MC77

1 SUMMARY

Let **A** be an $m \times n$ real matrix, and $\| \ \|_p,\ p \in [1,\infty]$ a p-norm for real vectors. The package computes **the scaling diagonal matrices $\mathbf{D}_r$ and $\mathbf{D}_c$** such that the p-norms of all columns and rows (seen as vectors) of $\overline{\mathbf{A}} = \mathbf{D}_r^{-1}\mathbf{A}\mathbf{D}_c^{-1}$ are approximately equal to 1. The iterative algorithm converges linearly with an asymptotic rate of $1/2$ for the ∞-norm. In the $m \neq n$ case, the package allows the use of the ∞-norm only. Furthermore, if $m = n$ and **A** is also *irreducible*, the algorithm will converge independently of the choice of the norm. Finally, when **A** is symmetric $\mathbf{D}_r = \mathbf{D}_c$.

ATTRIBUTES — Version: 1.0.1. **Types:** Real (single, double), Complex (single, double). **Calls:** None. **Date:** July 2004. **Origin:** Daniel Ruiz E.N.S.E.E.I.H.T., Toulouse (France). **Language:** Fortran 77.

2 HOW TO USE THE PACKAGE

2.1 Argument lists and calling sequences

There are four procedures for user calls:

(a) `MC77I/ID/IC/IZ` sets default values for the components of the arrays that hold controlling parameters. It should normally be called once prior to any other call.

(b) `MC77A/AD/AC/AZ` computes the scaling matrices when **A** is sparse and stored by columns.

(c) `MC77B/BD/BC/BZ` computes the scaling matrices when **A** is sparse and stored using the coordinate format. (This is more expensive than `MC77A/AD/AC/AZ` if checking of the input data is requested, see `ICNTL` in Section 2.1.3.)

(d) `MC77C/CD/CC/CZ` computes the scaling matrices when **A** is dense.

Hereafter, we will abbreviate `MC77I/ID/IC/IZ`, `MC77A/AD/AC/AZ`, `MC77B/BD/BC/BZ`, and `MC77C/CD/CC/CZ` respectively to `MC77I_`, `MC77A_`, `MC77B_`, and `MC77C_`.

2.1.1 To set default values of the controlling parameters

The single precision version

```
CALL MC77I(ICNTL, CNTL)
```

The double precision version

```
CALL MC77ID(ICNTL, CNTL)
```

The complex version

```
CALL MC77IC(ICNTL, CNTL)
```

The double precision complex version

```
CALL MC77IZ(ICNTL, CNTL)
```

`ICNTL` is an `INTEGER` array of length `10`. On return, it contains default values. Its elements control the actions of `MC77A_`, `MC77B_`, and `MC77C_`. The values from `ICNTL(8)` to `ICNTL(10)` are not currently used but are set to zero in this routine. For further information, see Section 2.2.

CNTL is a REAL (DOUBLE PRECISION in the D and Z versions) array of length 10. On return, it contains default values. Its elements control the action of MC77A_, MC77B_, and MC77C_. The values from CNTL(3) to CNTL(10) are not currently used but are set to zero in this routine. For further information, see Section 2.2.

2.1.2 Matrix stored by columns

The single precision version

```
CALL MC77A(JOB, M, N, NNZ, JCST, IRN, A, IW, LIW, DW, LDW, ICNTL, CNTL, INFO, RINFO)
```

The double precision version

```
CALL MC77AD(JOB, M, N, NNZ, JCST, IRN, A, IW, LIW, DW, LDW, ICNTL, CNTL, INFO, RINFO)
```

The complex version

```
CALL MC77AC(JOB, M, N, NNZ, JCST, IRN, A, IW, LIW, DW, LDW, ICNTL, CNTL, INFO, RINFO)
```

The double precision complex version

```
CALL MC77AZ(JOB, M, N, NNZ, JCST, IRN, A, IW, LIW, DW, LDW, ICNTL, CNTL, INFO, RINFO)
```

JOB is an INTEGER variable that must be set by the user to control the action. It is not altered by the subroutine. Possible values for JOB are:

- 0 Equilibrate the infinity norm of rows and columns in matrix **A**.
- p Equilibrate the p-th norm (p $\geq$ 1) of rows and columns in **A**.
- -1 Equilibrate the p-th norm of rows and columns in **A**, with $p \geq 1$ real. The value of p is given in CNTL(2).

When M $\neq$ N, JOB must be 0. **Restriction:** JOB $\geq$ -1.

M is an INTEGER variable that must be set by the user to the number of rows in matrix **A**. It is not altered by the subroutine. **Restriction:** M $\geq$ 1.

N is an INTEGER variable that must be set by the user to the number of columns in **A**. It is not altered by the subroutine. **Restrictions:** N $\geq$ 1. If ICNTL(6) $\neq$ 0 (symmetric matrix), M must be equal to N.

NNZ is an INTEGER variable that must be set by the user to the number of entries in **A** or in the lower triangular part of **A** if **A** is symmetric and ICNTL(6) $\neq 0$. NNZ is not altered by the subroutine. **Restriction:** NNZ $\geq$ 0.

