS = 'C' or 'c'   x := ATx.

[in] DIAG

          DIAG is CHARACTER1

           On entry, DIAG specifies whether or not A is unit

           triangular as follows:

              DIAG = 'U' or 'u'   A is assumed to be unit triangular.

              DIAG = 'N' or 'n'   A is not assumed to be unit

                                  triangular.

[in] N

          N is INTEGER

           On entry, N specifies the order of the matrix A.

           N must be at least zero.

[in] K

          K is INTEGER

           On entry with UPLO = 'U' or 'u', K specifies the number of

           super-diagonals of the matrix A.

           On entry with UPLO = 'L' or 'l', K specifies the number of

           sub-diagonals of the matrix A.

           K must satisfy  0 .le. K.

[in] A

          A is DOUBLE PRECISION array, 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.

           The following program segment will transfer an upper

           triangular band matrix from conventional full matrix storage

           to band storage:

                 DO 20, J = 1, N

                    M = K + 1 - J

                    DO 10, I = MAX( 1, J - K ), J

                       A( M + I, J ) = matrix( I, J )

              10    CONTINUE

              20 CONTINUE

           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.

           The following program segment will transfer a lower

           triangular band matrix from conventional full matrix storage

           to band storage:

                 DO 20, J = 1, N

                    M = 1 - J

                    DO 10, I = J, MIN( N, J + K )

                       A( M + I, J ) = matrix( I, J )

              10    CONTINUE

              20 CONTINUE

           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.

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in the calling (sub) program. LDA must be at least

           ( k + 1 ).

[in,out] X

          X is DOUBLE PRECISION array, 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.

[in] INCX

          INCX is INTEGER

           On entry, INCX specifies the increment for the elements of

           X. INCX must not be zero.

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date December 2016

Group: double\_blas\_level2

Further Details:

=====================

  Level 2 Blas routine.

  The vector and matrix arguments are not referenced when N = 0, or M = 0

  -- Written on 22-October-1986.

     Jack Dongarra, Argonne National Lab.

     Jeremy Du Croz, Nag Central Office.

     Sven Hammarling, Nag Central Office.

     Richard Hanson, Sandia National Labs.

## DTBSV

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DTBSV(UPLO,TRANS,DIAG,N,K,A,LDA,X,INCX)

       .. Scalar Arguments ..

       INTEGER INCX,K,LDA,N

       CHARACTER DIAG,TRANS,UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),X()

       ..

Purpose:

=============

 DTBSV  solves one of the systems of equations

    Ax = b,   or   ATx = 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.

 No test for singularity or near-singularity is included in this

 routine. Such tests must be performed before calling this routine.

Arguments:

==========

[in] UPLO

          UPLO is CHARACTER1

           On entry, UPLO specifies whether the matrix is an upper or

           lower triangular matrix as follows:

              UPLO = 'U' or 'u'   A is an upper triangular matrix.

              UPLO = 'L' or 'l'   A is a lower triangular matrix.

[in] TRANS

          TRANS is CHARACTER1

           On entry, TRANS specifies the equations to be solved as

           follows:

              TRANS = 'N' or 'n'   Ax = b.

              TRANS = 'T' or 't'   ATx = b.

              TRANS = 'C' or 'c'   ATx = b.

[in] DIAG

          DIAG is CHARACTER1

           On entry, DIAG specifies whether or not A is unit

           triangular as follows:

              DIAG = 'U' or 'u'   A is assumed to be unit triangular.

              DIAG = 'N' or 'n'   A is not assumed to be unit

                                  triangular.

[in] N

          N is INTEGER

           On entry, N specifies the order of the matrix A.

           N must be at least zero.

[in] K

          K is INTEGER

           On entry with UPLO = 'U' or 'u', K specifies the number of

           super-diagonals of the matrix A.

           On entry with UPLO = 'L' or 'l', K specifies the number of

           sub-diagonals of the matrix A.

           K must satisfy  0 .le. K.

[in] A

          A is DOUBLE PRECISION array, 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.

           The following program segment will transfer an upper

           triangular band matrix from conventional full matrix storage

           to band storage:

                 DO 20, J = 1, N

                    M = K + 1 - J

                    DO 10, I = MAX( 1, J - K ), J

                       A( M + I, J ) = matrix( I, J )

              10    CONTINUE

              20 CONTINUE

           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.

