



**Warning:** Subroutine MA32 has been superseded by subroutine MA42 which uses improved algorithms; the use of the latter routine is recommended. The superseded routine may be removed from later releases of the library.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 **solves one or more sets of sparse linear equations,  $\mathbf{Ax}=\mathbf{b}$** , by the frontal method, **optionally using disk storage** for the matrix factors. The equations may be input by the user in either of the two ways:

- (i) by elements in a finite-element calculation,
- (ii) by equations (matrix rows).

In both cases, the coefficient matrix and right hand side(s) are of the form

$$\mathbf{A} = \sum_{k=1}^m \mathbf{A}^{(k)}, \quad \mathbf{b} = \sum_{k=1}^m \mathbf{b}^{(k)}$$

In case (i), the summation is over finite elements.  $\mathbf{A}^{(k)}$  is nonzero only in those rows and columns which correspond to variables in the  $k$ -th element.  $\mathbf{b}^{(k)}$  is nonzero only in those rows which correspond to variables in element  $k$ .

In case (ii), the summation is over equations and  $\mathbf{A}^{(k)}$  and  $\mathbf{b}^{(k)}$  are nonzero only in row  $k$ .

In both cases, the user must supply for each  $k$  a list specifying which columns of  $\mathbf{A}$  are associated with  $\mathbf{A}^{(k)}$ , and arrays containing  $\mathbf{A}^{(k)}$  and  $\mathbf{b}^{(k)}$  in packed form. The nature of this packed form is defined more precisely in section 2.3.

A principal feature of this package is that it can solve large problems in a predetermined and relatively small amount of main store. At an intermediate stage of the solution,  $l$  say, the 'front' normally contains those variables associated with one or more of  $\mathbf{A}^{(k)}$ ,  $k=1,2,..,l$  which are also present in one or more of  $\mathbf{A}^{(k)}$ ,  $k=l+1,..,m$ , although for stability reasons it may include a few more variables from  $\mathbf{A}^{(k)}$ ,  $k=1,2,..,l$ . To keep the amount of main store low, it is important to order the  $\mathbf{A}^{(k)}$  so that the number of variables in the front (the 'front size') is small. For example, a very rectangular grid should be ordered pagewise parallel to the short side of the rectangle.

There is a facility for solving the equations  $\mathbf{A}^T \mathbf{x}=\mathbf{b}$  using the factors generated from the decomposition of  $\mathbf{A}$ .

This new version of MA32 (May 1983) should be upward compatible from the old version with two exceptions. The parameter NFRONT (MA32B/BD entry) is now an array of dimension two and the length of common block MA32P/PD is now 22 INTEGER variables. This version will run on the CRAY using blocked data files but there is a more efficient CRAY version using unblocked data files (see section 3.3).

**ATTRIBUTES** — **Version:** 1.0.0. **Types:** MA32A; MA32AD. **Calls:** I\_AMAX, \_DOT, \_AXPY, IM01, MC31, MC32. **Language:** There is also a CRAY 2 version. **Original date:** September 1980 (revised April 1983; May 1984) **Origin:** I.S. Duff, Harwell

## 2 HOW TO USE THE PACKAGE

The user must first call the initialization subroutine MA32I/ID, then for each element (or equation) must call the subroutine MA32A/AD to specify which variables are associated with it. Then, after all calls to MA32A/AD, MA32B/BD must be called for each element (or equation) to specify the nonzeros of  $\mathbf{A}^{(k)}$  and, optionally,  $\mathbf{b}^{(k)}$ . If  $\mathbf{b}^{(k)}$  are specified then MA32B/BD will solve the equations with right-hand side(s)  $\sum_{k=1}^m \mathbf{b}^{(k)}$ . If further systems with the same matrix are to be solved then MA32C/CD must be called, specifying the new right-hand side(s)  $\mathbf{b}$ . If systems of the form  $\mathbf{A}^T \mathbf{x} = \mathbf{b}$  are to be solved the user must first call MA32R/RD once only and then call MA32U/UD for each set of right-hand side(s).

### 2.1 Disk files and the initialization subroutine MA32I/ID

The frontal method is a variant of Gaussian elimination and involves the factorization

$$\mathbf{A} = \mathbf{P}\mathbf{L}\mathbf{U}\mathbf{Q}$$

where  $\mathbf{P}$  and  $\mathbf{Q}$  are permutation matrices,  $\mathbf{L}$  is a lower triangular matrix and  $\mathbf{U}$  is an upper triangular matrix. It is always necessary to store  $\mathbf{U}$  and  $\mathbf{Q}$  and it is necessary to store  $\mathbf{P}$  and  $\mathbf{L}$  if the whole factorization is to be held for rapid processing of further right-hand side(s) (MA32C/CD) or if subsequent solution of  $\mathbf{A}^T \mathbf{x} = \mathbf{b}$  is desired (MA32U/UD).

$\mathbf{P}\mathbf{L}$  and  $\mathbf{U}\mathbf{Q}$  require (on the IBM mainframe) files of length about  $(f+2.5)n$  and  $(f+3+nrhs)n$  double-precision variables, respectively, where  $f$  is the average front size (an estimate of which may be obtained from the geometry of the problem),  $n$  the order of the system, and  $nrhs$  the number of right-hand sides input to MA32B/BD. Normally these are temporary files allocated by MA32I/ID, but if the user wishes to preserve the factorization, the files must be explicitly allocated (see section 3.2). On the CRAY, the lengths should be  $(f+5)n$  and  $(f+5+nrhs)n$  CRAY 64-bit words respectively, and the user is advised to change the CRAY default buffer size (see section 3.3). MA32 does its own blocking so that all records written are of a fixed length, equal to that of an associated buffer. These buffers should be significantly longer than the maximum front size. Since logical records (corresponding to a row of  $\mathbf{U}\mathbf{Q}$  or a column of  $\mathbf{P}\mathbf{L}$ ) are not permitted to span the MA32 blocks, it is possible for data fragmentation to increase the total storage for the files. Thus, the user should be wary of calculating exact values for the lengths of  $\mathbf{P}\mathbf{L}$  and  $\mathbf{U}\mathbf{Q}$  based on the above formulae. If the buffers are made very long and the solution of  $\mathbf{A}^T \mathbf{x} = \mathbf{b}$  is not required, disk storage may be avoided.

