

Warning: Subroutine MA33 has been superseded by subroutine MA50 which uses improved algorithms; the use of the latter routine is recommended. The superseded routine may be removed from later releases of the library.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

These subroutines perform operations pertinent to the solution of a general sparse $n \times n$ system of linear equations

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

(i.e. solve $\mathbf{Ax}=\mathbf{b}$). Structurally singular matrices are permitted, including those with rows or columns consisting entirely of zeros (i.e. including rectangular matrices). It is assumed that the nonzero entries a_{ij} do not differ widely in size. If necessary, a prior call of the scaling subroutine MC19A/AD may be made.

- (a) MA33I/ID sets default values for control parameters prior to calls to the remaining subroutines.
- (b) MA33A/AD performs the **LU** decomposition of a user-input matrix **A**. The user can define input permutations **P**₁ and **Q**₁ so that **P**₁**AQ**₁ is in block lower triangular form, in which case only the diagonal blocks are factorized and any permutations for pivoting are confined to the diagonal blocks. In this case, in the resulting factorization

$$\mathbf{PAQ} = \mathbf{LU},$$

P and **Q** differ from **P**₁ and **Q**₁ only within the diagonal blocks. The input permutations **P**₁ and **Q**₁ may be found by calling MC23A/AD or the matrix may be treated as a single block by using **P**₁=**Q**₁=**I**. The matrix entries are held compactly by rows, although it should be noted that the user can supply the matrix by columns to get the **LU** decomposition of **A**^T.

- (c) MA33B/BD performs the **LU** decomposition of the diagonal blocks of a new matrix **PAQ** of the same sparsity pattern and the same **P** and **Q**, using information from a previous call to MA33A/AD. The entries of the input matrix must already be in their final positions in the **LU** decomposition structure. This routine typically executes about five times faster than MA33A/AD.
- (d) MA33C/CD uses the factors produced by MA33A/AD or MA33B/BD to solve $\mathbf{Ax}=\mathbf{b}$ or $\mathbf{A}^T \mathbf{x}=\mathbf{b}$.

ATTRIBUTES — **Version:** 1.0.0. **Remark:** This is a thread-safe version of the MA30 package. **Types:** MA33A, MA33AD. **Original date:** April 1977, modified March 2006. **Origin:** I. S. Duff, Harwell, thread-safe modifications, N. I. M. Gould, Rutherford Appleton Laboratory.

2 HOW TO USE THE PACKAGE

2.1 Argument lists

2.1.1 Argument list for MA33I/ID

The single precision version

```
CALL MA33I(ICNTL,CNTL)
```

The double precision version

```
CALL MA33ID(ICNTL,CNTL)
```

ICNTL is an INTEGER array of length 10 which contains integer control values used by the other MA33 subroutines. It is initialized to default values which may be reset by the user, see Section 2.2.1.

CNTL is an REAL (DOUBLE PRECISION in the D version) array of length 5 which contains real control values used by the other MA33 subroutines. It is initialized to default values which may be reset by the user, see Section 2.2.1.

2.1.2 Argument list for MA33A/AD

The single precision version

```
CALL MA33A(N,ICN,A,LICN,LENR,LENRL,IDISP,IP,IQ,IRN,LIRN,
+          LENC,IFIRST,LASTR,NEXTR,LASTC,NEXTC,IPTR,
+          IPC,U,IFLAG,ICNTL,CNTL,INFO,RINFO)
```

The double precision version

```
CALL MA33AD(N,ICN,A,LICN,LENR,LENRL,IDISP,IP,IQ,IRN,LIRN,
+          LENC,IFIRST,LASTR,NEXTR,LASTC,NEXTC,IPTR,
+          IPC,U,IFLAG,ICNTL,CNTL,INFO,RINFO)
```

N is an INTEGER variable which must be set by the user to the order of the matrix. N is not altered by MA33A/AD.

ICN is an INTEGER array of length LICN. Positions IDISP(2) to LICN must be set by the user to contain the column indices of the entries in the diagonal blocks of $\mathbf{P}_1 \mathbf{A} \mathbf{Q}_1$. Those belonging to a single row must be contiguous, but the ordering of column indices within each row is unimportant. The entries of row i precede those of row $i+1$, $i=1, \dots, n-1$ and no wasted space is allowed between the rows. On output the column indices of the LU decomposition of $\mathbf{P} \mathbf{A} \mathbf{Q}$ are held in positions IDISP(1) to IDISP(2), the rows are in pivotal order, and the column indices of the L part of each row are in pivotal order and precede those of U. Again there is no wasted space either within a row or between the rows. ICN(1) to ICN(IDISP(1)-1) are neither required nor altered. If MC23A/AD has been called, these will hold information about the off-diagonal blocks.

