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

```c
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

\[ B \leftarrow \alpha T^{-1}B, \quad B \leftarrow \alpha T^{-T}B, \]

\[ B \leftarrow \alpha BT^{-1}, \quad B \leftarrow \alpha BT^{-T}, \]

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

4 References


5 Parameters

1:  order  – Nag_OrderType  
    \[ \text{Input} \]
    On entry: the \text{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 \text{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: \text{order} = Nag_RowMajor or Nag_ColMajor.

2:  side  – Nag_SideType  
    \[ \text{Input} \]
    On entry: specifies whether \( B \) is operated on from the left or the right, as follows:
    if \text{side} = Nag_LeftSide, \( B \) is pre-multiplied from the left;
    if \text{side} = Nag_RightSide, \( B \) is post-multiplied from the right.
    Constraint: \text{side} = Nag_LeftSide or Nag_RightSide.

3:  uplo  – Nag_UploType  
    \[ \text{Input} \]
    On entry: specifies whether \( T \) is upper or lower triangular as follows:
    if \text{uplo} = Nag_Upper, \( T \) is upper triangular;
    if \text{uplo} = Nag_Lower, \( T \) is lower triangular.
    Constraint: \text{uplo} = Nag_Upper or Nag_Lower.
4: transt – Nag_TransType

\[\text{On entry:}\ \text{specifies the operation to be performed as follows:}\]

\[\begin{align*}
\text{if side = Nag_LeftSide and transt = Nag_Trans or Nag_ConjTrans, } & B \leftarrow \alpha T^{-T} B; \\
\text{if side = Nag_LeftSide and transt = Nag_NoTrans, } & B \leftarrow \alpha T^{-1} B; \\
\text{if side = Nag_RightSide and transt = Nag_Trans or Nag_ConjTrans, } & B \leftarrow \alpha BT^{-T}; \\
\text{if side = Nag_RightSide and transt = Nag_NoTrans, } & B \leftarrow \alpha BT^{-1}.
\end{align*}\]

\[\text{Constraint: transt = Nag_NoTrans, Nag_Trans or Nag_ConjTrans.}\]

5: diag – Nag_DiagType

\[\text{On entry:}\ \text{specifies whether A has non-unit or unit diagonal elements, as follows:}\]

\[\begin{align*}
\text{if diag = Nag_NonUnitDiag, } & \text{the diagonal elements are stored explicitly;} \\
\text{if diag = Nag_UnitDiag, } & \text{the diagonal elements are assumed to be 1, and are not referenced.}
\end{align*}\]

\[\text{Constraint: diag = Nag_NonUnitDiag or Nag_UnitDiag.}\]

6: m – Integer

\[\text{On entry:}\ m, \text{ the number of rows of the matrix } B; \text{ the order of } T \text{ if side = Nag_LeftSide.}\]

\[\text{Constraint: } m \geq 0.\]

7: n – Integer

\[\text{On entry:}\ n, \text{ the number of columns of the matrix } B; \text{ the order of } T \text{ if side = Nag_RightSide.}\]

\[\text{Constraint: } n \geq 0.\]

8: alpha – double

\[\text{On entry:}\ \text{the scalar } \alpha.\]

9: t[dim] – const double

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

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

\[\text{On entry:}\ \text{the } m \text{ by } m \text{ triangular matrix } T \text{ if side = Nag_LeftSide or } \text{n by } n \text{ triangular matrix } T \text{ if side = Nag_RightSide.}\]

\[\text{If uplo = Nag_Upper, } T \text{ is upper triangular and the elements of the array below the diagonal are not referenced; if uplo = Nag_Lower, } T \text{ is lower triangular and the elements of the array above the diagonal are not referenced.}\]

\[\text{If diag = Nag_UnitDiag, the diagonal elements of } T \text{ are not referenced, but are assumed to be 1.}\]

10: pdt – Integer

\[\text{On entry:}\ \text{the stride separating row or column elements (depending on the value of order) of the matrix } T \text{ in the array } t.\]

\[\text{Constraints:}\]

\[\begin{align*}
\text{if side = Nag_LeftSide, } & \text{pdt } \geq \text{max}(1, m); \\
\text{if side = Nag_RightSide, } & \text{pdt } \geq \text{max}(1, n).
\end{align*}\]

11: b[dim] – double

\[\text{Note:}\ \text{the dimension, } dim, \text{ of the array } b \text{ must be at least max}(1, \text{pdb} \times n) \text{ when order = Nag_ColMajor and at least max}(1, \text{pdb} \times m) \text{ when order = Nag_RowMajor.}\]
If order = Nag_ColMajor, the \((i, j)\)th element of the matrix \(B\) is stored in \(b[(j - 1) \times \text{pdb} + i - 1]\) and if order = Nag_RowMajor, the \((i, j)\)th element of the matrix \(B\) is stored in \(b[(i - 1) \times \text{pdb} + j - 1]\).

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

On exit: the updated matrix \(B\).

12: \(\text{pdb}\) – Integer

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

Constraints:

\[
\begin{align*}
\text{if order = Nag_ColMajor, pdb} & \geq \max(1, m); \\
\text{if order = Nag_RowMajor, pdb} & \geq \max(1, n).
\end{align*}
\]

13: \(\text{fail}\) – NagError *

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT

On entry, \(m = \langle value\rangle\).

Constraint: \(m \geq 0\).

On entry, \(n = \langle value\rangle\).

Constraint: \(n \geq 0\).

On entry, \(\text{pdt} = \langle value\rangle\).

Constraint: \(\text{pdt} \geq \max(1, n)\).

On entry, \(\text{pdb} = \langle value\rangle\).

Constraint: \(\text{pdb} \geq \max(1, m)\).

On entry, \(\text{pdb} = \langle value\rangle\).

Constraint: \(\text{pdb} \geq \max(1, n)\).

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.