

HSL ARCHIVE

Warning: Subroutine MC20 has been superseded by subroutine MC49 which uses improved algorithms; the use of the latter routine is recommended. The superseded routine may be removed from later releases of the library.

1 SUMMARY

Sorts the nonzeros of a sparse matrix from arbitrary order to an ordering by columns. An option exists for also ordering by rows within each column.

The subroutine allows users with sparse problems to input the nonzeros in a natural order more suited to the individual problem. The resulting ordered matrix can subsequently be presented to other sparse matrix subroutines in the library.

An in-place sort algorithm is used which handles each item to be sorted exactly three times, so the number of operations for the method is of the order of the number of nonzeros.

Described by I.S.Duff, Harwell report R.8730 (1977).

ATTRIBUTES — Version: 1.0.0. Types: MC20A, MC20AD. Original date: November 1975. Origin: J. K. Reid and I. S. Duff, Harwell.

2 HOW TO USE THE PACKAGE

2.1 Argument list

To sort the nonzeros of a sparse matrix from arbitrary order to column order, unordered within each column

The single precision version

CALL MC20A(NC, MAXA, A, INUM, JPTR, JNUM, JDISP)

The double precision version

CALL MC20AD(NC,MAXA,A,INUM,JPTR,JNUM,JDISP)

To sort the nonzeros within each column of a sparse matrix stored by columns

The single precision version

CALL MC20B(NC, MAXA, A, INUM, JPTR)

The double precision version

CALL MC20BD(NC, MAXA, A, INUM, JPTR)

- NC is an INTEGER which must be set by the user to the number of matrix columns. It is not altered by MC20A/AD or MC20B/BD.
- MAXA is an INTEGER which must be set by the user to the number of matrix nonzeros. It is not altered by MC20A/AD or MC20B/BD.
- A is a REAL (DOUBLE PRECISION in the D version) array of length MAXA. For entry to MC20A/AD the user must set it to contain the nonzeros in any order. On exit from MC20A/AD they are reordered so that column 1 precedes column 2 which precedes column 3, etc. but the order within columns is arbitrary. This format is required for

entry to MC20B/BD. On exit from MC20B/BD the nonzeros are also ordered within each column.

- INUM is an INTEGER array of length MAXA. On entry to and exit from MC20A/AD and MC20B/BD the absolute value of INUM(K) is the row index of the entry in A(K). The values, including signs, are moved so the user is at liberty to use these signs as flags attached to the nonzeros.
- JPTR is an INTEGER array of length NC. It is not required to be set for entry to MC20A/AD. On exit from MC20A/AD and on entry to and exit from MC20B/BD it contains the position in A of the first element of column J, J=1, 2,..., NC.
- JNUM is an INTEGER array of length MAXA. On entry to MC20A/AD, JNUM(K)+JDISP is the column index of the entry held in A(K). It is destroyed by MC20A/AD.
- JDISP is an INTEGER which must be set by the user to the required displacement for column indices. Normally zero will be suitable. JDISP is not altered by MC20A/AD.

2.2 Notes

It is expected that this subroutine will be called by other library subroutines but not by the user directly. There are no checks on the validity of the data and no error exits.

3 GENERAL INFORMATION

Workspace: None.

Use of common: None.

Other routines called directly: None.

Input/output: None.

Restrictions: None.

4 METHOD

MC20A/AD is an in-place sort algorithm which handles each item to be sorted exactly 3 times, so it is of order MAXA. The number of entries in each column is first obtained by a counting pass. The space needed by each column is allocated. Each entry in turn is made the *current entry* and examined to see if it is in place. If not, it is put into the next location allotted for the column it occurs in, and the entry displaced made the current entry. This chain of displacing entry continues until the first entry examined in the chain is located and stored. Then the next item is examined. A flag prevents an entry being moved twice.

MC20B/BD is a pairwise interchange algorithm of maximum order r(r-1)/2, for each column, where r is the number of entries in the column.

5 EXAMPLE OF USE

We illustrate the use of the package by reading the matrix

$$\mathbf{A} = \begin{pmatrix} 2 & 3 & 4 & 6 \\ & 1 & 5 & \\ & 5 & & 1 \end{pmatrix},$$

and sorting with each entry in turn.

Suitable code is as follows:-

C SIMPLE EXAMPLE OF USE OF MC20 PACKAGE

```
INUM(30), JPTR(30), JNUM(30)
      INTEGER
      DOUBLE PRECISION A(30)
С
C STORE VALUE FOR JDISP
      JDISP = 0
С
C READ MATRIX
      READ (5,*) NC,MAXA
      READ (5,*) (INUM(I),JNUM(I),A(I),I=1,MAXA)
С
C SORT MATRIX ENTRIES
      CALL MC20AD(NC,MAXA,A,INUM,JPTR,JNUM,JDISP)
С
C PRINT SORTED MATRIX
      JPTR(NC+1) = MAXA + 1
      DO 30 J = 1,NC
         K1 = JPTR(J)
         K2 = JPTR(J+1) - 1
         WRITE(6,20)J,(INUM(K),A(K),K=K1,K2)
 20
         FORMAT(' COLUMN', 12, ':', 3(14, F4.1))
30
      CONTINUE
С
C SORT WITHIN THE COLUMNS
      CALL MC20BD(NC, MAXA, A, INUM, JPTR)
С
C PRINT SORTED MATRIX
      WRITE(6,*)
      DO 40 J = 1,NC
         K1 = JPTR(J)
         K2 = JPTR(J+1) - 1
         WRITE(6,20)J,(INUM(K),A(K),K=K1,K2)
40
      CONTINUE
С
      STOP
      END
```

Suitable data is as follows:-

 $\begin{array}{cccccccc} 5 & 8 \\ 1 & 1 & 2 . 0 \\ 2 & 2 & 3 . 0 \\ 2 & 3 & 4 . 0 \\ 2 & 5 & 6 . 0 \\ 3 & 3 & 1 . 0 \\ 3 & 4 & 5 . 0 \\ 4 & 3 & 5 . 0 \\ 5 & 5 & 1 . 0 \end{array}$

This produces the following output

COLUMN COLUMN COLUMN COLUMN COLUMN	1: 2: 3: 4: 5:	1 2 3 2	2.0 3.0 4.0 5.0 6.0	4 5	5.0 1.0	3	1.0
COLUMN COLUMN COLUMN COLUMN COLUMN	1: 2: 3: 4: 5:	1 2 2 3 2	2.0 3.0 4.0 5.0 6.0	3 5	1.0 1.0	4	5.0

MC20