A is a REAL (DOUBLE PRECISION in the D version) array of length LICN whose entries IDISP(2) to LICN must be set by the user to the values of the nonzero entries of the matrix in the order indicated by ICN. On output A will hold the LU factors of the matrix, where again the position in the matrix is determined by the corresponding values in ICN. A(1) to A(IDISP(1)-1) are neither required nor altered.

LICN is an INTEGER variable which must be set by the user to the length of arrays ICN and A. It must be big enough for A and ICN to hold all the entries of L and U and leave some elbow room. It is possible to calculate a minimum value for LICN by a preliminary run of MA33A/AD (see INFO(5), Section 2.2). The adequacy of the elbow room can be judged by the size of the variable INFO(2) (see Section 2.2.2). LICN is not altered by MA33A/AD.

LENR is an INTEGER array of length N. On input, LENR(i) should equal the number of entries in row i , $i=1, \dots, N$ of the diagonal blocks of $\mathbf{P}_1 \mathbf{A} \mathbf{Q}_1$. On output, LENR(i) will equal the total number of entries in row i of L and row i of U (excluding the diagonal in the case of L).

LENRL is an INTEGER array of length N which need not be set by the user. On output from MA33A/AD, LENRL(i) will

hold the number of entries in row i of \mathbf{L} , excluding the diagonal.

IDISP is an INTEGER array of length 2. The user should set IDISP(1) to be the first available position in the array pair A/ICN for the LU decomposition and IDISP(2) to the position in A/ICN of the first entry in the diagonal blocks of $\mathbf{P}_1\mathbf{A}\mathbf{Q}_1$. On output, IDISP(1) will be unaltered while IDISP(2) will be set to the position in A/ICN of the last entry of the LU decomposition.

IP is an INTEGER array of length N. On input to MA33A/AD, the absolute value of IP(i) must be set to the row of \mathbf{A} which is row i of $\mathbf{P}_1\mathbf{A}\mathbf{Q}_1$. A negative value for IP(i) indicates that row i is at the end of a diagonal block. On output from MA33A/AD, IP(i) indicates the row of \mathbf{A} that is row i of \mathbf{PAQ} . IP(i) will still be negative for the last row of each block (except the last).

IQ is an INTEGER array of length N. On input to MA33A/AD, IQ(j) must be set to the column of \mathbf{A} which is column j of $\mathbf{P}_1\mathbf{A}\mathbf{Q}_1$. On output from MA33A/AD, the absolute value of IQ(j) indicates the column of \mathbf{A} that is column j in \mathbf{PAQ} . For a row, i say, in which a zero pivot is detected, IQ(i) is negated (see Section 2.6.1).

IRN is an INTEGER array of length LIRN used as workspace by MA33A/AD.

LIRN is an INTEGER variable which must be set to the length of IRN and is not altered. It must be greater than the largest number of entries in a diagonal block of $\mathbf{P}_1\mathbf{A}\mathbf{Q}_1$ but need not be as large as LICN. It should be large enough to hold the active part of any block, plus some elbowroom, the *a posteriori* adequacy of which can be estimated by examining the size of variable INFO(1) (see Section 2.2).

LENC, IFIRST, LASTR, NEXTR, LASTC, NEXTC are all INTEGER arrays of length N which are used as workspace by MA33A/AD. If ICNTL(5) (see Section 2.2.1) is set to a value less than or equal to N, arrays LASTC and NEXTC are not referenced by MA33A/AD and so can be dummied in the call to MA33A/AD.

IPTR, IPC are INTEGER arrays of length N which are used as workspace by MA33A/AD.

U is a REAL (DOUBLE PRECISION in the D version) variable which should be set by the user to a value between 0. and 1.0. If less than zero it is reset to zero and if its value is 1.0 or greater it is reset to 0.9999 (0.999999999 in the D version). It determines the balance between pivoting for sparsity and for stability, values near zero emphasizing sparsity and values near one emphasizing stability. We recommend $U=0.1$ as a possible first trial value. The stability can be judged by a later call to MC24A/AD or by setting ICNTL(6)=1 (see Section 2.2.1).

IFLAG is an INTEGER variable. It will have a non-negative value if MA33A/AD is successful. Negative values indicate error conditions while positive values indicate that the matrix has been successfully decomposed but is singular (see Section 2.3.1).

ICNTL is an INTEGER array of length 10 which contains integer control values, see Section 2.2.1. ICNTL is not altered by MA33A/AD.

CNTL is a REAL (DOUBLE PRECISION in the D version) array of length 5 which contains real control values, see Section 2.2.1. CNTL is not altered by MA33A/AD.

