NAG Library Function Document

nag_dtrtrs (f07tec)

1 Purpose

nag_dtrtrs (f07tec) solves a real triangular system of linear equations with multiple right-hand sides, $AX = B$ or $A^T X = B$.

2 Specification

```c
#include <nag.h>
#include <nagf07.h>

void nag_dtrtrs (Nag_OrderType order, Nag_UploType uplo,
                 Nag_TransType trans, Nag_DiagType diag, Integer n, Integer nrhs,
                 const double a[], Integer pda, double b[], Integer pdb, NagError *fail)
```

3 Description

nag_dtrtrs (f07tec) solves a real triangular system of linear equations $AX = B$ or $A^T X = B$.

4 References


5 Arguments

1: order – Nag_OrderType  

   *Input*

   On entry: the order argument 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 3.2.1.3 in the Essential Introduction for a more detailed explanation of the use of this argument.

   Constraint: order = Nag_RowMajor or Nag_ColMajor.

2: uplo – Nag_UploType  

   *Input*

   On entry: specifies whether $A$ is upper or lower triangular.

   uplo = Nag_Upper  
   $A$ is upper triangular.

   uplo = Nag_Lower  
   $A$ is lower triangular.

   Constraint: uplo = Nag_Upper or Nag_Lower.

3: trans – Nag_TransType  

   *Input*

   On entry: indicates the form of the equations.

   trans = Nag_NoTrans  
   The equations are of the form $AX = B$.
\( \mathbf{A} \mathbf{X} = \mathbf{B} \).

**Constraint:** \( \text{trans} = \text{Nag\_NoTrans, Nag\_Trans or Nag\_ConjTrans} \).

4: \(\text{diag} = \text{Nag\_DiagType} \)

*Input*

*On entry:* indicates whether \( \mathbf{A} \) is a nonunit or unit triangular matrix.

- \( \text{diag} = \text{Nag\_NonUnitDiag} \):
  - \( \mathbf{A} \) is a nonunit triangular matrix.

- \( \text{diag} = \text{Nag\_UnitDiag} \):
  - \( \mathbf{A} \) is a unit triangular matrix; the diagonal elements are not referenced and are assumed to be 1.

**Constraint:** \( \text{diag} = \text{Nag\_NonUnitDiag} \) or \( \text{Nag\_UnitDiag} \).

5: \( \mathbf{n} \)

*Input*

*On entry:* \( n \), the order of the matrix \( \mathbf{A} \).

**Constraint:** \( n \geq 0 \).

6: \( \mathbf{nrhs} \)

*Input*

*On entry:* \( r \), the number of right-hand sides.

**Constraint:** \( \mathbf{nrhs} \geq 0 \).

7: \( \mathbf{a} \)

*Input*

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

*On entry:* the \( n \) by \( n \) triangular matrix \( \mathbf{A} \).

- If \( \text{order} = \text{Nag\_ColMajor} \), \( A_{ij} \) is stored in \( \mathbf{a}[(j-1) \times \text{pda} + i - 1] \).
- If \( \text{order} = \text{Nag\_RowMajor} \), \( A_{ij} \) is stored in \( \mathbf{a}[(i-1) \times \text{pda} + j - 1] \).

- If \( \text{uplo} = \text{Nag\_Upper} \), the upper triangular part of \( \mathbf{A} \) must be stored and the elements of the array below the diagonal are not referenced.

- If \( \text{uplo} = \text{Nag\_Lower} \), the lower triangular part of \( \mathbf{A} \) must be stored and the elements of the array above the diagonal are not referenced.

- If \( \text{diag} = \text{Nag\_UnitDiag} \), the diagonal elements of \( \mathbf{A} \) are assumed to be 1, and are not referenced.

8: \( \text{pda} \)

*Input*

*On entry:* the stride separating row or column elements (depending on the value of \( \text{order} \)) of the matrix \( \mathbf{A} \) in the array \( \mathbf{a} \).

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

9: \( \mathbf{b} \)

*Input/Output*

*Note:* the dimension, \( \text{dim} \), of the array \( \mathbf{b} \) must be at least

- \( \max(1, \text{pdb} \times \mathbf{nrhs}) \) when \( \text{order} = \text{Nag\_ColMajor} \);
- \( \max(1, n \times \text{pdb}) \) when \( \text{order} = \text{Nag\_RowMajor} \).

*The \((i,j)\)th element of the matrix \( B \) is stored in*

- \( \mathbf{b}[(j-1) \times \text{pdb} + i - 1] \) when \( \text{order} = \text{Nag\_ColMajor} \);
- \( \mathbf{b}[(i-1) \times \text{pdb} + j - 1] \) when \( \text{order} = \text{Nag\_RowMajor} \).

*On entry:* the \( n \) by \( r \) right-hand side matrix \( \mathbf{B} \).