           The following program segment will transfer a lower

           triangular band matrix from conventional full matrix storage

           to band storage:

                 DO 20, J = 1, N

                    M = 1 - J

                    DO 10, I = J, MIN( N, J + K )

                       A( M + I, J ) = matrix( I, J )

              10    CONTINUE

              20 CONTINUE

           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.

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in the calling (sub) program. LDA must be at least

           ( k + 1 ).

[in,out] X

          X is DOUBLE PRECISION array, 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.

[in] INCX

          INCX is INTEGER

           On entry, INCX specifies the increment for the elements of

           X. INCX must not be zero.

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date December 2016

Group: double\_blas\_level2

Further Details:

=====================

  Level 2 Blas routine.

  -- Written on 22-October-1986.

     Jack Dongarra, Argonne National Lab.

     Jeremy Du Croz, Nag Central Office.

     Sven Hammarling, Nag Central Office.

     Richard Hanson, Sandia National Labs.

## DTPMV

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DTPMV(UPLO,TRANS,DIAG,N,AP,X,INCX)

       .. Scalar Arguments ..

       INTEGER INCX,N

       CHARACTER DIAG,TRANS,UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION AP(),X()

       ..

Purpose:

=============

 DTPMV  performs one of the matrix-vector operations

    x := Ax,   or   x := ATx,

 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.

Arguments:

==========

[in] UPLO

          UPLO is CHARACTER1

           On entry, UPLO specifies whether the matrix is an upper or

           lower triangular matrix as follows:

              UPLO = 'U' or 'u'   A is an upper triangular matrix.

              UPLO = 'L' or 'l'   A is a lower triangular matrix.

[in] TRANS

          TRANS is CHARACTER1

           On entry, TRANS specifies the operation to be performed as

           follows:

              TRANS = 'N' or 'n'   x := Ax.

              TRANS = 'T' or 't'   x := ATx.

              TRANS = 'C' or 'c'   x := ATx.

[in] DIAG

          DIAG is CHARACTER1

           On entry, DIAG specifies whether or not A is unit

           triangular as follows:

              DIAG = 'U' or 'u'   A is assumed to be unit triangular.

              DIAG = 'N' or 'n'   A is not assumed to be unit

                                  triangular.

[in] N

          N is INTEGER

           On entry, N specifies the order of the matrix A.

           N must be at least zero.

[in] AP

          AP is DOUBLE PRECISION array, 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.

[in,out] X

          X is DOUBLE PRECISION array, 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.

[in] INCX

          INCX is INTEGER

           On entry, INCX specifies the increment for the elements of

           X. INCX must not be zero.

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date December 2016

Group: double\_blas\_level2

Further Details:

=====================

  Level 2 Blas routine.

  The vector and matrix arguments are not referenced when N = 0, or M = 0

  -- Written on 22-October-1986.

     Jack Dongarra, Argonne National Lab.

     Jeremy Du Croz, Nag Central Office.

     Sven Hammarling, Nag Central Office.

     Richard Hanson, Sandia National Labs.

## DTPSV

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DTPSV(UPLO,TRANS,DIAG,N,AP,X,INCX)

       .. Scalar Arguments ..

       INTEGER INCX,N

       CHARACTER DIAG,TRANS,UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION AP(),X()

       ..

Purpose:

=============

 DTPSV  solves one of the systems of equations

    Ax = b,   or   ATx = 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.

 No test for singularity or near-singularity is included in this

 routine. Such tests must be performed before calling this routine.

Arguments:

==========

[in] UPLO

          UPLO is CHARACTER1

           On entry, UPLO specifies whether the matrix is an upper or

           lower triangular matrix as follows:

              UPLO = 'U' or 'u'   A is an upper triangular matrix.

              UPLO = 'L' or 'l'   A is a lower triangular matrix.

[in] TRANS

          TRANS is CHARACTER1

           On entry, TRANS specifies the equations to be solved as

           follows:

              TRANS = 'N' or 'n'   Ax = b.

              TRANS = 'T' or 't'   ATx = b.

              TRANS = 'C' or 'c'   ATx = b.

[in] DIAG

          DIAG is CHARACTER1

           On entry, DIAG specifies whether or not A is unit

           triangular as follows:

              DIAG = 'U' or 'u'   A is assumed to be unit triangular.

              DIAG = 'N' or 'n'   A is not assumed to be unit

                                  triangular.

[in] N

          N is INTEGER

           On entry, N specifies the order of the matrix A.

           N must be at least zero.

[in] AP

          AP is DOUBLE PRECISION array, 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.

[in,out] X

          X is DOUBLE PRECISION array, 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.

