NAG Library Routine Document

F07FRF (ZPOTRF)

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

F07FRF (ZPOTRF) computes the Cholesky factorization of a complex Hermitian positive definite matrix.

2 Specification

```
SUBROUTINE F07FRF (UPLO, N, A, LDA, INFO)

INTEGER                 N, LDA, INFO
COMPLEX (KIND=nag_wp) A(LDA,*)
CHARACTER(1)            UPLO
```

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

3 Description

F07FRF (ZPOTRF) forms the Cholesky factorization of a complex Hermitian positive definite matrix A either as $A = U^{\mathrm{H}}U$ if UPLO = 'U' or $A = LL^{\mathrm{H}}$ if UPLO = 'L', where U is an upper triangular matrix and L is lower triangular.

4 References

Demmel J W (1989) On floating-point errors in Cholesky *LAPACK Working Note No. 14* University of Tennessee, Knoxville

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Parameters

1: UPLO – CHARACTER(1) *Input*

On entry: specifies whether the upper or lower triangular part of A is stored and how A is to be factorized.

UPLO = 'U'

The upper triangular part of A is stored and A is factorized as $U^{\mathrm{H}}U$, where U is upper triangular.

UPLO = 'L'

The lower triangular part of A is stored and A is factorized as LL^{H}, where L is lower triangular.

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

2: N – INTEGER *Input*

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

Constraint: N $\geq$ 0.

3: A(LDA,$*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array A must be at least max(1, N).

On entry: the n by n Hermitian positive definite matrix A.

If UPLO = 'U', the upper triangular part of A must be stored and the elements of the array below the diagonal are not referenced.

If UPLO = 'L', the lower triangular part of A must be stored and the elements of the array above the diagonal are not referenced.

On exit: the upper or lower triangle of A is overwritten by the Cholesky factor U or L as specified by UPLO.

4: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F07FRF (ZPOTRF) is called.

Constraint: LDA $\geq$ max(1, N).

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 leading minor of order i 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. To factorize a matrix which is not positive definite, call F07MRF (ZHETRF) instead.

7 Accuracy

If UPLO = 'U', the computed factor U is the exact factor of a perturbed matrix $A + E$, where

$$|E| \leq c(n)\epsilon \left|U^{\mathrm{H}}\right||U|,$$

$c(n)$ is a modest linear function of n, and ϵ is the ***machine precision***. If UPLO = 'L', a similar statement holds for the computed factor L. It follows that $\left|e_{ij}\right| \leq c(n)\epsilon\sqrt{a_{ii}a_{jj}}$.

8 Further Comments

The total number of real floating point operations is approximately $\frac{4}{3}n^3$.

A call to F07FRF (ZPOTRF) may be followed by calls to the routines:

F07FSF (ZPOTRS) to solve $AX = B$;

F07FUF (ZPOCON) to estimate the condition number of A;

F07FWF (ZPOTRI) to compute the inverse of A.

The real analogue of this routine is F07FDF (DPOTRF).

9 Example

This example computes the Cholesky factorization of the matrix A, where

$$A = \begin{pmatrix} 3.23 + 0.00i & 1.51 - 1.92i & 1.90 + 0.84i & 0.42 + 2.50i \\ 1.51 + 1.92i & 3.58 + 0.00i & -0.23 + 1.11i & -1.18 + 1.37i \\ 1.90 - 0.84i & -0.23 - 1.11i & 4.09 + 0.00i & 2.33 - 0.14i \\ 0.42 - 2.50i & -1.18 - 1.37i & 2.33 + 0.14i & 4.29 + 0.00i \end{pmatrix}.$$

9.1 Program Text

```
    Program f07frfe

!     F07FRF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04dbf, zpotrf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, info, lda, n
      Character (1)                    :: uplo
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F07FRF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      Allocate (a(lda,n))

!     Read A from data file

      Read (nin,*) uplo
      If (uplo=='U') Then
        Read (nin,*)(a(i,i:n),i=1,n)
      Else If (uplo=='L') Then
        Read (nin,*)(a(i,1:i),i=1,n)
      End If

!     Factorize A
!     The NAG name equivalent of zpotrf is f07frf
      Call zpotrf(uplo,n,a,lda,info)

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

!       Print factor

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04dbf(uplo,'Nonunit',n,n,a,lda,'Bracketed','F7.4','Factor', &
          'Integer',rlabs,'Integer',clabs,80,0,ifail)

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

    End Program f07frfe
```

9.2 Program Data

```
F07FRF Example Program Data
  4                                                          :Value of N
  'L'                                                        :Value of UPLO
 (3.23, 0.00)
 (1.51, 1.92) ( 3.58, 0.00)
 (1.90,-0.84) (-0.23,-1.11) ( 4.09, 0.00)
 (0.42,-2.50) (-1.18,-1.37) ( 2.33, 0.14) ( 4.29, 0.00)  :End of matrix A
```

9.3 Program Results

```
 F07FRF Example Program Results

 Factor
                     1                 2                 3                 4
 1  ( 1.7972, 0.0000)
 2  ( 0.8402, 1.0683) ( 1.3164, 0.0000)
 3  ( 1.0572,-0.4674) (-0.4702, 0.3131) ( 1.5604, 0.0000)
 4  ( 0.2337,-1.3910) ( 0.0834, 0.0368) ( 0.9360, 0.9900) ( 0.6603, 0.0000)
```