If it is desired to use the factors to solve  $\mathbf{A}^T \mathbf{x} = \mathbf{b}$ , then files must be allocated to hold the factors  $(\mathbf{P}\mathbf{L})^T$  and  $(\mathbf{U}\mathbf{Q})^T$ . This allocation is discussed in section 2.5.

To specify the characteristics of the files and their associated buffers the user must make a call of the following form:

*The single precision version*

```
CALL MA32I(ISTRM,LENBUF,LENFLE,IFLAG)
```

*The double precision version*

```
CALL MA32ID(ISTRM,LENBUF,LENFLE,IFLAG)
```

ISTRM is an INTEGER array of length 2, which the user must set to specify the unit numbers of the files for  $\mathbf{U}\mathbf{Q}$  and  $\mathbf{P}\mathbf{L}$ , respectively. If either value is outside the range (1,99), then it is assumed that the user does not wish to have such a file. ISTRM is not altered by the subroutine. **Restriction:** ISTRM(1)  $\neq$  ISTRM(2) if they are in the range (1,99).

LENBUF is an INTEGER array of length 2, which the user must set to the main store buffer sizes associated with the files. A REAL (DOUBLE PRECISION in the D version) work-array of length LENBUF(1) + LENBUF(2) is required by MA32B/BD and MA32C/CD. If the user does not intend to call MA32C/CD or MA32U/UD, LENBUF(2) should be set to zero to indicate that  $\mathbf{P}$  and  $\mathbf{L}$  are not to be stored. A value for LENBUF(1) of  $10(maxf+nrhs+3)$  or greater is recommended where  $maxf$  is the maximum front size and  $nrhs$  the number of right-hand sides. If  $\mathbf{P}$  and  $\mathbf{L}$  are stored, then it is recommended that LENBUF(2) be at least as great as  $10(maxf+2.5)$ . On the CRAY, the

appropriate values are  $10(\maxf+nrhs+5)$  and  $10(\maxf+5)$  CRAY 64-bit words respectively. All read and write statements made by MA32 are of records of length  $LENBUF(I)$ ,  $I=1,2$ , so for efficiency these records should not be short. For our present 3350 disks, we recommend that  $LENBUF(1)$  (and  $LENBUF(2)$  if nonzero) should be set to the maximum of  $\maxf+nrhs+3$  and 2383 (the number of double precision words in a track). On the CRAY, a larger value is recommended to reduce the number of input/output requests. For large problems, a value of around 50 CRAY blocks (25600 CRAY words) has, in our experience, proved suitable.  $LENBUF$  is not altered by the subroutine. **Restriction:**  $LENBUF(1) > 0$ .

$LENFLE$  is an INTEGER array of length 2, which must be set to the lengths, in REAL (DOUBLE PRECISION in the D version) words, of the two files (see beginning of this section for advice on suitable lengths). If  $ISTRM(I)$  is not in the range (1,99), then  $LENFLE(I)$  is ignored.  $LENFLE$  is not altered by the subroutine.

$IFLAG$  is an INTEGER variable that need not be set by the user. On exit it has the value zero to indicate successful completion. A negative value indicates an error condition (see section 2.7).

## 2.2 Specification of which variables belong in each element or equation

A call of the following form must be made for each element (case (i)) or each equation (case (ii)):-

*The single precision version*

```
CALL MA32A(NVAR, IVAR, NDF, LAST, LENLST, IFLAG)
```

*The double precision version*

```
CALL MA32AD(NVAR, IVAR, NDF, LAST, LENLST, IFLAG)
```

$NVAR$  is an INTEGER variable which must be set by the user to the number of variables in the element or equation currently being input. It is unchanged by the subroutine.

$IVAR$  is an INTEGER array of length  $NVAR$  which must be set by the user to contain the indices of variables associated with the element or equation being input. These indices need not necessarily be in increasing order but must be distinct. It is not altered by MA32A/AD.

$NDF$  is an INTEGER variable which need not be set by the user. On output it will be set to the largest integer so far used to index a variable. It must not be changed by the user between calls to MA32A/AD nor prior to subsequent calls to MA32B/BD.

$LAST$  is an INTEGER array of length  $LENLST$  which need not be set by the user. On output, the first  $NDF$  entries indicate in which element or equation each variable appears for the last time. These  $NDF$  entries must not be changed between calls to MA32A/AD nor prior to subsequent calls to MA32B/BD.

$LENLST$  is an INTEGER variable which must be set by the user to the length of array  $LAST$ .  $LENLST$  must be greater than the largest integer used to index a variable. It is not altered by MA32A/AD.

$IFLAG$  is an INTEGER variable which must be set to 0 before the first call to MA32A/AD for a particular problem. On exits it has the value +1 to indicate success. Negative values indicate an error has been detected (see section 2.7).  $IFLAG$  must not be altered by the user between calls to MA32A/AD.

## 2.3 To factorize A and optionally solve $Ax=b$

A call of the following form must be made for each element (case (i)) or each equation (case (ii)). The elements/equations must be input in exactly the same order as when MA32A/AD was called.

Note that all the calls to MA32A/AD for a particular problem must be completed before the calls to MA32B/BD for that problem.

*The single precision version*

```
CALL MA32B(NVAR, IVAR, NDF, LAST, NMAXE, AVAR, NRHS, RHS, LRHS,
*          NFRONT, BUFR, LBUFR, LW, W, LIW, IW, IFLAG)
```

*The double precision version*

```
CALL MA32BD(NVAR,IVAR,NDF,LAST,NMAXE,AVAR,NRHS,RHS,LRHS,
*          NFRONT,BUFR,LBUFR,LW,W,LIW,IW,IFLAG)
```

NVAR, IVAR, NDF and LAST are as in the corresponding calls to MA32A/AD. IVAR is altered by MA32B/BD; NVAR and NDF are not altered by MA32B/BD. On exit from all calls to MA32B/BD, LAST will be unchanged. Between calls to MA32B/BD, LAST is used as a work array and will be changed.

NMAXE is an INTEGER variable which must be set by the user to the first dimension of the arrays AVAR and RHS. If it is 1, then input by equations rather than elements is assumed. It is unchanged by the subroutine.