INFO is an INTEGER array of length 10 in which MA33 returns integer information to the user, see Section 2.2.2.

RINFO is a REAL (DOUBLE PRECISION in the D version) array array of length 5 in which MA33 returns real information to the user, see Section 2.2.2.

2.1.3 Argument list for MA33B/BD

The single precision version

```
CALL MA33B(N, ICN, A, LICN, LENR, LENRL, IDISP, IP, IQ, W, IW, IFLAG,
+          ICNTL, CNTL, RINFO)
```

The double precision version

```
CALL MA33BD(N, ICN, A, LICN, LENR, LENRL, IDISP, IP, IQ, W, IW, IFLAG,
+          ICNTL, CNTL, RINFO)
```

- N** is an INTEGER variable set to the order of the matrix. It is not altered.
- ICN** is an INTEGER array of length LICN. It should be unchanged since the last call to MA33A/AD. It is not altered by MA33B/BD.
- A** is a REAL (DOUBLE PRECISION in the D version) array of length LICN. The user must set components IDISP(1) to IDISP(2) of A to contain the entries in the diagonal blocks of the matrix **PAQ** whose column numbers are held in ICN, using corresponding positions. Note that some zeros may need to be held explicitly. On output, entries IDISP(1) to IDISP(2) of array A contain the **LU** decomposition of the diagonal blocks of **PAQ**. Entries A(1) to A(IDISP(1)-1) are neither required nor altered by MA33B/BD.
- LICN** is an INTEGER variable which must be set by the user to the length of arrays A and ICN. It is not altered by MA33B/BD.
- LENR, LENRL** are INTEGER arrays of length N. They should be unchanged since the last call to MA33A/AD. They are not altered by MA33B/BD.
- IDISP** is an INTEGER array of length 2. It should be unchanged since the last call to MA33A/AD. It is not altered by MA33B/BD.
- IP, IQ** are INTEGER arrays of length N. They should be unchanged since the last call to MA33A/AD. They are not altered by MA33B/BD.
- W** is a REAL (DOUBLE PRECISION in the D version) array of length N which is used as workspace by MA33B/BD.
- IW** is an INTEGER array of length N which is used as workspace by MA33B/BD.
- IFLAG** is an INTEGER variable. On output from MA33B/BD, IFLAG has the value zero if the factorization was successful, has the value I if pivot I was very small (see Section 2.3.2) and has the value -I if an unexpected singularity was detected at stage I of the decomposition (see Section 2.3.2).
- ICNTL** is an INTEGER array of length 10 which contains integer control values, see Section 2.2.1. ICNTL is not altered by MA33A/AD.
- CNTL** is a REAL (DOUBLE PRECISION in the D version) array of length 5 which contains real control values, see Section 2.2.1. CNTL is not altered by MA33A/AD.
- RINFO** is a REAL (DOUBLE PRECISION in the D version) array array of length 5 in which MA33 returns real information to the user, see Section 2.2.2.

2.1.4 Argument list for MA33C/CD

The single precision version

```
CALL MA33C(N, ICN, A, LICN, LENR, LENRL, LENOFF,
+         IDISP, IP, IQ, X, W, MTYPE, RINFO)
```

The double precision version

```
CALL MA33CD(N, ICN, A, LICN, LENR, LENRL, LENOFF,
+         IDISP, IP, IQ, X, W, MTYPE, RINFO)
```

- N** is an INTEGER variable set to the order of the matrix. It is not altered by the subroutine.
- ICN** is an INTEGER array of length LICN. Entries IDISP(1) to IDISP(2) should be unchanged since the last call to MA33A/AD. If the matrix has more than one diagonal block, then column indices corresponding to entries in subdiagonal blocks of **PAQ** must appear in positions 1 to IDISP(1)-1. For each row these entries must be contiguous, with those in row i preceding those in row i+1 (i=1,..., N-1) and no wasted space between rows. Entries may be in any order within each row. ICN is not altered by MA33C/CD.
- A** is a REAL (DOUBLE PRECISION in the D version) array of length LICN. Entries IDISP(1) to IDISP(2) should be unchanged since the last call to MA33A/AD or MA33B/BD. If the matrix has more than one diagonal block, then the values of the entries in subdiagonal blocks must be in positions 1 to IDISP(1)-1 in the order given by

ICN. It is not altered by MA33C/CD.

LICN is an INTEGER variable set to the size of arrays ICN and A. It is not altered by MA33C/CD.

LENR, LENRL are INTEGER arrays of length N which should be unchanged since the last call to MA33A/AD. They are not altered by MA33C/CD.

