1 SUMMARY

This is for use in conjunction with the MA48 and MA50 packages for solving sparse unsymmetric sets of linear equations. If the matrix is singular or rectangular, it **identifies the rows and columns that are treated specially**. If the matrix is square, it **computes the determinant**.

It identifies which equations are ignored when solving $\mathbf{Ax}=\mathbf{b}$ and which solution components are always set to zero. The roles are reversed for $\mathbf{A}^T\mathbf{x}=\mathbf{b}$. There are such equations and/or components in the singular or rectangular case. Note that if $\mathbf{Ax}=\mathbf{b}$ or $\mathbf{A}^T\mathbf{x}=\mathbf{b}$ is not consistent, there may be large residuals for the equations that are ignored.

ATTRIBUTES — Version: 1.0.0. **Types:** Real (single, double). **Calls:** None. **Origin:** J.K. Reid, Rutherford Appleton Laboratory. **Original date:** May 1994. (i), (ii), (iii) and (iv).

2 HOW TO USE THE PACKAGE

2.1 Argument lists and calling sequences

There are four subroutines that can be called by the user:

1. MA51A/AD identifies the rows and columns that are treated specially and is for use following a call of MA48B/BD.
2. MA51B/BD identifies the rows and columns that are treated specially and is for use following a call of MA50B/BD.
3. MA51C/CD computes the determinant and is for use following a call of MA48B/BD.
4. MA51D/DD computes the determinant and is for use following a call of MA50B/BD.

2.1.1 To identify the rows and columns that are treated specially following a call of MA48B/BD

The single precision version

```
CALL MA51A(M,N,LA,IRN,KEEP,RANK,ROWS,COLS,W)
```

The double precision version

```
CALL MA51AD(M,N,LA,IRN,KEEP,RANK,ROWS,COLS,W)
```

M,N,LA are INTEGER variables that must be unchanged since the previous call to MA48B/BD. They are not altered by the subroutine.

IRN,KEEP are INTEGER arrays that must be unchanged since the previous call to MA48B/BD. They are not altered by the subroutine.

RANK is an INTEGER variable that need not be set by the user. On return, it holds the calculated rank of the matrix (it is the rank of the matrix actually factorized).

ROWS is an INTEGER array of length M that need not be set by the user. On return, it holds a permutation. The indices of the rows that are taken into account when solving $\mathbf{Ax}=\mathbf{b}$ are ROWS(i), $i \le$ RANK.

COLS is an INTEGER array of length N that need not be set by the user. On return, it holds a permutation. The indices of the columns that are taken into account when solving $\mathbf{Ax}=\mathbf{b}$ are COLS(j), $j \le$ RANK.

W is an INTEGER work array of length max (M,N).

2.1.2 To identify the rows and columns that are treated specially following a call of MA50B/BD

The single precision version

```
CALL MA51BD(M,N,IQ,NP,LFACT,IRNF,IPTRL,IPTRU,RANK,ROWS,COLS,W)
```

The double precision version

```
CALL MA51BD(M,N,IQ,NP,LFACT,IRNF,IPTRL,IPTRU,RANK,ROWS,COLS,W)
```

`M,N` are `INTEGER` variables that must be unchanged since the previous call to `MA50B/BD`. They are not altered by the subroutine.

`IQ` is an `INTEGER` array that must be unchanged since the previous call to `MA50B/BD`. `IQ` is not altered by the subroutine.

`NP,LFACT` are `INTEGER` variables that must be unchanged since the previous call to `MA50B/BD`. They are not altered by the subroutine.

`IRNF,IPTRL,IPTRU` are `INTEGER` arrays that must be unchanged since the previous call to `MA50B/BD`. They are not altered by the subroutine.

`RANK` is an `INTEGER` variable that need not be set by the user. On return, it holds the calculated rank of the matrix (it is the rank of the matrix actually factorized).

`ROWS` is an `INTEGER` array of length `M` that need not be set by the user. On return, it holds a permutation. The indices of the rows that are taken into account when solving $\mathbf{Ax}=\mathbf{b}$ are `ROWS`(i), $i \leq$ `RANK`.

`COLS` is an `INTEGER` array of length `N` that need not be set by the user. On return, it holds a permutation. The indices of the columns that are taken into account when solving $\mathbf{Ax}=\mathbf{b}$ are `COLS`(j), $j \leq$ `RANK`.

`W` is an `INTEGER` work array of length max (`M`,`N`).

2.1.3 To compute the determinant following a call of MA48B/BD

The single precision version

```
CALL MA51C(M,N,LA,A,IRN,KEEP,SGNDET,LOGDET,W)
```

The double precision version

```
CALL MA51CD(M,N,LA,A,IRN,KEEP,SGNDET,LOGDET,W)
```

`M,N,LA` are `INTEGER` variables that must be unchanged since the previous call to `MA48B/BD`. They are not altered by the subroutine.

`A` is a `REAL` (`DOUBLE PRECISION` in the D version) array that must be unchanged since the previous call to `MA48B/BD`. It is not altered by the subroutine.

`IRN,KEEP` are `INTEGER` arrays that must be unchanged since the previous call to `MA48B/BD`. They are not altered by the subroutine.

`SGNDET` is an `INTEGER` variable that need not be set by the user. On return, it has the value 1 if the determinant is positive, –1 if the determinant is negative, or 0 if the determinant is zero or the matrix is not square.

`LOGDET` is a `REAL` (`DOUBLE PRECISION` in the D version) variable that need not be set by the user. On return, it holds the logarithm of the absolute value of the determinant, or zero if the determinant is zero or the matrix is not square.

`W` is an `INTEGER` work array of length `N`.

