NAG Library Routine Document

F07WJF (DPFTRI)

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

F07WJF (DPFTRI) computes the inverse of a real symmetric positive definite matrix using the Cholesky factorization computed by F07WDF (DPFTRF) stored in Rectangular Full Packed (RFP) format. The RFP storage format is described in Section 3.3.3 in the F07 Chapter Introduction.

2 Specification

```
SUBROUTINE F07WJF (TRANSR, UPLO, N, A, INFO)

INTEGER              N, INFO
REAL (KIND=nag_wp) A(N*(N+1)/2)
CHARACTER(1)         TRANSR, UPLO
```

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

3 Description

F07WJF (DPFTRI) is used to compute the inverse of a real symmetric positive definite matrix A, the routine must be preceded by a call to F07WDF (DPFTRF), which computes the Cholesky factorization of A.

If UPLO = 'U', $A = U^{\mathrm{T}}U$ and A^{-1} is computed by first inverting U and then forming $\left(U^{-1}\right)U^{-\mathrm{T}}$.

If UPLO = 'L', $A = LL^{\mathrm{T}}$ and A^{-1} is computed by first inverting L and then forming $L^{-\mathrm{T}}\left(L^{-1}\right)$.

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 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: UPLO – CHARACTER(1) *Input*

On entry: specifies how A has been factorized.

UPLO = 'U'
$A = U^{\mathrm{T}}U$, where U is upper triangular.

UPLO = 'L'

$A = LL^{\mathrm{T}}$, where L is lower triangular.

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

3: N – INTEGER *Input*

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

Constraint: $\mathrm{N} \geq 0$.

4: $\mathrm{A}(\mathrm{N} \times (\mathrm{N}+1)/2)$ – REAL (KIND=nag_wp) array *Input/Output*

On entry: the Cholesky factorization of A stored in RFP format, as returned by F07WDF (DPFTRF).

On exit: the factorization is overwritten by the n by n matrix A^{-1} stored in RFP format.

5: INFO – INTEGER *Output*

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

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

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

INFO > 0

If INFO $= i$, the ith diagonal element of the Cholesky factor is zero; the Cholesky factor is singular and the inverse of A cannot be computed.

7 Accuracy

The computed inverse X satisfies

$$\|XA - I\|_2 \leq c(n)\epsilon\kappa_2(A) \qquad \text{and} \qquad \|AX - I\|_2 \leq c(n)\epsilon\kappa_2(A),$$

where $c(n)$ is a modest function of n, ϵ is the ***machine precision*** and $\kappa_2(A)$ is the condition number of A defined by

$$\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2.$$

8 Further Comments

The total number of floating point operations is approximately $\frac{2}{3}n^3$.

The complex analogue of this routine is F07WWF (ZPFTRI).

9 Example

This example computes the inverse of the matrix A, where

$$A = \begin{pmatrix} 4.16 & -3.12 & 0.56 & -0.10 \\ -3.12 & 5.03 & -0.83 & 1.18 \\ 0.56 & -0.83 & 0.76 & 0.34 \\ -0.10 & 1.18 & 0.34 & 1.18 \end{pmatrix}.$$

Here A is symmetric positive definite, stored in RFP format, and must first be factorized by F07WDF (DPFTRF).

9.1 Program Text

```
    Program f07wjfe

!     F07WJF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dpftrf, dpftri, dtfttr, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: ifail, info, ldf, lena, n
      Character (1)                    :: transr, uplo
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:), f(:,:)
!     .. Executable Statements ..
      Write (nout,*) 'F07WJF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, uplo, transr

      lena = n*(n+1)/2
      ldf = n
      Allocate (a(lena),f(ldf,n))

!     Read A from data file
      Read (nin,*) a(1:lena)

!     Factorize A
!     The NAG name equivalent of dpftrf is f07wdf
      Call dpftrf(transr,uplo,n,a,info)

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

!       Compute inverse of A
!       The NAG name equivalent of dpftri is f07wjf
        Call dpftri(transr,uplo,n,a,info)

!       Convert inverse to full array form, and print it
!       The NAG name equivalent of dtfttr is f01vgf
        Call dtfttr(transr,uplo,n,a,f,ldf,info)
        ifail = 0
        Call x04caf(uplo,'Nonunit',n,n,f,ldf,'Inverse',ifail)

      Else
        Write (nout,*) 'A is not positive definite'
      End If

    End Program f07wjfe
```

9.2 Program Data

```
F07WJF Example Program Data
  4 'L' 'N'                                                   : n, uplo, transr
  0.76  4.16 -3.12  0.56 -0.10  0.34  1.18  5.03 -0.83  1.18   : RFP matrix A
```

9.3 Program Results

```
 F07WJF Example Program Results

 Inverse
             1          2          3          4
 1      0.6995
 2      0.7769     1.4239
 3      0.7508     1.8255     4.0688
 4     -0.9340    -1.8841    -2.9342     3.4978
```