AVAR is a REAL (DOUBLE PRECISION in the D version) array of dimensions NMAXE by NVAR. It must be set by the user to contain  $\mathbf{A}^{(k)}$  in packed form. By packed form, we mean that the value AVAR(I,J) is the contribution to entry IVAR(I), IVAR(J) of the coefficient matrix from the element or equation currently being input to MA32B/BD,  $I=1,2,\dots,\min(\text{NMAXE},\text{NVAR})$ ,  $J=1,2,\dots,\text{NVAR}$ . Contributions to the same entry from different elements or equations are summed. (Note that the array AVAR is not required by MA32A/AD). It is altered by MA32B/BD. **Restriction:**  $\text{NVAR} \leq \text{NMAXE}$  (if  $\text{NMAXE} > 1$ ).

NRHS is an INTEGER variable which must be set by the user to the number of right-hand sides. Any value less than zero is treated as zero. It is not altered by the subroutine.

RHS is a REAL (DOUBLE PRECISION in the D version) array of dimensions NMAXE by LRHS. The user must set it to contain  $\mathbf{b}^{(k)}$  in packed form. By packed form, we mean that the value RHS(I,J) is the contribution to component IVAR(I) of the J-th right-hand side from the element or equation currently being input to MA32B/BD,  $I=1,2,\dots,\min(\text{NMAXE},\text{NVAR})$ ,  $J=1,2,\dots,\text{NRHS}$ . Contributions to the same component from different elements are summed. It is not referenced if  $\text{NRHS} \leq 0$ . If  $\text{NRHS} > 0$ , it is altered by the subroutine.

LRHS is an INTEGER variable which must be set by the user to the second dimension of RHS. It is not altered by the subroutine. **Restriction:**  $\text{LRHS} \geq \text{MAX}(1,\text{NRHS})$ .

NFRONT is an INTEGER array of length 2. NFRONT(I),  $I=1,2$  must be set to the greatest permissible number of rows and columns respectively allowed in the front. It will have a crucial effect on the amount of storage required, (see LW and LIW below). A suitable size is problem dependent. If input is by elements ( $\text{NMAXE} > 1$ ), NFRONT(2) will be set equal to NFRONT(1) by MA32B/BD. NFRONT will be unaltered unless there is insufficient space for the decomposition. If this happens ( $\text{IFLAG} = -12$ ) NFRONT will be a lower bound on the dimensions of the frontal matrix necessary for a successful decomposition. In order that the routine can choose pivots to avoid numerical instability, the user should set NFRONT somewhat larger than these values. (See also error returns 4, -12 and -18 in section 2.7).

BUFR is a REAL (DOUBLE PRECISION in the D version) array of length LBUFR. It need not be set by the user and must not be altered between calls to MA32B/BD. If the user is not using auxiliary storage (ISTRM(I),  $I=1,2$  not in the range (1,99) on entry to MA32I/ID), then it must not be altered between calls to MA32B/BD and a subsequent call to MA32C/CD.

LBUFR is an INTEGER variable which must be set by the user to the dimension of array BUFR. It must be at least as great as  $\text{LENBUF}(1) + \text{LENBUF}(2)$  as input to MA32I/ID.

LW is an INTEGER variable which indicates the amount of real workspace available. It must be greater than or equal to the maximum of  $\text{NFRONT}(1) * (\text{NFRONT}(2) + \text{LRHS} + 1) + \text{NFRONT}(2) + \text{LRHS}$  and  $\text{NRHS} * \text{NDF}$ . It is not altered by MA32B/BD.

W is a REAL (DOUBLE PRECISION in the D version) array of length LW used as workspace and to output the solution. It must be preserved between calls to MA32B/BD and, if  $\text{NRHS} > 0$ , on exit  $W(I + (J-1) * \text{NDF})$  will hold the I-th component of the solution to system J.

LIW is an INTEGER variable which must be set by the user to indicate the amount of integer workspace available. It must be greater than or equal to  $2 * \text{NFRONT}(1) + 4 * \text{NFRONT}(2)$ . It is not altered by MA32B/BD.

IW is an INTEGER array of length LIW used as workspace by MA32B/BD. It must be preserved between calls to MA32B/BD.

IFLAG is an INTEGER variable which must be set by the user to zero before the first call of MA32B/BD for a particular problem. If the user has just called MA32A/AD, he should note that, on successful completion of these calls, the corresponding parameter has value +1 and so *must be reset* before calling MA32B/BD. IFLAG will have value 1 on successful output from intermediate calls to MA32B/BD and have value 0 on successful exit from the final call. Negative values indicate a fatal error. Values greater than 1 are associated with a warning or non-terminal error. (see section 2.7).

## 2.4 To solve further systems of equations of the form $Ax = b$

*The single precision version*

```
CALL MA32C(NRHS, LB, B, X, BUFR, LBUFR, IW, LIW, IFLAG)
```

*The double precision version*

```
CALL MA32CD(NRHS, LB, B, X, BUFR, LBUFR, IW, LIW, IFLAG)
```

NRHS is an INTEGER variable which must be set by the user to the number of new systems which are to be solved. It is not altered by the subroutine. **Restriction:**  $NRHS \geq 1$ .

LB is an INTEGER variable which must be set by the user to the first dimensions of B and X. It must be at least as large as the number of variables in the system (NDF as output from MA32A/AD). It is not altered by the subroutine.

B is a REAL (DOUBLE PRECISION in the D version) array of dimensions LB by NRHS which must be set by the user so that  $B(I, J)$  is the I-th component of the right-hand side for the J-th system. It is altered by the subroutine.

X is a REAL (DOUBLE PRECISION in the D version) array of dimension LB by NRHS. On exit from the subroutine  $X(I, J)$  will hold the I-th component of the solution to system J.

BUFR is a REAL (DOUBLE PRECISION in the D version) array of length LBUFR. If auxiliary storage devices have been used (ISTRM(I),  $I=1, 2$  in the range (1,99) on input to MA32I/ID) then this is just a workspace array, otherwise it must be the same array as that of the same name in the previous calls to MA32B/BD. In this latter case, it will not be altered by MA32C/CD.

