

NAG Library Routine Document

F07AHF (DGERFS)

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

F07AHF (DGERFS) returns error bounds for the solution of a real system of linear equations with multiple right-hand sides, $AX = B$ or $A^T X = B$. It improves the solution by iterative refinement, in order to reduce the backward error as much as possible.

2 Specification

```

SUBROUTINE F07AHF(TRANS, N, NRHS, A, LDA, AF, LDAF, IPIV, B, LDB, X,
1          LDX, FERR, BERR, WORK, IWORK, INFO)
    INTEGER          N, NRHS, LDA, LDAF, IPIV(*), LDB, LDX, IWORK(N), INFO
    double precision A(LDA,*), AF(LDAF,*), B(LDB,*), X(LDX,*), FERR(NRHS),
1          BERR(NRHS), WORK(3*N)
    CHARACTER*1      TRANS

```

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

3 Description

F07AHF (DGERFS) returns the backward errors and estimated bounds on the forward errors for the solution of a real system of linear equations with multiple right-hand sides $AX = B$ or $A^T X = B$. The routine handles each right-hand side vector (stored as a column of the matrix B) independently, so we describe the function of F07AHF (DGERFS) 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: TRANS – CHARACTER*1 *Input*

On entry: indicates the form of the linear equations for which X is the computed solution.

TRANS = 'N'

The linear equations are of the form $AX = B$.

TRANS = 'T' or 'C'

The linear equations are of the form $A^T X = B$.

Constraint: TRANS = 'N', 'T' or 'C'.

- 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,*) – **double precision** array *Input*
Note: the second dimension of the array A must be at least $\max(1, N)$.
On entry: the n by n original matrix A as supplied to F07ADF (DGETRF).

- 5: LDA – INTEGER *Input*
On entry: the first dimension of the array A as declared in the (sub)program from which F07AHF (DGERFS) is called.
Constraint: LDA $\geq \max(1, N)$.

- 6: AF(LDAF,*) – **double precision** array *Input*
Note: the second dimension of the array AF must be at least $\max(1, N)$.
On entry: the LU factorization of A , as returned by F07ADF (DGETRF).

- 7: LDAF – INTEGER *Input*
On entry: the first dimension of the array AF as declared in the (sub)program from which F07AHF (DGERFS) 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: the pivot indices, as returned by F07ADF (DGETRF).

- 9: B(LDB,*) – **double precision** 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 F07AHF (DGERFS) is called.
Constraint: LDB $\geq \max(1, N)$.

- 11: X(LDX,*) – **double precision** 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 F07AEF (DGETRS).
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 F07AHF (DGERFS) is called.
Constraint: $LDX \geq \max(1, N)$.
- 13: FERR(NRHS) – **double precision** array *Output*
On exit: 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: 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($3 \times N$) – **double precision** array *Workspace*
- 16: IWORK(N) – INTEGER 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 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 $4n^2$ floating-point operations. Each step of iterative refinement involves an additional $6n^2$ 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 $Ax = b$ or $A^T x = b$; the number is usually 4 or 5 and never more than 11. Each solution involves approximately $2n^2$ operations.

The complex analogue of this routine is F07AVF (ZGERFS).

9 Example

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

$$A = \begin{pmatrix} 1.80 & 2.88 & 2.05 & -0.89 \\ 5.25 & -2.95 & -0.95 & -3.80 \\ 1.58 & -2.69 & -2.90 & -1.04 \\ -1.11 & -0.66 & -0.59 & 0.80 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 9.52 & 18.47 \\ 24.35 & 2.25 \\ 0.77 & -13.28 \\ -6.22 & -6.21 \end{pmatrix}.$$

Here A is nonsymmetric and must first be factorized by F07ADF (DGETRF).

9.1 Program Text

```

*      F07AHF Example Program Text
*      Mark 15 Release. NAG Copyright 1991.
*      .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, NRHMAX, LDA, LDAF, LDB, LDX
      PARAMETER        (NMAX=8,NRHMAX=NMAX,LDA=NMAX,LDAF=NMAX,LDB=NMAX,
+                      LDX=NMAX)
      CHARACTER        TRANS
      PARAMETER        (TRANS='N')
*      .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, N, NRHS
*      .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), AF(LDAF,NMAX), B(LDB,NRHMAX),
+                      BERR(NRHMAX), FERR(NRHMAX), WORK(3*NMAX),
+                      X(LDX,NMAX)
      INTEGER          IPIV(NMAX), IWORK(NMAX)
*      .. External Subroutines ..
      EXTERNAL         DGERFS, DGETRF, DGETRS, F06QFF, X04CAF
*      .. Executable Statements ..
      WRITE (NOUT,*) 'F07AHF 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,*) ((A(I,J),J=1,N),I=1,N)
*         READ (NIN,*) ((B(I,J),J=1,NRHS),I=1,N)
*
*         CALL F06QFF('General',N,N,A,LDA,AF,LDAF)
*
*         CALL F06QFF('General',N,NRHS,B,LDB,X,LDX)
*
*         Factorize A in the array AF
*
*         CALL DGETRF(N,N,AF,LDAF,IPIV,INFO)
*
*         WRITE (NOUT,*)
*         IF (INFO.EQ.0) THEN
*
*             Compute solution in the array X
*
*             CALL DGETRS(TRANS,N,NRHS,AF,LDAF,IPIV,X,LDX,INFO)
*
*             Improve solution, and compute backward errors and
*             estimated bounds on the forward errors
*
*             CALL DGERFS(TRANS,N,NRHS,A,LDA,AF,LDAF,IPIV,B,LDB,X,LDX,
+                         FERR,BERR,WORK,IWORK,INFO)
*
*             Print solution
*
*             IFAIL = 0
*
*             CALL X04CAF('General',' ',N,NRHS,X,LDX,'Solution(s)',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 U is singular'

```

```

        END IF
    END IF
*
99999 FORMAT ((3X,1P,7E11.1))
END

```

9.2 Program Data

F07AHF Example Program Data

```

4 2                                     :Values of N and NRHS
1.80 2.88 2.05 -0.89
5.25 -2.95 -0.95 -3.80
1.58 -2.69 -2.90 -1.04
-1.11 -0.66 -0.59 0.80 :End of matrix A
9.52 18.47
24.35 2.25
0.77 -13.28
-6.22 -6.21 :End of matrix B

```

9.3 Program Results

F07AHF Example Program Results

Solution(s)

	1	2
1	1.0000	3.0000
2	-1.0000	2.0000
3	3.0000	4.0000
4	-5.0000	1.0000

Backward errors (machine-dependent)

9.4E-17	3.7E-17
---------	---------

Estimated forward error bounds (machine-dependent)

2.4E-14	3.3E-14
---------	---------