

NAG Library Routine Document

F07MVF (ZHERFS)

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

F07MVF (ZHERFS) returns error bounds for the solution of a complex Hermitian indefinite system of linear equations with multiple right-hand sides, $AX = B$. It improves the solution by iterative refinement, in order to reduce the backward error as much as possible.

2 Specification

```

SUBROUTINE F07MVF(UPLO, N, NRHS, A, LDA, AF, LDAF, IPIV, B, LDB, X, LDX,
1             FERR, BERR, WORK, RWORK, INFO)
    INTEGER          N, NRHS, LDA, LDAF, IPIV(*), LDB, LDX, INFO
    double precision FERR(NRHS), BERR(NRHS), RWORK(N)
    complex*16       A(LDA,*), AF(LDAF,*), B(LDB,*), X(LDX,*), WORK(2*N)
    CHARACTER*1      UPLO

```

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

3 Description

F07MVF (ZHERFS) returns the backward errors and estimated bounds on the forward errors for the solution of a complex Hermitian indefinite system of linear equations with multiple right-hand sides $AX = B$. The routine handles each right-hand side vector (stored as a column of the matrix B) independently, so we describe the function of F07MVF (ZHERFS) in terms of a single right-hand side b and solution x .

Given a computed solution x , the routine computes the *component-wise backward error* β . This is the size of the smallest relative perturbation in each element of A and b such that x is the exact solution of a perturbed system

$$(A + \delta A)x = b + \delta b$$

$$|\delta a_{ij}| \leq \beta |a_{ij}| \quad \text{and} \quad |\delta b_i| \leq \beta |b_i|.$$

Then the routine estimates a bound for the *component-wise forward error* in the computed solution, defined by:

$$\max_i |x_i - \hat{x}_i| / \max_i |x_i|$$

where $\hat{x}$ is the true solution.

For details of the method, see the F07 Chapter Introduction.

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 is to be factorized.
UPLO = 'U'
The upper triangular part of A is stored and A is factorized as $PUDU^H P^T$, where U is upper triangular.
UPLO = 'L'
The lower triangular part of A is stored and A is factorized as $PLDL^H P^T$, where L is lower triangular.
Constraint: UPLO = 'U' or 'L'.
- 2: N – INTEGER *Input*
On entry: n , the order of the matrix A .
Constraint: $N \geq 0$.
- 3: NRHS – INTEGER *Input*
On entry: r , the number of right-hand sides.
Constraint: NRHS ≥ 0 .
- 4: A(LDA,*) – **complex*16** array *Input*
Note: the second dimension of the array A must be at least $\max(1, N)$.
On entry: the n by n original Hermitian matrix A as supplied to F07MRF (ZHETRF).
- 5: LDA – INTEGER *Input*
On entry: the first dimension of the array A as declared in the (sub)program from which F07MVF (ZHERFS) is called.
Constraint: LDA $\geq \max(1, N)$.
- 6: AF(LDAF,*) – **complex*16** array *Input*
Note: the second dimension of the array AF must be at least $\max(1, N)$.
On entry: details of the factorization of A , as returned by F07MRF (ZHETRF).
- 7: LDAF – INTEGER *Input*
On entry: the first dimension of the array AF as declared in the (sub)program from which F07MVF (ZHERFS) is called.
Constraint: LDAF $\geq \max(1, N)$.
- 8: IPIV(*) – INTEGER array *Input*
Note: the dimension of the array IPIV must be at least $\max(1, N)$.
On entry: details of the interchanges and the block structure of D , as returned by F07MRF (ZHETRF).

- 9: B(LDB,*) – **complex*16** array *Input*
Note: the second dimension of the array B must be at least $\max(1, \text{NRHS})$.
On entry: the n by r right-hand side matrix B .
- 10: LDB – INTEGER *Input*
On entry: the first dimension of the array B as declared in the (sub)program from which F07MVF (ZHERFS) is called.
Constraint: $\text{LDB} \geq \max(1, N)$.
- 11: X(LDX,*) – **complex*16** array *Input/Output*
Note: the second dimension of the array X must be at least $\max(1, \text{NRHS})$.
On entry: the n by r solution matrix X , as returned by F07MSF (ZHETRS).
On exit: the improved solution matrix X .
- 12: LDX – INTEGER *Input*
On entry: the first dimension of the array X as declared in the (sub)program from which F07MVF (ZHERFS) is called.
Constraint: $\text{LDX} \geq \max(1, N)$.
- 13: FERR(NRHS) – **double precision** array *Output*
On exit: $\text{FERR}(j)$ contains an estimated error bound for the j th solution vector, that is, the j th column of X , for $j = 1, 2, \dots, r$.
- 14: BERR(NRHS) – **double precision** array *Output*
On exit: $\text{BERR}(j)$ contains the component-wise backward error bound β for the j th solution vector, that is, the j th column of X , for $j = 1, 2, \dots, r$.
- 15: WORK($2 \times N$) – **complex*16** array *Workspace*
- 16: RWORK(N) – **double precision** array *Workspace*
- 17: INFO – INTEGER *Output*
On exit: $\text{INFO} = 0$ unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

$\text{INFO} < 0$

If $\text{INFO} = -i$, the i th parameter had an illegal value. An explanatory message is output, and execution of the program is terminated.

