Warning: Subroutine ME20 has been superseded by subroutine MF49 which uses improved algorithms; the use of the latter routine is recommended. The superseded routine may be removed from later releases of the library.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 entries of a sparse complex matrix from arbitrary order to an ordering by columns or a complete ordering by columns and rows.

The subroutine allows users with sparse problems to input the entries 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 entries.

ATTRIBUTES — **Version:** 1.0.0. **Types:** ME20A, ME20AD. **Remark:** Superseded by MF49. **Original date:** August 1979. **Origin:** I.S.Duff, Harwell.

2 HOW TO USE THE PACKAGE

2.1 Argument lists and calling sequences

There are two entries

(a) to sort the entries of a sparse matrix from arbitrary order to column order, unordered within each column.

The single precision version.

```
CALL ME20A(NC,MAXA,A,INUM,JPTR,JNUM,JDISP)
```

The double precision version.

```
CALL ME20AD(NC,MAXA,A,INUM,JPTR,JNUM,JDISP)
```

(b) to sort the entries within each column of a sparse matrix stored by columns.

The single precision version.

```
CALL ME20B(NC,MAXA,A,INUM,JPTR)
```

The double precision version.

```
CALL ME20BD(NC,MAXA,A,INUM,JPTR)
```

NC is an INTEGER variable which must be set by the user to the number of matrix columns. This argument is not altered by the subroutine.

MAXA is an INTEGER variable which must be set by the user to the number of entries in the matrix This argument is not altered by the subroutine.

A is a REAL (DOUBLE PRECISION in the D version) two dimensional array of first dimension 2 and second dimension of length at least MAXA. On entry to ME20A/AD the user must set it to contain the entries in any order. (A(1,*), A(2,*) containing the real and imaginary parts respectively). On exit from ME20A/AD they are

reordered so that column 1 precedes column 2 which precedes column 3, etc., but the order within the columns is arbitrary. This format is required for entry to ME20B/BD. On exit from ME20B/BD the entries are also ordered within each column.

INUM is an INTEGER array of length at least MAXA. On entry to and on exit from ME20A/AD and ME20B/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 entries.

JPTR is an INTEGER array of length at least NC. It need not be set on entry to ME20A/AD. On exit from ME20A/AD and on entry to and on exit from ME20B/BD it contains the position in A of the first element of column J, J =1,..., NC.

JNUM is an INTEGER array of length at least MAXA. On entry to ME20A/AD, JNUM(K)+JDISP is the column index of the element held in A(K). The contents of this array are destroyed by ME20A/AD.

JDISP is an INTEGER variable which must be set by the user to the required displacement for column indices. Normally zero will be suitable. This argument is not altered by the subroutine.

2.2 Remark

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

Use of common: none.

Workspace: none.

Other subprograms: none.

Input/Output: none.

Restrictions: none.

4 METHOD

ME20A/AD uses an in place sort algorithm which handles each item to be sorted exactly three times, so it is of order MAXA. The number of elements in each column is first obtained by a counting pass. Space needed by each column is allocated. Each element in turn is made the *current element* 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 element displaced made the current element. This chain of displacing elements continues until the first element examined in the chain is located and stored. Then the next item is examined. A flag prevents an element being moved twice.

ME20B/BD uses a pairwise interchange algorithm of maximum order $r(r-1)/2$ for each column, where r is the number of elements in the column.