

## 1 SUMMARY

To **write a sparse matrix in Rutherford-Boeing format**. The matrix can be input as an assembled matrix in either column-oriented or coordinate form, or as an unassembled finite-element matrix. The HSL routine MC55 can be used to write supplementary files in Rutherford-Boeing format and the HSL routine MC56 can be used to read files held in Rutherford-Boeing format. For information on the Rutherford-Boeing Sparse Matrix Collection (RBSMC), the user should consult the report

The Rutherford-Boeing Sparse Matrix Collection. I.S. Duff, R.G. Grimes, and J.G. Lewis. Report RAL-TR-97-031, Revision 1, 1999, Rutherford Appleton Laboratory.

**ATTRIBUTES** — **Version:** 1.0.0. **Types:** Real (single, double), Integer, Complex (single, double). **Remark:** This package is also included in the HSL Archive. On April 7 2001, calls to MC49 and MF49 were replaced by calls to MC59 to make the routine threadsafe. **Calls:** MC59. **Language:** Fortran 77. **Original date:** September 2000. **Origin:** I.S. Duff, Rutherford Appleton Laboratory. **Licence:** A third-party licence for this package is available without charge.

## 2 HOW TO USE THE PACKAGE

### 2.1 The argument list

There is a single subroutine for both assembled and unassembled matrices.

*The single precision version*

```
CALL MC54A( ICNTL, TITLE, KEY, M, N, NE, IP, IND, VALUE, IW, INFO )
```

*The double precision version*

```
CALL MC54AD( ICNTL, TITLE, KEY, M, N, NE, IP, IND, VALUE, IW, INFO )
```

*The integer version*

```
CALL MC54AI( ICNTL, TITLE, KEY, M, N, NE, IP, IND, VALUE, IW, INFO )
```

*The complex version*

```
CALL MC54AC( ICNTL, TITLE, KEY, M, N, NE, IP, IND, VALUE, IW, INFO )
```

*The double precision complex version*

```
CALL MC54AZ( ICNTL, TITLE, KEY, M, N, NE, IP, IND, VALUE, IW, INFO )
```

ICNTL is an INTEGER array of length 10 whose first 5 components must be set by the user. It holds information on the input matrix and output file and controls the action of the subroutine. This argument is not altered by the subroutine.

ICNTL(1) must be set by the user to the unit number for the output file on which the matrix in Rutherford-Boeing format will be written.

ICNTL(2) must be set by the user to indicate the precision of the output reals. It should be set to the number of significant decimal digits in the numerical value of the entries. Values of ICNTL(2) less than or equal to one or greater than 17 are treated as if they were 17. ICNTL(2) is not accessed if ICNTL(3)=0. Not used or accessed by MC54AI.

ICNTL(3) must be set by the user to indicate whether only the pattern is being supplied (ICNTL(3)=0) or whether real values are supplied (ICNTL(3)≠0).

ICNTL(4) must be set by the user to indicate whether the matrix is symmetric (ICNTL(4)=0), skew symmetric

(ICNTL(4)=1), rectangular (ICNTL(4)=2), or square unsymmetric (ICNTL(4)=3). For complex matrices (calls to MC54AC/AZ), ICNTL(4)=4 indicates matrix is Hermitian. Note that the Rutherford-Boeing format does not allow the definition of a skew symmetric matrix without numerical values. Values of ICNTL(4) outside the range 0 to 3 (4 for the case of MC54AC or MC54AZ) will result in an error return.

ICNTL(5) must be set by the user to indicate the format of the input matrix: ICNTL(5)=0 for an assembled matrix in column-oriented form, ICNTL(5)=1 for an assembled matrix in coordinate form, and ICNTL(5) = 2 for an unassembled finite-element matrix. Values of ICNTL(5) outside the range 0 to 2 will result in an error return.

ICNTL(6) to ICNTL(10) are not accessed by the routine.

TITLE is a CHARACTER\*72 variable that must be set by the user to the text for the title line for the matrix. This argument is not altered by the subroutine.

KEY is a CHARACTER\*8 variable that must be set by the user to the identifying key for the matrix. This should be distinct from that of any other matrix. This argument is not altered by the subroutine.

M is an INTEGER variable that must be set by the user. If the matrix is assembled, M is the number of rows of the matrix; if the matrix is an unassembled finite-element matrix, M is the value of the largest index used. This argument is not altered by the subroutine.