JCST is an INTEGER array of length N+1. JCST(J), J = 1,...,N, that must be set by the user to the position in array IRN of the first row index of an entry in column J. JCST(N+1) must be set to NNZ+1. The array JCST is not altered by the subroutine.

IRN is an INTEGER array of length NNZ. IRN(K), K = 1,...,NNZ, must be set by the user to hold the row indices of the entries in the matrix or its lower triangular part of if it is symmetric and ICNTL(6) $\neq$ 0. The entries belonging to column J must be stored contiguously in the positions from JCST(J) to JCST(J+1)-1. The ordering of the row indices within each column is unimportant. The array IRN is not altered by the subroutine.

A is a REAL (DOUBLE PRECISION in the D version, COMPLEX in the C version, or COMPLEX*16 in the Z version) array of length NNZ. The user must set A(K) to the numerical value of the entry that corresponds to IRN(K), K = 1,...,NNZ. The array A is not altered by the subroutine.

IW is an INTEGER array of length LIW that is used for workspace.

`LIW` is an `INTEGER` variable that must be set by the user to the length of array `IW`. It is not altered by the subroutine.
Restrictions: If `ICNTL(6) = 0`: `LIW` $\geq$ `M+N`. If `ICNTL(6)` $\neq$ `0`: `LIW` $\geq$ `M`.

`DW` is a `REAL` (`DOUBLE PRECISION` in the `D` and `Z` versions) array of length `LDW` that need not be set by the user. On return, `DW(I), I = 1,...,M` contains the diagonal entries in the row-scaling matrix $\mathbf{D}_r$, and if `ICNTL(6)` is equal to zero `DW(M+J), J = 1,...,N` contains the diagonal entries in the column-scaling matrix $\mathbf{D}_c$.

`LDW` is an `INTEGER` variable that must be set by the user to the length of array `DW`. It is not altered by the subroutine.
Restrictions:
If `ICNTL(5) = 0` and `ICNTL(6) = 0`: `LDW` $\geq$ `NNZ + 2*(M+N)`
If `ICNTL(5) = 1` and `ICNTL(6) = 0`: `LDW` $\geq$ `2*(M+N)`
If `ICNTL(5) = 0` and `ICNTL(6)` $\neq$ `0`: `LDW` $\geq$ `NNZ + 2*M`
If `ICNTL(5) = 1` and `ICNTL(6)` $\neq$ `0`: `LDW` $\geq$ `2*M`.

`ICNTL` is an `INTEGER` array of length `10`, that must be set by the user to control the action of `MC77A_`. It is not altered by the subroutine. See Section 2.2 for details.

`CNTL` is a `REAL` (`DOUBLE PRECISION` in the `D` and `Z` versions) array of length `10` that controls the action of `MC77A_`. It is not altered by the subroutine. See Section 2.2 for details.

`INFO` is an `INTEGER` array of length `10` that need not be set by the user and returns information. `INFO(1)` is set zero to indicate success. For the meaning of nonzero values, see Section 2.4. Other information is returned in `INFO(2)` and `INFO(3)`, see Section 2.3. The positions from `INFO(4)` to `INFO(10)` are not currently used and are set to zero by the routine.

`RINFO` is a `REAL` (`DOUBLE PRECISION` in the `D` and `Z` versions) array of length `10` that need not be set by the user and returns information (see Section 2.3). The positions from `RINFO(3)` to `RINFO(10)` are not currently used and are set to zero.

2.1.3 The coordinate format

The single precision version

```
CALL MC77B(JOB, M, N, NNZ, IRN, JCN, A, IW, LIW, DW, LDW, ICNTL, CNTL, INFO, RINFO)
```

The double precision version

```
CALL MC77BD(JOB, M, N, NNZ, IRN, JCN, A, IW, LIW, DW, LDW, ICNTL, CNTL, INFO, RINFO)
```

The complex version

```
CALL MC77BC(JOB, M, N, NNZ, IRN, JCN, A, IW, LIW, DW, LDW, ICNTL, CNTL, INFO, RINFO)
```

The double precision complex version

```
CALL MC77BZ(JOB, M, N, NNZ, IRN, JCN, A, IW, LIW, DW, LDW, ICNTL, CNTL, INFO, RINFO)
```

`JOB` is an `INTEGER` variable that must be set by the user to control the action. It is not altered by the subroutine. Possible values for `JOB` are:

- `0` Equilibrate the infinity norm of rows and columns in matrix **A**.
- `p` Equilibrate the `p`-th norm (`p` $\geq$ `1`) of rows and columns in **A**.
- `-1` Equilibrate the p-th norm of rows and columns in **A**, with $p \geq 1$ real. The value of p is given in `CNTL(2)`.

When M $\neq$ N, JOB must be 0. **Restriction:** JOB $\geq$ -1.

M is an INTEGER variable that must be set by the user to the number of rows in matrix **A**. It is not altered by the subroutine. **Restriction:** M $\geq$ 1.

