1 SUMMARY

This subroutine **estimates the 1-norm** ($\max_j \sum_{i=1}^{n} |a_{ij}|$) of an $n \times n$ matrix **A** given the ability to multiply a vector by both the matrix and its transpose. Because the explicit form of **A** is not required, the subroutine can be used for estimating the norm of matrix functions such as the inverse. Additionally this subroutine is potentially useful for estimating condition numbers of a matrix when the the matrix is sparse or not available explicitly.

ATTRIBUTES — Version: 1.0.0. (12 July 2004) **Types:** Real (single, double). **Calls:** `I_AMAX`. **Original date:** June 2001. **Remark:** `MC71` is a threadsafe version of `MC41`. **Origin:** M. Arioli, I.S. Duff, Harwell.

2 HOW TO USE THE PACKAGE

'Reverse communication' is used to provide the subroutine with the values of $\mathbf{A}\,\mathbf{x}$ or $\mathbf{A}^T\mathbf{x}$ for a given vector **x**. When the subroutine requires the values of these products, it sets **x** in the array `X` and returns to the user's program with a flag, called `KASE`, set to `1` if $\mathbf{A}\,\mathbf{x}$ is required or set to `2` if $\mathbf{A}^T\mathbf{x}$ is required. The user's program must set `X` to the value of the matrix-vector product required by the subroutine and it must not alter the value of any other arguments of the subroutine. Initially the user must set the value of `KASE` to `0;` the subroutine sets `KASE` to `0` at the end of the computation. An example of use is shown in Section 5.

2.1 Argument list

The single precision version:

```
CALL MC71A(N,KASE,X,EST,W,IW,KEEP)
```

The double precision version:

```
CALL MC71AD(N,KASE,X,EST,W,IW,KEEP)
```

`N` is an `INTEGER` variable that must be set by the user to the order of the matrix. This argument is not altered by the subroutine. **Restriction:** `N` >0.

`KASE` is an `INTEGER` variable that must be set by the user to `0` on the initial call. The subroutine sets `KASE` to `1` or `2` in the intermediate returns and to `0` for the final return. `KASE` $=-$`1` indicates an error condition (see §2.5).
When `KASE`=`1` the user must supply the product of `X` by the matrix **A**.
When `KASE`=`2` the user must supply the product of `X` by the transpose of the matrix **A**.

`X` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length `N` that need not be set by the user initially. In the intermediate returns, `X` must be overwritten by the product of the output value of `X` by the matrix (`KASE`=`1`) or by the transpose of the matrix (`KASE`=`2`).

`EST` is a `REAL` (`DOUBLE PRECISION` in the D version) variable. It is set by the subroutine to contain a lower bound estimate for the 1-norm of the matrix.

`W` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length `N` that is used by the routine as private workspace and must not be altered by the user.

`IW` is an `INTEGER` array of length `N` that is used by the routine as private workspace and must not be altered by the user.

`KEEP` is an `INTEGER` array of length 5 which is used by the routine as private workspace and must not be altered by the user.

2.2 Finding the ∞–norm

Since $\|\mathbf{B}\|_\infty = \|\mathbf{B}^T\|_1$, the subroutine can be used to estimate the ∞–norm of **B** by working with $\mathbf{A} = \mathbf{B}^T$.

2.3 Finding condition numbers

The subroutine can be used to estimate the classical condition number $\kappa(\mathbf{B}) = \|\mathbf{B}\|\,\|\mathbf{B}^{-1}\|$ (for either 1 or ∞–norm) of a matrix **B** assuming it is possible to solve both the systems $\mathbf{B}\,\mathbf{y} = \mathbf{x}$ and $\mathbf{B}^T\mathbf{y} = \mathbf{x}$ for different right-hand sides **x**. The products of $\mathbf{A} = \mathbf{B}^{-1}$ or $\mathbf{A}^T = \mathbf{B}^{-T}$ times a given vector **x** can be computed by solving the corresponding systems.

2.4 Nonsquare matrices

It is also possible to use `MC71` for computing the 1-norm or the ∞–norm of an $m \times n$ matrix **B**. This can be easily calculated by computing the norm of the square matrix obtained by bordering the matrix **B** with $(m-n)$ zero columns if $m > n$ or with $(n-m)$ zero rows if $m < n$.

2.5 Errors and diagnostic messages

The value `−1` for `KASE` indicates that `N` ≤ `0`.

3 GENERAL INFORMATION

Workspace: Provided by user, see arguments `IW` and `W`.

Use of common: None.

Other routines called directly: `MC71A/AD` calls `I_AMAX`. The user does not need to call these subroutines directly.

Input/output: None.

Restrictions: `N` > `0`.

