## 1 SUMMARY

To **write a supplementary file in Rutherford-Boeing format**. The HSL routine MC54 can be used to write matrices 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 — Versions:** MC55A, MC55AD MC55AI, MC55AC, MC55AZ. **Remark:** This package is also included in the HSL Archive. **Language:** Fortran 77. **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 MC55A(ICNTL,TITLE,KEY,CASEID,DATTYP,POSITN,ORGNIZ,M,NVEC,NAUXD,
     *           IP,IND,VALUE,INFO)
```

*The double precision version*

```
      CALL MC55AD(ICNTL,TITLE,KEY,CASEID,DATTYP,POSITN,ORGNIZ,M,NVEC,NAUXD,
     *           IP,IND,VALUE,INFO)
```

*The integer version*

```
      CALL MC55AI(ICNTL,TITLE,KEY,CASEID,DATTYP,POSITN,ORGNIZ,M,NVEC,NAUXD,
     *           IP,IND,VALUE,INFO)
```

*The complex version*

```
      CALL MC55AC(ICNTL,TITLE,KEY,CASEID,DATTYP,POSITN,ORGNIZ,M,NVEC,NAUXD,
     *           IP,IND,VALUE,INFO)
```

*The double precision complex version*

```
      CALL MC55AZ(ICNTL,TITLE,KEY,CASEID,DATTYP,POSITN,ORGNIZ,M,NVEC,NAUXD,
     *           IP,IND,VALUE,INFO)
```

ICNTL is an INTEGER array of length 10 whose first 2 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 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 DATTYP is equal to ord, ipt, or icv, nor on a call to MC55AI.

ICNTL(3) 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 supplementary data. 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 associated with this supplementary data. This argument is not altered by the subroutine.

CASEID is a CHARACTER*8 variable that must be set by the user to an identifying key for the data. Different supplementary files of the same type for the same matrix must have different values for CASEID. This argument is not altered by the subroutine.

DATTYP is a CHARACTER*3 variable that must be set by the user to indicate the type of supplementary data being written. Current possible values are:

ord ... orderings

rhs ... right-hand sides

sln ... solutions

est ... estimates

evl ... eigenvalues

svl ... singular-values

evc ... eigenvectors

svc ... singular-vectors

sbv ... Schur basis vectors

sbm ... Schur basis matrix

sbp ... Schur basis parameters

ipt ... partition

icv ... covering

lvl ... Laplacian-values

lvc ... Laplacian-vectors

geo ... geometry

avl ... auxiliary values

POSITN is a CHARACTER*1 variable that indicates position attributes of the problem defined by the supplementary data. It need only be set for values of DATTYP equal to ord, rhs, sln, est, evc, svc, lvc, ipt, icv, and geo. The meaning of POSITN in each of these cases is as follows:

ord POSITN has the value s, l, or r according to whether the ordering is applied symmetrically, on the left, or on the right.

rhs, sln, est POSITN has the value l or r according to whether the linear system is of the form **Ax=b** or $\mathbf{A}^T\mathbf{x}=\mathbf{b}$ respectively.

evc, svc, lvc POSITN has the value s, l, or r according to whether the vectors represent both left and right eigensolutions (lvc and symmetric evc case), or the left or right vectors respectively.

ipt, icv POSITN has the value s, l, or r according to whether the partition or covering should be applied symmetrically, or on the left (rows) or right (columns) respectively.

geo POSITN has the value s, l, or r according to whether the geometry data applies symmetrically, or represents the left/row vertices or right/column vertices respectively.

In these cases, this argument is not altered by the subroutine; otherwise it is set to a blank character.

ORGNIZ is a CHARACTER*1 variable that need only be set if DATTYP is equal to rhs. It indicates whether the right-hand side is sparse (s), dense (d), or elemental (e). In this case, this argument is not altered by the subroutine; otherwise it is set to a blank character.

M is an INTEGER variable that must be set by the user to the row dimension of the vector data. The actual meaning depends on the value of DATTYP. For ord, rhs, sln, est, evc, svc, lvc, sbv, ipt, icv, avl it is the row dimension of the vectors involved; for evl, svl, lvl, sbm, sbp, geo it is the number of values, vectors, or points involved. This argument is not altered by the subroutine.

NVEC is an INTEGER variable that indicates the number of vectors involved. It must be set by the user unless DATTYP equals evl, svl, lvl, or sbp (in which case it is set by the subroutine to have the value 1). In the case of DATTYP equal to geo, it should be set to the geometric dimensionality. Unless set to 1, as above, this argument is not altered by the subroutine.

NAUXD is an INTEGER variable that need only be set if DATTYP is equal to rhs and ORGNIZ is equal to e. In this case it must be set to the total number of numerical entries in all right-hand sides; In this case, this argument is not altered by the subroutine; in all other cases it is altered.

IP is an INTEGER array that need only be set by the user if DATTYP is equal to rhs (ORGNIZ equal to s), ipt, or icv. For rhs (ORGNIZ equal to s), it must be used to hold the positions of the starts of each sparse right-hand side in IND and VALUE; for ipt and icv it must be set to hold the positions of the beginnings of each partition or covering in IND.

IND is an INTEGER array that must be set by the user if DATTYP has the value ord, rhs (ORGNIZ equal to s), ipt, or icv. For ord, it must be set to the orderings; for rhs (ORGNIZ equal to s), it must be set to the row indices of the entries in the sparse right-hand sides; and for ipt and icv to the indices for each partitioning or covering.

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. It need not be set by the user if DATTYP is equal to ord, ipt, or icv. Otherwise it must be set to the value of the numerical data.

INFO is an INTEGER array of length 5 that need not be set by the user. INFO(1) will be set to 0 in the event of a successful call to MC55A/AD/AC/AZ/AI and will have a negative value if an error is detected (see Section 2.2). In the case of an error, INFO(2) will hold further information. INFO(3) to INFO(5) are currently not used by the routine.

### 2.2 Error diagnostics

A successful return from MC55A/AD/AC/AZ/AI is indicated by a value of INFO(1) equal to zero. Possible nonzero values for INFO(1) are given below.

–1 M < 1. Value of M returned in INFO(2).

–2 NVEC < 1. Value of NVEC returned in INFO(2).

–3 Value of DATTYP not supported by this code.

## 3 GENERAL INFORMATION

**Workspace:** None.

**Use of common:** None.

**Other routines called directly:** `MC55A/AD/AC/AZ/AI` calls internal routines `MC55B/BD/BC/BZ/BI`, `MC55C/CD/CC/CZ/CI`, `MC55D/DD/DC/DZ/DI`, and `MC55E/ED/EC/EZ/EI`, respectively. These need not be called by the user.

**Input/output:** Only the Rutherford-Boeing file for the matrix that is output to unit `ICNTL(1)`.

## 4 METHOD

The code calculates data values for the header lines and then outputs them and writes the data in the required format to the file on unit `ICNTL(1)`.

## 5 EXAMPLES OF USE

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