N is an INTEGER variable that must be set by the user to the number of columns in **A**. It is not altered by the subroutine. **Restrictions:** N $\geq$ 1. If ICNTL(6) $\neq$ 0 (symmetric matrix), M must be equal to N.

NNZ is an INTEGER variable that must be set by the user to the number of entries in **A** or in the lower triangular part of **A** if **A** is symmetric and ICNTL(6) $\neq$ 0. NNZ is not altered by the subroutine. **Restriction:** NNZ $\geq$ 0.

IRN is an INTEGER array of length NNZ. IRN(K), K = 1,...,NNZ, must be set by the user to hold the row indices of the entries in the matrix or its lower triangular part of if it is symmetric and ICNTL(6) $\neq$ 0. The array IRN is not altered by the subroutine.

JCN is an INTEGER array of length NNZ. JCN(K) must be set by the user to hold the column index of the entry whose row index is held in IRN(K), K = 1, ... , NNZ. The array JCN is not altered by the subroutine.

A is a REAL (DOUBLE PRECISION in the D version, COMPLEX in the C version, or COMPLEX*16 in the Z version) array of length NNZ. The user must set A(K) to the numerical value of the entry that corresponds to (IRN(K),JCN(K)), K=1,...,NNZ. The array A is not altered by the subroutine.

IW is an INTEGER array of length LIW that is used for workspace.

LIW is an INTEGER variable that must be set by the user to the length of array IW. It is not altered by the subroutine. **Restrictions:** If ICNTL(6) = 0: LIW $\geq$ M+N. If ICNTL(6) $\neq$ 0: LIW $\geq$ M.

DW is a REAL (DOUBLE PRECISION in the D and Z versions) array of length LDW that need not be set by the user. On return, DW(I), I = 1,...,M contains the diagonal entries in the row-scaling matrix $\mathbf{D}_r$, and if ICNTL(6) is equal to zero DW(M+J), J = 1,...,N contains the diagonal entries in the column-scaling matrix $\mathbf{D}_c$.

LDW is an INTEGER variable that must be set by the user to the length of array DW. It is not altered by the subroutine.
Restrictions:
If ICNTL(5) = 0 and ICNTL(6) = 0: LDW $\geq$ NNZ + 2*(M+N)
If ICNTL(5) = 1 and ICNTL(6) = 0: LDW $\geq$ 2*(M+N)
If ICNTL(5) = 0 and ICNTL(6) $\neq$ 0: LDW $\geq$ NNZ + 2*M
If ICNTL(5) = 1 and ICNTL(6) $\neq$ 0: LDW $\geq$ 2*M.

ICNTL is an INTEGER array of length 10, that must be set by the user to control the action of MC77B_. It is not altered by the subroutine. See Section 2.2 for details.
Note: If ICNTL(4) = 0 (the default) a significant overhead is incurred in checking for duplicates. It is often more efficient to convert to column format and use MC77A_.

CNTL is a REAL (DOUBLE PRECISION in the D and Z versions) array of length 10 that controls the action of MC77B_. It is not altered by the subroutine. See Section 2.2 for details.

INFO is an INTEGER array of length 10 that need not be set by the user and returns information. INFO(1) is set zero to indicate success. For the meaning of nonzero values, see Section 2.4. Other information is returned in INFO(2) and INFO(3), see Section 2.3. The positions from INFO(4) to INFO(10) are not currently used and are set to zero by the routine.

RINFO is a REAL (DOUBLE PRECISION in the D and Z versions) array of length 10 that need not be set by the user and returns information (see Section 2.3). The positions from RINFO(3) to RINFO(10) are not currently used and are set to zero.

2.1.4 The dense format

The single precision version

```
CALL MC77C(JOB, M, N, A, LDA, IW, LIW, DW, LDW, ICNTL, CNTL, INFO, RINFO)
```

The double precision version

```
CALL MC77CD(JOB, M, N, A, LDA, IW, LIW, DW, LDW, ICNTL, CNTL, INFO, RINFO)
```

The complex version

```
CALL MC77CC(JOB, M, N, A, LDA, IW, LIW, DW, LDW, ICNTL, CNTL, INFO, RINFO)
```

The double precision complex version

```
CALL MC77CZ(JOB, M, N, A, LDA, IW, LIW, DW, LDW, ICNTL, CNTL, INFO, RINFO)
```

JOB is an INTEGER variable that must be set by the user to control the action. It is not altered by the subroutine. Possible values for JOB are:

0 Equilibrate the infinity norm of rows and columns in matrix **A**.

p Equilibrate the p-th norm (p $\geq$ 1) of rows and columns in **A**.

-1 Equilibrate the p-th norm of rows and columns in **A**, with $p \geq 1$ real. The value of p is given in CNTL(2).

When M $\neq$ N, JOB must be 0. **Restriction:** JOB $\geq$ -1.

M is an INTEGER variable that must be set by the user to the number of rows in matrix **A**. It is not altered by the subroutine. **Restriction:** M $\geq$ 1.

