Warning: Subroutine MC19 has been superseded by subroutine MC29 which uses improved algorithms; the use of the latter routine is recommended. The superseded routine may be removed from later releases of the library.

1 SUMMARY

This subroutine **calculates scaling factors for a sparse matrix A = $\{a_{ij}\}_{n\times n}$.** They may be used, for instance, to scale the matrix prior to solving a corresponding set of linear equations, and are chosen so that the scaled matrix has its nonzeros near to unity in the sense that the sum of the squares of the logarithms of the nonzeros is minimized. The natural logarithms of the scaling factors r_i, c_j, $i,j = 1, 2, ..., n$ for the rows and columns are returned so that the scaled matrix has elements

$$b_{ij} = a_{ij} \exp(r_i + c_j).$$

The subroutine expects a square $n\times n$ matrix but tolerates rows or columns consisting entirely of zeros. Therefore a rectangular matrix can be handled by embedding it in a square matrix whose additional rows or columns consist entirely of zeros. Only the nonzeros, stored in any order, need be passed to the subroutine.

The method is described by Curtis and Reid, J. Inst. Maths. Applics. (1972), **10**, pp. 118-124.

ATTRIBUTES — Version: 1.0.0. **Types:** MC19A, MC19AD. **Original date:** March 1977. **Origin:** J. K. Reid, Harwell.

2 HOW TO USE THE PACKAGE

2.1 Argument list

The single precision version

```
CALL MC19A(N,NZ,A,IRN,ICN,R,C,W)
```

The double precision version

```
CALL MC19AD(N,NZ,A,IRN,ICN,R,C,W)
```

N is an INTEGER variable which must be set by the user to the order n of the matrix **A**. **Restriction:** $n \geq 1$.

NZ is an INTEGER variable which must be set by the user to the number of nonzeros in **A**. If NZ≤0 then **A** is regarded as the zero matrix.

A is a REAL array (DOUBLE PRECISION in the D version) of length NZ which must be set by the user to the nonzero elements of the matrix **A**. They may be in any order and the subroutine does not change their values.

IRN,ICN are two INTEGER arrays of length NZ which the user must set to the row and column numbers of the nonzeros. If the matrix element a_{ij} is held in A(K) then IRN(K) must contain i and ICN(K) must contain j.

R,C are two REAL arrays (single precision in both versions) of length n which need not be set by the user. On return R(i) and C(j) contain r_i and c_j, respectively.

W is a REAL array (single precision in both versions) of length $5n$ used for workspace.

2.2 Common

The single precision version

```
COMMON/MC19B/ LP,IFAIL
```

The double precision version

```
COMMON/MC19BD/ LP,IFAIL
```

`LP` is an `INTEGER` variable which specifies the Fortran stream number to be used for the error messages. The default value is 6. To suppress the printing of error messages set `LP` to zero.

`IFAIL` is an `INTEGER` variable which is always set by the subroutine to indicate success or failure. On exit from the subroutine, `IFAIL` will take one of the following values.

- 0 successful entry,
- 1 $n < 1$,
- 3 one or more nonzeros have their row or column indices outside the range 1 to n.

3 GENERAL INFORMATION

Use of common: the subroutine uses common block `MC19B/BD`, see §2.3.

Other routines called directly: `MC19C/CD` (block data).

Input/output: in the event of errors, diagnostic messages are printed. The output stream for these may be changed or the messages suppressed by altering the `COMMON` variable `LP`, see § 2.2.

Restrictions: $n \geq 1$.

4 METHOD

The variables r_i and c_j are chosen to minimize the function

$$\Phi = \sum_{i,j} (f_{ij} + r_i + c_j)^2$$

where

$$f_{ij} = \log|a_{ij}|$$

and the summation is over pairs i, j for which $a_{ij} \neq 0$. This done to sufficient accuracy in only a few matrix-by-vector multiplications. For further information, see Curtis and Reid, On the Automatic Scaling of Matrices for Gaussian Elimination, J. Inst. Maths. Applics. (1972), **10**, pp. 118-124.

Use of this method gives far better results on sparse matrices than scaling to equilibrate row and column norms.

5 EXAMPLE OF USE

The following program reads a sparse matrix, scales it and prints the result.

```
      REAL A(1000),R(100),C(100),W(500)
      INTEGER IRN(1000),ICN(1000)
C  READ ORDER AND NUMBER OF NONZEROS
      READ(5,*) N,NZ
C  CHECK THAT N AND NZ ARE WITHIN BOUNDS
      IF(N.LE.0.OR.N.GT.100) GO TO 40
      IF(NZ.LE.0.OR.NZ.GT.1000) GO TO 40
C  READ MATRIX NONZEROS
      READ(5,*) (IRN(I),ICN(I),A(I),I=1,NZ)
C
```

```
      CALL MC19A(N,NZ,A,IRN,ICN,R,C,W)
C
C  CALCULATE ROW AND COLUMN SCALING MULTIPLIERS
      DO 10 I=1,N
         R(I)=EXP(R(I))
         C(I)=EXP(C(I))
   10 CONTINUE
C  SCALE THE MATRIX
      DO 20 K=1,NZ
         I=IRN(K)
         J=ICN(K)
         A(K)=A(K)*R(I)*C(J)
   20 CONTINUE
C  PRINT THE SCALED NONZEROS
      WRITE(6,30) (IRN(I),ICN(I),A(I),I=1,NZ)
   30 FORMAT(2I5,F10.4)
      GO TO 60
C  DEAL WITH ERROR CONDITION
   40 WRITE(6,50)
   50 FORMAT(' N OR NZ OUT OF PERMITTED RANGE')
   60 STOP
      END
```

To scale the following matrix

$$\begin{pmatrix} 100. & 0. & 0. & 4. \\ 0. & 6. & 0. & 8. \\ 900. & 0. & 110000. & 0. \\ 0. & 14000. & 0 & 16000. \end{pmatrix}$$

we could have as input

```
   4       8
   4       4          16000.
   1       1            100.
   4       2          14000.
   2       2            6.
   3       1            900.
   2       4            8.
   3       3          110000.
   1       4            4.
```

and we would get the following output

```
   4    4    1.4055
   1    1    1.2847
   4    2    0.8942
   2    2    1.1184
   3    1    0.7784
   2    4    2.0509
   3    3    1.0000
   1    4    0.3469
```

which corresponds to the scaled matrix

$$\begin{pmatrix} 1.2846 & 0. & 0. & 0.3469 \\ 0. & 1.1184 & 0. & 2.0509 \\ 0.7784 & 0. & 1.0000 & 0. \\ 0. & 0.8942 & 0. & 1.4055 \end{pmatrix}$$