LBUFR is an INTEGER variable which must be set by the user to the size of array BUFR. It must be at least as large as  $LNUFR(1) + LNUFR(2)$  as input to MA32I/ID. It is not altered by MA32C/CD.

IW is an INTEGER array of length LIW which is used as workspace by MA32C/CD.

LIW is an INTEGER variable which must be set by the user to a value at least as great as the maximum of  $NFRONT(1)$  and  $NFRONT(2)$  (see call to MA32B/BD). It is not altered by MA32C/CD.

IFLAG is an INTEGER variable which need not be set by the user. On exit, IFLAG will equal zero if the run has been successful. Negative values indicate an error has been detected (see section 2.7).

## 2.5 To solve systems of equations of the form $A^T x = b$

The user must call MA32R/RD once only after all calls to MA32B/BD and before calling MA32U/UD. Thereafter MA32U/UD may be called to solve any sets of equations whose coefficient matrix is  $A^T$ .

Subroutine MA32R/RD creates the files holding the factors of the transpose matrix from the files holding the factors of  $A$  generated by MA32B/BD.

*The single precision version*

```
CALL MA32R( ISTRMT, LENBUT, LENFLT, BUFR, LBUFR, IW, LIW, IFLAG)
```

*The double precision version*

```
CALL MA32RD( ISTRMT, LENBUT, LENFLT, BUFR, LBUFR, IW, LIW, IFLAG)
```

ISTRMT is an INTEGER array of length 2, which the user must set to specify the unit numbers of the files for the factors of  $\mathbf{A}^T$ . ISTRMT is not altered by the subroutine. **Restrictions:**  $1 \leq \text{ISTRMT}(I) \leq 99$ ,  $I=1, 2$ . ISTRMT(1), ISTRMT(2) must differ from each other and from the ISTRM(1), ISTRM(2) most recently passed to MA32I/ID.

LENBUT is an INTEGER array of length 2, which the user must set to the main store buffer sizes associated with the files. A workspace array, BUFR, of length the maximum of  $\text{LENBUT}(1) + \text{LENBUF}(1)$  and  $\text{LENBUT}(2) + \text{LENBUF}(2)$ , where  $\text{LENBUF}(I)$ ,  $I=1, 2$  are as input to MA32I/ID, is required by MA32R/RD. We recommend a value for  $\text{LENBUT}(1)$  of  $10(1.5 \times \text{maxf} + 2)$  or greater, where *maxf* is the maximum front size, and a value for  $\text{LENBUT}(2)$  of  $10(\text{maxf} + 2)$  or more. LENBUT is not altered by the subroutine. **Restriction:**  $\text{LENBUT}(I) > 0$ ,  $I=1, 2$ .

LENFLT is an INTEGER array of length 2, which must be set to the lengths (in double precision words) of the files for the factors of the transpose. Values for LENFLT equal to those for LENFLE in the call to MA32I/ID will usually suffice. LENFLT is not altered by the subroutine. **Restriction:**  $\text{LENFLT}(I) \geq \text{LENBUT}(I)$ ,  $I=1, 2$ .

BUFR is a REAL (DOUBLE PRECISION in the D version) array of length LBUFR which is used as workspace by MA32R/RD.

LBUFR is an INTEGER variable which must be set by the user to the size of array BUFR. It must be at least as large as the maximum of  $\text{LENBUT}(1) + \text{LENBUF}(1)$  and  $\text{LENBUT}(2) + \text{LENBUF}(2)$  where  $\text{LENBUF}(I)$  and  $\text{LENBUT}(I)$ ,  $I=1, 2$  are as input to MA32I/ID and MA32R/RD. It is not altered by MA32R/RD.

IW is an INTEGER array of length LIW which is used as workspace by MA32R/RD.

LIW is an INTEGER variable which must be set by the user to a value at least as great as the maximum of NFRONT(1) and NFRONT(2) (as input to MA32B/BD). It is not altered by MA32R/RD.

IFLAG is an INTEGER variable which need not be set by the user. On exit, IFLAG will equal zero if the run has been successful. Negative values indicate an error has been detected (see section 2.7).

After a call to MA32R/RD, the user must call MA32U/UD to solve systems of equations  $\mathbf{A}^T \mathbf{x} = \mathbf{b}$

*The single precision version*

```
CALL MA32U(NRHS, LB, B, X, BUFR, LBUFR, IW, LIW, IFLAG)
```

*The double precision version*

```
CALL MA32UD(NRHS, LB, B, X, BUFR, LBUFR, IW, LIW, IFLAG)
```

NRHS is an INTEGER variable which must be set by the user to the number of new systems to be solved. It is not altered by the subroutine. **Restriction:**  $\text{NRHS} \geq 1$ .

LB is an INTEGER variable which must be set by the user to the first dimensions of B and X. It must be at least as large as the number of variables in the system (NDF as output from MA32A/AD). It is not altered by the subroutine.

B is a REAL (DOUBLE PRECISION in the D version) array of dimensions LB by NRHS which must be set by the user so that  $B(I, J)$  is the I-th component of the right-hand side for the J-th system. It is altered by the subroutine.

X is a REAL (DOUBLE PRECISION in the D version) array of dimension LB by NRHS. On exit from the subroutine  $X(I, J)$  will hold the I-th component of the solution to system J.

BUFR is a REAL (DOUBLE PRECISION in the D version) array of length LBUFR which is used as workspace by MA32U/UD.

LBUFR is an INTEGER variable which must be set by the user to the size of array BUFR. It must be at least as large as the maximum of LENBUT(1) and LENBUT(2) as input to MA32R/RD. It is not altered by MA32U/UD.

IW is an INTEGER array of length LIW which is used as workspace by MA32U/UD.

LIW is an INTEGER variable which must be set by the user to a value at least as great as the maximum of NFRONT(1) and NFRONT(2) (as input to MA32B/BD). It is not altered by MA32U/UD.

IFLAG is an INTEGER variable which need not be set by the user. On exit, IFLAG will equal zero if the run has been successful. Negative values indicate an error has been detected (see section 2.7).

## 2.6 Common blocks

Common blocks are used to preserve and communicate values between calls to the subroutines and between the various auxiliary subroutines used by the package. There are two common blocks which may be of interest to the user.

