NAG C Library Function Document

nag_dtrsm (f16yjc)

1   Purpose

nag_dtrsm (f16yjc) solves one of the matrix equations

$$TX = \alpha B, \quad T^T X = \alpha B, \quad XT = \alpha B \quad \text{or} \quad XT^T = \alpha B,$$

where X and B are m by n real matrices and T is a real triangular matrix.

2   Specification

```
void nag_dtrsm (Nag_OrderType order, Nag_SideType side, Nag_UploType uplo,
    Nag_TransType transt, Nag_DiagType diag, Integer m, Integer n, double alpha,
    const double t[], Integer pdt, double b[], Integer pdb, NagError *fail)
```

3   Description

nag_dtrsm (f16yjc) performs one of the matrix-matrix operations

$$B \leftarrow \alpha T^{-1}B, \qquad B \leftarrow \alpha T^{-T}B,$$
$$B \leftarrow \alpha BT^{-1} \quad \text{or} \quad B \leftarrow \alpha BT^{-T},$$

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

4   References

The BLAS Technical Forum Standard (2001) *www.netlib.org/blas/blast-forum*

5   Parameters

1:    **order** – Nag_OrderType   *Input*

On entry: the **order** parameter specifies the two-dimensional storage scheme being used, i.e., row-major ordering or column-major ordering. C language defined storage is specified by **order** = **Nag_RowMajor**. See Section 2.2.1.4 of the Essential Introduction for a more detailed explanation of the use of this parameter.

Constraint: **order** = **Nag_RowMajor** or **Nag_ColMajor**.

2:    **side** – Nag_SideType   *Input*

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

  if **side** = **Nag_LeftSide**, B is pre-multiplied from the left;

  if **side** = **Nag_RightSide**, B is post-multiplied from the right.

Constraint: **side** = **Nag_LeftSide** or **Nag_RightSide**.

3:    **uplo** – Nag_UploType   *Input*

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

  if **uplo** = **Nag_Upper**, T is upper triangular;

  if **uplo** = **Nag_Lower**, T is lower triangular.

Constraint: **uplo** = **Nag_Upper** or **Nag_Lower**.

4:     **transt** – Nag_TransType   *Input*

On entry: specifies the operation to be performed as follows:

if **side** = **Nag_LeftSide** and **transt** = **Nag_Trans** or **Nag_ConjTrans**, $B \leftarrow \alpha T^{-T} B$;

if **side** = **Nag_LeftSide** and **transt** = **Nag_NoTrans**, $B \leftarrow \alpha T^{-1} B$;

if **side** = **Nag_RightSide** and **transt** = **Nag_Trans** or **Nag_ConjTrans**, $B \leftarrow \alpha B T^{-T}$;

if **side** = **Nag_RightSide** and **transt** = **Nag_NoTrans**, $B \leftarrow \alpha B T^{-1}$.

Constraint: **transt** = **Nag_NoTrans**, **Nag_Trans** or **Nag_ConjTrans**.

5:     **diag** – Nag_DiagType   *Input*

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

if **diag** = **Nag_NonUnitDiag**, the diagonal elements are stored explicitly;

if **diag** = **Nag_UnitDiag**, the diagonal elements are assumed to be 1, and are not referenced.

Constraint: **diag** = **Nag_NonUnitDiag** or **Nag_UnitDiag**.

6:     **m** – Integer   *Input*

On entry: m, the number of rows of the matrix B; the order of T if **side** = **Nag_LeftSide**.

Constraint: **m** ≥ 0.

7:     **n** – Integer   *Input*

On entry: n, the number of columns of the matrix B; the order of T if **side** = **Nag_RightSide**.

Constraint: **n** ≥ 0.

8:     **alpha** – double   *Input*

On entry: the scalar α.

9:     **t**$[dim]$ – const double   *Input*

Note: the dimension, dim, of the array **t** must be at least $\max(1, \mathbf{pdt} \times \mathbf{m})$ when **side** = **Nag_LeftSide** and at least $\max(1, \mathbf{pdt} \times \mathbf{n})$ when **side** = **Nag_RightSide**.

If **order** = **Nag_ColMajor**, the (i, j)th element of the matrix T is stored in **t**$[(j-1) \times \mathbf{pdt} + i - 1]$ and if **order** = **Nag_RowMajor**, the (i, j)th element of the matrix T is stored in **t**$[(i-1) \times \mathbf{pdt} + j - 1]$.

On entry: the m by m triangular matrix T if **side** = **Nag_LeftSide** or n by n triangular matrix T if **side** = **Nag_RightSide**. If **uplo** = **Nag_Upper**, T is upper triangular and the elements of the array below the diagonal are not referenced; if **uplo** = **Nag_Lower**, T is lower triangular and the elements of the array above the diagonal are not referenced. If **diag** = **Nag_UnitDiag**, the diagonal elements of T are not referenced, but are assumed to be 1.

10:    **pdt** – Integer   *Input*

On entry: the stride separating row or column elements (depending on the value of **order**) of the matrix T in the array **t**.

Constraints:

if **side** = **Nag_LeftSide**, **pdt** $\geq \max(1, \mathbf{m})$;
if **side** = **Nag_RightSide**, **pdt** $\geq \max(1, \mathbf{n})$.

11:    **b**$[dim]$ – double   *Input/Output*

Note: the dimension, dim, of the array **b** must be at least $\max(1, \mathbf{pdb} \times \mathbf{n})$ when **order** = **Nag_ColMajor** and at least $\max(1, \mathbf{pdb} \times \mathbf{m})$ when **order** = **Nag_RowMajor**.

If **order** = **Nag_ColMajor**, the (i, j)th element of the matrix B is stored in $\mathbf{b}[(j − 1) \times \mathbf{pdb} + i − 1]$ and if **order** = **Nag_RowMajor**, the (i, j)th element of the matrix B is stored in $\mathbf{b}[(i − 1) \times \mathbf{pdb} + j − 1]$.

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

On exit: the updated matrix B.

12:     **pdb** – Integer   *Input*

On entry: the stride separating matrix row or column elements (depending on the value of **order**) in the array **b**.

Constraints:

> if **order** = **Nag_ColMajor**, **pdb** $\geq$ max$(1, \mathbf{m})$;
> if **order** = **Nag_RowMajor**, **pdb** $\geq$ max$(1, \mathbf{n})$.

13:     **fail** – NagError *   *Input/Output*

The NAG error parameter (see the Essential Introduction).

6   Error Indicators and Warnings

NE_INT

On entry, $\mathbf{m} = \langle value \rangle$.
Constraint: $\mathbf{m} \geq 0$.

On entry, $\mathbf{n} = \langle value \rangle$.
Constraint: $\mathbf{n} \geq 0$.

On entry, $\mathbf{pdt} = \langle value \rangle$.
Constraint: $\mathbf{pdt} \geq$ max$(1, \mathbf{n})$.

On entry, $\mathbf{pdb} = \langle value \rangle$.
Constraint: $\mathbf{pdb} \geq$ max$(1, \mathbf{m})$.

On entry, $\mathbf{pdb} = \langle value \rangle$.
Constraint: $\mathbf{pdb} \geq$ max$(1, \mathbf{n})$.

NE_BAD_PARAM

On entry, parameter $\langle value \rangle$ had an illegal value.

7   Accuracy

The BLAS standard requires accurate implementations which avoid unnecessary over/underflow (see section 2.7 of The BLAS Technical Forum Standard (2001)).

8   Further Comments

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

9   Example

None.