NAG C Library Function Document

nag_dtrsv (f16pjc)

1 Purpose

nag_dtrsv (f16pjc) solves one of the systems of equations

$$Tx = \alpha b \quad \text{or} \quad T^T x = \alpha b,$$

where T is a real triangular matrix.

2 Specification

```
void nag_dtrsv (Nag_OrderType order, Nag_UploType uplo, Nag_TransType trans,
     Nag_DiagType diag, Integer n, double alpha, const double t[], Integer pdt,
     double x[], Integer incx, NagError *fail)
```

3 Description

nag_dtrsv (f16pjc) performs one of the matrix-vector operations

$$x \leftarrow \alpha T^{-1} x \quad \text{or} \quad x \leftarrow \alpha T^{-T} x,$$

where T is an n by n real triangular matrix, x is an n element real vector 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: **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**.

3: **trans** – Nag_TransType *Input*

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

if **trans** = **Nag_NoTrans**, $x \leftarrow \alpha T^{-1} x$;

if **trans** = **Nag_Trans** or **Nag_ConjTrans**, $x \leftarrow \alpha T^{-T} x$.

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

4: **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**.

5: **n** – Integer *Input*

On entry: n, the order of the matrix T.

Constraint: $\mathbf{n} \geq 0$.

6: **alpha** – double *Input*

On entry: the scalar α.

7: **t**[dim] – const double *Input*

Note: the dimension, dim, of the array **t** must be at least $\max(1, \mathbf{pdt} \times \mathbf{n})$.

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

On entry: the n by n triangular matrix T. 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.

8: **pdt** – Integer *Input*

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

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

9: **x**[dim] – double *Input/Output*

Note: the dimension, dim, of the array **x** must be at least $1 + (\mathbf{n} - 1)|\mathbf{incx}|$.

On entry: the right hand side vector b.

On exit: the solution vector x.

10: **incx** – Integer *Input*

On entry: the increment in the subscripts of **x** between successive elements of x.

Constraint: $\mathbf{incx} \neq 0$.

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

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT

On entry, **n** = ⟨*value*⟩.
Constraint: $\mathbf{n} \geq 0$.

On entry, **incx** = ⟨*value*⟩.
Constraint: $\mathbf{incx} \neq 0$.

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

NE_BAD_PARAM

On entry, parameter ⟨*value*⟩ 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.