N is an INTEGER variable that must be set by the user to the number of columns in **A**. It is not altered by the subroutine. **Restrictions:** N $\geq$ 1. If ICNTL(6) $\neq$ 0 (symmetric matrix), M must be equal to N.

A is a REAL (DOUBLE PRECISION in the D version, COMPLEX in the C version, or COMPLEX*16 in the Z version) array containing the matrix or its lower triangular part of if it is symmetric and ICNTL(6) $\neq$ 0. In the former case, it is of rank two with leading dimension LDA and A(i, j) holds A_{ij}, i = 1,..., m, j =1,..., n. In the latter case, it is of rank one with size LDA and the lower triangular part is held packed by columns. A is not altered by the subroutine.

LDA is a INTEGER variable containing the leading (or only) dimension of the array A. It is not altered by the subroutine.

IW is an INTEGER array of length LIW that is used for workspace.

LIW is an INTEGER variable that must be set by the user to the length of array IW. It is not altered by the subroutine. **Restrictions:** If ICNTL(6) = 0: LIW $\geq$ M+N. If ICNTL(6) $\neq$ 0: LIW $\geq$ M.

DW is a REAL (DOUBLE PRECISION in the D and Z versions) array of length LDW that need not be set by the user. On return, DW(I), I = 1,...,M contains the diagonal entries in the row-scaling matrix $\mathbf{D}_r$, and if ICNTL(6) is equal to zero DW(M+J), J = 1,...,N contains the diagonal entries in the column-scaling matrix $\mathbf{D}_c$.

LDW is an INTEGER variable that must be set by the user to the length of array DW. It is not altered by the subroutine.
Restrictions:
If ICNTL(5) = 0 and ICNTL(6) = 0: LDW $\geq$ LDA*N + 2*(M+N).
If ICNTL(5) = 1 and ICNTL(6) = 0: LDW $\geq$ 2*(M+N)
If ICNTL(5) = 0 and ICNTL(6) $\neq$ 0: LDW $\geq$ (M*(M+1))/2 + 2*M.
If ICNTL(5) = 1 and ICNTL(6) $\neq$ 0: LDW $\geq$ 2*M.

`ICNTL` is an `INTEGER` array of length `10`, that must be set by the user to control the action of `MC77C_`. It is not altered by the subroutine. See Section 2.2 for details.

`CNTL` is a `REAL` (`DOUBLE PRECISION` in the `D` and `Z` versions) array of length `10` that controls the action of `MC77C_`. It is not altered by the subroutine. See Section 2.2 for details.

`INFO` is an `INTEGER` array of length `10` that need not be set by the user and returns information. `INFO(1)` is set zero to indicate success. For the meaning of nonzero values, see Section 2.4. Other information is returned in `INFO(2)` and `INFO(3)`, see Section 2.3. The positions from `INFO(4)` to `INFO(10)` are not currently used and are set to zero by the routine.

`RINFO` is a `REAL` (`DOUBLE PRECISION` in the `D` and `Z` versions) array of length `10` that need not be set by the user and returns information (see Section 2.3). The positions from `RINFO(3)` to `RINFO(10)` are not currently used and are set to zero.

2.2 Control parameters

`ICNTL(1)` has default value `6`. It is the unit for error messages. If it is negative, these messages will be suppressed.

`ICNTL(2)` has default value `6`. It is the unit for warning messages. If it is negative, these messages are suppressed.

`ICNTL(3)` has default value `-1`. It is the unit for monitoring printing. If it is negative, these messages are suppressed.

`ICNTL(4)` has default value `0`. If left at the default value, the incoming data is checked for out-of-range indices, duplicates and if `ICNTL(6)` $\neq$ `0` for entries in the upper triangular part of the matrix. If these are found, the routine will exit with an error. Setting `ICNTL(4)` to any other value will avoid the checks but is likely to cause problems later if out-of-range indices or duplicates are present. The user should only set `ICNTL(4)` nonzero if the data is known to be in range without duplicates.

`ICNTL(5)` has default value `0`. If left at the default value, it indicates that the absolute value of `A` must be computed internally. The user should only set `ICNTL(5)` nonzero if `A` is `REAL` (`DOUBLE PRECISION` in the `D` version) and non-negative.

`ICNTL(6)` has default value `0`. If nonzero, the input matrix `A` is symmetric and the user must only supply the lower triangular part of `A` in the appropriate format. Entries in the upper triangular part of a symmetric matrix will be considered as out-of-range producing an error. The user can use `ICNTL(4)` to avoid this check if the data is known to be correct.

`ICNTL(7)` has a default value of `10`. It specifies the maximum number of scaling iterations that may be performed. **Restriction:** `ICNTL(7)` $>$ `0`.

`CNTL(1)` has a default value of `0`. It specifies the tolerance value when stopping the iterations, that is it is the desired value such that all row and column norms in the scaled matrix lie between (`1` $\pm$ `CNTL(1)`). If `CNTL(1)` is less than or equal to zero, tolerance is not checked, and the algorithm will stop when the maximum number of iterations given in `ICNTL(7)` is reached. **Restriction:** If `ICNTL(7)`=0, `CNTL(1)`>0.