2.1.4 To compute the determinant following a call of MA50B/BD

The single precision version

```
CALL MA51DD(M,N,IQ,NP,LFACT,FACT,IRNF,IPTRL,IPTRU,SGNDET,LOGDET,W)
```

The double precision version

```
CALL MA51DD(M,N,IQ,NP,LFACT,FACT,IRNF,IPTRL,IPTRU,SGNDET,LOGDET,W)
```

`M,N` are `INTEGER` variables that must be unchanged since the previous call to `MA50B/BD`. They are not altered by the subroutine.

`IQ` is an `INTEGER` array that must be unchanged since the previous call to `MA50B/BD`. `IQ` is not altered by the subroutine.

`NP,LFACT` are `INTEGER` variables that must be unchanged since the previous call to `MA50B/BD`. They are not altered by the subroutine.

`FACT` is a `REAL` (`DOUBLE PRECISION` in the D version) array that must be unchanged since the previous call to `MA50B/BD`. It is not altered by the subroutine.

`IRNF,IPTRL,IPTRU` are `INTEGER` arrays that must be unchanged since the previous call to `MA50B/BD`. They are not altered by the subroutine.

`SGNDET` is an `INTEGER` variable that need not be set by the user. On return, it has the value 1 if the determinant is positive, –1 if the determinant is negative, or 0 if the determinant is zero or the matrix is not square.

`LOGDET` is a `REAL` (`DOUBLE PRECISION` in the D version) variable that need not be set by the user. On return, it holds the logarithm of the absolute value of the determinant, or zero if the determinant is zero or the matrix is not square.

3 GENERAL INFORMATION

Use of common: None.

Other routines called directly: `MA51` calls `MA51X/XD`, `MA51Y/YD`, and `MA51Z/ZD`, which never need to be called directly by the user.

Input/output: None.

4 METHOD

The subroutines that this package supports all identify a square submatrix of **A** that they consider to be nonsingular. When solving $\mathbf{Ax}=\mathbf{b}$, equations outside this submatrix are ignored and solution components that correspond to columns outside the submatrix are set to zero. This package identifies such rows and columns from stored integer data.

For the determinant, the subroutine computes the parity of the permutations that have been applied and the product of the diagonal entries of the triangular factors.

It should be emphasized that the primary purpose of the `MA48` and `MA50` packages is to solve square nonsingular sets of equations. The rank is determined from any zero or small pivots encountered. There are more reliable (but much more expensive) ways of determining numerical rank.

5 EXAMPLE OF USE

In the example code shown below, we read two matrices. Each is factorized, and its determinant computed and printed. If the determinant is zero, we determine which rows and columns are associated with the nonsingular submatrix of the factorization.

```
       INTEGER  LA, MAXN
      PARAMETER (LA=200, MAXN=20)
      DOUBLE PRECISION A(LA),CNTL(5),LOGDET,RINFO,W(4*MAXN)
      INTEGER  COLS(MAXN),I,ICNTL(9),INFO(12),IRN(LA),IW(9*MAXN),
     *         JCN(LA),KEEP(7+10*MAXN),M,N,NE,RANK,ROWS(MAXN),SGNDET,
     *         WORK(MAXN)

      DO KASE = 1,2
C     Read in input matrix.
        READ(5,*) M,N,NE
        READ(5,*) (IRN(I),JCN(I),A(I),I=1,NE)

C     Factorize matrix
        CALL MA48ID(CNTL,ICNTL)
        CALL MA48AD(M,N,NE,1,LA,A,IRN,JCN,KEEP,CNTL,ICNTL,IW,INFO,RINFO)
        CALL MA48BD(M,N,NE,1,LA,A,IRN,JCN,KEEP,CNTL,ICNTL,W,IW,INFO,
     *            RINFO)

C     Compute the determinant
        CALL MA51CD(M,N,LA,A,IRN,KEEP,SGNDET,LOGDET,WORK)

C     Print result
        IF(SGNDET.GT.0)THEN
          WRITE(6,'(/,A,F9.3)')
     +    ' Determinant is positive; log(determinant) =',LOGDET
        ELSE IF(SGNDET.LT.0)THEN
          WRITE(6,'(/,A,F9.3)')
     +  ' Determinant is negative; log(-determinant) =',LOGDET
        ELSE
          WRITE(6,'(/,A)')' Determinant is zero'

C     Determine the nonsingular submatrix of the factorization
          CALL MA51AD(M,N,LA,IRN,KEEP,RANK,ROWS,COLS,WORK)

C     Print out rank and identifying vectors.
          WRITE(6,'(A,I2)')' Rank =',RANK
          WRITE(6,'(A,9I2)') ' ROWS =',(ROWS(I),I=1,RANK)
          WRITE(6,'(A,9I2)') ' COLS =',(COLS(I),I=1,RANK)
        END IF
      END DO

      END
```

For the matrices:

$$\begin{pmatrix} 7.5 & - & - & - & 3.1 \\ - & - & - & - & - \\ 1.0 & - & 4.1 & - & - \\ - & - & 0.3 & 4.1 & - \end{pmatrix} \text{ and } \begin{pmatrix} 7.5 & 3.1 & - & - \\ - & - & 0.3 & - \\ - & - & 4.1 & 4.1 \\ 1.0 & - & - & - \end{pmatrix}$$

we have as input

```
4     5     6
1     5     3.1
4     3     0.3
3     3     4.1
4     4     4.1
1     1     7.5
3     1     1.0

4     4     6
1     2     3.1
2     3     0.3
3     3     4.1
3     4     4.1
1     1     7.5
4     1     1.0
```

and the output would be

```
Determinant is zero
Rank = 3
ROWS = 1 3 4
COLS = 1 3 4

Determinant is negative; log(-determinant) =    1.338
```