*On exit:* the \( n \) by \( r \) solution matrix \( \mathbf{X} \).
10:  **pdb** — Integer

   **Input**

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

   **Constraints**:

   if **order** = Nag_ColMajor, \( pdb \geq \max(1, n) \);
   if **order** = Nag_RowMajor, \( pdb \geq \max(1, nrhs) \).

11:  **fail** — NagError

   **Input/Output**

   The NAG error argument (see Section 3.6 in the Essential Introduction).

6  **Error Indicators and Warnings**

**NE_ALLOC_FAIL**

   Dynamic memory allocation failed.

   See Section 3.2.1.2 in the Essential Introduction for further information.

**NE_BAD_PARAM**

   On entry, argument \( \langle \text{value} \rangle \) had an illegal value.

**NE_INT**

   On entry, \( n = \langle \text{value} \rangle \).
   Constraint: \( n \geq 0 \).
   On entry, \( nrhs = \langle \text{value} \rangle \).
   Constraint: \( nrhs \geq 0 \).
   On entry, \( pda = \langle \text{value} \rangle \).
   Constraint: \( pda > 0 \).
   On entry, \( pdb = \langle \text{value} \rangle \).
   Constraint: \( pdb > 0 \).

**NE_INT_2**

   On entry, \( pda = \langle \text{value} \rangle \) and \( n = \langle \text{value} \rangle \).
   Constraint: \( pda \geq \max(1, n) \).
   On entry, \( pdb = \langle \text{value} \rangle \) and \( n = \langle \text{value} \rangle \).
   Constraint: \( pdb \geq \max(1, n) \).
   On entry, \( pdb = \langle \text{value} \rangle \) and \( nrhs = \langle \text{value} \rangle \).
   Constraint: \( pdb \geq \max(1, nrhs) \).

**NE_INTERNAL_ERROR**

   An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

   An unexpected error has been triggered by this function. Please contact NAG.

   See Section 3.6.6 in the Essential Introduction for further information.

**NE_NO_LICENCE**

   Your licence key may have expired or may not have been installed correctly.

   See Section 3.6.5 in the Essential Introduction for further information.
NE_SINGULAR

Element \(\text{value}\) of the diagonal is exactly zero. \(A\) is singular and the solution has not been computed.

7 Accuracy

The solutions of triangular systems of equations are usually computed to high accuracy. See Higham (1989).

For each right-hand side vector \(b\), the computed solution \(x\) is the exact solution of a perturbed system of equations \((A + E)x = b\), where

\[
|E| \leq c(n)\epsilon|A|,
\]

\(c(n)\) is a modest linear function of \(n\), and \(\epsilon\) is the machine precision.

If \(\hat{x}\) is the true solution, then the computed solution \(x\) satisfies a forward error bound of the form

\[
\frac{\|x - \hat{x}\|_\infty}{\|x\|_\infty} \leq c(n)\text{cond}(A,x)\epsilon, \quad \text{provided} \quad c(n)\text{cond}(A,x)\epsilon < 1,
\]

where \(\text{cond}(A,x) = \|\|A^{-1}\|\|A\|\|\|_\infty / \|x\|_\infty\|

Note that \(\text{cond}(A,x) \leq \text{cond}(A) = \|\|A^{-1}\|\|A\|\|_\infty \leq \kappa_\infty(A)\); \(\text{cond}(A,x)\) can be much smaller than \(\text{cond}(A)\) and it is also possible for \(\text{cond}(A^T)\) to be much larger (or smaller) than \(\text{cond}(A)\).

Forward and backward error bounds can be computed by calling \text{nag_dtrrfs} (f07thc), and an estimate for \(\kappa_\infty(A)\) can be obtained by calling \text{nag_dtrcon} (f07tgc) with \text{norm} = \text{Nag InfNorm}.

8 Parallelism and Performance

\text{nag_dtrtrs} (f07tec) is not threaded by NAG in any implementation.

\text{nag_dtrtrs} (f07tec) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this function. Please also consult the Users’ Note for your implementation for any additional implementation-specific information.

9 Further Comments

The total number of floating-point operations is approximately \(n^2r\).

The complex analogue of this function is \text{nag_ztrtrs} (f07tsc).

10 Example

This example solves the system of equations \(AX = B\), where

\[
A = \begin{pmatrix}
4.30 & 0.00 & 0.00 & 0.00 \\
-3.96 & -4.87 & 0.00 & 0.00 \\
0.40 & 0.31 & -8.02 & 0.00 \\
-0.27 & 0.07 & -5.95 & 0.12
\end{pmatrix}
\quad \text{and} \quad
B = \begin{pmatrix}
-12.90 & -21.50 \\
16.75 & 14.93 \\
-17.55 & 6.33 \\
-11.04 & 8.09
\end{pmatrix}.
\]
10.1 Program Text

/* nag_dtrtrs (f07tec) Example Program. */
* Copyright 2014 Numerical Algorithms Group.
*/
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagf07.h>
#include <nagx04.h>

int main(void)
{
    /* Scalars */
    Integer i, j, n, nrhs, pda, pdb;
    Integer exit_status = 0;
    Naq_UploType uplo;
    NagError fail;
    Nag_OrderType order;
    /* Arrays */
    char nag_enum_arg[40];
    double a[0], b[0];

