

**Warning:** Subroutine MC20 has been superseded by subroutine MC49 which uses improved algorithms; the use of the latter routine is recommended. The superseded routine may be removed from later releases of the library.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



```

      INTEGER      INUM(30),JPTR(30),JNUM(30)
      DOUBLE PRECISION A(30)
C
C STORE VALUE FOR JDISP
      JDISP = 0
C
C READ MATRIX
      READ (5,*) NC,MAXA
      READ (5,*) (INUM(I),JNUM(I),A(I),I=1,MAXA)
C
C SORT MATRIX ENTRIES
      CALL MC20AD(NC,MAXA,A,INUM,JPTR,JNUM,JDISP)
C
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',I2,':',3(I4,F4.1))
30      CONTINUE
C
C SORT WITHIN THE COLUMNS
      CALL MC20BD(NC,MAXA,A,INUM,JPTR)
C
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
C
      STOP
      END

```

Suitable data is as follows:-

```

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

```

This produces the following output

```

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

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

```