NAG Library Routine Document

F07WKF (DTFTRI)

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

F07WKF (DTFTRI) computes the inverse of a real triangular matrix stored in Rectangular Full Packed (RFP) format.

2 Specification

```
SUBROUTINE F07WKF (TRANSR, UPLO, DIAG, N, AR, INFO)
INTEGER            N, INFO
REAL (KIND=nag_wp) AR(N*(N+1)/2)
CHARACTER(1)       TRANSR, UPLO, DIAG
```

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

3 Description

F07WKF (DTFTRI) forms the inverse of a real triangular matrix A, stored using RFP format. The RFP storage format is described in Section 3.3.3 in the F07 Chapter Introduction. Note that the inverse of an upper (lower) triangular matrix is also upper (lower) triangular.

4 References

Du Croz J J and Higham N J (1992) Stability of methods for matrix inversion *IMA J. Numer. Anal.* **12** 1–19

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 Arguments

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: 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'.

3: DIAG – CHARACTER(1) *Input*

On entry: indicates whether A is a nonunit or unit triangular matrix.

DIAG = 'N'
A is a nonunit triangular matrix.

DIAG = 'U'
A is a unit triangular matrix; the diagonal elements are not referenced and are assumed to be 1.

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

4: N – INTEGER *Input*

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

Constraint: N ≥ 0.

5: AR(N $\times$ (N + 1)/2) – REAL (KIND=nag_wp) array *Input/Output*

On entry: the upper or lower triangular part (as specified by UPLO) of the n by n symmetric matrix A, in either normal or transposed RFP format (as specified by TRANSR). The storage format is described in detail in Section 3.3.3 in the F07 Chapter Introduction.

On exit: A is overwritten by A^{-1}, in the same storage format as A.

6: INFO – INTEGER *Output*

On exit: INFO = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

INFO < 0

If INFO $= -i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO > 0

Diagonal element $\langle value \rangle$ of A is exactly zero. A is singular its inverse cannot be computed.

7 Accuracy

The computed inverse X satisfies

$$|XA - I| \leq c(n)\epsilon|X||A|,$$

where $c(n)$ is a modest linear function of n, and ϵ is the ***machine precision***.

Note that a similar bound for $|AX - I|$ cannot be guaranteed, although it is almost always satisfied.

The computed inverse satisfies the forward error bound

$$\left|X - A^{-1}\right| \leq c(n)\epsilon\left|A^{-1}\right||A||X|.$$

See Du Croz and Higham (1992).

8 Parallelism and Performance

F07WKF (DTFTRI) 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

The total number of floating-point operations is approximately $\frac{1}{3}n^3$.

The complex analogue of this routine is F07WXF (ZTFTRI).

10 Example

This example computes the inverse of the matrix A, where

$$A = \begin{pmatrix} 4.30 & 0.00 & 0.00 & 0.00 \\ -3.96 & -4.87 & 0.00 & 0.00 \\ 0.40 & 0.31 & -8.02 & 0.00 \\ -0.27 & 0.07 & -5.95 & 0.12 \end{pmatrix}$$

and is stored using RFP format.

10.1 Program Text

```
    Program f07wkfe

!     F07WKF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dtftri, dtfttr, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, info, k, lar1, lda, lenar, &
                                          n, q
      Character (1)                    :: diag, transr, uplo
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), ar(:)
!     .. Executable Statements ..
      Write (nout,*) 'F07WKF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, uplo, transr, diag

      lenar = n*(n+1)/2
      lda = n
      Allocate (ar(lenar),a(lda,n))

!     Setup notional dimensions of RFP matrix AR
      k = n/2
      q = n - k
      If (transr=='N' .Or. transr=='n') Then
        lar1 = 2*k + 1
      Else
        lar1 = q
      End If

!     Read an RFP matrix into array AR
      Do i = 1, lar1
        Read (nin,*) ar(i:lenar:lar1)
      End Do

!     Compute inverse of A
!     The NAG name equivalent of dtftri is f07wkf
```

```
      Call dtftri(transr,uplo,diag,n,ar,info)

      Write (nout,*)
      Flush (nout)
      If (info==0) Then

!       Convert inverse to full array form, and print it
!       The NAG name equivalent of dtfttr is f01vgf
        Call dtfttr(transr,uplo,n,ar,a,lda,info)
        ifail = 0
        Call x04caf(uplo,'Nonunit',n,n,a,lda,'Inverse',ifail)
      Else
        Write (nout,*) 'A is singular'
      End If

    End Program f07wkfe
```

10.2 Program Data

```
F07WKF Example Program Data
  4       'L'   'N'   'N' : n, uplo, transr, diag
 -8.02   -5.95
  4.30    0.12
 -3.96   -4.87
  0.40    0.31
 -0.27    0.07            : AR
```

10.3 Program Results

```
 F07WKF Example Program Results

 Inverse
             1          2          3          4
 1      0.2326
 2     -0.1891    -0.2053
 3      0.0043    -0.0079    -0.1247
 4      0.8463    -0.2738    -6.1825     8.3333
```