1 SUMMARY

Given a sparse symmetric matrix **A** of order *n*, this subroutine **computes a set of element matrices** that, if assembled, would yield the same matrix. Note that this set of elements is not unique. The lower triangular entries of the matrix must be input by the user either in compressed column format (column pointer/row index scheme) or by row and column index pairs in any order (coordinate scheme). Diagonal entries must be present and out-of-range indices (indices less than 1 or greater than *n*) and duplicate entries are not allowed.

ATTRIBUTES — **Version:** 1.0.0. (12 July 2004) **Types:** Real (single, double). **Language:** Fortran 77. **Original date:** October 1995. **Calls:** MC59. **Origin:** I. S. Duff, Rutherford Appleton Laboratory.

2 HOW TO USE THE PACKAGE

2.1 Argument lists

There is only one user-callable entry to this package.

The single precision version

```
CALL MC37A(N,NE,IRN,JCN,A,IP,LIELT,NUMELT,PTRELT,IELT,
           LAELT,AELT,IW,ICNTL,INFO)
```

The double precision version

```
CALL MC37AD(N,NE,IRN,JCN,A,IP,LIELT,NUMELT,PTRELT,IELT,
            LAELT,AELT,IW,ICNTL,INFO)
```

N is an INTEGER variable that must be set by the user to the order of the matrix **A**. It is not altered by the subroutine. **Restriction:** N ≥ 1.

NE is an INTEGER variable that must be set by the user to the number of entries in the lower triangle (including the diagonal) of the matrix being input. It is not altered by the subroutine. **Restriction:** NE ≥ 1.

IRN is an INTEGER array of length NE. IRN(K) must be set by the user to hold the row index of entry K of the lower triangle of the matrix (including the diagonal), K=1, ..., NE. Out-of-range indices and duplicate entries are not allowed. IRN is not altered if input is by columns. If the coordinate scheme is used, on exit IRN holds the row indices of the matrix reordered by columns.

JCN is an INTEGER array of length NE that must be set by the user and is altered by the subroutine. If the matrix is held as row and column indices, JCN(K) must hold the column index of entry K, K=1, ..., NE. If the matrix is held by columns, then JCN(1) must be set to -1.

A is a REAL (DOUBLE PRECISION in the D version) array of length NE. If LAELT > 1, then A must be set by the user to hold the real values so that entry A(K) is in row IRN(K), K=1, ..., NE. If LAELT=1, then A is not accessed by MC37A/AD. A is not altered if input is by columns. If the coordinate scheme is used, on exit A holds the row indices of the matrix reordered by columns.

IP is an INTEGER array of length N+1. If the user inputs the matrix by columns, then IP(J) must be set to the position in IRN and A (if held) of the first entry in column J, J=1, ..., N, and IP(N+1) must be set to the position immediately following the last entry in column N. In this case, IP is not altered. IP need not be set if input is by row and column indices (i.e. JCN(1) ≠ -1). In this case, on output, IP(J) will be equal to the position in the reordered IRN and A of the first entry in column J, J=1, ..., N, and IP(N+1)-1 is the position of the last entry in column N.

LIELT is an INTEGER variable that must be set by the user to the length of array IELT. The value of LIELT required will sometimes be less than NE but will never exceed 2**noff*, where *noff* is the number of off-diagonal matrix

entries. It is not altered by the subroutine. The space used, or needed in the event of failure with INFO(1)=−10, is returned in INFO(2). **Restriction:** LIELT ≥ N.

NUMELT is an INTEGER variable that need not be set by the user. On output, NUMELT is the number of elements created. It will never exceed *noff*, where *noff* is the number of off-diagonal matrix entries.

PTRELT is an INTEGER array that need not be set by the user. On output, PTRELT(I) will be set to the position in IELT of the first entry for the indices of element I, I=1, ..., NUMELT. The last entry of the NUMELT element is in position PTRELT(NUMELT+1)−1 of IELT.

IELT is an INTEGER array of length LIELT. On output from MC37A/AD, IELT holds indexing information for the element representation of the matrix. Each element is stored contiguously, the first entry of element I, I=1, ..., NUMELT being in position PTRELT(I) and the last in position PTRELT(I+1)−1. The indices of the variables in the element are in ascending order.

LAELT is an INTEGER variable that must be set by the user to the length of array AELT. If no reals are provided, LAELT must have the value 1. In this case, arrays A and AELT are not accessed by the routine. The value of LAELT required will sometimes be not much more than NE but will never exceed 3**noff*, where *noff* is the number of off-diagonal matrix entries. It is not altered by the subroutine. The space used, or needed in the event of failure with INFO(1)=−11, is returned in INFO(3). **Restriction:** LAELT ≥ N, if LAELT ≠ 1.

AELT is a REAL (DOUBLE PRECISION in the D version) array of length LAELT. If LAELT is greater than 1, AELT will hold, on output, the real information for the element representation of the matrix. The elements are stored contiguously. For each element, the lower triangle of the packed matrix is stored by columns.