LENOFF is an INTEGER array of length N. If the matrix \mathbf{PAQ} (or $\mathbf{P}_1\mathbf{AQ}_1$) has more than one diagonal block, then $\text{LENOFF}(i)$, $i=1, \dots, N$ should be set to the number of entries in row i of the matrix \mathbf{PAQ} which are in subdiagonal blocks. If there is only one diagonal block then $\text{LENOFF}(1)$ may be set to -1 , in which case the other entries of LENOFF are never accessed. LENOFF is not altered by MA33C/CD.

IDISP is an INTEGER array of length 2 which should be unchanged since the last call to MA33A/AD. It is not altered by MA33C/CD.

IP, IQ are INTEGER arrays of length N which should be unchanged since the last call to MA33A/AD. They are not altered by MA33C/CD.

X is a REAL (DOUBLE PRECISION in the D version) array of length N. It must be set by the user to the values of the right-hand side vector \mathbf{b} for the equations being solved. On exit from MA33C/CD it will be equal to the solution \mathbf{x} .

W is a REAL (DOUBLE PRECISION in the D version) array of length N which is used as workspace by MA33C/CD.

MTYPE is an INTEGER variable which must be set by the user. If $\text{MTYPE}=1$, the solution to the system $\mathbf{Ax}=\mathbf{b}$ is returned; any other value for MTYPE will return the solution to the system $\mathbf{A}^T\mathbf{x}=\mathbf{b}$. MTYPE is not altered by MA33C/CD.

RINFO is a REAL (DOUBLE PRECISION in the D version) array array of length 5 in which MA33 returns real information to the user, see Section 2.2.2.

2.2 The control and information arrays

2.2.1 Control arrays

The INTEGER array ICNTL contains integer control values used by the other MA33 entries. It is initialized to default values by MA33I/ID and may subsequently be reset by the user.

ICNTL(1) (was LP in MA30) is used by the subroutines as the unit number for error messages. If it is set to less-than-or-equal-to zero these messages will be suppressed. The default value is $\text{ICNTL}(1)=6$.

ICNTL(2) (was LOGICAL ABORT1 in MA30) is used by MA33A/AD and MA33B/BD to warn of structural singularity. If ICNTL(2) is set 1, MA33A/AD will exit immediately on detecting structural singularity. Otherwise, the computation will continue. The default value is $\text{ICNTL}(2)=1$.

ICNTL(3) (was LOGICAL ABORT2 in MA30) is used by MA33A/AD and MA33B/BD to warn of numerical singularity. If ICNTL(3) is set 1, MA33A/AD will exit immediately on detecting numerical singularity. Otherwise, the computation will continue. The default value is $\text{ICNTL}(3)=1$.

ICNTL(4) (was LOGICAL ABORT3 in MA30) is used by MA33A/AD and MA33B/BD to warn if the available space in the array pair A/ICN has been filled up by the previously decomposed, active, and undecomposed parts of the matrix. If ICNTL(3) is set 0, MA33A/AD will exit immediately, as soon as A/ICN has been filled. Otherwise, the computation will continue. The default value is $\text{ICNTL}(4)=0$.

ICNTL(5) (was NSRCH in MA30) is used by MA33A/AD to limit the pivot search. If ICNTL(5) is set to a value less than or equal to N, the pivot search is limited to ICNTL(5) rows. This may result in different fill-in and execution time for MA33A/AD. If ICNTL(5) is less than or equal to N, the workspace arrays LASTC and NEXTC (see Section 2.1.3) are not referenced by MA33A/AD. The default value for ICNTL(5) is 32768.

ICNTL(6) (was LOGICAL LBIG in MA30) is used by MA33A/AD and MA33B/BD to control the computation of the

largest entry encountered in the factorization. If $ICNTL(6)$ is set to 1, the value of the largest entry encountered in the factorization by MA33A/AD is returned in $RINFO(2)$. Setting $ICNTL(6)=1$ will marginally increase the factorization time. The default value is $ICNTL(6)=0$.

Components $ICNTL(7)$ to $ICNTL(10)$ are not currently used by the package.

The REAL (DOUBLE PRECISION in the D version) array contains real control values used by the other MA33 entries. It is initialized to default values by MA33I/ID and may subsequently be reset by the user.

$CNTL(1)$ (was EPS in MA30) is used by MA33B/BD to test for small pivots. If the user sets $CNTL(1)$ to any value greater than 1.0, no check is made on the size of the pivots. Although the absence of such a check would fail to warn the user of bad instability, its absence will enable MA33B/BD to run slightly faster. An *a posteriori* check on the stability of the factorization can be obtained from MC24A/AD. The default value is $CNTL(1)=0.0001$.