4 METHOD

The method used is based on that developed by Hager (1984) and incorporates the modifications suggested by Higham (1987). Because $\|\mathbf{A}\|_1$ is the global maximum of the function $f(\mathbf{x}) = \|\mathbf{A}\,\mathbf{x}\|_1$ over the set $S = \{\, \mathbf{x} : \|\mathbf{x}\|_1 \le 1 \,\}$, Hager (1984) introduces an iterative method which, at each step, moves from a point in S to another one where the value of $f(\mathbf{x})$ increases. In a finite number of iterations ($\le n$) the algorithm guarantees the convergence to a local maximum of $f(\mathbf{x})$. In practice, 2 or 3 iterations are sufficient to obtain a local maximum. In `MC71A/AD`, the number of iterations is limited to 5. Each iteration requires one matrix-vector product of the form **Ax** and one of the form $\mathbf{A}^T\mathbf{x}$ so the maximum number of calls to `MC71A/AD` that the user need make is 12.

`MC71A/AD` does not require the user to supply the matrix **A** but does require the user to compute products **Ax** and $\mathbf{A}^T\mathbf{x}$. Hence, it is possible to use `MC71A/AD` to compute the norm of a matrix which is a function of matrices without computing the resulting matrix explicitly. For example, the norm of the inverse of a matrix **B** may be estimated if the systems $\mathbf{B}\,\mathbf{y} = \mathbf{x}$ and $\mathbf{B}^T\mathbf{y} = \mathbf{x}$ can be solved. Arioli, Demmel, and Duff (1988) use this method for estimating different kinds of condition numbers of a sparse matrix.

References

Arioli, M., Demmel J. W., and Duff, I. S. (1988). Solving sparse systems with sparse backward error. Report CSS 214, CSS Division, Harwell Laboratory, England.

Hager, W. W. (1984). Condition Estimates. *SIAM J. Sci. Stat. Comput.* **5**, 311-316.

Higham, N. J. (1987). Fortran codes for estimating the 1-norm of a real or complex matrix, with applications to condition estimation. Numerical Analysis Report 135, University of Manchester M13 9PL, England.

5 EXAMPLE OF USE

The following example shows the use of the subroutine MC71AD for computing the ∞–norm of the matrix

$$\mathbf{A} = \begin{pmatrix} 1 & -2 & 1 & -2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

When the program of Figure 1 is run on the input

```
4
1.0 -2.0  1.0 -2.0
0.0  1.0  0.0  0.0
0.0  0.0  1.0  0.0
1.0  0.0  0.0  1.0
```

it produces the output

```
 ESTIMATE OF THE NORM OF THE MATRIX  0.6000D+01
```

```
      DOUBLE PRECISION A(10,10), W(10), X(10), V(10), EST
      INTEGER IW(10),KEEP(5)
      INTEGER N,I,J,LA,KASE,ITER,ITYPE
      READ(5,10) N
      READ(5,20) ((A(I,J),J=1,N),I=1,N)
   10 FORMAT(I3)
   20 FORMAT(4F5.1)
C
      LA = 10
      KASE = 0
      DO 100 ITER = 1,12
        CALL MC71AD(N,KASE,X,EST,V,IW,KEEP)
        IF (KASE .LT. 0) GO TO 600
        IF (KASE .EQ. 0) GO TO 200
        ITYPE = 2 * KASE - 3
        CALL MAPX(LA, N, A, X, W, ITYPE)
  100 CONTINUE
  200 WRITE(6,9998) EST
      GO TO 1000
  600 WRITE(6,9997)
 1000 STOP
 9997 FORMAT(' MC71AD- N .LE. 0 ')
 9998 FORMAT(' ESTIMATE OF THE NORM OF THE MATRIX ',E10.4)
      END
      SUBROUTINE MAPX(LA, N, A, X, W, ITYPE)
      INTEGER LA,N,ITYPE,I,J
      DOUBLE PRECISION A(LA,LA), X(N), W(N), ZERO
      DATA ZERO /0.0D0/
      IF (ITYPE .EQ. 1) THEN
        DO 100 I = 1,N
          W(I) = ZERO
          DO 200 J = 1,N
            W(I) = W(I) + A(I,J) * X(J)
  200     CONTINUE
  100   CONTINUE
      ELSE
        DO 300 I = 1,N
          W(I) = ZERO
          DO 400 J = 1,N
            W(I) = W(I) + A(J,I) * X(J)
  400     CONTINUE
  300   CONTINUE
      ENDIF
      DO 500 I = 1,N
        X(I) = W(I)
  500 CONTINUE
      RETURN
      END
```

Figure 1. Code to estimate the infinity norm of a matrix