7 Accuracy

The bounds returned in FERR are not rigorous, because they are estimated, not computed exactly; but in practice they almost always overestimate the actual error.

8 Further Comments

For each right-hand side, computation of the backward error involves a minimum of $16n^2$ real floating-point operations. Each step of iterative refinement involves an additional $24n^2$ real operations. At most five steps of iterative refinement are performed, but usually only one or two steps are required.

Estimating the forward error involves solving a number of systems of linear equations of the form $Ax = b$; the number is usually 5 and never more than 11. Each solution involves approximately $8n^2$ real operations.

The real analogue of this routine is F07MHF (DSYRFS).

9 Example

This example solves the system of equations $AX = B$ using iterative refinement and to compute the forward and backward error bounds, where

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

and

$$B = \begin{pmatrix} 7.79 + 5.48i & -35.39 + 18.01i \\ -0.77 - 16.05i & 4.23 - 70.02i \\ -9.58 + 3.88i & -24.79 - 8.40i \\ 2.98 - 10.18i & 28.68 - 39.89i \end{pmatrix}.$$

Here A is Hermitian indefinite and must first be factorized by F07MRF (ZHETRF).

9.1 Program Text

```
*      F07MVF Example Program Text
*      Mark 15 Release. NAG Copyright 1991.
*      .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, NRHMAX, LDA, LWORK, LDAF, LDB, LDX
      PARAMETER        (NMAX=8,NRHMAX=NMAX,LDA=NMAX,LWORK=64*NMAX,
+                      LDAF=NMAX,LDB=NMAX,LDX=NMAX)
*      .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, N, NRHS
      CHARACTER        UPLO
*      .. Local Arrays ..
      COMPLEX *16      A(LDA,NMAX), AF(LDAF,NMAX), B(LDB,NRHMAX),
+                      WORK(LWORK), X(LDX,NMAX)
      DOUBLE PRECISION BERR(NRHMAX), FERR(NRHMAX), RWORK(NMAX)
      INTEGER          IPIV(NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*      .. External Subroutines ..
      EXTERNAL         F06TFF, X04DBF, ZHERFS, ZHETRF, ZHETRS
*      .. Executable Statements ..
      WRITE (NOUT,*) 'F07MVF Example Program Results'
*      Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, NRHS
      IF (N.LE.NMAX .AND. NRHS.LE.NRHMAX) THEN
*
*          Read A and B from data file, and copy A to AF and B to X
*
      READ (NIN,*) UPLO
      IF (UPLO.EQ.'U') THEN
        READ (NIN,*) ((A(I,J),J=1,N),I=1,N)
      ELSE IF (UPLO.EQ.'L') THEN
        READ (NIN,*) ((A(I,J),J=1,I),I=1,N)
```

```

      END IF
      READ (NIN,*) ((B(I,J),J=1,NRHS),I=1,N)
      CALL F06TFF(UPLO,N,N,A,LDA,AF,LDAF)
      CALL F06TFF('General',N,NRHS,B,LDB,X,LDX)
*
*      Factorize A in the array AF
*
      CALL ZHETRF(UPLO,N,AF,LDAF,IPIV,WORK,LWORK,INFO)
*
      WRITE (NOUT,*)
      IF (INFO.EQ.0) THEN
*
*          Compute solution in the array X
*
      CALL ZHETRS(UPLO,N,NRHS,AF,LDAF,IPIV,X,LDX,INFO)
*
*          Improve solution, and compute backward errors and
*          estimated bounds on the forward errors
*
      CALL ZHERFS(UPLO,N,NRHS,A,LDA,AF,LDAF,IPIV,B,LDB,X,LDX,FERR,
+              BERR,WORK,RWORK,INFO)
*
*          Print solution
*
      IFAIL = 0
      CALL X04DBF('General',' ',N,NRHS,X,LDX,'Bracketed','F7.4',
+              'Solution(s)','Integer',RLABS,'Integer',CLABS,
+              80,0,IFAIL)
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Backward errors (machine-dependent)'
      WRITE (NOUT,99999) (BERR(J),J=1,NRHS)
      WRITE (NOUT,*)
+      'Estimated forward error bounds (machine-dependent)'
      WRITE (NOUT,99999) (FERR(J),J=1,NRHS)
      ELSE
      WRITE (NOUT,*) 'The factor D is singular'
      END IF
      END IF
*
99999 FORMAT ((5X,1P,4(E11.1,7X)))
      END

```

9.2 Program Data

F07MVF Example Program Data

```

4 2                                     :Values of N and NRHS
'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
( 7.79,  5.48) (-35.39, 18.01)
(-0.77,-16.05) ( 4.23,-70.02)
(-9.58,  3.88) (-24.79, -8.40)
( 2.98,-10.18) ( 28.68,-39.89)
                                     :End of matrix B

```

9.3 Program Results

F07MVF Example Program Results

```

Solution(s)
          1          2
1 ( 1.0000,-1.0000) ( 3.0000,-4.0000)
2 (-1.0000, 2.0000) (-1.0000, 5.0000)
3 ( 3.0000,-2.0000) ( 7.0000,-2.0000)
4 ( 2.0000, 1.0000) (-8.0000, 6.0000)

```

Backward errors (machine-dependent)

8.0E-17 7.0E-17

Estimated forward error bounds (machine-dependent)

2.5E-15 3.0E-15
