

Warning: Subroutine MA01 performs functions which are adequately treated by routines in other standard subroutine libraries (for example, LAPACK). The use of this routine is not recommended, and it may be removed from future releases of this library.

1 SUMMARY

To solve a system of n linear algebraic equations in n unknowns with one or more right-hand sides, i.e. solve

$$\sum_{j=1}^n a_{ij} x_{ji} = b_{il} \quad i=1,2,\dots,n \quad l=1,2,\dots,k$$

and optionally compute the inverse matrix \mathbf{A}^{-1} of the equation coefficient matrix $\mathbf{A}=\{a_{ij}\}_{n \times n}$.

Gaussian elimination with partial pivoting is used, see 'Modern Computing Methods', NPL, 1957, with double length accumulation of inner products.

No equation balancing or solution refinement is performed and the accuracy of the solution may be poor when the equations are not well conditioned.

ATTRIBUTES — **Version:** 1.0.0. **Remark:** Superseded by MA21. **Types:** MA01B; MA01BD. **Calls:** ZE01. **Original date:** February 1963. **Origin:** E.J.York*, Harwell.

2 HOW TO USE THE PACKAGE

2.1 The argument list and calling sequence

The single precision version:

```
CALL MA01B(A,B,N,K,INV,IA,IB)
```

The double precision version:

```
CALL MA01BD(A,B,N,K,INV,IA,IB)
```

- A is a two-dimensional REAL (DOUBLE PRECISION in the D version) array of first dimension IA. It is used to pass the left hand sides of the equations to the subroutine, then it is used by the subroutine as workspace and also to return the inverse matrix. On entry the user must set $A(I,J)$, $I=1,N$, $J=1,N$ to the elements a_{ij} , $i=1,2,\dots,n$, $j=1,2,\dots,n$. The inverse matrix \mathbf{A}^{-1} , if requested, is returned in the array A, otherwise, the array contains the factorization of a version of the matrix \mathbf{A} that has undergone row interchanges. Details of the interchanges are not returned. **N.B.** the array A is always overwritten by the subroutine.
- B is a two-dimensional REAL (DOUBLE PRECISION in the D version) array of first dimension IB. It is used to pass the right hand sides of the equations and to return the solutions. The user must set the columns of B to the k right hand sides of the equations, i.e. set $B(I,L)$, $I=1,N$, to b_{il} $i=1,2,\dots,n$ for L and $l=1,2,\dots,k$. If there is only one right hand side, B can be a one-dimensional array. When solutions have been requested the elements $B(I,L)$, $I=1,N$, are returned set to x_{il} $i=1,2,\dots,n$ for L and $l=1,2,\dots,k$. If solutions are not requested B is not used or altered.
- N is an INTEGER variable and must be set by the user to n the number of equations. **Restriction:** $2 \leq n \leq 100$.
- K is an INTEGER variable and must be set by the user to k the number of right hand side vectors.

INV is an INTEGER variable set by the user to select the inversion option. The following options can be selected:

INV=0 no inverse is computed.

INV>0 the inverse is computed and returned in A.

INV<0 the inverse is computed by re-entering the subroutine with INV<0 after a previous call with INV=0. In this case most of the work has been done on the previous call and the subroutine completes the inverse but only on the assumption that the arguments A, N, IA and IB are unaltered.

IA is an INTEGER variable set by the user to the first dimension of the array A. **N.B.** this is required because the first dimension may be greater than n . **Restriction:** $IA \geq n$.

IB is an INTEGER variable set by the user to the first dimension of the array B. **Restriction:** $IB \geq 1$.

3 GENERAL INFORMATION

Use of Common: none.

Workspace: two private arrays of length 100 are used for workspace. These are the cause of the restriction $n \leq 100$ but the limit can be increased by recompiling with larger dimensions. One of these arrays holds the row interchanges.

Other subroutines: ZE01 is called.

Input Output: none.

Restrictions: $2 \leq n \leq 100$.

4 METHOD

The method is that of simple Gaussian elimination with partial pivoting followed by back substitution to obtain the solution. No balancing of equations is performed and no solution refinement is carried out.