The first common block provides information to control the decomposition, to direct the output messages, and to return information to the user. All the input parameters have default values set by a block data subprogram. To reference this common block, the user must include in the calling program the COMMON statement:

*The single precision version*

```
COMMON/MA32M/ TOL, ALPHA, LP, MP, IAUTO, ISRCH, INFD(10), INFP(6)
```

*The double precision version*

```
COMMON/MA32MD/ TOL, ALPHA, LP, MP, IAUTO, ISRCH, INFD(10), INFP(6)
```

TOL is a REAL (DOUBLE PRECISION in the D version) variable with default value zero. The matrix will be declared singular if the largest element in any column during decomposition is less than or equal to TOL.

ALPHA is a REAL (DOUBLE PRECISION in the D version) variable with default value 0.1. Unless IAUTO is operative (see IAUTO), an element of the frontal matrix will only be considered suitable for use as a pivot if it is at least as large as |ALPHA| times the largest element in its column. If ALPHA is negative, the static condensation option will not be employed. In that case, the arrays IVAR and AVAR will not be altered by MA32B/BD.

LP is an INTEGER variable with default value 6. This is the stream number for error messages. Printing of these messages is suppressed if LP is set to 0.

MP is an INTEGER variable with default value 6. This is the stream on which warning messages are output. Printing of these messages is suppressed if MP is set to 0.

IAUTO is an INTEGER variable with default value 0. If it is set by the user to have a positive value, then, when the number of potential pivot columns is greater than or equal to IAUTO, an elimination will be performed even if the best pivot candidate does not satisfy the threshold criterion determined by ALPHA. Under such a condition we say IAUTO is operative.

ISRCH is an INTEGER variable with default value 0. If IAUTO is positive and ISRCH is also, then only ISRCH potential pivot columns will be searched for a pivot when IAUTO is operative. The best pivot candidate (largest relative to other nonzeros in its column) from these ISRCH columns will then be used as a pivot.

INFD is an INTEGER array of length 10 which, on exit from the final call to MA32B/BD, provides the user with information on the decomposition

```
INFD(1): Number of variables in the problem.
INFD(2): Number of nonzeros in UQ (see section 2.1)
INFD(3): Total storage for UQ (in DOUBLE PRECISION words)
INFD(4): Number of nonzeros in PL (see section 2.1)
INFD(5): Total storage for PL (in DOUBLE PRECISION words)
INFD(6): Number of operations (each operation is a multiplication and an addition) in the innermost loop.
```

This count includes any operations performed during static condensation.

INFD(7): Maximum front-width (rows).

INFD(8): Maximum front-width (columns). For an element entry, this value will equal INFD(7).

INFD(9): Number of buffers output to **UQ** file.

INFD(10): Number of buffers output to **PL** file.

INFP is an INTEGER array of length 6 which, on exit from all calls to MA32B/BD, provides the user with information on the pivot selection process.

INFP(1): Number of columns searched during pivot searches.

INFP(2): Number of nonzeros tested for stability as pivots.

INFP(3): Number of nonzeros accessed during pivot selection process.

INFP(4): Number of pivots chosen which did not satisfy the threshold criterion based on the value of |ALPHA|.

INFP(5): Number of static condensations performed. These are eliminations performed within an individual element/equation when a variable appears only in that single element/equation.

INFP(6): Potential number of static condensations. This may be greater than INFP(5) because numerical considerations may prevent us from eliminating all internal variables immediately.

The second common block will, on successful completion, contain the determinant of the matrix.

#### *The single precision version*

```
COMMON/MA32Y/ DET, IDET
```

#### *The double precision version*

```
COMMON/MA32YD/ DET, IDET
```

DET is a REAL (DOUBLE PRECISION in the D version) variable. On successful exit from MA32B/BD, DET will be set to the natural logarithm of the modulus of the determinant of the matrix.

IDET is an INTEGER variable. On successful exit from MA32B/BD, IDET will be set to the sign of the determinant of the matrix.

There are three other common blocks used by this package. In some computing environments, the user is required to define these common blocks and their lengths in his main program, although this is not the case for the IBM mainframe. Common block MA32N/ND must remain unchanged between a call to MA32I/ID and subsequent calls to MA32B/BD, or MA32C/CD. Common blocks MA32P/PD and MA32Q/QD must only remain unchanged between individual calls to MA32A/AD and between individual calls to MA32B/BD.

The common blocks are

#### *The single precision version*

```
COMMON/MA32N/ IDUMMY(10)
```

```
COMMON/MA32P/ JDUMMY(22)
```

```
COMMON/MA32Q/ DUMMY(2), KDUMMY(6), LDUMMY
```

```
COMMON/MA32X/ MDUMMY(10)
```

#### *The double precision version*

```
COMMON/MA32ND/ IDUMMY(10)
```

```
COMMON/MA32PD/ JDUMMY(22)
```

```
COMMON/MA32QD/ DUMMY(2), KDUMMY(6), LDUMMY
```

```
COMMON/MA32XD/ MDUMMY(10)
```

Where

IDUMMY is an INTEGER array.

JDUMMY is an INTEGER array.

DUMMY is a REAL (DOUBLE PRECISION in the D version) array.

KDUMMY is an INTEGER array.

LDUMMY is a LOGICAL variable.

MDUMMY is an INTEGER array.

and the array lengths are as explicitly dimensioned.

## 2.7 Error diagnostics

On successful completion, the subroutines in this package will exit with the parameter IFLAG set to zero or +1. Other values for IFLAG and the reasons for them are given below. In each case the associated message is output only if the appropriate stream number (LP for errors, MP for warnings, see section 2.6) is positive.

We list possible error returns and the reasons for them below. A self-explanatory message is, in each case, output on unit LP.

