NAG Library Routine Document

F07MNF (ZHESV)

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

F07MNF (ZHESV) computes the solution to a complex system of linear equations

$$AX = B,$$

where A is an n by n Hermitian matrix and X and B are n by r matrices.

2 Specification

```
SUBROUTINE F07MNF (UPLO, N, NRHS, A, LDA, IPIV, B, LDB, WORK, LWORK, INFO)

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

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

3 Description

F07MNF (ZHESV) uses the diagonal pivoting method to factor A as $A = UDU^{\mathrm{H}}$ if UPLO = 'U' or $A = LDL^{\mathrm{H}}$ if UPLO = 'L', where U (or L) is a product of permutation and unit upper (lower) triangular matrices, and D is Hermitian and block diagonal with 1 by 1 and 2 by 2 diagonal blocks. The factored form of A is then used to solve the system of equations $AX = B$.

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia http://www.netlib.org/lapack/lug

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: if UPLO = 'U', the upper triangle of A is stored.

If UPLO = 'L', the lower triangle of A is stored.

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

2: N – INTEGER *Input*

On entry: n, the number of linear equations, i.e., the order of the matrix A.

Constraint: N ≥ 0.

3: NRHS – INTEGER *Input*

On entry: r, the number of right-hand sides, i.e., the number of columns of the matrix B.

Constraint: NRHS ≥ 0.

4: 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 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: if INFO $= 0$, the block diagonal matrix D and the multipliers used to obtain the factor U or L from the factorization $A = UDU^{\mathrm{H}}$ or $A = LDL^{\mathrm{H}}$ as computed by F07MRF (ZHETRF).

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F07MNF (ZHESV) is called.

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

6: 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 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$(i) = -l < 0$, $\begin{pmatrix} d_{i-1,i-1} & \bar{d}_{i,i-1} \\ \bar{d}_{i,i-1} & d_{ii} \end{pmatrix}$ 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 IPIV$(i) = $ 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.

7: B(LDB,$*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

Note: To solve the equations $Ax = b$, where b is a single right-hand side, B may be supplied as a one-dimensional array with length LDB $= \max(1, \text{N})$.

On entry: the n by r right-hand side matrix B.

On exit: if INFO $= 0$, the n by r solution matrix X.

8: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F07MNF (ZHESV) is called.

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

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

On exit: if INFO $= 0$, WORK(1) returns the optimal LWORK.

10: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F07MNF (ZHESV) is called.

LWORK ≥ 1, and for best performance LWORK $\geq \max(1, \mathrm{N} \times nb)$, where *nb* is the optimal block size for F07MRF (ZHETRF).

If LWORK $= -1$, a workspace query is assumed; the routine only calculates the optimal size of the WORK array, returns this value as the first entry of the WORK array, and no error message related to LWORK is issued.

11: 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 argument 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, so the solution could not be computed.

7 Accuracy

The computed solution for a single right-hand side, $\hat{x}$, satisfies an equation of the form

$$(A + E)\hat{x} = b,$$

where

$$\|E\|_1 = O(\epsilon)\|A\|_1$$

and ϵ is the ***machine precision***. An approximate error bound for the computed solution is given by

$$\frac{\|\hat{x} - x\|_1}{\|x\|_1} \leq \kappa(A)\frac{\|E\|_1}{\|A\|_1},$$

where $\kappa(A) = \left\|A^{-1}\right\|_1 \|A\|_1$, the condition number of A with respect to the solution of the linear equations. See Section 4.4 of Anderson *et al.* (1999) for further details.

F07MPF (ZHESVX) is a comprehensive LAPACK driver that returns forward and backward error bounds and an estimate of the condition number. Alternatively, F04CHF solves $Ax = b$ and returns a forward error bound and condition estimate. F04CHF calls F07MNF (ZHESV) to solve the equations.

8 Further Comments

The total number of floating point operations is approximately $\frac{4}{3}n^3 + 8n^2r$, where r is the number of right-hand sides.

The real analogue of this routine is F07MAF (DSYSV). The complex symmetric analogue of this routine is F07NNF (ZSYSV).

9 Example

This example solves the equations

$$Ax = b,$$

where A is the Hermitian matrix

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

and

$$b = \begin{pmatrix} 2.98 - 10.18i \\ -9.58 + 3.88i \\ -0.77 - 16.05i \\ 7.79 + 5.48i \end{pmatrix}.$$

Details of the factorization of A are also output.

9.1 Program Text

```
    Program f07mnfe

!     F07MNF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04dbf, zhesv
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nb = 64, nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, info, lda, lwork, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), b(:), work(:)
      Integer, Allocatable             :: ipiv(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F07MNF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      lwork = nb*n
      Allocate (a(lda,n),b(n),work(lwork),ipiv(n))

!     Read the upper triangular part of the matrix A from data file

      Read (nin,*)(a(i,i:n),i=1,n)

!     Read b from data file

      Read (nin,*) b(1:n)

!     Solve the equations Ax = b for x
!     The NAG name equivalent of zhesv is f06mnf
      Call zhesv('Upper',n,1,a,lda,ipiv,b,n,work,lwork,info)

      If (info==0) Then

!       Print solution

        Write (nout,*) 'Solution'
        Write (nout,99999) b(1:n)

!       Print details of factorization

        Write (nout,*)
        Flush (nout)
```

```
!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04dbf('Upper','Non-unit diagonal',n,n,a,lda,'Bracketed','F7.4', &
          'Details of the factorization','Integer',rlabs,'Integer',clabs,80,0, &
          ifail)

!       Print pivot indices

        Write (nout,*)
        Write (nout,*) 'Pivot indices'
        Write (nout,99998) ipiv(1:n)

      Else
        Write (nout,99997) 'The diagonal block ', info, ' of D is zero'
      End If

99999 Format ((3X,4(' (',F7.4,',',F7.4,')':)))
99998 Format (1X,7I11)
99997 Format (1X,A,I3,A)
    End Program f07mnfe
```

9.2 Program Data

```
F07MNF Example Program Data
   4                                                            :Value of N
 ( -1.84,  0.00) (  0.11, -0.11) ( -1.78, -1.18) (  3.91, -1.50)
                 ( -4.63 , 0.00) ( -1.84,  0.03) (  2.21,  0.21)
                                 ( -8.87,  0.00) (  1.58, -0.90)
                                                 ( -1.36 , 0.00) :End matrix A
 (  2.98,-10.18) ( -9.58,  3.88) ( -0.77,-16.05) (  7.79,  5.48) :End vector b
```

9.3 Program Results

```
 F07MNF Example Program Results

 Solution
    ( 2.0000, 1.0000) ( 3.0000,-2.0000) (-1.0000, 2.0000) ( 1.0000,-1.0000)

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

 Pivot indices
           1          2         -1         -1
```