`CNTL(2)` has a default value of `1`. It is used in conjunction with parameter `JOB` set to `-1`, to specify a `REAL` (`DOUBLE PRECISION` in the `D` and `Z` versions) value for the power p of the norm under consideration.
Restriction: `CNTL(2)` $\geq$ `1`.

2.3 Information returned to the user

INFO(1) has the value zero if the call was successful, is positive in the case of warning, and has a negative value in the event of an error (see Section 2.4)

INFO(2) holds further information on the error (see Section 2.4).

INFO(3) returns the number of iterations performed.

INFO(4) ... INFO(10) are not used and they are set to zero.

RINFO(1) will contain on exit the maximum distance of all row norms to 1.

RINFO(2) will contain on exit the maximum distance of all column norms to 1. If ICNTL(6) $\neq 0$ (indicating that the input matrix is symmetric), RINFO(2) is set to the value of RINFO(1).

RINFO(3) ... RINFO(10) are not used and they are set to zero.

2.4 Warning and error messages

If INFO(1) has zero value on output, the code was successful. Otherwise, a negative value indicates one of the following errors (INFO(2) holds further information on the error):

-1 M $<$ 1. Value of M held in INFO(2).

-2 N $<$ 1. Value of N held in INFO(2).

-3 ICNTL(6) $\neq$ 0 (input matrix is symmetric) and M $\neq$ N. Value of (M-N) held in INFO(2).

-4 In MC77A_ and MC77B_, NNZ < 1. Value of NNZ held in INFO(2).
In MC77C_ for the non-symmetric case, LDA < M. Value of LDA-M held in INFO(2).
In MC77C_ for the symmetric case, LDA < (M*(M+1))/2. Value of LDA - (M*(M+1))/2 held in INFO(2).

-5 The defined length LIW violates the restriction on LIW. Value of LIW required given by INFO(2).

-6 The defined length LDW violates the restriction on LDW. Value of LDW required given by INFO(2).

-7 An entry is found whose row or column indices are out of range. INFO(2) contains the index of a row or a column in which such an entry is found.

-8 A repeated entry is found. For MC77A_, INFO(2) contains the index of a column in which such entries are found. For MC77B_, INFO(2) contains an index pointing to a value in IRN and JCN in which such an entry is found.

-9 An entry corresponding to an element in the upper triangular part of the matrix is found in the input data (ICNTL(6) $\neq$ 0).

-10 ICNTL(7) is out of range and INFO(2) contains its value.

-11 CNTL(2) is out of range.

-12 JOB < -1. Value of JOB held in INFO(2).

-13 JOB $\neq 0$ and M $\neq$ N. Value of JOB held in INFO(2).

A positive value of INFO(1) gives the following warning message.

+1 The maximum number of iterations given by ICNTL(7) has been reached. This warning is issued only when CNTL(1) is greater than zero, meaning that convergence is actually checked. If CNTL(1) is less than or equal to zero, the algorithm stops when the maximum number of iterations given in ICNTL(7) is reached, with successful exit and INFO(1) set to 0.

3 GENERAL INFORMATION

Workspace: IW

Other routines called directly: MC77J_, MC77K_, MC77L_, MC77M_, MC77N_, MC77O_, MC77P_, MC77Q_, MC77R_, MC77S_, MC77T_, MC77U_.

Input/output: Output is provided under the control of ICNTL(1), ICNTL(2), and ICNTL(3) for error and warning messages respectively.

Restrictions: For MC77A_ and MC77B_:

JOB $\geq$ -1; M $\geq$ 1; N $\geq$ 1; NNZ $\geq$ 0; ICNTL(7) $>$ 0;

If JOB = -1: CNTL(2) $\geq$ 1.

If ICNTL(6) = 0 (general case):

LIW $\geq$ M+N;

If (ICNTL(5) = 0 or JOB $\notin \{0,1\}$): LDW $\geq$ NNZ + 2*(M+N);

If (ICNTL(5) $\neq$ 0 and JOB $\in \{0,1\}$): LDW $\geq$ 2*(M+N);

If ICNTL(6) $\neq$ 0 (symmetric case):

M = N; LIW $\geq$ M;

If (ICNTL(5) = 0 or JOB $\notin \{0,1\}$): LDW $\geq$ NNZ + 2*M;

If (ICNTL(5) $\neq$ 0 and JOB $\in \{0,1\}$): LDW $\geq$ 2*M;

For MC77C_: same restrictions as above with NNZ replaced by LDA*N when ICNTL(6) = 0, and NNZ replaced by (M*(M+1))/2 when ICNTL(6) $\neq$ 0.

Portability: ISO Fortran 77.

