NAG Fortran Library Routine Document

F11MHF

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

F11MHF returns error bounds for the solution of a real sparse system of linear equations with multiple right-hand sides, $AX = B$ or $A^TX = B$. It improves the solution by iterative refinement in standard precision, in order to reduce the backward error as much as possible.

2 Specification

```
 SUBROUTINE F11MHF (TRANS, N, ICOLZP, IROWIX, A, IPRM, IL, LVAL, IU,
1                   UVAL, NRHS, B, LDB, X, LDX, FERR, BERR, IFAIL)
 INTEGER            N, ICOLZP(*), IROWIX(*), IPRM(7*N), IL(*), IU(*),
1                   NRHS, LDB, LDX, IFAIL
 double precision   A(*), LVAL(*), UVAL(*), B(LDB,*), X(LDX,*), FERR(*),
1                   BERR(*)
 CHARACTER*1        TRANS
```

3 Description

F11MHF 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 $A^TX = B$. The routine handles each right-hand side vector (stored as a column of the matrix B) independently, so we describe the function of F11MHF 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 if x is the exact solution of a perturbed system:

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

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.

The routine uses the LU factorization $P_rAP_c = LU$ computed by F11MEF and the solution computed by F11MFF.

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: specifies whether $AX = B$ or $A^T X = B$ is solved:

if TRANS = 'N', then $AX = B$ is solved;
if TRANS = 'T', then $A^T X = B$ is solved.

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

2: N – INTEGER *Input*

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

Constraint: $\mathrm{N} \geq 0$.

3: ICOLZP($*$) – INTEGER array *Input*

On entry: $\mathrm{ICOLZP}(i)$ contains the index in A of the start of a new column. See Section 2.1.3 in the F11 Chapter Introduction.

4: IROWIX($*$) – INTEGER array *Input*

Note: the dimension of the array IROWIX must be at least $\mathrm{ICOLZP}(\mathrm{N}+1)-1$, the number of non-zeros of the sparse matrix A.

On entry: the row index array of the sparse matrix A.

5: A($*$) – ***double precision*** array *Input*

Note: the dimension of the array A must be at least $\mathrm{ICOLZP}(\mathrm{N}+1)-1$, the number of non-zeros of the sparse matrix A.

On entry: the array of non-zero values in the sparse matrix A.

6: IPRM($7 \times \mathrm{N}$) – INTEGER array *Input*

On entry: the column permutation which defines P_c, the row permutation which defines P_r, plus associated data structures as computed by F11MEF.

7: IL($*$) – INTEGER array *Input*

On entry: records the sparsity pattern of matrix L as computed by F11MEF.

8: LVAL($*$) – ***double precision*** array *Input*

On entry: records the non-zero values of matrix L and some non-zero values of matrix U as computed by F11MEF.

9: IU($*$) – INTEGER array *Input*

On entry: records the sparsity pattern of matrix U as computed by F11MEF.

10: UVAL($*$) – ***double precision*** array *Input*

On entry: records some non-zero values of matrix U as computed by F11MEF.

11: NRHS – INTEGER *Input*

On entry: $nrhs$, the number of right-hand sides in B.

Constraint: $\mathrm{NRHS} \geq 0$.

12: B(LDB,$*$) – ***double precision*** array *Input*

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

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

13: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F11MHF is called.

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

14: X(LDX,$*$) – ***double precision*** array *Input/Output*

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

On entry: the n by $nrhs$ solution matrix X, as returned by F11MFF.

On exit: the n by $nrhs$ improved solution matrix X.

15: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which F11MHF is called.

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

16: FERR($*$) – ***double precision*** array *Output*

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

17: BERR($*$) – ***double precision*** array *Output*

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

18: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: $\mathrm{IFAIL} = 0$ unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry $\mathrm{IFAIL} = 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

$\mathrm{IFAIL} = 1$

On entry, TRANS $\neq$ 'N' or 'T',
or N < 0,
or NRHS < 0,
or $\mathrm{LDB} < \max(1, \mathrm{N})$,
or $\mathrm{LDX} < \max(1, \mathrm{N})$.

IFAIL = 2

Ill-defined row permutation in array IPRM. Internal checks have revealed that the IPRM array is corrupted.

IFAIL = 3

Ill-defined column permutations in array IPRM. Internal checks have revealed that the IPRM array is corrupted.

IFAIL = 301

Unable to allocate required internal workspace.

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

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 $A^T x = b$;

9 Example

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

$$A = \begin{pmatrix} 2.00 & 1.00 & 0 & 0 & 0 \\ 0 & 0 & 1.00 & -1.00 & 0 \\ 4.00 & 0 & 1.00 & 0 & 1.00 \\ 0 & 0 & 0 & 1.00 & 2.00 \\ 0 & -2.00 & 0 & 0 & 3.00 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1.56 & 3.12 \\ -0.25 & -0.50 \\ 3.60 & 7.20 \\ 1.33 & 2.66 \\ 0.52 & 1.04 \end{pmatrix}.$$

Here A is nonsymmetric and must first be factorized by F11MEF.