[in] INCX

          INCX is INTEGER

           On entry, INCX specifies the increment for the elements of

           X. INCX must not be zero.

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date December 2016

Group: double\_blas\_level2

Further Details:

=====================

  Level 2 Blas routine.

  -- Written on 22-October-1986.

     Jack Dongarra, Argonne National Lab.

     Jeremy Du Croz, Nag Central Office.

     Sven Hammarling, Nag Central Office.

     Richard Hanson, Sandia National Labs.

## DTRMM

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DTRMM(SIDE,UPLO,TRANSA,DIAG,M,N,ALPHA,A,LDA,B,LDB)

       .. Scalar Arguments ..

       DOUBLE PRECISION ALPHA

       INTEGER LDA,LDB,M,N

       CHARACTER DIAG,SIDE,TRANSA,UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),B(LDB,)

       ..

Purpose:

=============

 DTRMM  performs one of the matrix-matrix operations

    B := alphaop( A )B,   or   B := alphaBop( 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 ) = AT.

Arguments:

==========

[in] SIDE

          SIDE is CHARACTER1

           On entry,  SIDE specifies whether  op( A ) multiplies B from

           the left or right as follows:

              SIDE = 'L' or 'l'   B := alphaop( A )B.

              SIDE = 'R' or 'r'   B := alphaBop( A ).

[in] UPLO

          UPLO is CHARACTER1

           On entry, UPLO specifies whether the matrix A is an upper or

           lower triangular matrix as follows:

              UPLO = 'U' or 'u'   A is an upper triangular matrix.

              UPLO = 'L' or 'l'   A is a lower triangular matrix.

[in] TRANSA

          TRANSA is CHARACTER1

           On entry, 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 ) = AT.

              TRANSA = 'C' or 'c'   op( A ) = AT.

[in] DIAG

          DIAG is CHARACTER1

           On entry, DIAG specifies whether or not A is unit triangular

           as follows:

              DIAG = 'U' or 'u'   A is assumed to be unit triangular.

              DIAG = 'N' or 'n'   A is not assumed to be unit

                                  triangular.

[in] M

          M is INTEGER

           On entry, M specifies the number of rows of B. M must be at

           least zero.

[in] N

          N is INTEGER

           On entry, N specifies the number of columns of B.  N must be

           at least zero.

[in] ALPHA

          ALPHA is DOUBLE PRECISION.

           On entry,  ALPHA specifies the scalar  alpha. When  alpha is

           zero then  A is not referenced and  B need not be set before

           entry.

[in] A

           A is DOUBLE PRECISION array, 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.

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in the calling (sub) program.  When  SIDE = 'L' or 'l'  then

           LDA  must be at least  max( 1, m ),  when  SIDE = 'R' or 'r'

           then LDA must be at least max( 1, n ).

[in,out] B

          B is DOUBLE PRECISION array, 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.

[in] LDB

          LDB is INTEGER

           On entry, LDB specifies the first dimension of B as declared

           in  the  calling  (sub)  program.   LDB  must  be  at  least

           max( 1, m ).

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date December 2016

Group: double\_blas\_level3

Further Details:

=====================

  Level 3 Blas routine.

  -- Written on 8-February-1989.

     Jack Dongarra, Argonne National Laboratory.

     Iain Duff, AERE Harwell.

     Jeremy Du Croz, Numerical AlgorithmsGroup: Ltd.

     Sven Hammarling, Numerical AlgorithmsGroup: Ltd.

## DTRMV

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DTRMV(UPLO,TRANS,DIAG,N,A,LDA,X,INCX)

       .. Scalar Arguments ..

       INTEGER INCX,LDA,N

       CHARACTER DIAG,TRANS,UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),X()

       ..

Purpose:

=============

 DTRMV  performs one of the matrix-vector operations

    x := Ax,   or   x := ATx,

 where x is an n element vector and  A is an n by n unit, or non-unit,

 upper or lower triangular matrix.

Arguments:

==========

[in] UPLO

          UPLO is CHARACTER1

           On entry, UPLO specifies whether the matrix is an upper or

           lower triangular matrix as follows:

              UPLO = 'U' or 'u'   A is an upper triangular matrix.

              UPLO = 'L' or 'l'   A is a lower triangular matrix.

[in] TRANS

          TRANS is CHARACTER1

           On entry, TRANS specifies the operation to be performed as

           follows:

              TRANS = 'N' or 'n'   x := Ax.

              TRANS = 'T' or 't'   x := ATx.