IW is an INTEGER array of length N+1 that need not be set by the user. IW is used as workspace by MC37A/AD.

ICNTL is an INTEGER array of length 5. Component ICNTL(1) must be set by the user to the output stream for error messages. These are suppressed if ICNTL(1) is negative. The components 2 to 5 are not currently accessed by MC37A/AD.

INFO is an INTEGER array of length 10 that need not be set by the user.
The elements of the array INFO provide information on the action of MC37A/AD:–

INFO(1) has the value zero if the call was successful and a negative value in the event of an error:–

−1	N < 1.
−2	NZ < 1.
−3	LIELT < N.
−4	LAELT < N, if LAELT ≠ 1.
−5	Error in sort routine (MC59A/AD). Probably caused by indices out-of-range or duplicates.
−6	Indices out-of-range.
−7	Entry in upper triangle.
−8	Duplicates present.
−9	Missing diagonal entry.
−10	LIELT too small. Value needed given by INFO(2).
−11	LAELT too small. Value needed given by INFO(3).

INFO(2) is the number of entries in IELT holding the integer data for the elements or the number required in the case of an error with INFO(1)=−10.

INFO(3) is the number of entries in AELT holding the real data for the elements or the number required in the case of an error with INFO(1)=−11.

3 GENERAL INFORMATION

Use of common: None.

Workspace: IW(N+1)

Other routines called directly: MC37A/AD calls an internal routine MC37B/BD that need not be called by the user. MC59A/AD is also called.

Input/output: Error messages printed on unit ICNTL(1) and are suppressed if ICNTL(1) < 0.

Restrictions: N ≥ 1, NE ≥ 1, LIELT ≥ N, LAELT ≥ N, if LAELT ≠ 1.

4 METHOD

A representation of the lower triangle of the matrix sorted by rows is first formed. Column indices are in order within these rows. The element creation algorithm is a greedy algorithm that accesses the rows of this sorted matrix in reverse order building as large elements as it can starting with the first currently unassembled entry encountered. A separate (optional) pass, using also the original data is used to generate the real data for the elements.

5 EXAMPLES OF USE

We illustrate the use of MC37 on the following matrix:

$$\mathbf{A} = \begin{pmatrix} 1.1 & & & \\ & 2.2 & & \\ 3.1 & & 3.3 & \\ & 4.2 & & 4.4 \end{pmatrix}$$

Program

```
C Simple example of use of MC37
C
      PROGRAM TEST
C Define array dimensions
      INTEGER LIWMAX, NMAX, NZMAX, MAXELT
      PARAMETER (LIWMAX=200, NMAX=10, NZMAX=100, MAXELT=20)
C Declaration of subroutine parameters
      INTEGER N, NE, IRN(NZMAX), JCN(NZMAX)
      DOUBLE PRECISION A(NZMAX)
      INTEGER IP(NMAX+1), NUMELT, PTRELT(MAXELT+1), IELT(LIWMAX)
      DOUBLE PRECISION AELT(LIWMAX)
      INTEGER IW(NMAX+1), ICNTL(5), INFO(10)
C Local variables
      INTEGER ELT, INFILE, K, KALT, KELT, LP, NALL, NELL

C Set reading and printing streams
      INFILE = 5
      LP     = 6

C
C Read input data N, IPE and IW
      READ(INFILE,*) N,NE
      READ(INFILE,*) (IRN(K),K=1,NE)
      READ(INFILE,*) (JCN(K),K=1,NE)
      READ(INFILE,*) (A(K),K=1,NE)
C
C Perform MC37 ordering
      ICNTL(1) = LP
```

```
       CALL MC37AD (N, NE, IRN, JCN, A, IP, LIWMAX, NUMELT, PTRELT,
      *             IELT, LIWMAX, AELT, IW, ICNTL, INFO)

       IF (INFO(1).LT.0) THEN
         WRITE (LP,*) ' ERROR detected during ordering'
       ELSE

 C Print output element matrix
         WRITE(LP,'(A,I2/)') 'Number of elements = ',NUMELT
         KALT = 0
         DO 10 ELT = 1,NUMELT
           WRITE(LP,'(/A,I2)') 'Element ',ELT
           KELT = PTRELT(ELT)
           NELL = PTRELT(ELT+1) - KELT
           NALL = (NELL*(NELL+1))/2
           WRITE(LP,'(A/(20I4))') 'Indices',(IELT(K),K=KELT,KELT+NELL-1)
           WRITE(LP,'(A/(8D10.3))') 'Reals',(AELT(K),K=KALT+1,KALT+NALL)
           KALT = KALT + NALL
    10   CONTINUE
       ENDIF

       STOP
       END
```

Input Data

```
   4   6
   1   3   2   4   3   4
   1   1   2   2   3   4
 1.1 3.1 2.2 4.2 3.3 4.4
```

Output data

```
Number of elements =  2


Element  1
Indices
   2   4
Reals
 0.220D+01 0.420D+01 0.440D+01

Element  2
Indices
   1   3
Reals
 0.110D+01 0.310D+01 0.330D+01
```