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,       &
                   FERR, BERR, WORK, RWORK, INFO)
INTEGER                  N, NRHS, LDA, LDAF, IPIV(*), LDB, LDX, INFO
REAL (KIND=nag_wp)       FERR(NRHS), BERR(NRHS), RWORK(N)
COMPLEX (KIND=nag_wp)  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

$$(A + \delta A)x = b + \delta b$$
$$\left|\delta a_{ij}\right| \leq \beta\left|a_{ij}\right| \quad \text{and} \quad \left|\delta b_i\right| \leq \beta\left|b_i\right|.$$

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

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

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: 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 $PUDU^{H}P^{T}$, 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 ≥ 0.

3: NRHS – INTEGER *Input*

On entry: r, the number of right-hand sides.

Constraint: NRHS ≥ 0.

4: A(LDA,$*$) – COMPLEX (KIND=nag_wp) array *Input*

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

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

6: AF(LDAF,$*$) – COMPLEX (KIND=nag_wp) array *Input*

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

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

8: IPIV($*$) – INTEGER array *Input*

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

On entry: details of the interchanges and the block structure of D, as returned by F07MRF (ZHETRF).

9: B(LDB,$*$) – COMPLEX (KIND=nag_wp) array *Input*

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

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

11: X(LDX,$*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

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

13: FERR(NRHS) – REAL (KIND=nag_wp) array *Output*

On exit: FERR(j) contains an estimated error bound for the jth solution vector, that is, the jth column of X, for $j = 1, 2, \ldots, r$.

14: BERR(NRHS) – REAL (KIND=nag_wp) array *Output*

On exit: BERR(j) contains the component-wise backward error bound β for the jth solution vector, that is, the jth column of X, for $j = 1, 2, \ldots, r$.

15: WORK($2 \times \mathrm{N}$) – COMPLEX (KIND=nag_wp) array *Workspace*

16: RWORK(N) – REAL (KIND=nag_wp) array *Workspace*

17: 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.

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

```
    Program f07mvfe

!     F07MVF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: f06tff, nag_wp, x04dbf, zherfs, zhetrf, zhetrs
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                           :: i, ifail, info, lda, ldaf, ldb, ldx, &
                                           lwork, n, nrhs
      Character (1)                     :: uplo
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), af(:,:), b(:,:), work(:),  &
                                             x(:,:)
      Real (Kind=nag_wp), Allocatable   :: berr(:), ferr(:), rwork(:)
      Integer, Allocatable              :: ipiv(:)
      Character (1)                     :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F07MVF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, nrhs
      lda = n
      ldaf = n
      ldb = n
      ldx = n
      lwork = 64*n
      Allocate (a(lda,n),af(ldaf,n),b(ldb,nrhs),work(lwork),x(ldx,n), &
        berr(nrhs),ferr(nrhs),rwork(n),ipiv(n))

!     Read A and B from data file, and copy A to AF and B to X

      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
      Read (nin,*)(b(i,1:nrhs),i=1,n)
```

```
      Call f06tff(uplo,n,n,a,lda,af,ldaf)
      x(1:n,1:nrhs) = b(1:n,1:nrhs)

!     Factorize A in the array AF
!     The NAG name equivalent of zhetrf is f07mrf
      Call zhetrf(uplo,n,af,ldaf,ipiv,work,lwork,info)

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

!       Compute solution in the array X
!       The NAG name equivalent of zhetrs is f07msf
        Call zhetrs(uplo,n,nrhs,af,ldaf,ipiv,x,ldx,info)

!       Improve solution, and compute backward errors and
!       estimated bounds on the forward errors

!       The NAG name equivalent of zherfs is f07mvf
        Call zherfs(uplo,n,nrhs,a,lda,af,ldaf,ipiv,b,ldb,x,ldx,ferr,berr,work, &
          rwork,info)

!       Print solution

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        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(1:nrhs)
        Write (nout,*) 'Estimated forward error bounds (machine-dependent)'
        Write (nout,99999) ferr(1:nrhs)
      Else
        Write (nout,*) 'The factor D is singular'
      End If

99999 Format ((5X,1P,4(E11.1,7X)))
    End Program f07mvfe
```

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)
        5.1E-17           3.5E-17
 Estimated forward error bounds (machine-dependent)
        2.5E-15           3.0E-15
```