N is an INTEGER variable that must be set by the user. If the matrix is assembled, N is the number of columns of the matrix; if the matrix is an unassembled finite-element matrix, N is the number of elements. This argument is not altered by the subroutine.

NE is an INTEGER variable that must be set by the user. If the matrix is assembled, NE is the number of entries in the matrix; if the matrix is an unassembled finite-element matrix, NE is the number of indices in the element lists. This argument is not altered by the subroutine.

IP is an INTEGER array of length  $\text{MAX}(M,N)+1$  or  $2*N+1$  if input is an unassembled finite-element matrix with rectangular element matrices. If the matrix is stored in column-oriented form, IP must be set by the user so that  $IP(J)$ ,  $J=1, \dots, N$ , is the position in IND of the first entry in column J and  $IP(N+1)-1$  holds the position of the last entry in column N. For unassembled finite-element problems (ICNTL(5)  $\neq$  0 or 1), IP must hold the position of the start of each element in array IND. If elements are rectangular, then the row indices for element J are in positions  $IP(2*J-1)$  to  $IP(2*J)-1$  of IND and the column indices for element J are in positions  $IP(2*J)$  to  $IP(2*J+1)-1$ . In all these cases, this argument is not altered by the subroutine. If the matrix is held in coordinate form, IP need not be set on entry and, on exit, will be set to column pointers for the reordered matrix.

IND is an INTEGER array whose first NE positions must be set by the user. If the matrix is assembled, IND must hold the row indices of the entries. If storage is column-oriented, these must be ordered by columns. The row indices within each column must be in ascending order. If storage is in coordinate form, the corresponding column indices must be in positions  $IND(NE+I)$ ,  $I=1, \dots, NE$ . In this case, the matrix entries can be in any order. For a symmetric matrix, only the lower triangle should be stored. If the matrix is an unassembled finite-element matrix, IND must hold the list of indices for each element in turn. IND is not altered by the subroutine, unless input is in coordinate form (ICNTL(5)=1), in which case it will be reset to hold the row indices of the reordered matrix.

VALUE is a REAL (DOUBLE PRECISION in the D version, INTEGER in the I version, COMPLEX in the C version, or DOUBLE PRECISION COMPLEX in the Z version) array. If ICNTL(3)  $\neq$  0, it must be set by the user to contain the numerical values of the entries. For an assembled matrix, the first NE positions must contain the values of the entries whose row indices are in corresponding positions in IND. For an unassembled finite-element matrix, the first entries of VALUE must contain contiguously the numerical values of each element held by columns. If these are symmetric, only the lower triangle should be held ordered by columns. VALUE is not altered by the subroutine, unless the matrix is input in coordinate form (ICNTL(5)=1), in which case it will be reset to hold

the values for the reordered matrix.

**IW** is an INTEGER work array of length  $\text{MAX}(M,N)+1$  that is only accessed if the matrix is input in coordinate form ( $\text{ICNTL}(5)=1$ ).

**INFO** is an INTEGER array of length 5 that need not be set by the user.  $\text{INFO}(1)$  will be set to 0 in the event of a successful call to MC54A/AD/AI/AC/AZ and will have a negative value if an error is detected (see Section 2.2). In the case of an error,  $\text{INFO}(2)$  will hold further information.  $\text{INFO}(3)$  to  $\text{INFO}(5)$  are currently not used by the routine.

## 2.2 Error diagnostics

A successful return from MC54A/AD/AI/AC/AZ is indicated by a value of  $\text{INFO}(1)$  equal to zero. Possible nonzero values for  $\text{INFO}(1)$  are given below.

- 1  $M < 1$ . Value of  $M$  returned in  $\text{INFO}(2)$ .
- 2  $N < 1$ . Value of  $N$  returned in  $\text{INFO}(2)$ .
- 3  $NE < 1$ . Value of  $NE$  returned in  $\text{INFO}(2)$ .
- 4  $\text{ICNTL}(4)$  less than 0 or greater than 3. Value of  $\text{ICNTL}(4)$  returned in  $\text{INFO}(2)$ .
- 5  $\text{ICNTL}(5)$  less than 0 or greater than 2. Value of  $\text{ICNTL}(5)$  returned in  $\text{INFO}(2)$ .
- 6  $\text{ICNTL}(4)$  is set to 1 when  $\text{ICNTL}(3)$  is equal to 0.  $\text{INFO}(2)$  has the value zero.
- 7 Matrix was submitted in column-oriented form with  $\text{IP}(I+1) < \text{IP}(I)$  for some  $I$ ,  $1 \leq I \leq N$ . Value of  $I$  returned in  $\text{INFO}(2)$ .
- 8 Index out-of-range. Value of first out-of-range component returned in  $\text{INFO}(2)$ .
- 9 Matrix was submitted in column-oriented form with rows not in order within each column. Column index returned in  $\text{INFO}(2)$ .

