NAG Fortran Library Routine Document

F11MEF

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

F11MEF computes the LU factorization of a real sparse matrix in compressed column (Harwell–Boeing), column-permuted format.

2 Specification

```
 SUBROUTINE F11MEF (N, IROWIX, A, IPRM, THRESH, NZLMX, NZLUMX, NZUMX, IL,
1                   LVAL, IU, UVAL, NNZL, NNZU, FLOP, IFAIL)
 INTEGER            N, IROWIX(*), IPRM(7*N), NZLMX, NZLUMX, NZUMX,
1                   IL(7*N+NZLMX+4), IU(2*N+NZUMX+1), NNZL, NNZU, IFAIL
 double precision   A(*), THRESH, LVAL(*), UVAL(NZUMX), FLOP
```

3 Description

Given a real sparse matrix A, F11MEF computes an LU factorization of A with partial pivoting, $P_rAP_c = LU$, where P_r is a row permutation matrix (computed by F11MEF), P_c is a (supplied) column permutation matrix, L is unit lower triangular and U is upper triangular. The column permutation matrix, P_c, must be computed by a prior call to F11MDF. The matrix A must be presented in the column permuted, compressed column (Harwell–Boeing) format.

The LU factorization is output in the form of four one-dimensional arrays: integer arrays IL and IU and real-valued arrays LVAL and UVAL. These describe the sparsity pattern and numerical values in the L and U matrices. The minimum required dimensions of these arrays cannot be given as a simple function of the size parameters (order and number of non-zero values) of the matrix A. This is due to unpredictable fill-in created by partial pivoting. F11MEF will, on return, indicate which dimensions of these arrays were not adequate for the computation or (in the case of one of them) give a firm bound. The user should then allocate more storage and try again.

4 References

Demmel J W, Eisenstat S C, Gilbert J R, Li X S and Li J W H (1999) A Supernodal Approach to Sparse Partial Pivoting *SIAM J. Matrix Anal. Appl.* **20** 720–755 URL: http://citeseer.nj.nec.com/demmel95supernodal.html

Demmel J W, Gilbert J R and Li X S (1999) An Asynchronous Parallel Supernodal Algorithm for Sparse Gaussian Elimination *SIAM J. Matrix Anal. Appl.* **20** 915–952 URL: http://citeseer.nj.nec.com/demmel97asynchronous.html

5 Parameters

1: N – INTEGER *Input*

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

Constraint: N ≥ 0.

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

Note: the dimension of the array IROWIX must be at least nnz, the number of non-zeros of the sparse matrix A.

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

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

Note: the dimension of the array A must be at least nnz, the number of non-zeros of the sparse matrix A.

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

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

On entry: contains the column permutation which defines the permutation P_c and associated data structures as computed by routine F11MDF.

On exit: part of the array is modified to record the row permutation P_r determined by pivoting.

5: THRESH – ***double precision*** *Input*

On entry: the diagonal pivoting threshold, t. At step j of the Gaussian elimination, if $\left|A_{jj}\right| \geq t\left(\max_{i \geq j} \left|A_{ij}\right|\right)$, use A_{jj} as a pivot, otherwise use $\max_{i \geq j} \left|A_{ij}\right|$. A value of $t = 1$ corresponds to partial pivoting, a value of $t = 0$ corresponds to always choosing the pivot on the diagonal (unless it is zero).

Constraint: $0 \leq t \leq 1$.

Suggested value: THRESH $= 1.0$. Smaller values may result in a faster factorization, but the benefits are likely to be small in most cases. It might be possible to use THRESH $= 0.0$ if the user is confident about the stability of the factorization, for example, if A is diagonally dominant.

6: NZLMX – INTEGER *Input*

On entry: indicates the available size of array IL. The dimension of IL should be at least $7 \times \mathrm{N} + \mathrm{NZLMX} + 4$. A good range for NZLMX that works for many problems is nnz to $8 \times nnz$, where nnz is the number of non-zeros in the sparse matrix A. If, on exit, IFAIL is set to 2, the given NZLMX was too small and the user should attempt to provide more storage and call the routine again.

Constraint: NZLMX ≥ 1.

7: NZLUMX – INTEGER *Input/Output*

On entry: indicates the available size of array LVAL. The dimension of LVAL should be at least NZLUMX.

Constraint: NZLUMX ≥ 1.

On exit: if IFAIL is set to 4, the given NZLUMX was too small and is reset to a value that will be sufficient. The user should then provide the indicated storage and call the routine again.

8: NZUMX – INTEGER *Input*

On entry: indicates the available sizes of arrays IU and UVAL. The dimension of IU should be at least $2 \times \mathrm{N} + \mathrm{NZUMX} + 1$ and the dimension of UVAL should be at least NZUMX. A good range for NZUMX that works for many problems is nnz to $8 \times nnz$, where nnz is the number of non-zeros in the sparse matrix A. If, on exit, IFAIL is set to 3, the given NZUMX was too small and the user should attempt to provide more storage and call the routine again.