4 METHOD

Consider a general $m \times n$ real matrix $\mathbf{A}$, and denote by $\mathbf{r}_i = \mathbf{a}_{i.}^T \in I\!R^{n\times 1}$, $i = 1,\ldots,m$, the row-vectors from $\mathbf{A}$ and by $\mathbf{c}_j = \mathbf{a}_{.j} \in I\!R^{n\times 1}$, $j = 1,\ldots,n$, the column-vectors from $\mathbf{A}$. Denote by $\mathbf{D}_R$ and $\mathbf{D}_C$ the $m\times m$ and $n\times n$ diagonal matrices given by:

$$\mathbf{D}_R = \mathrm{diag}(\sqrt{\|\mathbf{r}_i\|_\infty})_{i=1,\ldots,m} \quad \text{and} \quad \mathbf{D}_C = \mathrm{diag}(\sqrt{\|\mathbf{c}_j\|_\infty})_{j=1,\ldots,n} \tag{4.1}$$

where $\|.\|_\infty$ stands for the infinity-norm of a real vector (that is the maximum entry in absolute value, and also sometimes called the max norm). If a row (or column) in $\mathbf{A}$ has all entries equal to zero, we replace the diagonal entry in $\mathbf{D}_R$ (or $\mathbf{D}_C$ respectively) by 1. In the following, we will assume that this does not happen, considering that such cases are fictitious in the sense that zero rows or columns should be taken away and the system reduced.

We then scale matrix $\mathbf{A}$ on both sides, forming the scaled matrix $\widehat{\mathbf{A}}$ in the following way

$$\widehat{\mathbf{A}} = \mathbf{D}_R^{-1}\mathbf{A}\mathbf{D}_C^{-1} \ . \tag{4.2}$$

Now, the idea of the proposed algorithm is to iterate on that process, viz.

Algorithm 1 (Simultaneous row and column iterative scaling)

$$\widehat{\mathbf{A}}^{(0)} = \mathbf{A}, \quad \mathbf{D}_1^{(0)} = \mathbf{I}_m, \quad \text{and } \mathbf{D}_2^{(0)} = \mathbf{I}_n$$

for $k = 0,1,2,\ldots,$ until convergence do :

$$\mathbf{D}_R = \mathrm{diag}(\sqrt{\|\mathbf{r}_i^{(k)}\|_\infty})_{i=1,\ldots,m}, \quad \text{and } \mathbf{D}_C = \mathrm{diag}(\sqrt{\|\mathbf{c}_j^{(k)}\|_\infty})_{j=1,\ldots,n}$$

$$\widehat{\mathbf{A}}^{(k+1)} = \mathbf{D}_R^{-1}\widehat{\mathbf{A}}^{(k)}\mathbf{D}_C^{-1}$$

$$\mathbf{D}_1^{(k+1)} = \mathbf{D}_1^{(k)}\mathbf{D}_R, \quad \text{and } \mathbf{D}_2^{(k+1)} = \mathbf{D}_2^{(k)}\mathbf{D}_C$$

Algorithm 1 computes the scaling diagonal matrices $\mathbf{D}_1^{(k)}$ and $\mathbf{D}_2^{(k)}$ such that the max-norm of all columns and rows (seen as vectors) of $\widehat{\mathbf{A}}^{(k)} = \mathbf{D}_1^{(k)^{-1}} \mathbf{A} \mathbf{D}_2^{(k)^{-1}}$ tend to 1 as $k \to +\infty$. Convergence is obtained when

$$\max_{1 \le i \le m} \left\{ |(1 - \|\mathbf{r}_i^{(k)}\|_\infty)| \right\} \le \varepsilon \text{ and } \max_{1 \le j \le n} \left\{ |(1 - \|\mathbf{c}_j^{(k)}\|_\infty)| \right\} \le \varepsilon \tag{4.3}$$

for a given value of $\varepsilon > 0$.

The properties of this algorithm are discussed in [1], and can be summarized as follows:

- this iterative procedure asymptotically scales the infinity norm of both rows and columns in a matrix to 1, with an asymptotic linear rate of convergence of $\frac{1}{2}$,
- it preserves symmetry when this is the case,
- in the case of square matrices ($m = n$) with support (e.g. for which a transversal exists), it can also be generalized to any other vector norm for both rows and columns, replacing for instance $\|\cdot\|_\infty$ above by $\|\cdot\|_p$, $p = 1, 2, \ldots$,
- in this latter case, convergence of the algorithm is only ensured when the matrix $\mathbf{A}$ is irreducible or has total support.

References

[1] D. R. RUIZ A scaling algorithm to equilibrate both row and column norms in matrices. Technical Report RAL-TR-2001-034, RAL, Oxfordshire, England, and Report RT/APO/01/4, ENSEEIHT-IRIT, Toulouse, France, (2001).

