NAG Library Routine Document

F06WBF (DTFSM)

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

F06WBF (DTFSM) performs one of the matrix-matrix operations

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

where A is a real triangular matrix stored in Rectangular Full Packed (RFP) format, B is an m by n real matrix, and α is a real scalar. $A^{-\mathrm{T}}$ denotes $(A^{\mathrm{T}})^{-1}$ or equivalently $(A^{-1})^{\mathrm{T}}$.

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 F06WBF (TRANSR, SIDE, UPLO, TRANS, DIAG, M, N, ALPHA, AR, B,    &
                   LDB)

INTEGER             M, N, LDB
REAL (KIND=nag_wp) ALPHA, AR(*), B(LDB,*)
CHARACTER(1)        TRANSR, SIDE, UPLO, TRANS, DIAG
```

The routine may be called by its LAPACK name ***dtfsm***.

3 Description

F06WBF (DTFSM) solves (for X) a triangular linear system of one of the forms

$$AX = \alpha B, \quad A^{\mathrm{T}}X = \alpha B, \quad \text{or} \quad XA = \alpha B, \quad XA^{\mathrm{T}} = \alpha B,$$

where A is a real triangular matrix stored in RFP format, B, X are m by n real matrices, and α is a real scalar. The RFP storage format is described in Section 3.3.3 in the F07 Chapter Introduction.

4 References

Gustavson F G, Waśniewski J, Dongarra J J and Langou J (2010) Rectangular full packed format for Cholesky's algorithm: factorization, solution, and inversion *ACM Trans. Math. Software* **37, 2**

5 Parameters

1: TRANSR – CHARACTER(1) *Input*

On entry: specifies whether the RFP representation of A is normal or transposed.

TRANSR = 'N'
The matrix A is stored in normal RFP format.

TRANSR = 'T'
The matrix A is stored in transposed RFP format.

Constraint: TRANSR = 'N' or 'T'.

2: SIDE – CHARACTER(1) *Input*

On entry: specifies whether B is operated on from the left or the right, or similarly whether A (or its transpose) appears to the left or right of the solution matrix in the linear system to be solved.

SIDE = 'L'
B is pre-multiplied from the left. The system to be solved has the form $AX = \alpha B$ or $A^{\mathrm{T}}X = \alpha B$.

SIDE = 'R'
B is post-multiplied from the right. The system to be solved has the form $XA = \alpha B$ or $XA^{\mathrm{T}} = \alpha B$.

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

3: UPLO – CHARACTER(1) *Input*

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

UPLO = 'U'
A is upper triangular.

UPLO = 'L'
A is lower triangular.

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

4: TRANS – CHARACTER(1) *Input*

On entry: specifies whether the operation involves A^{-1} or $A^{-\mathrm{T}}$, i.e., whether or not A is transposed in the linear system to be solved.

TRANS = 'N'
The operation involves A^{-1}, i.e., A is not transposed.

TRANS = 'T'
The operation involves $A^{-\mathrm{T}}$, i.e., A is transposed.

Constraint: TRANS = 'N' or 'T'.

5: DIAG – CHARACTER(1) *Input*

On entry: specifies whether A has nonunit or unit diagonal elements.

DIAG = 'N'
The diagonal elements of A are stored explicitly.

DIAG = 'U'
The diagonal elements of A are assumed to be 1, the corresponding elements of AR are not referenced.

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

6: M – INTEGER *Input*

On entry: m, the number of rows of the matrix B.

Constraint: M ≥ 0.

7: N – INTEGER *Input*

On entry: n, the number of columns of the matrix B.

Constraint: N ≥ 0.

8: ALPHA – REAL (KIND=nag_wp) *Input*

On entry: the scalar α.

9: AR$(*)$ – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array AR must be at least $\max(1, \mathrm{M} \times (\mathrm{M}+1)/2)$ if SIDE = 'L' and at least $\max(1, \mathrm{N} \times (\mathrm{N}+1)/2)$ if SIDE = 'R'.

On entry: A, the m by m triangular matrix A if SIDE = 'L' or the n by n triangular matrix A if SIDE = 'R', stored in RFP format (as specified by TRANSR). The storage format is described in detail in Section 3.3.3 in the F07 Chapter Introduction. If ALPHA = 0.0, AR is not referenced.

10: B(LDB, $*$) – REAL (KIND=nag_wp) 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, or similarly the solution matrix X.

11: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F06WBF (DTFSM) is called.

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

6 Error Indicators and Warnings

None.

7 Accuracy

Not applicable.

8 Parallelism and Performance

F06WBF (DTFSM) is not threaded by NAG in any implementation.

F06WBF (DTFSM) 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 routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

None.

10 Example

This example reads in the lower triangular part of a symmetric matrix A which it converts to RFP format. It also reads in α and a 6 by 4 matrix B and then performs the matrix-matrix operation $B \leftarrow \alpha A^{-1} B$.

10.1 Program Text