- 1 LENLST is less than or equal to zero on entry to MA32A/AD (MA32A/AD entry only).
- 2 Non-positive number of variables in element or equation (MA32A/AD and MA32B/BD entries). This error is also raised if the number of variables (NVAR) is greater than the first dimension of the real arrays supplying the element information (NMAXE) (MA32B/BD element entry only i.e. NMAXE > 1).
- 3 Variable index out of range in element or equation (MA32A/AD and MA32B/BD entries).
- 4 Duplicate occurrences of same variable found in the one element or equation (MA32A/AD entry only).
- 5 Defined length of in-core buffer is too small. LBUFR must be set to a value of at least LENBUF(1) + LENBUF(2) as input to MA32I/ID (MA32B/BD, MA32C/CD entries), to a value of at least the maximum of LENBUT(1) and LENBUT(2) as input to MA32R/RD (MA32U/UD entry), or to a value of at least the maximum of LENBUT(1)+LENBUF(1) and LENBUT(2)+LENBUF(2) (MA32R/RD entry). This error is also provoked by an attempt to solve for subsequent right-hand sides when factors of **L** have not been stored (MA32C/CD entry only).
- 6 Defined length of real work array insufficient on entry to MA32B/BD (MA32B/BD entry only).
- 7 Defined length of integer work array insufficient on entry to MA32B/BD (MA32B/BD entry only).
- 8 First dimension of real arrays supplying element or equation information (NMAXE) is non-positive or was 1 and has now been changed. (MA32B/BD entry only).
- 9 User has changed number of right-hand sides between calls to MA32B/BD. (MA32B/BD entry only).
- 10 LRHS < max(1, NRHS). Array storage only allocated for LRHS right-hand sides but solution for NRHS requested on entry to MA32B/BD. (MA32B/BD entry only).
- 11 Non-positive dimensions (NFRONT) for in-core frontal matrix or space allocated to it has been changed between calls to MA32B/BD. (MA32B/BD entry only).
- 12 Not enough space has been allocated to the in-core frontal matrix to permit decomposition. However, a symbolic decomposition has been performed and a lower bound on the space required is given in the output message and in parameter NFRONT. See also error + 4. (MA32B/BD entry only).
- 13 Variable appears again after it has been fully summed. (MA32B/BD entry only)
- 14 Singularity detected in matrix during decomposition. (MA32B/BD entry only).
- 15 Buffer size too small to hold the single pivot row or column currently generated. (MA32B/BD and MA32R/RD entries).
- 16 No auxiliary storage requested and in-core buffer not long enough to hold factors. The message also gives

information on how much storage is necessary to prevent this failure on a subsequent run on identical data. This is also given by  $INFD(3) + INFD(5)$  from common block MA32M/MD (see section 2.6) Of course, for subsequent runs it may be better to use auxiliary storage. See also error +5. (MA32B/BD entry only).

- 17 Insufficient space has been allocated to the direct access data sets to permit a successful decomposition. The space necessary for subsequent success is output and is also given by common block variables  $INFD(9)$  and  $INFD(10)$  (section 2.6) which give the space which must be allocated to files  $ISTRM(1)$  and  $ISTRM(2)$  respectively. To avoid this error on a subsequent run on the same data, the user must set  $LENFLE(I)$  to be at least as great as  $LENBUF(I)$  times these values,  $I=1,2$ . See also error +6. (MA32B/BD entry only). This error is also raised if there is insufficient space in files  $ISTRMT(1)$  or  $ISTRMT(2)$  when setting up the files for the solution of transpose equations. (MA32R/RD entry only).
- 18 Insufficient core allocated to the in-core frontal matrix.  $NFRONT$  should have values greater than  $LW$  as input to MA32B/BD and perhaps much greater (see also error -12). (MA32B/BD entry only).
- 19 Request to solve for non-positive number of right-hand sides. (MA32C/CD and MA32U/UD entries only).
- 20  $LENBUF(1) \leq 0$  or  $LENBUF(I) > LENFLE(I)$ ,  $I=1,2$  on entry to MA32I/ID. (MA32I/ID entry).  
 $LENBUT(I) \leq 0$  or  $LENBUT(I) > LENFLT(I)$ ,  $I=1,2$  on entry to MA32R/RD. (MA32R/RD entry).
- 21 Stream numbers for **PL** and **LQ** files equal (i.e.  $ISTRM(1) = ISTRM(2)$ ). (MA32I/ID entry only)
- 22 Error in Fortran OPEN statement ... usually caused by opening the same file with different record length. Iostat parameter printed. ( MA32I/ID and MA32R/RD entries).
- 23 Attempt to solve transpose equations when factors not stored on auxiliary storage. (MA32R/RD entry only).
- 24 Stream numbers of files for factors of transposed matrix are either out of the range 1 to 99, or are equal to each other or to one of the stream numbers for the factors themselves (i.e.  $ISTRMT(1) = ISTRMT(2)$  or  $ISTRMT(I) = ISTRM(J)$ ,  $I, J=1, 2$ ). (MA32R/RD entry only).
- 26 Error detected in direct access read. Iostat value is printed.
- 27 Error detected in direct access write. Iostat value is printed.

There are only two warning messages. These return a positive value of IFLAG. They are:

- +2 Change in the first dimension of element or equation arrays during a sequence of calls to MA32B/BD. (MA32B/BD entry only).
- +3 We have chosen  $INFP(4)$  (see common block MA32M/MD, section 2.6) pivots not satisfying the numerical tolerance condition. (MA32B/BD entry only).

There are three error returns associated with a positive value of IFLAG. In each case, the user is encouraged to continue calling MA32B/BD. At the end of the sequence of calls, information to enable success on a subsequent run will be output although the decomposition will not have been completed. Note that if the user exits immediately before completing the sequence of calls, the array LAST (see section 2) is probably corrupted.

- +4 Insufficient space has been allocated to the in-core frontal matrix. If the user continues to call MA32B/BD, a lower bound will be obtained for the size required (error -12) unless the workspace allocation is far too small (error -18). (MA32B/BD entry only).
- +5 No auxiliary storage requested and in-core buffer not long enough to hold factors. If the user completes his sequence of calls to MA32B/BD, the amount of storage required for a successful run will be output (error -16). (MA32B/BD entry only).
- +6 Insufficient space has been allocated to the direct access data sets. If the user completes his sequence of calls to MA32B/BD, the amount of space required for subsequent success will be given (error -17). (MA32B/BD entry only).

### 3 GENERAL INFORMATION

#### 3.1 Summary of information.

**Use of common:** The subroutines use common areas MA32M/MD, MA32N/ND, MA32P/PD, MA32Q/QD, MA32X/XD and MA32Y/YD.

**Other routines called directly:** Subroutines internal to the package are MA32D/DD, MA32E/ED, MA32F/FD, MA32G/GD, MA32H/HD, MA32J/JD, MA32K/KD, MA32L/LD, MA32R/RD, MA32S/SD, MA32T/TD, MA32U/UD, MA32V/VD, MA32W/WD. In the CRAY version, the subroutines call a CAL routine in SCILIB, ISAMAX, and the CAL routines IM01AD, MC31AD and MC32AD.