5 EXAMPLE OF USE

The following programs display the use of `MC77AD`, `MC77BD`, and `MC77CD` on the matrix $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} 100 & 10 & \\ 4 & -1000 & 5 \\ & 23 & 0.01 \end{bmatrix}.$$

The scaled matrix in ∞-norm is:

$$\begin{aligned} \mathbf{B} &= \begin{bmatrix} 10 & & \\ & 31.623 & \\ & & 0.729 \end{bmatrix}^{-1} \begin{bmatrix} 100 & 10 & \\ 4 & -1000 & 5 \\ & 23 & 0.01 \end{bmatrix} \begin{bmatrix} 10 & & \\ & 31.623 & \\ & & 0.159 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 1.00 & 0.0316 & 0 \\ 0.0126 & -1.00 & 0.9944 \\ 0 & 0.9977 & 0.0863 \end{bmatrix}. \end{aligned}$$

The scaled matrix in 1-norm is:

$$\begin{aligned} \mathbf{C} &= \begin{bmatrix} 10.479 & & \\ & 56.578 & \\ & & 0.452 \end{bmatrix}^{-1} \begin{bmatrix} 100 & 10 & \\ 4 & -1000 & 5 \\ & 23 & 0.01 \end{bmatrix} \begin{bmatrix} 9.650 & & \\ & 66.675 & \\ & & 0.115 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 0.9889 & 0.0143 & 0 \\ 0.0073 & -0.2651 & 0.7685 \\ 0 & 0.7632 & 0.1924 \end{bmatrix}. \end{aligned}$$

MC77AD example.

```
      INTEGER MAXN,MAXNZ,LIW,LDW
      PARAMETER (MAXN=10,MAXNZ=100)
      INTEGER IRN(MAXNZ),JCOLST(MAXN+1),IW(2*MAXN),ICNTL(10),INFO(10)
      DOUBLE PRECISION CNTL(10),DW(MAXNZ+4*MAXN),RINFO(10)
      DOUBLE PRECISION A(MAXNZ)
      INTEGER JOB, N, NNZ, I
C     .. External Subroutines ..
      EXTERNAL MC77AD,MC77ID
C     .. read in input matrix
      READ(5,*) N,NNZ
      IF (N.GT.MAXN .OR. NNZ.GT.MAXNZ) THEN
         WRITE(6,'(A,2I5)')'N and NNZ too large = ',N,NNZ
         STOP
      ENDIF
      READ(5,*) (JCOLST(I),I=1,N+1)
      READ(5,*) (IRN(I),A(I),I=1,NNZ)
C     set default controls
      CALL MC77ID(ICNTL,CNTL)
C     set LIW and LDW
      LIW = 2*MAXN
      LDW = MAXNZ+4*MAXN
      write(6,'(a)') '++++++++++++++++++++++++++++++++++++++++++'
      write(6,'(a)') '     M = N unsymmetric case inf norm     '
      write(6,'(a)') '------------------------------------------'
C
      JOB = 0
      CALL MC77AD(JOB,N,N,NNZ,JCOLST,IRN,A,IW,LIW,DW,LDW,
     &                  ICNTL,CNTL,INFO,RINFO)
      WRITE(6,9042) ICNTL(7), RINFO(1), RINFO(2)
      WRITE(6,9031) (DW(I),I=1,N)
      WRITE(6,9032) (DW(N+J),J=1,N)
      write(6,'(a)') '++++++++++++++++++++++++++++++++++++++++++'
      write(6,'(a)') '     M = N unsymmetric case 1 norm     '
      write(6,'(a)') '------------------------------------------'
C
      JOB = 1
      CALL MC77AD(JOB,N,N,NNZ,JCOLST,IRN,A,IW,LIW,DW,LDW,
     &                  ICNTL,CNTL,INFO,RINFO)
      WRITE(6,9042) ICNTL(7), RINFO(1), RINFO(2)
      WRITE(6,9031) (DW(I),I=1,N)
      WRITE(6,9032) (DW(N+J),J=1,N)
 9031 FORMAT (' DW(1:M)     = ',5(F11.3)/(15X,5(F11.3)))
 9032 FORMAT (' DW(M+1:M+N) = ',5(F11.3)/(15X,5(F11.3)))
 9041 FORMAT (' INFO(1:3)   = ',3(I8,5X))
 9042 FORMAT (' MAXIT = ',I8,'  RINFO(1) = ',1PD14.4,
     &'  RINFO(2) = ',1PD14.4)
      END
```