```
    Program f06wbfe

!     F06WBF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: dtfsm, dtrttf, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: alpha
      Integer                          :: i, ifail, info, lda, ldb, m, n
      Character (1)                    :: side, trans, transr, uplo
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), ar(:), b(:,:), work(:)
!     .. Executable Statements ..
      Write (nout,*) 'F06WBF Example Program Results'

!     Skip heading in data file
      Read (nin,*)

      Read (nin,*) m, n, uplo, transr, side, alpha, trans

      lda = m
      ldb = m
      Allocate (a(lda,m),ar((m*(m+1))/2),work(m),b(ldb,n))

!     Read upper or lower triangle of matrix A from data file

      If (uplo=='L' .Or. uplo=='l') Then
        Do i = 1, m
          Read (nin,*) a(i,1:i)
        End Do
      Else
        Do i = 1, m
          Read (nin,*) a(i,i:m)
        End Do
      End If

!     Read matrix B from data file

      Read (nin,*)(b(i,1:n),i=1,m)

!     Convert A to rectangular full packed storage in ar

!     The NAG name equivalent of dtrttf is f01vef
      Call dtrttf(transr,uplo,m,a,lda,ar,info)

      Write (nout,*)
      Flush (nout)

!     Perform the matrix-matrix operation

!     The NAG name equivalent of dtfsm is f06wbf
      Call dtfsm(transr,side,uplo,trans,'N',m,n,alpha,ar,b,ldb)

!     Print the result

      ifail = 0
      Call x04caf('General',' ',m,n,b,ldb,'The Solution',ifail)

    End Program f06wbfe
```

10.2 Program Data

```
F06WBF Example Program Data
  6 4 'L' 'N' 'L' 4.21 'N'      : M, N, UPLO, TRANSR, SIDE, ALPHA, TRANS
  1.0
  2.0  2.0
  3.0  3.0  3.0
  4.0  4.0  4.0  4.0
  5.0  5.0  5.0  5.0  5.0
  6.0  6.0  6.0  6.0  6.0  6.0  : Matrix A
  3.22   1.37   2.31   0.29
  1.64   1.80   0.38  -1.52
  1.87   2.87   2.02  -0.80
  5.20  -2.99  -0.91  -3.87
  1.83  -2.71  -2.81  -1.13
 -1.10  -0.63  -0.50   0.81     : End of matrix B
```

10.3 Program Results

```
 F06WBF Example Program Results

 The Solution
             1          2          3          4
 1     13.5562     5.7677     9.7251     1.2209
 2    -10.1040    -1.9787    -8.9252    -4.4205
 3     -0.8280     0.2386     2.0348     2.0769
 4      2.8488    -7.1745    -3.7925    -2.9505
 5     -3.9321     0.8652    -1.4082     3.1217
 6     -2.3127     1.8398     2.0152     1.5198
```