NAG Fortran Library Routine Document

F11MDF

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

F11MDF computes a column permutation suitable for LU factorization (by F11MEF) of a real sparse matrix in compressed column (Harwell–Boeing) format and applies it to the matrix. This routine must be called prior to F11MEF.

2 Specification

```
SUBROUTINE F11MDF (SPEC, N, ICOLZP, IROWIX, IPRM, IFAIL)
INTEGER            N, ICOLZP(*), IROWIX(*), IPRM(7*N), IFAIL
CHARACTER*1        SPEC
```

3 Description

Given a sparse matrix in compressed column (Harwell–Boeing) format A and a choice of column permutation schemes, the routine computes those data structures that will be needed by the LU factorization routine F11MEF and associated routines F11MMF, F11MFF and F11MHF. The column permutation choices are:

original order (that is, no permutation);

user-supplied permutation;

a permutation, computed by the routine, designed to minimise fill-in during the LU factorization.

The algorithm for this computed permutation is based on the approximate minimum degree column ordering algorithm COLAMD. The computed permutation is not sensitive to the magnitude of the non-zero values of A.

4 References

Amestoy P R, Davis T A and Duff I S (1996) An approximate minimum degree ordering algorithm *SIAM J. Matrix Anal. Appl.* **17** 886–905 URL: http://citeseer.nj.nec.com/amestoy96approximate.html

Davis T A, Gilbert J R, Larimore S I and Ng E G (2000) Algorithm 8xx: COLAMD, a column approximate minimum degree ordering algorithm *Technical Report TR–00–006* Department of Computer and Information Science and Engineering, University of Florida URL: http://citeseer.nj.nec.com/391100.html

Davis T A, Gilbert J R, Larimore S I and Ng E G (2000) A column approximate minimum degree ordering algorithm *Technical Report TR–00–005* Department of Computer and Information Science and Engineering, University of Florida URL: http://citeseer.nj.nec.com/citeseer.nj.nec.com/davis00column.html

5 Parameters

1: SPEC – CHARACTER*1 *Input*

On entry: indicates the permutation to be applied as follows:

if SPEC = 'N', the identity permutation is used (i.e., the columns are not permuted);
if SPEC = 'U', the permutation in the IPRM array is used, as supplied by the user;
if SPEC = 'M', the permutation computed by the COLAMD algorithm is used

Constraint: SPEC = 'N', 'U' or 'M'.

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: 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: IROWIX(i) conatins the row index in A for elment $A(i)$. See Section 2.1.3 in the F11 Chapter Introduction.

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

On entry: The first N entries contain the column permutation supplied by the user. This will be used if SPEC = 'U', and ignored otherwise. If used, it must consist of a permutation of all the integers in the range $[0,(\mathrm{N}-1)]$, the leftmost column of the matrix A denoted by 0 and the rightmost by $\mathrm{N}-1$. Labelling columns in this way, $\mathrm{IPRM}(i)=j$ means that column $i-1$ of A is in position j in AP_c, where $P_rAP_c=LU$ expresses the factorization to be performed.

On exit: a new permutation is returned in the first N entries. The rest of the array contains data structures that will be used by other routines. The routine computes the column elimination tree for A and a post-order permutation on the tree. It then compounds the IPRM permutation given or computed by the COLAMD algorthm with the post-order permutation. This array is needed by the LU factorization routine F11MEF and associated routines F11MMF, F11MFF and F11MHF and should be passed to them unchanged.

6: 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, SPEC $\neq$ 'N', 'U' or 'M',
or $\mathrm{N}<0$.

$\mathrm{IFAIL}=2$

On entry, SPEC = 'U', but IPRM does not represent a valid permutation of the integers in $[0,(\mathrm{N}-1)]$. An input value of IPRM is either out of range or repeated.

IFAIL = 3

Unspecified failure of the COLAMD algorithm. This should not happen and should be reported to NAG.

IFAIL = 4

On entry, the array ICOLZP failed to satisfy the following constraints:

$\mathrm{ICOLZP}(1) = 1;$

$\mathrm{ICOLZP}(i-1) \leq \mathrm{ICOLZP}(i)$, for $i = 2, 3, \ldots, \mathrm{N}+1;$

$\mathrm{ICOLZP}(i) \leq \mathrm{N} * \mathrm{N} + 1$, for $i = 1, 2, \ldots, \mathrm{N}+1.$

IFAIL = 5

On entry, the array IROWIX failed to satisfy the following constraints:

$1 \leq \mathrm{IROWIX}(i) \leq \mathrm{N}$ for $i = 1, 2, \ldots, \mathrm{ICOLZP}(\mathrm{N}+1);$

for each column $i = 1 \ldots \mathrm{N}$, the row indices $\mathrm{IROWIX}(j)$, where $j = \mathrm{ICOLZP}(i) \ldots \mathrm{ICOLZP}(i+1) - 1$, do not repeat.

IFAIL = 301

Unable to allocate required internal workspace.

7 Accuracy

Not applicable. This computation does not use floating point numbers.

8 Further Comments

We recommend calling this routine with SPEC='M' before calling F11MEF. The COLAMD algorithm computes a sparsity-preserving permutation P_c solely from the pattern of A such that the LU factorization $P_r A P_c = LU$ remains as sparse as possible, regardless of the subsequent choice of P_r. The algorithm takes advantage of the existence of super-columns (columns with the same sparsity pattern) to reduce running time.

9 Example

To compute a sparsity preserving column permutation for 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.

```
*     F11MDF 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)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, N, NNZ
      CHARACTER        SPEC
```

```
*     .. Local Arrays ..
      INTEGER          ICOLZP(NMAX+1), IPRM(7*NMAX), IROWIX(LA)
*     .. External Subroutines ..
      EXTERNAL         F11MDF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F11MDF 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,*) IROWIX(I)
   40    CONTINUE

*
*        Calculate COLAMD permutation
*
         SPEC = 'M'
         IFAIL = 0
         CALL F11MDF(SPEC,N,ICOLZP,IROWIX,IPRM,IFAIL)
*
*        Output results
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'COLAMD Permutation'
         WRITE (NOUT,'(10I6)') (IPRM(I),I=1,N)
*
*        Calculate user permutation
*
         SPEC = 'U'
         IFAIL = 0
         IPRM(1) = 4
         IPRM(2) = 3
         IPRM(3) = 2
         IPRM(4) = 1
         IPRM(5) = 0
         CALL F11MDF(SPEC,N,ICOLZP,IROWIX,IPRM,IFAIL)
*
*        Output results
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'User Permutation'
         WRITE (NOUT,'(10I6)') (IPRM(I),I=1,N)
*
*        Calculate natural permutation
*
         SPEC = 'N'
         IFAIL = 0
         CALL F11MDF(SPEC,N,ICOLZP,IROWIX,IPRM,IFAIL)
*
*        Output results
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Natural Permutation'
         WRITE (NOUT,'(10I6)') (IPRM(I),I=1,N)

*
      END IF
      END
```

9.2 Program Data

```
F11MDF Example Program Data
  5   N
 1
 3
 5
 7
 9
 12   ICOLZP(I) I=1,..,N+1
 1
 3
 1
 5
 2
 3
 2
 4
 3
 4
 5    IROWIX(I) I=1,...,NNZ
```

9.3 Program Results

```
 F11MDF Example Program Results

 COLAMD Permutation
     1     0     4     3     2

 User Permutation
     4     3     2     1     0

 Natural Permutation
     0     1     2     3     4
```