$CNTL(2)$ (was TOL in MA30) is used by MA33A/AD and MA33B/BD to remove small entries during the factorization. If it is set to a positive value, MA33A/AD will drop from the factors any entry whose modulus is less than $CNTL(2)$. The factorization will then require less storage but will be inaccurate. After a run of MA33A/AD where entries have been dropped, MA33B/BD should not be called. The default value is $CNTL(2)=0.0$.

Components $CNTL(3)$ to $ICNTL(5)$ are not currently used by the package.

2.2.2 Information arrays

On return from the three principal MA33 subroutines the INTEGER array INFO and REAL (DOUBLE PRECISION in the D version) array RINFO contain information for the user.

$INFO(1)$ (was IRNCP in MA30) monitors the adequacy of the allocated space in array IRN by recording the number of data management compresses performed on the array during the execution of MA33A/AD. If $INFO(1)$ is fairly large (say greater than $N/10$), it may be advantageous to increase the size of IRN.

$INFO(2)$ (was ICNCP in MA30) monitors the adequacy of the allocated space in arrays A/ICN by recording the number of data management compresses performed on these arrays during the execution of MA33A/AD. If $INFO(2)$ is fairly large (say greater than $N/10$), it may be advantageous to increase the size of the corresponding arrays.

$INFO(3)$ (was IRANK in MA30) gives an estimate (actually an upper bound) of the rank of the matrix (see Section 2.6.1) on exit from MA33A/AD. If $IFLAG=0$, this will be equal to N .

$INFO(4)$ (was MINIRN in MA30) indicates, following a successful call to MA33A/AD, the minimum length to which IRN could be reduced while still permitting a successful decomposition of the given matrix. If, however, the user were to decrease the length of IRN to that size, the number of compresses ($INFO(1)$) may be very high and quite costly. If $LIRN$ is not large enough to begin the decomposition on a diagonal block, $INFO(4)$ will be equal to the value required to continue the decomposition and $IFLAG$ will be set to -3 or -6 (see Section 2.3.1). A value of $LIRN$ slightly greater than this (say about $N/2$) will usually provide enough space to complete the decomposition on that block. In the event of any other failure $INFO(4)$ gives the minimum size of IRN required for a successful decomposition up to the failure.

$INFO(5)$ (was MINICN in MA30) indicates, following a successful call to MA33A/AD, the minimum size of LICN required to enable a successful decomposition. In the event of failure with $IFLAG=-5$ (see Section 2.3.1), $INFO(5)$ will, if $ICNTL(4)=0$, indicate the minimum length that would be sufficient to prevent this error in a subsequent run on an identical matrix. Again the user may prefer to use a value of ICN slightly greater than $INFO(5)$ for subsequent runs to avoid too many compresses ($INFO(2)$). In the event of failure with $IFLAG$ equal to any negative value except -4 , $INFO(5)$ will give the minimum length to which LICN could be reduced to enable a successful decomposition to the point at which failure occurred. Notice that, on a successful entry, $IDISP(2)$ gives the amount of space in A/ICN required for the decomposition while $INFO(5)$ will usually be slightly greater because of the need for elbow room. If the user is very unsure how large to make LICN, the variable $INFO(5)$ can be used to provide that information. A preliminary run should be performed with

ICNTL(4)=0 and LICN about $1\frac{1}{2}$ times as big as the number of entries in the original matrix. Unless the initial problem is very sparse (when the run will be successful) or fills in extremely badly (giving an error return with IFLAG equal to -4), an error return with IFLAG equal to -5 should result and INFO(5) will give the amount of space required for a successful decomposition.

INFO(6) (was NDROP in MA30) holds the number of entries dropped from the data structure by MA33A/AD if CNTL(2) GT 0. Otherwise it will be set to zero.

RINFO(1) (was RMIN in MA30) gives the user some information about the stability of the decomposition in MA33B/BD. At each stage, i say, of the LU decomposition, the magnitude of the pivot $APIV_i$ is compared with the largest off-diagonal entry ($ROWMAX_i$) currently in its row (row i of \mathbf{U}). If the smallest ratio $APIV_i/ROWMAX_i$ is less than CNTL(1), RINFO(1) is set to this value and IFLAG is returned with the value I (see Section 2.3), which is the index of the row in which this occurs. If the user sets CNTL(1) greater than one, this test is not performed. In this case, and when there are no small pivots, RINFO(1) will be set equal to CNTL(1).

RINFO(2) (was BIG in MA30) will, if, ICNTL(6)=1, be set to the largest entry encountered during the factorization on exit from MA33A/AD and MA33B/BD. Otherwise it will be set to zero.