              TRANS = 'C' or 'c'   x := ATx.

[in] DIAG

          DIAG is CHARACTER1

           On entry, DIAG specifies whether or not A is unit

           triangular as follows:

              DIAG = 'U' or 'u'   A is assumed to be unit triangular.

              DIAG = 'N' or 'n'   A is not assumed to be unit

                                  triangular.

[in] N

          N is INTEGER

           On entry, N specifies the order of the matrix A.

           N must be at least zero.

[in] A

          A is DOUBLE PRECISION array, 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.

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in the calling (sub) program. LDA must be at least

           max( 1, n ).

[in,out] X

          X is DOUBLE PRECISION array, 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.

[in] INCX

          INCX is INTEGER

           On entry, INCX specifies the increment for the elements of

           X. INCX must not be zero.

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date December 2016

Group: double\_blas\_level2

Further Details:

=====================

  Level 2 Blas routine.

  The vector and matrix arguments are not referenced when N = 0, or M = 0

  -- Written on 22-October-1986.

     Jack Dongarra, Argonne National Lab.

     Jeremy Du Croz, Nag Central Office.

     Sven Hammarling, Nag Central Office.

     Richard Hanson, Sandia National Labs.

## DTRSM

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DTRSM(SIDE,UPLO,TRANSA,DIAG,M,N,ALPHA,A,LDA,B,LDB)

       .. Scalar Arguments ..

       DOUBLE PRECISION ALPHA

       INTEGER LDA,LDB,M,N

       CHARACTER DIAG,SIDE,TRANSA,UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),B(LDB,)

       ..

Purpose:

=============

 DTRSM  solves one of the matrix equations

    op( A )X = alphaB,   or   Xop( A ) = alphaB,

 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 ) = AT.

 The matrix X is overwritten on B.

Arguments:

==========

[in] SIDE

          SIDE is CHARACTER1

           On entry, SIDE specifies whether op( A ) appears on the left

           or right of X as follows:

              SIDE = 'L' or 'l'   op( A )X = alphaB.

              SIDE = 'R' or 'r'   Xop( A ) = alphaB.

[in] UPLO

          UPLO is CHARACTER1

           On entry, UPLO specifies whether the matrix A is an upper or

           lower triangular matrix as follows:

              UPLO = 'U' or 'u'   A is an upper triangular matrix.

              UPLO = 'L' or 'l'   A is a lower triangular matrix.

[in] TRANSA

          TRANSA is CHARACTER1

           On entry, 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 ) = AT.

              TRANSA = 'C' or 'c'   op( A ) = AT.

[in] DIAG

          DIAG is CHARACTER1

           On entry, DIAG specifies whether or not A is unit triangular

           as follows:

              DIAG = 'U' or 'u'   A is assumed to be unit triangular.

              DIAG = 'N' or 'n'   A is not assumed to be unit

                                  triangular.

[in] M

          M is INTEGER

           On entry, M specifies the number of rows of B. M must be at

           least zero.

[in] N

          N is INTEGER

           On entry, N specifies the number of columns of B.  N must be

           at least zero.

[in] ALPHA

          ALPHA is DOUBLE PRECISION.

           On entry,  ALPHA specifies the scalar  alpha. When  alpha is

           zero then  A is not referenced and  B need not be set before

           entry.

[in] A

          A is DOUBLE PRECISION array, dimension ( LDA, k ),

           where k is m when SIDE = 'L' or 'l'

             and k 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.

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in the calling (sub) program.  When  SIDE = 'L' or 'l'  then

           LDA  must be at least  max( 1, m ),  when  SIDE = 'R' or 'r'

           then LDA must be at least max( 1, n ).

[in,out] B

          B is DOUBLE PRECISION array, dimension ( LDB, N )

           Before entry,  the leading  m by n part of the array  B must

           contain  the  right-hand  side  matrix  B,  and  on exit  is

           overwritten by the solution matrix  X.

[in] LDB

          LDB is INTEGER

           On entry, LDB specifies the first dimension of B as declared

           in  the  calling  (sub)  program.   LDB  must  be  at  least

           max( 1, m ).

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date December 2016

Group: double\_blas\_level3

Further Details:

=====================

  Level 3 Blas routine.

  -- Written on 8-February-1989.

     Jack Dongarra, Argonne National Laboratory.

     Iain Duff, AERE Harwell.

     Jeremy Du Croz, Numerical AlgorithmsGroup: Ltd.

     Sven Hammarling, Numerical AlgorithmsGroup: Ltd.

