NAG Fortran Library Routine Document

F06YJF (DTRSM)

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

F06YJF (DTRSM) performs one of the matrix-matrix operations

$$B \leftarrow \alpha A^{-1}B, \quad B \leftarrow \alpha A^{-T}B,$$
$$B \leftarrow \alpha BA^{-1} \quad \text{or} \quad B \leftarrow \alpha BA^{-T},$$

where A is a real triangular matrix, B is an m by n real matrix, and α is a real scalar. A^{-T} denotes $\left(A^T\right)^{-1}$ or equivalently $\left(A^{-1}\right)^T$.

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

2 Specification

```
SUBROUTINE F06YJF (SIDE, UPLO, TRANSA, DIAG, M, N, ALPHA, A, LDA, B,
1                  LDB)
INTEGER            M, N, LDA, LDB
double precision   ALPHA, A(LDA,*), B(LDB,*)
CHARACTER*1        SIDE, UPLO, TRANSA, DIAG
```

The routine may be called by its BLAS name ***dtrsm***.

3 Description

None.

4 References

None.

5 Parameters

1: SIDE – CHARACTER*1 *Input*

On entry: specifies whether B is operated on from the left or the right, as follows:

if SIDE = 'L', B is pre-multiplied from the left;
if SIDE = 'R', B is post-multiplied from the right.

Constraint: SIDE = 'L' or 'R'.

2: UPLO – CHARACTER*1 *Input*

On entry: specifies whether A is upper or lower triangular as follows:

if UPLO = 'U', A is upper triangular;
if UPLO = 'L', A is lower triangular.

Constraint: UPLO = 'U' or 'L'.

3: TRANSA – CHARACTER*1 *Input*

On entry: specifies whether the operation involves A^{-1} or A^{-T}, as follows:

if TRANSA = 'N', it involves A^{-1};
if TRANSA = 'T' or 'C', it involves A^{-T}.

Constraint: TRANSA = 'N', 'T' or 'C'.

4: DIAG – CHARACTER*1 *Input*

On entry: specifies whether A has non-unit or unit diagonal elements, as follows:

if DIAG = 'N', the diagonal elements are stored explicitly;
if DIAG = 'U', the diagonal elements are assumed to be 1, and are not referenced.

Constraint: DIAG = 'N' or 'U'.

5: M – INTEGER *Input*

On entry: m, the number of rows of the matrix B; the order of A if SIDE = 'L'.

Constraint: $\mathrm{M} \geq 0$.

6: N – INTEGER *Input*

On entry: n, the number of columns of the matrix B; the order of A if SIDE = 'R'.

Constraint: $\mathrm{N} \geq 0$.

7: ALPHA – ***double precision*** *Input*

On entry: the scalar α.

8: A(LDA,$*$) – ***double precision*** array *Input*

Note: the second dimension of the array A must be at least $\max(1,\mathrm{M})$ if SIDE = 'L' and at least $\max(1,\mathrm{N})$ if SIDE = 'R'.

On entry: the triangular matrix A; A is m by m if SIDE = 'L', or n by n if SIDE = 'R'. If UPLO = 'U', A is upper triangular and the elements of the array below the diagonal are not referenced; if UPLO = 'L', A is lower triangular and the elements of the array above the diagonal are not referenced. If DIAG = 'U', the diagonal elements of A are not referenced, but are assumed to be 1.

9: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F06YJF (DTRSM) is called.

Constraint: $\mathrm{LDA} \geq \max(1,\mathrm{M})$ if SIDE = 'L'; $\mathrm{LDA} \geq \max(1,\mathrm{N})$ if SIDE = 'R'.

10: B(LDB,$*$) – ***double precision*** array *Input/Output*

Note: the second dimension of the array B must be at least $\max(1,\mathrm{N})$.

On entry: the m by n matrix B. If ALPHA = 0, B need not be set.

On exit: the updated matrix B.

11: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F06YJF (DTRSM) is called.

Constraint: $\mathrm{LDB} \geq \max(1,\mathrm{M})$.

6 Error Indicators and Warnings

None.