**Workspace:** provided by the arrays:

BUFR(LBUFR) (MA32B/BD, MA32C/CD, MA32R/RD and MA32U/UD. If, however, auxiliary files are not used, BUFR holds the matrix factors for entries MA32B/BD and MA32C/CD).

W(LW) (MA32B/BD only).

IW(LIW) (MA32B/BD, MA32C/CD, MA32R/RD and MA32U/UD)

**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 variables LP and MP (see section 2.6). Stream LP is used for error messages and stream MP for warnings.

**Restrictions:**

NVAR > 0,

LENLST > 0,

NMAXE > 0

NVAR ≤ NMAXE (if NMAXE > 1, MA32B/BD entry only),

NRHS > 0 (MA32C/CD and MA32U/UD entries only),

NFRONT(I) > 0, I=1,2 (if NMAXE > 1, NFRONT(2) need not be defined),

LRHS ≥ MAX0(1, NRHS),

LENBUF(1) > 0,

LENBUT(I) > 0, I=1,2,

ISTRM(1) ≠ ISTRM(2) if both in the range (1,99)

ISTRMT(1) ≠ ISTRMT(2),

1 ≤ ISTRMT(I) ≤ 99, I=1,2,

ISTRMT(I) ≠ ISTRM(J) I, J=1, 2,

LBUFR ≥ LENBUF(1) + LENBUF(2) (MA32B/BD, MA32C/CD),

LBUFR ≥ max{LENBUF(1) + LENBUT(1), LENBUF(2) + LENBUT(2)} (MA32R/RD),

LBUFR ≥ max{LENBUT(1), LENBUT(2)} (MA32U/UD),

LW ≥ maximum of NFRONT(1) \* (NFRONT(2) + LRHS + 1) + LRHS + NFRONT(2) and NRHS \* NDF,

LIW ≥ 2 \* NFRONT(1) + 4 \* NFRONT(2).

**Portability:** There is also a CRAY version.

#### 3.2 Storage of files of factors on IBM mainframe.

If the user has not requested the use of auxiliary storage, then the factors will be found in array BUFR at the end of the sequence of calls to MA32B/BD. Otherwise the subroutine will output **PL** and **UQ** (see section 2.1) as files on

Fortran streams ISTRM(2) and ISTRM(1) respectively. Similarly, if MA32R/RD is called prior to solving  $\mathbf{A}^T \mathbf{x} = \mathbf{b}$ , the subroutine will output  $(\mathbf{PL})^T$  and  $(\mathbf{UQ})^T$  (see section 2.5) as files on Fortran streams ISTRMT(2) and ISTRMT(1) respectively. The following dd card must be included for each file (if the files are only being used temporarily UNIT=DISK can be replaced by UNIT=TEMP).

```
//G.FT<nI>F001 DD DISP=(NEW,CATLG),DSN=<dataset_name>,
// UNIT=DISK,SPACE=(<ll>,<bl>,RLSE),
// DCB=(BUFNO=1,RECFM=F,BLKSIZE=1600)
```

where (the values for file **UQ** are given in parenthesis)

<nI> is the stream number for the file (=ISTRM(1))

<ll> is the length of the buffer in bytes (=8\*LENBUF(1))

<bl> is the number of buffers output (=LENFLE(1)/LENBUF(1))

Note that, when subsequently using these files in future runs, the user must call MA32I/ID for both the solution of  $\mathbf{Ax} = \mathbf{b}$  and  $\mathbf{A}^T \mathbf{x} = \mathbf{b}$  before calling MA32C/CD or MA32U/UD so that appropriate parameters in internal common blocks are initialized.

### 3.3 The use of MA32 on the CRAY.

For the CRAY version, the inner loops have been replaced by CRAY Assembler Language code. The CAL code is invoked by calls to the library routines IM01AD, MC31AD and MC32AD. The main inner loop runs at over 300 Megaflops.

### 3.4 The use of the Fortran 77 version.

Standard Fortran direct access has been used.

When running this version on the IBM, the data sets must be allocated using DD cards of the form shown in section 3.2 since dynamic allocation is not permitted on the IBM from standard Fortran.

Dynamic allocation is possible on the CRAY so no files need be ASSIGNED in the JCL. If, however, a file is ASSIGNED (for example it may be needed later), then it must be blocked since such a structure is required when using standard Fortran direct access on the CRAY. This is the default although for efficiency it may be advisable to increase the buffer size to be comparable with the buffer size being used for the factors.

## 4 METHOD

The method used is a modification of the unsymmetric frontal scheme of Hood (1976). Our code has been developed from work by Cliffe et al (1978).

The equations are assembled into an in-core frontal matrix either an element at a time (see section 1) or an equation at a time. A variable which has appeared for the last time (i.e. does not occur in future elements or equations) is available for use as a pivot in Gaussian elimination.

Eliminations are performed whenever a variable in this frontal matrix is available for use as a pivot and satisfies a numerical tolerance. The pivot row and, optionally, the pivot column are output to an in-core buffer and thence, if requested, to auxiliary storage. In order to prevent the amount of in-core storage required becoming too large, the user should try to order his elements so that the same variable does not occur in different elements which are widely apart in the ordering. Thus, for example, in a narrow pipe geometry, elements should be ordered across the cross-section of the pipe rather than along its length. From a consideration of this geometry suitable values for NFRONT can be obtained. See Duff (1981,1983) for more details of this package.

**References.**

Cliffe, K.A., Jackson, C.P., Rae, J. and Winters, K.H. (1978) Finite element flow modelling using velocity and pressure variables. Report AERE R.9202.

Duff, I.S. (1981) MA32 – A package for solving sparse unsymmetric systems using the frontal method. Report AERE R.10079.

Duff, I.S. (1983) Enhancements to the MA32 package for solving sparse unsymmetric equations. Report AERE R.11009.

Hood, P. (1976) Frontal solution program for unsymmetric matrices. *Int. J. Num. Meth. Engng* **10** pp 379-399.