MC77BD example.

```
      INTEGER MAXN,MAXNZ,LIW,LDW
      PARAMETER (MAXN=10,MAXNZ=100)
      INTEGER IRN(MAXNZ),JCN(MAXNZ),IW(2*MAXN),ICNTL(10),INFO(10)
      DOUBLE PRECISION CNTL(10),DW(MAXNZ+4*MAXN),RINFO(10)
      DOUBLE PRECISION A(MAXNZ)
      INTEGER JOB, N, NNZ, I
C     .. External Subroutines ..
      EXTERNAL MC77AD,MC77ID
C     .. read in input matrix
      READ(5,*) N,NNZ
      IF (N.GT.MAXN .OR. NNZ.GT.MAXNZ) THEN
         WRITE(6,'(A,2I5)')'N and NNZ too large = ',N,NNZ
         STOP
      ENDIF
      READ(5,*) (JCN(I),IRN(I),A(I),I=1,NNZ)
C     set default controls
      CALL MC77ID(ICNTL,CNTL)
C     set LIW and LDW
      LIW = 2*MAXN
      LDW = MAXNZ+4*MAXN
      write(6,'(a)') '+++++++++++++++++++++++++++++++++++++++++'
      write(6,'(a)') '     M = N unsymmetric case inf norm     '
      write(6,'(a)') '-----------------------------------------'
C
      JOB = 0
      CALL MC77BD(JOB,N,N,NNZ,IRN,JCN,A,IW,LIW,DW,LDW,
     &                  ICNTL,CNTL,INFO,RINFO)
      WRITE(6,9042) ICNTL(7), RINFO(1), RINFO(2)
      WRITE(6,9031) (DW(I),I=1,N)
      WRITE(6,9032) (DW(N+J),J=1,N)
      write(6,'(a)') '+++++++++++++++++++++++++++++++++++++++++'
      write(6,'(a)') '     M = N unsymmetric case 1 norm     '
      write(6,'(a)') '-----------------------------------------'
C
      JOB = 1
      CALL MC77BD(JOB,N,N,NNZ,IRN,JCN,A,IW,LIW,DW,LDW,
     &                  ICNTL,CNTL,INFO,RINFO)
      WRITE(6,9042) ICNTL(7), RINFO(1), RINFO(2)
      WRITE(6,9031) (DW(I),I=1,N)
      WRITE(6,9032) (DW(N+J),J=1,N)
 9031 FORMAT (' DW(1:M)     = ',5(F11.3)/(15X,5(F11.3)))
 9032 FORMAT (' DW(M+1:M+N) = ',5(F11.3)/(15X,5(F11.3)))
 9041 FORMAT (' INFO(1:3)   = ',3(I8,5X))
 9042 FORMAT (' MAXIT = ',I8,'  RINFO(1) = ',1PD14.4,
     &'  RINFO(2) = ',1PD14.4)
      END
```

`MC77CD` example

```
      INTEGER MAXN,LIW,LDW
      PARAMETER (MAXN=10)
      INTEGER IW(2*MAXN),ICNTL(10),INFO(10)
      DOUBLE PRECISION CNTL(10),DW(MAXN*MAXN+4*MAXN),RINFO(10)
      DOUBLE PRECISION A(MAXN,MAXN)
      INTEGER JOB, N, I, J
C     .. External Subroutines ..
      EXTERNAL MC77AD,MC77ID
C     ..read in input matrix
      READ(5,*) N
      IF (N.GT.MAXN) THEN
         WRITE(6,'(A,2I5)')'N > MAXN  ',N,MAXN
         STOP
      ENDIF
      READ(5,*) ((A(I,J),I=1,N),J=1,N)
C     set default controls
      CALL MC77ID(ICNTL,CNTL)
C     set LIW and LDW
      LIW = 2*MAXN
      LDW = MAXN*MAXN+4*MAXN
      write(6,'(a)') '++++++++++++++++++++++++++++++++++++++++'
      write(6,'(a)') '     M = N unsymmetric case inf norm     '
      write(6,'(a)') '----------------------------------------'
C
      JOB = 0
      CALL MC77CD(JOB,N,N,A,MAXN,IW,LIW,DW,LDW,
     &                  ICNTL,CNTL,INFO,RINFO)
      WRITE(6,9042) ICNTL(7), RINFO(1), RINFO(2)
      WRITE(6,9031) (DW(I),I=1,N)
      WRITE(6,9032) (DW(N+J),J=1,N)
      write(6,'(a)') '++++++++++++++++++++++++++++++++++++++++'
      write(6,'(a)') '     M = N unsymmetric case 1 norm     '
      write(6,'(a)') '----------------------------------------'
C
      JOB = 1
      CALL MC77CD(JOB,N,N,A,MAXN,IW,LIW,DW,LDW,
     &                  ICNTL,CNTL,INFO,RINFO)
      WRITE(6,9042) ICNTL(7), RINFO(1), RINFO(2)
      WRITE(6,9031) (DW(I),I=1,N)
      WRITE(6,9032) (DW(N+J),J=1,N)
 9031 FORMAT (' DW(1:M)     = ',5(F11.3)/(15X,5(F11.3)))
 9032 FORMAT (' DW(M+1:M+N) = ',5(F11.3)/(15X,5(F11.3)))
 9041 FORMAT (' INFO(1:3)   = ',3(I8,5X))
 9042 FORMAT (' MAXIT = ',I8,'  RINFO(1) = ',1PD14.4,
     &'  RINFO(2) = ',1PD14.4)
      END
```

The three programs produce the following output:

```
+++++++++++++++++++++++++++++++++++++++++
     M = N unsymmetric case inf norm
-----------------------------------------
 MAXIT =       10  RINFO(1) =     3.6771E-03  RINFO(2) =     5.1608E-03
 DW(1:M)     =      10.000     31.623      0.729
 DW(M+1:M+N) =      10.000     31.623      0.159
+++++++++++++++++++++++++++++++++++++++++
     M = N unsymmetric case 1 norm
-----------------------------------------
 MAXIT =       10  RINFO(1) =     5.8022E-02  RINFO(2) =     5.4572E-02
 DW(1:M)     =      10.479     56.578      0.452
 DW(M+1:M+N) =       9.650     66.675      0.115
```