#ifdef NAG_LOAD_FP
    /* The following line is needed to force the Microsoft linker to load floating point support */
    float force_loading_of_ms_float_support = 0;
#endif /* NAG_LOAD_FP */

#ifdef NAG_COLUMN_MAJOR
    #define A(I, J) a[(J-1)*pda +I-1]
    #define B(I, J) b[(J-1)*pdb +I-1]
    order = Nag_ColMajor;
#else
    #define A(I, J) a[(I-1)*pda +J-1]
    #define B(I, J) b[(I-1)*pdb +J-1]
    order = Nag_RowMajor;
#endif

INIT_FAIL(fail);
printf("nag_dtrtrs (f07tec) Example Program Results\n\n");

#ifdef _WIN32
    scanf_s("%*_n\n");
#else
    scanf("%*_n\n");
#endif

#ifdef NAG_COLUMN_MAJOR
    pda = n;
    pdb = n;
#else
    pda = n;
    pdb = nrhs;
#endif

    if (!a) NAG_ALLOC(n * n, double)) ||
    if (!b = NAG_ALLOC(n * nrhs, double))
    {
        printf("Allocation failure\n");
    }
exit_status = -1;
goto END;
}

/* Read A and B from data file */

#ifdef _WIN32
    scanf_s(" %39s%[\n] ", nag_enum_arg, _countof(nag_enum_arg));
#else
    scanf(" %39s%[\n] ", nag_enum_arg);
#endif

/* nag_enum_name_to_value (x04nac).
* Converts NAG enum member name to value
*/
uplo = (Nag_UploType) nag_enum_name_to_value(nag_enum_arg);

if (uplo == Nag_Upper)
{
    for (i = 1; i <= n; ++i)
    {
        for (j = i; j <= n; ++j)
        {
            #ifdef _WIN32
                scanf_s("%lf", &A(i, j));
            #else
                scanf("%lf", &A(i, j));
            #endif
        }
        #ifdef _WIN32
            scanf_s("%*[\n] ");
        #else
            scanf("%*[\n] ");
        #endif
    }
}
else
{
    for (i = 1; i <= n; ++i)
    {
        for (j = 1; j <= i; ++j)
        {
            #ifdef _WIN32
                scanf_s("%lf", &A(i, j));
            #else
                scanf("%lf", &A(i, j));
            #endif
        }
        #ifdef _WIN32
            scanf_s("%*[\n] ");
        #else
            scanf("%*[\n] ");
        #endif
    }
}

for (i = 1; i <= nrhs; ++i)
{
    for (j = 1; j <= n; ++j)
    {
        #ifdef _WIN32
            scanf_s("%lf", &B(i, j));
        #else
            scanf("%lf", &B(i, j));
        #endif
    }
    #ifdef _WIN32
        scanf_s("%*[\n] ");
    #else
        scanf("%*[\n] ");
    #endif

    /* Compute solution */
    /* nag_dtrtrs (f07tec).
     * Solution of real triangular system of linear equations,
     * multiple right-hand sides
     */
}
*/
{nag_dtrtrs(order, uplo, Nag_NoTrans, Nag_NonUnitDiag, n, nrhs, a, pda, b, pdb, &fail);
if (fail.code != NE_NOERROR)
    {
        printf("Error from nag_dtrtrs (f07tec).\n%s\n", fail.message);
        exit_status = 1;
        goto END;
    }
/* Print solution */
/* nag_gen_real_mat_print (x04cac).
 * Print real general matrix (easy-to-use)
 */
fflush(stdout);
nag_gen_real_mat_print(order, Nag_GeneralMatrix, Nag_NonUnitDiag, n, nrhs, b, pdb, "Solution(s)", 0, &fail);
if (fail.code != NE_NOERROR)
    {
        printf("Error from nag_gen_real_mat_print (x04cac).\n%s\n", fail.message);
        exit_status = 1;
        goto END;
    }
END:
NAG_FREE(a);
NAG_FREE(b);
return exit_status;
}

10.2 Program Data

nag_dtrtrs (f07tec) Example Program Data
4 2 :Values of n and nrhs
Nag_Lower :Value of uplo
4.30
-3.96 -4.87
0.40 0.31 -8.02
-0.27 0.07 -5.95 0.12 :End of matrix A
-12.90 -21.50
16.75 14.93
-17.55 6.33
-11.04 8.09 :End of matrix B

10.3 Program Results

nag_dtrtrs (f07tec) Example Program Results

Solution(s)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-3.0000</td>
<td>-5.0000</td>
</tr>
<tr>
<td>2</td>
<td>-1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>2.0000</td>
<td>-1.0000</td>
</tr>
<tr>
<td>4</td>
<td>1.0000</td>
<td>6.0000</td>
</tr>
</tbody>
</table>