1 SUMMARY

To **solve a system** of n **linear algebraic equations** in n unknowns,

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \qquad i=1, 2,..., n,$$

where the coefficient matrix $\mathbf{A}=\{a_{ij}\}_{n\times n}$ is a **band matrix** of bandwidth k.

The equations are solved by the method of Gaussian elimination with partial pivoting. The matrix **A** is passed to the subroutine in a compact form.

When several systems with identical left-hand side matrices **A** are to be solved the subroutine may be re-entered in a way that avoids repeating the elimination phase.

ATTRIBUTES — **Version:** 1.0.0. **Remark:** Formerly MA07B. For positive definite band systems see also MA36. **Types:** MA35A; MA35AD. **Calls:** _AXPY, _COPY, _DOT, _ROT and _ROTG. **Original date:** January 1970. **Origin:** J.K.Reid, Harwell.

2 HOW TO USE THE PACKAGE

2.1 Argument list

The single precision version

```
CALL MA35A(A,B,IA,N,K,PT)
```

The double precision version

```
CALL MA35AD(A,B,IA,N,K,PT)
```

A is a REAL (DOUBLE PRECISION in the D version) two-dimensional array of first dimension IA which must be set by the user to contain the elements a_{ij} of the coefficient matrix **A**. The elements are to be stored in compact form. Suppose the nonzero part of the matrix **A** to be made up of k diagonals (see argument K), then all elements in the same diagonal should be stored in the same column of the array A, and all elements of the same row of the matrix **A** should be in the same row of the array A. In fact, the matrix element a_{ij} is stored in A(i,j-i+(k+1)/2). The elements of the top right-hand corner and the bottom left-hand corner need not be set. Note that the dimensions of A must be at least n by $(3k+1)/2$ and that A is altered by the subroutine.

B is a REAL (DOUBLE PRECISION in the D version) array which must be set by the user to the right-hand side b_i, i=1, 2,..., n. The subroutine will overwrite the elements of B with the solution x_i, i=1, 2,..., n.

IA is an INTEGER variable which must be set by the user to the first dimension of the array A. For example, if the space allocation for A was specified by

```
DIMENSION A(100,50)
```

then IA would have to be set to 100. **Restriction:** IA $\geq n$. IA is not altered by the subroutine.

N is an INTEGER variable which must be set by the user to n the number of equations. N is not altered by the subroutine.

K is an INTEGER which must be set by the user to k the bandwidth of the matrix A (see argument A). The band is assumed to be centered about the diagonal and therefore k must be odd. K is not altered by the subroutine. **Restriction:** k must be odd.

PT is a REAL variable which must be set by the user to tell the subroutine if the elimination phase has already been done for this matrix by a previous call. If this is a fresh matrix, which will be the normal case, PT must be set to a nonzero value. If it is set to zero, the subroutine assumes that the elimination phase has been completed, i.e. that on a previous call to the subroutine, equations with the same coefficient matrix were solved, and the resultant elements returned in A have been undisturbed since. This is an efficient procedure to solve equations with more than one right-hand side.

On return from the subroutine PT is set to the value of the smallest pivot. If a pivot of zero is found a diagnostic message is printed and the attempt to solve the equations is abandoned.

Note that if the subroutine is entered several times care must be taken to reset PT before re-entry.

3 GENERAL INFORMATION

Workspace: None.

Use of common: None.

Other routines called directly: None.

Input/output: Diagnostic message (see argument PT).

Restrictions: IA $\geq n$, k must be odd.

4 METHOD

Gaussian elimination is used to triangularize the equations, after which a back substitution is carried out to obtain the solution.

Row interchanges are used at each step of the elimination. This is a sensible strategy if the errors in the matrix elements are all of a similar size (these errors may be caused by the finite machine word length or by errors in experimental data or in the way the matrix was constructed). If the smallest pivot is of a comparable size with the errors, then the errors in the solution x_i, i=1, 2,..., n, will be large. In the limiting case of a zero pivot, no sensible answer can be obtained.