**5 EXAMPLE OF USE**

We give a sketch of the code required to solve a set of equations using this package when input is by elements (figure 5.1) and when input is by equations (figure 5.2). In each case this is not an example of a full working program but rather an indication of how the routines might be called in practice.

For the element case (figure 5.1), we wish to solve for one right hand side at the same time as the decomposition and do not wish to retain factors for subsequent solution. In the equation case (figure 5.2), we do not supply right hand sides with the equation but thereafter we want to solve for three right hand sides.

```

      INTEGER ISTRM(2),LENBUF(2),LENFLE(2),IVAR(20),LAST(2000),IW(300)
      INTEGER NFRONT(2)
      DOUBLE PRECISION AVAR(20,20),RHS(20,1),BUFR(1600),
      *W(3000),DUM
      COMMON/MA32MD/ DUM(2),IDUM(4),INFD(10),INFP(6)
C INITIALIZE BOUNDS ON ARRAYS AND PROBLEM SIZE
      LENLST=2000
      NMAXE=20
      NRHS=1
      LRHS=1
      NFRONT(1)=50
      NFRONT(2)=50
      LBUFR=1600
      LW=3000
      LIW=300
C INITIALIZATION OF FILES, WE DO NOT WISH TO SOLVE FOR
C FURTHER SYSTEMS
      ISTRM(1)=2
      ISTRM(2)=0
      LENBUF(1)=1600
      LENBUF(2)=0
      LENFLE(1)=16000
      CALL MA32ID(ISTRM,LENBUF,LENFLE,IFLAG)
      IF(IFLAG.NE.0) GO TO 500
C CALLS TO ESTABLISH WHEN EACH VARIABLE IS FULLY ASSEMBLED
C NELT IS NUMBER OF ELEMENTS (THIS MUST BE SET BY THE USER)
      DO 10 IELT=1,NELT
      <generate (perhaps from a subroutine call) a list (in IVAR)
      of the variables appearing in element IELT (setting NVAR,
      and IVAR)>
      CALL MA32AD(NVAR,IVAR,NDF,LAST,LENLST,IFLAG)
      IF(IFLAG.LT.0) GO TO 500
      10 CONTINUE
C INPUT ELEMENTS AND RIGHT HAND SIDES
      IFLAG=0
      DO 20 IELT=1,NELT
      <generate (perhaps from a subroutine call) element and right
      hand side of element IELT (setting NVAR, IVAR, AVAR and RHS)>
      CALL MA32BD (NVAR,IVAR,NDF,LAST,NMAXE,AVAR,NRHS,RHS,
      *LRHS,NFRONT,BUFR,LBUFR,LW,W,LIW,IW,IFLAG)
C ONLY TRAP FATAL ERRORS
      IF(IFLAG.LT.0.AND.IELT.NE.NELT) GO TO 500
      20 CONTINUE
      IF(IFLAG.LT.0) GO TO 400
      <solution is in first NDF locations of W>
      GO TO 1000
      400 <extract information from INFD in common block MA32M/MD to
      enable prevention of failure on subsequent run>
      GO TO 1000
      500 <print appropriate fatal error diagnostic>
      1000 STOP
      END

```

**Figure 5.1 Use of element input**

```

      INTEGER ISTRM(2),LENBUF(2),LENFLE(2),IVAR(15),LAST(1500),IW(300)
      INTEGER NFRONT(2)
      DOUBLE PRECISION AVAR(1,15),RHS(1,1),BUFR(2406),
      *W(3000),B(1500,3),X(1500,3),DUM
      COMMON/MA32MD/ DUM(2),IDUM(4),INFD(10),INFP(6)
C INITIALIZE BOUNDS ON ARRAYS AND PROBLEM SIZE
      LENLST=1500
      NMAXE=1
      NRHS=0
      LRHS=1
      NFRONT(1)=50
      NFRONT(2)=50
      LBUFR=2406
      LW=3000
      LIW=300
      LB=1500
C INITIALIZATION OF FILES
      ISTRM(1)=2
      ISTRM(2)=3
      LENBUF(1)=1203
      LENBUF(2)=1203
      LENFLE(1)=6015
      LENFLE(2)=6015
      CALL MA32ID(ISTRM,LENBUF,LENFLE,IFLAG)
      IF(IFLAG.LT.0) GO TO 500
C CALLS TO DETERMINE WHEN EACH VARIABLE IS FULLY ASSEMBLED
C NEQ IS NUMBER OF EQUATIONS (THIS MUST BE SET BY THE USER)
      DO 10 IEQ=1,NEQ
      <generate, perhaps using a subroutine, a list of columns
      (variables) in row (equation) IEQ ..
      ... thus setting IVAR and NVAR>
      CALL MA32AD(NVAR,IVAR,NDF,LAST,LENLST,IFLAG)
      IF(IFLAG.LT.0) GO TO 500
      10 CONTINUE
C INPUT EQUATIONS
      IFLAG=0
      DO 20 IEQ=1,NEQ
      <set nonzeros in equation IEQ into AVAR and IVAR such that
      for the NVAR nonzeros in this row, the value of the
      nonzero in column IVAR(I) is AVAR(I), I=1,..NVAR>
      CALL MA32BD(NVAR,IVAR,NDF,LAST,NMAXE,AVAR,NRHS,RHS,
      *LRHS,NFRONT,BUFR,LBUFR,LW,W,LIW,IW,IFLAG)
C ONLY TRAP FATAL ERRORS
      IF(IFLAG.LT.0.AND.IEQ.NE.NEQ) GO TO 500
      20 CONTINUE
      IF(IFLAG.LT.0) GO TO 400
      <place jth right hand side in B(.,j),j=1,3>
      NRHS=3
      CALL MA32CD(NRHS,LB,B,X,BUFR,LBUFR,IW,NFRONT,IFLAG)
      IF(IFLAG.LT.0) GO TO 500
      <solution to system j is in X(.,j),j=1,3>
      GO TO 1000
      400 <extract information from INFD in common block MA32MD to
      prevent similar failure on subsequent run>
      GO TO 1000
      500 <print appropriate fatal error diagnostic>
      1000 STOP
      END

```

Figure 5.2 Use of equation input