NAG Library Routine Document

F07MRF (ZHETRF)

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

F07MRF (ZHETRF) computes the Bunch–Kaufman factorization of a complex Hermitian indefinite matrix.

2 Specification

```
SUBROUTINE F07MRF (UPLO, N, A, LDA, IPIV, WORK, LWORK, INFO)

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

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

3 Description

F07MRF (ZHETRF) factorizes a complex Hermitian matrix A, using the Bunch–Kaufman diagonal pivoting method. A is factorized either as $A = PUDU^{\mathrm{H}}P^{\mathrm{T}}$ if UPLO = 'U' or $A = PLDL^{\mathrm{H}}P^{\mathrm{T}}$ if UPLO = 'L', where P is a permutation matrix, U (or L) is a unit upper (or lower) triangular matrix and D is an Hermitian block diagonal matrix with 1 by 1 and 2 by 2 diagonal blocks; U (or L) has 2 by 2 unit diagonal blocks corresponding to the 2 by 2 blocks of D. Row and column interchanges are performed to ensure numerical stability while keeping the matrix Hermitian.

This method is suitable for Hermitian matrices which are not known to be positive definite. If A is in fact positive definite, no interchanges are performed and no 2 by 2 blocks occur in D.

4 References

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 $PUDU^{\mathrm{H}}P^{\mathrm{T}}$, where U is upper triangular.

UPLO = 'L'
: The lower triangular part of A is stored and A is factorized as $PLDL^{\mathrm{H}}P^{\mathrm{T}}$, where L is lower triangular.

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

2: N – INTEGER *Input*

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

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

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

Note: the second dimension of the array A must be at least $\max(1, \text{N})$.

On entry: the n by n Hermitian indefinite 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 details of the block diagonal matrix D and the multipliers used to obtain the 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 F07MRF (ZHETRF) is called.

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

5: IPIV($*$) – INTEGER array *Output*

Note: the dimension of the array IPIV must be at least $\max(1, \text{N})$.

On exit: details of the interchanges and the block structure of D. More precisely,

if $\text{IPIV}(i) = k > 0$, d_{ii} is a 1 by 1 pivot block and the ith row and column of A were interchanged with the kth row and column;

if UPLO = 'U' and $\text{IPIV}(i-1) = \text{IPIV}(i) = -l < 0$, $\begin{pmatrix} d_{i-1,i-1} & \bar{d}_{i,i-1} \\ \bar{d}_{i,i-1} & d_{ii} \end{pmatrix}$ is a 2 by 2 pivot block and the $(i-1)$th row and column of A were interchanged with the lth row and column;

if UPLO = 'L' and $\text{IPIV}(i) = \text{IPIV}(i+1) = -m < 0$, $\begin{pmatrix} d_{ii} & d_{i+1,i} \\ d_{i+1,i} & d_{i+1,i+1} \end{pmatrix}$ is a 2 by 2 pivot block and the $(i+1)$th row and column of A were interchanged with the mth row and column.

6: WORK($\max(1, \text{LWORK})$) – COMPLEX (KIND=nag_wp) array *Workspace*

On exit: if INFO = 0, WORK(1) contains the minimum value of LWORK required for optimum performance.

7: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F07MRF (ZHETRF) is called, unless $\text{LWORK} = -1$, in which case a workspace query is assumed and the routine only calculates the optimal dimension of WORK (using the formula given below).

Suggested value: for optimum performance LWORK should be at least $\text{N} \times nb$, where nb is the ***block size***.

Constraint: $\text{LWORK} \geq 1$ or $\text{LWORK} = -1$.

8: 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 $= i$, $d(i, i)$ is exactly zero. The factorization has been completed, but the block diagonal matrix D is exactly singular, and division by zero will occur if it is used to solve a system of equations.

7 Accuracy

If UPLO = 'U', the computed factors U and D are the exact factors of a perturbed matrix $A + E$, where

$$|E| \leq c(n)\epsilon P|U||D||U^{\mathrm{H}}|P^{\mathrm{T}},$$

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

If UPLO = 'L', a similar statement holds for the computed factors L and D.

8 Further Comments

The elements of D overwrite the corresponding elements of A; if D has 2 by 2 blocks, only the upper or lower triangle is stored, as specified by UPLO.

The unit diagonal elements of U or L and the 2 by 2 unit diagonal blocks are not stored. The remaining elements of U or L are stored in the corresponding columns of the array A, but additional row interchanges must be applied to recover U or L explicitly (this is seldom necessary). If IPIV$(i) = i$, for $i = 1, 2, \ldots, n$ (as is the case when A is positive definite), then U or L is stored explicitly (except for its unit diagonal elements which are equal to 1).

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

A call to F07MRF (ZHETRF) may be followed by calls to the routines:

F07MSF (ZHETRS) to solve $AX = B$;

F07MUF (ZHECON) to estimate the condition number of A;

F07MWF (ZHETRI) to compute the inverse of A.

The real analogue of this routine is F07MDF (DSYTRF).

9 Example

This example computes the Bunch–Kaufman factorization of the matrix A, where

$$A = \begin{pmatrix} -1.36 + 0.00i & 1.58 + 0.90i & 2.21 - 0.21i & 3.91 + 1.50i \\ 1.58 - 0.90i & -8.87 + 0.00i & -1.84 - 0.03i & -1.78 + 1.18i \\ 2.21 + 0.21i & -1.84 + 0.03i & -4.63 + 0.00i & 0.11 + 0.11i \\ 3.91 - 1.50i & -1.78 - 1.18i & 0.11 - 0.11i & -1.84 + 0.00i \end{pmatrix}.$$

9.1 Program Text

```
    Program f07mrfe

!     F07MRF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04dbf, zhetrf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, info, lda, lwork, n
      Character (1)                    :: uplo
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), work(:)
      Integer, Allocatable             :: ipiv(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F07MRF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      lwork = 64*n
      Allocate (a(lda,n),work(lwork),ipiv(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 zhetrf is f07mrf
      Call zhetrf(uplo,n,a,lda,ipiv,work,lwork,info)

      Write (nout,*)
      Flush (nout)

!     Print details of factorization

!     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', &
        'Details of factorization','Integer',rlabs,'Integer',clabs,80,0,ifail)

!     Print pivot indices

      Write (nout,*)
      Write (nout,*) 'IPIV'
      Write (nout,99999) ipiv(1:n)

      If (info/=0) Write (nout,*) 'The factor D is singular'

99999 Format ((1X,I12,3I18))
    End Program f07mrfe
```

9.2 Program Data

```
F07MRF Example Program Data
  4                                                         :Value of N
  'L'                                                       :Value of UPLO
 (-1.36, 0.00)
 ( 1.58,-0.90) (-8.87, 0.00)
 ( 2.21, 0.21) (-1.84, 0.03) (-4.63, 0.00)
 ( 3.91,-1.50) (-1.78,-1.18) ( 0.11,-0.11) (-1.84, 0.00)  :End of matrix A
```

9.3 Program Results

```
 F07MRF Example Program Results

 Details of factorization
                    1                  2                  3                  4
 1  (-1.3600, 0.0000)
 2  ( 3.9100,-1.5000) (-1.8400, 0.0000)
 3  ( 0.3100, 0.0433) ( 0.5637, 0.2850) (-5.4176, 0.0000)
 4  (-0.1518, 0.3743) ( 0.3397, 0.0303) ( 0.2997, 0.1578) (-7.1028, 0.0000)

 IPIV
           -4                 -4                  3                  4
```