1 SUMMARY

This subroutine **sorts the sparsity pattern of a complex matrix to an ordering by columns.** There is an option for ordering the entries within each column by increasing row indices and an option for checking the user-supplied matrix entries for indices which are out-of-range or duplicated.

ATTRIBUTES — Version: 1.0.0. **Remark:** This supersedes ME20. **Types:** MF49A, MF49AD. **Original date:** April 1993. **Origin:** J.A.Scott, Rutherford Appleton Laboratory.

2 HOW TO USE THE PACKAGE

2.1 The argument list

The single precision version

```
CALL MF49A(IND,NC,NR,NNZ,IRN,JCN,YESA,LA,A,LIP,IP,LIW,IW,IFLAG)
```

The double precision version

```
CALL MF49AD(IND,NC,NR,NNZ,IRN,JCN,YESA,LA,A,LIP,IP,LIW,IW,IFLAG)
```

IND is an INTEGER variable which must be set on entry to specify whether the user-supplied matrix entries are to be checked for errors and whether ordering by increasing row indices within each column is required. If IND is set to 1, error checking is performed and the ordering within each column will be arbitrary. If IND is set to 2, error checking is performed and the ordering within each column will be by increasing row indices. If IND is set to -1, only the scalar input data are checked for errors and the ordering within each column will be arbitrary. If IND is set to -2, only the scalar input data is checked for errors and the ordering within each column will be by increasing row indices. This argument is unchanged by the routine. **Restriction:** IND=-2, -1, 1, or 2.

NC is an INTEGER variable which must be set to the number of columns in the matrix. This argument is unchanged by the routine. **Restriction:** NC ≥ 1.

NR is an INTEGER variable which must be set to the number of rows in the matrix. This argument is unchanged by the routine. **Restriction:** NR ≥ 1.

NNZ is an INTEGER variable which must be set to the number of entries in the matrix. This argument is unchanged by the routine. **Restriction:** NNZ ≥ 1.

IRN is an INTEGER array of length NNZ which must be set to contain the row indices of the entries in the matrix. The entries may be in any order. On exit, the row indices are reordered so that the entries of a single column are contiguous with column J preceding column J+1 (J=1,2,...,N-1), with no space between columns. If IND=-1 o 1, the order within each column is arbitrary; if IND=−2 or 2, the order within each column is by increasing row indices.

JCN is an INTEGER array of length NNZ. On entry, JCN(K) must be set to the column index of the entry whose row index is held in IRN(K) (K=1,2,...,NNZ). The contents of this array are altered by the routine.

YESA is a LOGICAL variable which must be set by the user. If YESA = .FALSE., only the sparsity pattern of the matrix will be ordered by columns. If YESA = .TRUE., the numerical values of the entries of the matrix will also be ordered by columns. This argument is unchanged by the routine.

LA is an INTEGER variable which defines the length of the array A. This argument is unchanged by the routine. **Restrictions:** If YESA = .TRUE., LA ≥ NNZ; if YESA = .FALSE., LA ≥ 1.

A is a COMPLEX (COMPLEX*16 in the D version) array of length LA. If YESA = .TRUE., A(K) must be set by the user

to hold the value of the entry with indices IRN(K) and JCN(K) (K=1,2,...,NNZ). On exit, the array will have been permuted in the same way as the array IRN. If YESA = .FALSE., the array is not accessed.

LIP is an INTEGER variable which defines the length of the array IP. This argument is unchanged by the routine. **Restrictions:** If IND=-1 or 1, LIP ≥ NC+1; if IND=-2 or 2, LIP ≥ MAX(NC, NR)+1.

IP is an INTEGER array of length LIP which need not be set on entry. On exit, IP(J) contains the position in the array IRN of the first entry in column J (J=1,2,...,NC), and IP(NC+1) is set one greater than the number of entries in the matrix.

LIW is an INTEGER variable which defines the length of the array IW. This argument is unchanged by the routine. **Restrictions:** If IND=-1 or 1, LIW ≥ NC+1; if IND=-2 or 2, LIW ≥ NR+1.

IW is an INTEGER array of length LIW. This array is used by the routine as workspace.

IFLAG is an INTEGER variable which need not be set on entry. On exit, a negative value of IFLAG is used to signal a fatal error in the input data, a positive value of IFLAG is used to indicate that a warning has been issued, and a zero value is used to indicate a successful call to the routine. A positive value of IFLAG can only be returned if IND=1 or 2. Possible nonzero values of IFLAG and their consequences are as follows;

–1 – IND violates the restrictions on it.
–2 – NC, or NR, or NNZ is less than 1.
–3 – LA is too small.
–4 – LIW is too small.
–5 – LIP is too small.
+1 – Multiple entries have been input. One copy of each multiple entry is kept and, if YESA = .TRUE., the values of these entries are added together. Initially IDUP in COMMON is set to zero. If an entry appears *k* times, IDUP is incremented by *k*–1 and NZOUT in COMMON is set to the revised number of entries in the matrix. Note that this warning will overwrite an IFLAG = 2 or an IFLAG = 3 warning.
+2 – One or more of the entries in IRN is out-of-range. These entries are removed by the routine. IOUT in COMMON is set to the number of entries which were out-of-range and NZOUT in COMMON is set to the revised number of entries in the matrix.
+3 – One or more of the entries in JCN are out-of-range. These entries are removed by the routine. JOUT in COMMON is set to the number of entries which were out-of-range and NZOUT in COMMON is set to the revised number of entries in the matrix. Note that this warning will overwrite an IFLAG = 2 warning.

