NAG Fortran Library Routine Document

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

2 Specification

```
SUBROUTINE F07MRF(UPLO, N, A, LDA, IPIV, WORK, LWORK, INFO)
ENTRY      chetrf (UPLO, N, A, LDA, IPIV, WORK, LWORK, INFO)
INTEGER           N, LDA, IPIV(*), LWORK, INFO
complex           A(LDA,*), WORK(*)
CHARACTER*1       UPLO
```

The ENTRY statement enables the routine to be called by its LAPACK name.

3 Description

This routine factorizes a complex Hermitian matrix A, using the Bunch–Kaufman diagonal pivoting method. A is factorized as either $A = PUDU^HP^T$ if UPLO = 'U', or $A = PLDL^HP^T$ 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: indicates whether the upper or lower triangular part of A is stored and how A has been factorized, as follows:

if UPLO = 'U', the upper triangular part of A is stored and A is factorized as $PUDU^HP^T$, where U is upper triangular;

if UPLO = 'L', the lower triangular part of A is stored and A is factorized as $PLDL^HP^T$, where L is lower triangular.

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

2: N – INTEGER *Input*

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

Constraint: N ≥ 0.

3: A(LDA,*) – ***complex*** array *Input/Output*

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

On entry: the n by n Hermitian matrix A. If UPLO = 'U', the upper triangle of A must be stored and the elements of the array below the diagonal are not referenced; if UPLO = 'L', the lower triangle 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 (CHETRF/ZHETRF) is called.

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

5: IPIV($*$) – INTEGER array *Output*

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

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

More precisely, if 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 IPIV$(i-1) =$ IPIV$(i) = -l < 0$, $\begin{pmatrix} d_{i-1,i-1} & \bar{d}_{i,i-1} \\ d_{i,i-1} & d_{ii} \end{pmatrix}$ 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 IPIV$(i) =$ IPIV$(i+1) = -m < 0$, $\begin{pmatrix} d_{ii} & \bar{d}_{i+1,i} \\ d_{i+1,i} & d_{i+1,i+1} \end{pmatrix}$ 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($*$) – ***complex*** array *Workspace*

Note: the dimension of the array WORK must be at least $\max(1, \mathrm{LWORK})$.

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 (CHETRF/ZHETRF) is called, unless 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 $\mathrm{N} \times nb$, where nb is the ***blocksize***.

Constraint: LWORK ≥ 1 or 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_{ii} 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 subsequently used to solve a system of linear equations or to compute A^{-1}.

7 Accuracy

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

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

$c(n)$ 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 this routine may be followed by calls to the routines:

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

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

F07MWF (CHETRI/ZHETRI) to compute the inverse of A.

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

9 Example

To compute 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

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     F07MRF Example Program Text
*     Mark 15 Release. NAG Copyright 1991.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDA, LWORK
      PARAMETER        (NMAX=8,LDA=NMAX,LWORK=64*NMAX)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, N
      CHARACTER        UPLO
*     .. Local Arrays ..
      complex          A(LDA,NMAX), WORK(LWORK)
```

```
      INTEGER          IPIV(NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Subroutines ..
      EXTERNAL         chetrf, X04DBF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F07MRF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read A from data file
*
         READ (NIN,*) UPLO
         IF (UPLO.EQ.'U') THEN
            READ (NIN,*) ((A(I,J),J=I,N),I=1,N)
         ELSE IF (UPLO.EQ.'L') THEN
            READ (NIN,*) ((A(I,J),J=1,I),I=1,N)
         END IF
*
*        Factorize A
*
         CALL chetrf(UPLO,N,A,LDA,IPIV,WORK,LWORK,INFO)
*
         WRITE (NOUT,*)
*
*        Print details of factorization
*
         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(I),I=1,N)
*
         IF (INFO.NE.0) WRITE (NOUT,*) 'The factor D is singular'
*
      END IF
      STOP
*
99999 FORMAT ((1X,I12,3I18))
      END
```

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