RINFO(3) (was RESID in MA30) will be set on exit from MA33C/CD, in the case of singular or rectangular matrices, to the maximum residual $|b_i - \sum_j a_{ij}x_j|$ for the unsatisfied equations. Otherwise it will be set to zero.

Components INFO(7) to INFO(10), RINFO(4) and RINFO(5) are not currently used but set to zero.

2.3 Error diagnostics

2.3.1 Error diagnostics for MA33A/AD

If the subroutine performs the LU decomposition without any complications or errors, the value of IFLAG will be non-negative on exit from MA33A/AD. The complications which can arise are given below. Some are more disastrous than others and the user must decide in each instance what further action to take. In some cases messages are output on unit number ICNTL(1) (unless ICNTL(1)=0, see Section 2.2.1). Possible negative values for IFLAG are as follows:

- 1 The matrix is structurally singular with rank given by INFO(3) (see Section 2.2.2). The message: Error return from MA33A/AD because matrix is structurally singular will be output. If, however, the user wants the LU decomposition of a structurally singular matrix (see Section 2.6) and sets the variable ICNTL(2)=0, then, in the event of singularity and a successful decomposition, IFLAG is returned with the value +1 and no message is output.
- 2 The matrix is numerically singular (it may also be structurally singular) with estimated rank given by INFO(3) (see Section 2.2.2). The message: Error return from MA33A/AD because matrix is numerically singular will be output. The user can choose to continue the decomposition even when a zero pivot is encountered by setting variable ICNTL(3)=0. If a singularity is encountered, IFLAG will then return with a value of +2, and no message is output if the decomposition has been completed successfully.
- 3 LIRN has not been large enough to continue with the decomposition. Should this happen, the message: ERROR Return from MA33A/AD because LIRN not big enough at stage ... in block ... with first row ... and last row ... is output. If the stage was zero, then variable INFO(4) (see Section 2.2) gives the length sufficient to start the decomposition on this block and the message: To continue set LIRN to at least ... is output. For a successful decomposition on this block the user should make LIRN slightly (say about $N/2$) greater than this value, which is that of INFO(4).
- 4 LICN has not been large enough to continue with the decomposition. Should this happen, the message: Error return from MA33A/AD because LICN not big enough at stage ... in block ... with first row ... and last row ... is output.

- 5 The decomposition has been completed but some of the **LU** factors have been discarded to create enough room in **A/ICN**. The variable **INFO(5)** (see Section 2.2) then gives the size that **LICN** should be to enable the factorization to be successful. Each time any of the **LU** factors are destroyed, the message: **LU decomposition destroyed to create more space** is output. If the user sets the variable **ICNTL(4)=1**, the subroutine will exit immediately instead of destroying any factors and continuing.
- 6 Both **LICN** and **LIRN** are too small. Termination has been caused by lack of space in **IRN** (see error **IFLAG= -3**), but already some of the **LU** factors in **A/ICN** have been lost (see error **IFLAG= -5**). **INFO(5)** gives the minimum amount of space required in **A/ICN** for decomposition up to this point and the message: **Error return from MA33A/AD LIRN and LICN too small** is output.

2.3.2 Error diagnostics for MA33B/BD

If MA33B/BD performs a successful decomposition, **IFLAG** will have a non-negative value on exit. Positive values indicate a warning; negative values indicate an error. Possible values for **IFLAG** are now described.

- I The routine has terminated at stage **I** for one of the following reasons.
 - (i) A zero pivot was found at stage **I**, where MA33A/AD found a nonzero pivot.
 - (ii) An entry was found in a part of row **I** which lay in a submatrix which was entirely zero after decomposition using MA33A/AD (see Section 2.6.1). With this error return the message: **Error return from MA33B/BD singularity detected in row ...** is output.
- +I Although the decomposition has been successful, this return indicates that there is a small pivot in row **I** with possible consequent stability problems. The variable **RINFO(1)** gives an idea of how severe this is. See the description of **RINFO(1)** for further details on this return.

2.3.3 Error diagnostics for MA33C/CD

There are no error returns from this subroutine.

2.4 Parameter usage summary

2.4.1 MA33I/ID

Output from MA33I/ID: **ICNTL**, **CNTL**.

2.4.2 MA33A/AD

Input by user: **N**, **ICN**, **A(IDISP(1):LICN)**, **LICN**, **LENR(N)**, **IDISP(2)**, **IP(N)**, **LIRN**, **U**, **ICNTL**, **CNTL**.

Unchanged by MA33A/AD: **N**, **LICN**, **LIRN**, **ICNTL**, **CNTL**.