## DTRSV

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       SUBROUTINE DTRSV(UPLO,TRANS,DIAG,N,A,LDA,X,INCX)

       .. Scalar Arguments ..

       INTEGER INCX,LDA,N

       CHARACTER DIAG,TRANS,UPLO

       ..

       .. Array Arguments ..

       DOUBLE PRECISION A(LDA,),X()

       ..

Purpose:

=============

 DTRSV  solves one of the systems of equations

    Ax = b,   or   ATx = 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.

 No test for singularity or near-singularity is included in this

 routine. Such tests must be performed before calling this routine.

Arguments:

==========

[in] UPLO

          UPLO is CHARACTER1

           On entry, UPLO specifies whether the matrix is an upper or

           lower triangular matrix as follows:

              UPLO = 'U' or 'u'   A is an upper triangular matrix.

              UPLO = 'L' or 'l'   A is a lower triangular matrix.

[in] TRANS

          TRANS is CHARACTER1

           On entry, TRANS specifies the equations to be solved as

           follows:

              TRANS = 'N' or 'n'   Ax = b.

              TRANS = 'T' or 't'   ATx = b.

              TRANS = 'C' or 'c'   ATx = b.

[in] DIAG

          DIAG is CHARACTER1

           On entry, DIAG specifies whether or not A is unit

           triangular as follows:

              DIAG = 'U' or 'u'   A is assumed to be unit triangular.

              DIAG = 'N' or 'n'   A is not assumed to be unit

                                  triangular.

[in] N

          N is INTEGER

           On entry, N specifies the order of the matrix A.

           N must be at least zero.

[in] A

          A is DOUBLE PRECISION array, 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.

[in] LDA

          LDA is INTEGER

           On entry, LDA specifies the first dimension of A as declared

           in the calling (sub) program. LDA must be at least

           max( 1, n ).

[in,out] X

          X is DOUBLE PRECISION array, 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.

[in] INCX

          INCX is INTEGER

           On entry, INCX specifies the increment for the elements of

           X. INCX must not be zero.

  Level 2 Blas routine.

  -- Written on 22-October-1986.

     Jack Dongarra, Argonne National Lab.

     Jeremy Du Croz, Nag Central Office.

     Sven Hammarling, Nag Central Office.

     Richard Hanson, Sandia National Labs.

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date December 2016

Group: double\_blas\_level1

## DZASUM

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       DOUBLE PRECISION FUNCTION DZASUM(N,ZX,INCX)

       .. Scalar Arguments ..

       INTEGER INCX,N

       ..

       .. Array Arguments ..

       COMPLEX16 ZX()

       ..

Purpose:

=============

    DZASUM takes the sum of the (|Re(.)| + |Im(.)|)'s of a complex vector and

    returns a single precision result.

Arguments:

==========

[in] N

          N is INTEGER

         number of elements in input vector(s)

[in,out] ZX

          ZX is COMPLEX16 array, dimension ( 1 + ( N - 1 )abs( INCX ) )

[in] INCX

          INCX is INTEGER

         storage spacing between elements of ZX

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date November 2017

Group: double\_blas\_level1

Further Details:

=====================

     jack dongarra, 3/11/78.

     modified 3/93 to return if incx .le. 0.

     modified 12/3/93, array(1) declarations changed to array()

## DZNRM2

=========== DOCUMENTATION ===========

 Online html documentation available at

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

Definition:

===========

       DOUBLE PRECISION FUNCTION DZNRM2(N,X,INCX)

       .. Scalar Arguments ..

       INTEGER INCX,N

       ..

       .. Array Arguments ..

       COMPLEX16 X()

       ..

Purpose:

=============

 DZNRM2 returns the euclidean norm of a vector via the function

 name, so that

    DZNRM2 := sqrt( xHx )

Arguments:

==========

[in] N

          N is INTEGER

         number of elements in input vector(s)

[in] X

          X is COMPLEX16 array, dimension (N)

         complex vector with N elements

[in] INCX

          INCX is INTEGER

         storage spacing between elements of X

Authors:

========

 \author Univ. of Tennessee

 \author Univ. of California Berkeley

 \author Univ. of Colorado Denver

 \author NAG Ltd.

 \date November 2017

Group: double\_blas\_level1

Further Details:

=====================

  -- This version written on 25-October-1982.

     Modified on 14-October-1993 to inline the call to ZLASSQ.

     Sven Hammarling, Nag Ltd.

# 3 SINGLE