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.

2 Specification

```
SUBROUTINE F07WJF (TRANSR, UPLO, N, AR, INFO)
INTEGER             N, INFO
REAL (KIND=nag_wp) AR(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, stored in RFP format. The RFP storage format is described in Section 3.3.3 in the F07 Chapter Introduction. 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 $(U^{-1})U^{-\mathrm{T}}$.

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

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 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: N ≥ 0.

4: AR(N $\times$ (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

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

The leading minor of order ⟨*value*⟩ is not positive definite and the factorization could not be completed. Hence A itself is not positive definite. This may indicate an error in forming the matrix A. There is no routine specifically designed to invert a symmetric matrix stored in RFP format which is not positive definite; the matrix must be treated as a full symmetric matrix, by calling F07MJF (DSYTRI).

7 Accuracy

The computed inverse X satisfies

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

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

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

8 Parallelism and Performance

F07WJF (DPFTRI) 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 $\frac{2}{3}n^3$.

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

10 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).

10.1 Program Text

```
    Program f07wjfe

!     F07WJF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. 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                          :: i, ifail, info, k, lar1, lda, lenar, &
                                          n, q
      Character (1)                    :: transr, uplo
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), ar(:)
!     .. Executable Statements ..
      Write (nout,*) 'F07WJF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, uplo, transr

      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

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

!       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 not positive definite'
      End If

    End Program f07wjfe
```

10.2 Program Data

```
F07WJF Example Program Data
  4       'L'       'N'   : n, uplo, transr

  0.76    0.34
  4.16    1.18
 -3.12    5.03
  0.56   -0.83
 -0.10    1.18            : AR
```

10.3 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
```