## 3 GENERAL INFORMATION

**Workspace:** **IW** of length  $N+1$ . Only used if  $\text{ICNTL}(5)=1$ .

**Use of common:** None.

**Other routines called directly:** The entries MC54A/AD/AI/AC/AZ call internal routines MC54B/BD/BI/BC/BZ, MC54C/CD/CI/CC/CZ, MC54D/DD/DI/DC/DZ, and MC54E/ED/EI/EC/EZ, respectively. These need not be called by the user. The entries MC54A/AD/AC/AZ/AI also call the sorting routines MC59A/AD/AC/AZ/AI (only if  $\text{ICNTL}(5)=1$ ).

**Input/output:** Only the Rutherford-Boeing file for the matrix that is output to unit  $\text{ICNTL}(1)$ .

## 4 METHOD

The code calculates data values for the header lines and then outputs them to the file  $\text{ICNTL}(1)$ . It then sorts the matrix to column order if input is in coordinate form and writes the matrix in the Rutherford-Boeing format to the file  $\text{ICNTL}(1)$ .

## 5 EXAMPLES OF USE

We illustrate the use of the subroutine by a small program that reads a matrix from a file and outputs it in Rutherford-Boeing format.

```
C This program reads an unsymmetric or rectangular matrix in triplet form
C from the file on unit 5, then outputs the Rutherford-Boeing format
C to file on unit 6.
```

```
PROGRAM SPEC

INTEGER          NZMAX,NMAX
PARAMETER        (NZMAX=1000,NMAX=100)
DOUBLE PRECISION A(NZMAX)
INTEGER          IND(2*NZMAX),IP(NMAX+1)
INTEGER          ICNTL(10),M,N,NE,IW(NMAX+1),INFO(5),K
CHARACTER        TITLE*72,KEY*8,CARD*80

EXTERNAL        MC54AD

C Read in triplets from file
DO 10 K=1,NZMAX
  READ(5,*,END=20) IND(K),IND(NZMAX+K),A(K)
  IF (IND(K).EQ.0 .AND. IND(NZMAX+K).EQ.0) GO TO 20
10 CONTINUE
  WRITE(6,'(A)') 'Failure .. must increase value of NZMAX'
  GO TO 500

C Compute values of M, N, and NE and test they are within range.
20 NE = K-1
  M = 0
  N = 0
  DO 30 K=1,NE
    M = MAX(M,IND(K))
    N = MAX(N,IND(NZMAX+K))
    IND(NE+K) = IND(NZMAX+K)
30 CONTINUE
  IF (M.GT.NMAX .OR. N.GT.NMAX) THEN
    WRITE(6,'(A)') 'Failure .. must increase value of NMAX'
    GO TO 500
  ENDIF

C Set parameters for call.
  TITLE = 'spec_sheet_test_matrix'
  KEY   = 'TEST_01'

C Output unit
  ICNTL(1) = 6
C Non-default format for reals
  ICNTL(2) = 2
C Reals present.
  ICNTL(3) = 1
C Unsymmetric
  ICNTL(4) = 3
C Rectangular
  IF (M.NE.N) ICNTL(4) = 2
C Coordinate entry
  ICNTL(5) = 1
C Call subroutine and check for errors
  CALL MC54AD(ICNTL,TITLE,KEY,M,N,NE,IP,IND,A,IW,INFO)
  IF (INFO(1).LT.0) WRITE(6,'(A,2I12)') 'Value of INFO is: ',INFO
```

```
500 STOP
    END
```

With the input:

```
1 3 1.3
1 1 1.1
2 2 2.2
2 3 2.3
1 4 1.4
```

this program produces, in file on unit 6:

```
spec_sheet_test_matrix                                TEST_01
              3              1              1              1
rra              2              4              5              0
(40I2)          (40I2)          (8E10.1E3)
1 2 3 5 6
1 2 1 2 1
1.1E+000 2.2E+000 1.3E+000 2.3E+000 1.4E+000
```