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.

```
*     F11MHF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          LA, NMAX, MMAX
      PARAMETER        (LA=10000,NMAX=1000,MMAX=10)
      DOUBLE PRECISION ONE
      PARAMETER        (ONE=1.D0)
*     .. Local Scalars ..
      DOUBLE PRECISION FLOP, THRESH
      INTEGER          I, IFAIL, J, N, NNZ, NNZL, NNZU, NRHS, NZLMX,
     +                 NZLUMX, NZUMX
      CHARACTER        SPEC, TRANS
*     .. Local Arrays ..
      DOUBLE PRECISION A(LA), B(NMAX,MMAX), BERR(MMAX), FERR(MMAX),
     +                 LVAL(8*LA), UVAL(8*LA), X(NMAX,MMAX)
      INTEGER          ICOLZP(NMAX+1), IL(7*NMAX+8*LA+4), IPRM(7*NMAX),
     +                 IROWIX(LA), IU(2*NMAX+8*LA+1)
      CHARACTER        CLABS(1), RLABS(1)
```

```
*     .. External Subroutines ..
      EXTERNAL         F11MDF, F11MEF, F11MFF, F11MHF, X04CAF, X04CBF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F11MHF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
*
*     Read order of matrix and number of right hand sides
*
      READ (NIN,*) N, NRHS
      IF (N.LE.NMAX .AND. NRHS.LE.MMAX) THEN
*
*        Read the matrix A
*
         DO 20 I = 1, N + 1
            READ (NIN,*) ICOLZP(I)
   20    CONTINUE
         NNZ = ICOLZP(N+1) - 1
         DO 40 I = 1, NNZ
            READ (NIN,*) A(I), IROWIX(I)
   40    CONTINUE
*
*        Read the tright hand sides
*
         DO 80 J = 1, NRHS
            READ (NIN,*) (X(I,J),I=1,N)
            DO 60 I = 1, N
               B(I,J) = X(I,J)
   60       CONTINUE
   80    CONTINUE

*
*        Calculate COLAMD permutation
*
         SPEC = 'M'
         IFAIL = 0
         CALL F11MDF(SPEC,N,ICOLZP,IROWIX,IPRM,IFAIL)
*
*        Factorise
*
         THRESH = ONE
         IFAIL = 0
         NZLMX = 8*NNZ
         NZLUMX = 8*NNZ
         NZUMX = 8*NNZ
         CALL F11MEF(N,IROWIX,A,IPRM,THRESH,NZLMX,NZLUMX,NZUMX,IL,
     +              LVAL,IU,UVAL,NNZL,NNZU,FLOP,IFAIL)
*
*        Compute solution in array X
*
         TRANS = 'N'
         IFAIL = 0
         CALL F11MFF(TRANS,N,IPRM,IL,LVAL,IU,UVAL,NRHS,X,NMAX,IFAIL)
*
*        Improve solution, and compute backward errors and estimated
*        bounds on the forward errors
*
         CALL F11MHF(TRANS,N,ICOLZP,IROWIX,A,IPRM,IL,LVAL,IU,UVAL,NRHS,
     +              B,NMAX,X,NMAX,FERR,BERR,IFAIL)

*
*        Print solution
*
         WRITE (NOUT,*)
         CALL X04CAF('G',' ',N,NRHS,X,NMAX,'Solutions',IFAIL)
         CALL X04CBF('G','X',NRHS,1,FERR,NRHS,'1PE8.1',
     +              'Estimated Forward Error','N',RLABS,'N',CLABS,80,0,
     +              IFAIL)
         CALL X04CBF('G','X',NRHS,1,BERR,NRHS,'1PE8.1','Backward Error',
     +              'N',RLABS,'N',CLABS,80,0,IFAIL)
```

```
*
      END IF
      END
```

9.2 Program Data

```
F11MHF Example Program Data
  5 2  N, NRHS
 1
 3
 5
 7
 9
 12   ICOLZP(I) I=1,..,N+1
  2.   1
  4.   3
  1.   1
 -2.   5
  1.   2
  1.   3
 -1.   2
  1.   4
  1.   3
  2.   4
  3.   5    A(I), IROWIX(I) I=1,NNZ
 1.56 -.25 3.6 1.33 .52
 3.12 -.50 7.2 2.66 1.04 X(I,J) J=1,NRHS I=1,N
```

9.3 Program Results

```
 F11MHF Example Program Results

 Solutions
             1          2
 1      0.7000     1.4000
 2      0.1600     0.3200
 3      0.5200     1.0400
 4      0.7700     1.5400
 5      0.2800     0.5600
 Estimated Forward Error
   5.0E-15
   5.0E-15
 Backward Error
   3.6E-17
   3.6E-17
```