Work arrays: **IRN(LIRN)**, **LENC(N)**, **IFIRST(N)**, **LASTR(N)**, **NEXTR(N)**, **LASTC(N)**, **NEXTC(N)**, **IPTR(N)**, **IPC(N)**.

Output from MA33A/AD: **ICN**, **A**, **LENR**, **LENRL(N)**, **IDISP**, **IP**, **IQ**, **IFLAG**, **INFO**, **RINFO**.

2.4.3 MA33B/BD

Input by user: **A(IDISP(1):IDISP(2))**, **ICNTL**, **CNTL**.

Input (from MA33A/AD): **N**, **ICN(IDISP(1):IDISP(2))**, **LICN**, **LENR(N)**, **LENRL(N)**, **IDISP(2)**.

Unchanged by MA33B/BD: **N**, **ICN**, **LICN**, **LENR**, **LENRL**, **IDISP**, **ICNTL**, **CNTL**.

Work arrays: **W(N)**, **IW(N)**.

Output from MA33B/BD: **A**, **IFLAG**, **RINFO**.

In any singular block of the matrix, MA33A/AD will permute a null or zero block to the end, viz:

$$\mathbf{P}_k \mathbf{A}_{kk} \mathbf{Q}_k = \begin{pmatrix} \mathbf{A}_1 & \mathbf{C} \\ \mathbf{D} & \mathbf{0} \end{pmatrix},$$

performing an LU decomposition on the first r_k rows and columns, where the $r_k \times r_k$ matrix \mathbf{A}_1 is nonsingular. The final rank INFO(3) equals $\sum_k r_k$ where the sum is over the diagonal blocks. For each row of \mathbf{A} in which a singularity is detected the corresponding entry in IQ is made negative.

2.6.2 MA33B/BD

MA33B/BD will produce the LU decomposition of a singular or rectangular matrix provided that a previous matrix was successfully decomposed by MA33A/AD and the same singularities are found by both the MA33B/BD and the MA33A/AD runs. Thus the zero block of the above figure must be zero after MA33B/BD and the block corresponding to the first r_k rows and columns must be nonsingular.

2.6.3 MA33C/CD

If MA33A/AD or MA33B/BD have been used to decompose a singular or rectangular matrix, then MA33C/CD will still perform the solution process using the information in array IQ which has a negative value for any row in which a singularity occurs. Because of the mode of operation of these earlier codes, these singularities will always occur at the ends of blocks. The maximum residual for the unsatisfied equations is given by the variable RINFO(3) (see Section 2.2.2).

3 GENERAL INFORMATION

Workspace:

MA33A/AD uses the INTEGER work arrays IRN of length LIRN ($> NZ$), LENC, IFIRST, LASTR, NEXTR, LASTC, NEXTC all of length N, and the INTEGER work arrays IPTR and IPC both of length N. Note that arrays LASTC and NEXTC are not referenced if ICNTL(5) (see Section 2.2.1) has been set to a value less than or equal to N.

MA33B/BD uses an INTEGER array IW of length N and a REAL (DOUBLE PRECISION in the D version) array W also of length N.

MA33C/CD uses a REAL (DOUBLE PRECISION in the D version) array W of length N.

Other routines called directly: MA33D/DD.

MA33D/DD need never be called directly by the user.

There are no subroutines called by MA33B/BD or MA33C/CD.

Input/output: Error messages are output on unit number ICNTL(1). The default value for ICNTL(1) is 6 and these messages can be suppressed by setting $ICNTL(1) \leq 0$ (see Section 2.2.1).

4 METHOD

MA33A/AD uses a sparse variant of Gaussian elimination to decompose each diagonal block into its LU factors.

MA33B/BD utilizes knowledge of the pivotal sequence and the sparsity structure of the LU factors from a previous call to MA33A/AD to factorize the new matrix into its LU factors by row Gauss elimination.

MA33C/CD performs simple forward elimination and back-substitution on each block in turn performing back-substitution on the off-diagonal parts when required. The code for the solution of the direct problem and its transpose are entirely separate.

A discussion of the design of these subroutines is given by Duff and Reid (ACM Trans Math Software 5 pp. 18-35, 1979) while fuller details of the implementation are given in Duff (Harwell Report R 8730, 1977). The additional

pivoting option in MA33A/AD and the use of drop tolerances (see CNTL(1), CNTL(2), ICNTL(5) and ICNTL(6) in Section 2.2) were added to the package after joint work by Duff, Reid, Schaumburg, Wasniewski and Zlatev.