2.2 Common blocks

One common block is used by MF49A/AD. The common block is:

The single precision version

```
COMMON/ MF49E/ LP, MP, IOUT, JOUT, IDUP, NZOUT
```

The double precision version

```
COMMON/ MF49ED/ LP, MP, IOUT, JOUT, IDUP, NZOUT
```

where the parameters are given default values by a block data subprogram MF49D/DD.

LP is an INTEGER variable used as the unit number of the device to which error messages are sent. The default value is 6. Error messages can be suppressed by setting LP ≤ 0. This parameter is unchanged by the routine.

MP is an INTEGER variable used as the unit number of the device to which warning messages are sent. The default value is 6. Warning messages can be suppressed by setting MP ≤ 0. This parameter is unchanged by the routine.

IOUT, JOUT, IDUP, and NZOUT are INTEGER variables. In case of a warning message, information on the number of entries supplied by the user which were out-of-range or were duplicated is placed in one or more of these variables. For details, see §2.1.

3 GENERAL INFORMATION

Use of common: MF49A/AD uses the common block MF49E/ED; see §2.2.

Workspace: An integer array IW of length LIW is used by MF49A/AD as workspace see §2.1.

Other routines called directly: The routine MF49A/AD calls the internal subroutine MF49B/BD and, if IND=-2 or 2, the internal subroutine MF49C/CD. MF49A/AD uses a block data subprogram MF49D/DD.

Input/output: Error messages on unit LP and warning messages on unit MP. These have default value 6. Printing of error and warning messages is suppressed by setting LP and MP less than or equal to 0.

Restrictions: The routine MF49A/AD has the following restrictions:

IND=-2, -1, 1 or 2,
NC≥1, NR≥1, NNZ≥1,
if YESA = .TRUE., LA≥NNZ,
if YESA = .FALSE., LA≥1,
if IND=-1 or 1, LIP≥NC+1 and LIW≥NC+1,
if IND=-2 or 2, LIP≥MAX(NC,NR)+1 and LIW≥NR+1.

4 METHOD

MF49 is a complex version of MC49. For details of the method, the user is referred to MC49.

5 EXAMPLE OF USE

As a simple example, suppose we wish to order the sparse matrix

$$\begin{pmatrix} 1+i & 1+2i & & 1+4i \\ & 2+2i & & \\ 3+i & & 3+3i & 3+4i \\ & & & 4+4i \\ & 5+2i & & \end{pmatrix}$$

by columns, with the entries ordered by increasing row indices within each column. Using the following program:

```
C     .. Parameters ..
      INTEGER MAXN,MAXNZ
      PARAMETER (MAXN=5,MAXNZ=25)
C     ..
C     .. Local Scalars ..
      INTEGER IFLAG,IND,J,K,K1,K2,LA,LIP,LIW,NC,NNZ,NR
      LOGICAL YESA
C     ..
C     .. Local Arrays ..
      COMPLEX A(MAXNZ)
      INTEGER IP(MAXN+1),IRN(MAXNZ),IW(MAXN+1),JCN(MAXNZ)
C     ..
C     .. External Subroutines ..
      EXTERNAL MF49A
C     ..
C     .. Intrinsic Functions ..
      INTRINSIC MAX
C     ..
      IND = 2
      YESA = .TRUE.
      READ (5,FMT=*) NR,NC,NNZ
      DO 10 K = 1,NNZ
         READ (5,FMT=*) IRN(K),JCN(K),A(K)
```

```
     10 CONTINUE
        LA = NNZ
        LIP = MAX(NC,NR) + 1
        LIW = NR + 1
        CALL MF49A(IND,NC,NR,NNZ,IRN,JCN,YESA,LA,A,LIP,IP,LIW,IW,IFLAG)
        WRITE (6,FMT=9000) IFLAG
        IF (IFLAG.LT.0) GO TO 40
        DO 30 J = 1,NC
           K1 = IP(J)
           K2 = IP(J+1) - 1
           IF (K1.LE.K2) THEN
              WRITE (6,FMT=9010) J
              DO 20 K = K1,K2
                 WRITE (6,FMT=9020) IRN(K),A(K)
     20       CONTINUE
           END IF
     30 CONTINUE
     40 CONTINUE
        STOP
   9000 FORMAT ('  IFLAG = ',I2,' on exit from MF49A ')
   9010 FORMAT ('  Column ',I2)
   9020 FORMAT ('     Row ',I2,' Value ',2F4.1)
        END
```

on the data

```
     5   4   9
     1   1   (1., 1.)
     3   3   (3., 3.)
     3   4   (3., 4.)
     3   1   (3., 1.)
     1   2   (1., 2.)
     1   4   (1., 4.)
     2   2   (2., 2.)
     4   4   (4., 4.)
     5   2   (5., 2.)
```

produces the output:

```
    IFLAG =  0 on exit from MF49A
    Column  1
       Row  1 Value  1.0 1.0
       Row  3 Value  3.0 1.0
    Column  2
       Row  1 Value  1.0 2.0
       Row  2 Value  2.0 2.0
       Row  5 Value  5.0 2.0
    Column  3
       Row  3 Value  3.0 3.0
    Column  4
       Row  1 Value  1.0 4.0
       Row  3 Value  3.0 4.0
       Row  4 Value  4.0 4.0
```