```
C   This program reads two sparse right-hand sides from standard input
C   (unit 5), and then outputs the Rutherford-Boeing format to file
C   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,NVEC,INFO(5),K,NAUXD
      CHARACTER        FILEIN*80,TITLE*72,KEY*8,CARD*80,CASEID*8,
     *                 DATTYP*3, POSITN*1, ORGNIZ*1

      EXTERNAL         MC55AD

      READ(5,*) M, NVEC
C Read in rhs from file
      READ(5,*) (IP(K),K=1,NVEC+1)
      READ(5,*) (IND(K),K=1,IP(NVEC+1)-1)
      READ(5,*) (A(K),K=1,IP(NVEC+1)-1)


C Set parameters for call.
      ICNTL(1) = 6
      ICNTL(2) = 3
      TITLE = 'spec_sheet_rhs'
      KEY   = 'TEST_01'
      DATTYP = 'rhs'
      CASEID = 'RHS_01'
      POSITN = 'l'
      ORGNIZ = 's'
      CALL MC55AD(ICNTL,TITLE,KEY,CASEID,DATTYP,POSITN,ORGNIZ,
     *            M,NVEC,NAUXD,IP,IND,A,INFO)
      IF (INFO(1).LT.0) WRITE(6,'(A,2I12)') 'Value of INFO is: ',INFO

      STOP
      END
```

With the input in file mc55ads.data:

```
3 2
```

```
1 3 5
1 2 2 3
1.1 2.1 2.2 2.3
```

this program produces, in file rbfile and on unit 6:

```
spec_sheet_rhs                                                    TEST_01
rhsls RHS_01   r             3             2             4
(40I2)              (40I2)              (7E11.2E3)
 1 3 5
 1 2 2 3
  1.10E+000  2.10E+000  2.20E+000  2.30E+000
```