5 EXAMPLE OF USE

In the example code shown below, we read in the entries of a sparse matrix, which is then sorted into row order. We should point out that MA33A/AD expects that the matrix entries are in row order, though the columns within each row need not be in any particular order. We then decompose the linear equations and finally solve for the unknowns.

```

PARAMETER( MAXN = 50 , MAXNZ = MAXN*MAXN ,
+          LICN = 500 , LIRN = 500 )
DOUBLE PRECISION A(LICN), W(MAXN), X(MAXN), U, CNTL(5), RINFO(5)
INTEGER ICN(LICN), IRN(LIRN), IPTR(MAXN), IPC(MAXN), LENC(MAXN),
+       LENR(MAXN), LENRL(MAXN), IFIRST(MAXN), LASTR(MAXN),
+       NEXTR(MAXN), LASTC(MAXN), NEXTC(MAXN), IP(MAXN),
+       IQ(MAXN), IDISP(2), LENOFF(MAXN), IFLAG,
+       IROW(MAXNZ), ICOL(MAXNZ), JPTR(MAXN), N, NZ, I,
+       ICNTL(10), INFO(10)
READ(5,*) N , NZ
IF ( ( N .GT. MAXN ) .OR. ( NZ .GT. MAXNZ ) ) THEN
  WRITE(6,"( /, A )" ) ' Error in the input data for N and/or NZ '
  STOP
END IF
C Read in data in any order into the temporary arrays IROW and ICOL.
READ(5,*) ( IROW(I) , ICOL(I) , A(I) , I=1,NZ )
JDISP = 0
C Sort the input matrix into row order (this is how MA33A/AD must have
C its input data).
CALL MC20AD( N, NZ, A, ICOL, JPTR, IROW, JDISP )
C JPTR(I) points to the start of the I-th row of the matrix stored in
C the array A(). Now setup the input data arrays for MA33A/AD.
DO 100 I=1,NZ
  ICN(I) = ICOL(I)
100 CONTINUE
  ISTART = LICN - NZ + 1
  IDISP(2) = ISTART
  IDISP(1) = NZ + 1
C LENR(I) holds the number of entries in the I-th row
DO 150 I=1,N-1
  LENR(I) = JPTR(I+1) - JPTR(I)
150 CONTINUE
  LENR(N) = NZ - JPTR(N) + 1
C MA33AD expects the block that it is to reduce to LU form
C to be contained in the ends of the arrays A() and ICN()
C so move the matrix we wish to have reduced to the end.
DO 155 I = 1 , NZ
  A(ISTART+I-1) = A(I)
  ICN(ISTART+I-1) = ICN(I)
155 CONTINUE
C Treat the input matrix A() as a single block, so set the permutation
C arrays IP() and IQ() to reflect this fact.
DO 200 I=1,N
  IP(I) = I
  IQ(I) = I
200 CONTINUE
  U = 0.1D0
C Remember to call MA33ID to initialize control parameters
CALL MA33ID( ICNTL, CNTL )
C Factorize the matrix

```

```

      CALL MA33AD( N, ICN, A, LICN, LENR, LENRL, IDISP, IP, IQ, IRN,
+               LIRN, LENC, IFIRST, LASTR, NEXTR, LASTC, NEXTC,
+               IPTR, IPC, U, IFLAG, ICNTL, CNTL, INFO, RINFO )
      IF (IFLAG .LT. 0) THEN
        WRITE(6,"( A, I2 )" ) ' Error return from MA33AD with IFLAG =', IFLAG
        STOP
      END IF
C  Read in the right-hand side vector
      READ(5,*) ( X(I), I=1,N )
      MTYPE = 1
      LENOFF(1) = -1
C  Solve the system
      CALL MA33CD( N, ICN, A, LICN, LENR, LENRL, LENOFF, IDISP, IP,
+               IQ, X, W, MTYPE, RINFO )
      WRITE(6,"( /, A, / )" ) ' The solution vector is:'
      WRITE(6,"( ' X(',I2,')= ', 1P, E13.5 )" ) ( I,X(I),I=1,N)
      END

```

If we wish to solve the following problem:

$$\begin{pmatrix} 1. & & & 4. \\ & 6. & 7. & 8. \\ 9. & & 11. & 12. \\ & 14. & & 16. \end{pmatrix} \mathbf{x} = \begin{pmatrix} 5. \\ 21. \\ 32. \\ 30. \end{pmatrix}$$

we might have as input

4	10	
4	4	16.
3	3	11.
2	2	6.
1	1	1.
4	2	14.
2	3	7.
3	1	9.
2	4	8.
3	4	12.
1	4	4.
5.		
21.		
32.		
30.		

and we would get the following output

The solution vector is:

```

X( 1)= 1.00000E-00
X( 2)= 1.00000E-00
X( 3)= 1.00000E+00
X( 4)= 1.00000E+00

```