Constraint: NZUMX ≥ 1.

9: IL($7 \times \mathrm{N} + \mathrm{NZLMX} + 4$) – INTEGER array *Output*

On exit: encapsulates the sparsity pattern of matrix L.

10: LVAL($*$) – ***double precision*** array *Output*

Note: the dimension of the array LVAL must be at least NZLUMX.

On exit: records the non-zero values of matrix L and some of the non-zero values of matrix U.

11: IU($2 \times \mathrm{N} + \mathrm{NZUMX} + 1$) – INTEGER array *Output*

On exit: encapsulates the sparsity pattern of matrix U.

12: UVAL(NZUMX) – ***double precision*** array *Output*

On exit: records some of the non-zero values of matrix U.

13: NNZL – INTEGER *Output*

On exit: the number of non-zero values in the matrix L.

14: NNZU – INTEGER *Output*

On exit: the number of non-zero values in the matrix U.

15: FLOP – ***double precision*** *Output*

On exit: the number of floating-point operations performed.

16: 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: 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 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:

IFAIL $= 1$

On entry, N < 0,
or NZLMX < 1,
or NZLUMX < 1,
or NZUMX < 1,
or THRESH < 0.0,
or THRESH > 1.0.

IFAIL $= 2$

NZLMX was not large enough. You should repeat the call with a larger value of NZLMX, providing more storage for the output array IL.

IFAIL = 3

NZUMX was not large enough. You should repeat the call with a larger value of NZUMX, providing more storage for the output arrays IU and UVAL.

IFAIL = 4

NZLUMX was not large enough. You should repeat the call with the value of NZLUMX returned on exit, providing more storage for the output array LVAL.

IFAIL = 5

The matrix A is singular and no factorization will be attempted.

IFAIL = 301

Unable to allocate required internal workspace.

7 Accuracy

The computed factors L and U are the exact factors of a perturbed matrix $A + E$, where

$$|E| \leq c(n)\epsilon|L||U|,$$

$c(n)$ is a modest linear function of n, and ϵ is the ***machine precision***, when partial pivoting is used. If no partial pivoting is used, the factorization accuracy can be considerably worse. A call to F11MMF after F11MEF can help determine the quality of the factorization.

8 Further Comments

The total number of floating-point operations depends on the sparsity pattern of the matrix A.

A call to this routine may be followed by calls to the routines:

F11MFF to solve $AX = B$ or $A^TX = B$;

F11MGF to estimate the condition number of A;

F11MMF to estimate the reciprocal pivot growth of the LU factorization.

9 Example

To compute the LU factorization of the matrix A, where

$$A = \begin{pmatrix} 2.00 & 1.00 & 0 & 0 & 0 \\ 0 & 0 & 1.00 & -1.00 & 0 \\ 4.00 & 0 & 1.00 & 0 & 1.00 \\ 0 & 0 & 0 & 1.00 & 2.00 \\ 0 & -2.00 & 0 & 0 & 3.00 \end{pmatrix}.$$

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.

```
*     F11MEF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          LA, NMAX
      PARAMETER        (LA=10000,NMAX=1000)
      DOUBLE PRECISION ONE
      PARAMETER        (ONE=1.D0)
*     .. Local Scalars ..
      DOUBLE PRECISION FLOP, THRESH
```

```
      INTEGER          I, IFAIL, N, NNZ, NNZL, NNZU, NZLMX, NZLUMX,
     +                 NZUMX
      CHARACTER        SPEC
*     .. Local Arrays ..
      DOUBLE PRECISION A(LA), LVAL(8*LA), UVAL(8*LA)
      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, X04CBF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F11MEF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
*
*     Read order of matrix
*
      READ (NIN,*) N
      IF (N.LE.NMAX) 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

*
*        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)
*
*        Output results
*
         WRITE (NOUT,*)
         WRITE (NOUT,*)
     +     'Number of nonzeros in factors (excluding unit diagonal)'
         WRITE (NOUT,'(I8)') NNZL + NNZU - N

         CALL X04CBF('G','X',1,10,LVAL,1,'F7.2',
     +              'Factor elements in LVAL','N',RLABS,'N',CLABS,80,0,
     +              IFAIL)
         CALL X04CBF('G','X',1,4,UVAL,1,'F7.2',
     +              'Factor elements in UVAL','N',RLABS,'N',CLABS,80,0,
     +              IFAIL)

*
      END IF
      END
```

9.2 Program Data

```
F11MEF Example Program Data
  5                 N
 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
```

9.3 Program Results

```
 F11MEF Example Program Results

 Number of nonzeros in factors (excluding unit diagonal)
      14
 Factor elements in LVAL
    -2.00  -0.50   4.00   0.50   2.00   0.50  -1.00   0.50   1.00  -1.00
 Factor elements in UVAL
     1.00   3.00   1.00   1.00
```