1 SUMMARY

This subroutine **estimates the 1-norm or the ∞–norm** of an $n \times n$ matrix **A** given the ability to multiply a vector by both the matrix and its transpose. This subroutine is useful for estimating different kinds of condition numbers of a matrix.

ATTRIBUTES — **Versions:** MC41A, MC41AD **Calls:** MX06A **Language:** Fortran 77. **Date:** February 1988. **Size:** 2612 bytes; 195 cards. **Origin:** M. Arioli, I.S. Duff, Harwell. **Conditions on external use:** (i), (ii), (iii) and (iv).

2 HOW TO USE THE ROUTINE

'Reverse Communication' is used to provide the subroutine with the values of $\mathbf{A}x$ or $\mathbf{A}^T x$, which means that computer instructions for their computations must be present in the part of the user's program that calls the subroutine. When the subroutine requires the values of these products, it sets the appropriate values in the vector X and returns to the user's program with a flag, called KASE, set to 1 if $\mathbf{A}x$ is required or set to 2 if $\mathbf{A}^T x$ is required. Initially, the user must set the value of KASE to 0. In the intermediate return the user's program must set X to the value of the matrix-vector product required by the subroutine and it may not alter the value of any other arguments of the subroutine. The subroutine sets KASE to 0 at the end of the computation. An example is shown in Section 5.

2.1 Argument list

The single precision version:

```
CALL MC41A(N,KASE,X,EST,W,IW)
```

The double precision version:

```
CALL MC41AD(N,KASE,X,EST,W,IW)
```

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

KASE is an INTEGER variable that must be set by the user to 0 on the initial call. The routine sets KASE to 1 or 2 in the intermediate returns and to 0 for the final return. Negative values of KASE indicate an error condition (see § 2.5).

KASE = 1 the user must supply the product of X by the matrix **A**.

KASE = 2 the user must supply the product of X by the transpose of the matrix **A**.

X is a REAL array (DOUBLE PRECISION in D version) of length N. The routine sets the value of X on the initial call. In the intermediate returns X must be overwritten by the product of the previous value of X by the matrix (KASE = 1) or by the transpose of the matrix (KASE = 2).

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

W is a REAL array (DOUBLE PRECISION in D version) of length N that is used as workspace.

IW is an INTEGER array of length N that is used as workspace.

2.2 Finding the ∞–norm

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

2.3 Nonsquare matrices

It is also possible to use MC41 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 $m < n$.

2.4 Common

No common areas are used by the routine.

2.5 Errors and diagnostic messages

A negative value for KASE indicates that $N \leq 0$.

3 GENERAL INFORMATION

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

Use of common: None.

Other routines called directly: MC41A/AD calls MX06A. The user does not need to call this subroutine directly.

Input/output: None.

Portability: Fortran 77.

Restrictions: $N > 0$.

4 METHOD

The method used is based on that developed by Hager (1984). The version implemented is based on 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 $S = \{\,\mathbf{x} : \|\mathbf{x}\|_1 \leq 1\,\}$, 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 ($\leq n$) the algorithm guarantees the convergence to a local maximum of $f(\mathbf{x})$. In practice usually 2 or 3 iterations are needed to obtain a local maximum. The limit on the number of iterations for the method is set to 10, although our empirical evidence suggests that we never need more than 5 iterations.

The dependence of the method on the matrix A is isolated in the ability of computing the products of a vector by the matrix or by its transpose. Hence, it is possible to compute the norm of a matrix which is a function of matrices without computing the resulting matrix explicitly. For example, it is possible to estimate the norm of the inverse of a matrix A given the ability of solving the systems $\mathbf{A}\,\mathbf{x} = \mathbf{b}$ and $\mathbf{A}^T\mathbf{x} = \mathbf{b}$. 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 MC41AD for computing the ∞–norm of the matrix

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

Running the program

```
      DOUBLE PRECISION A(10,10), W(10), X(10), V(10), EST
      INTEGER IW(10)
      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
      NDUMMY = N + 1
      KASE = 0
      DO 100 ITER = 1, NDUMMY
        CALL MC41AD(N,KASE,X,EST,V,IW)
        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(' MC41AD- N .LE. 0 ')
 9998 FORMAT(' ESTIMATE OF THE NORM OF THE MATRIX ',D10.4)
      END
      SUBROUTINE MAPX(LA, N, A, X, W, ITYPE)
      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
```

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
```

we obtain the output:

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