



1 SUMMARY

This subroutine **solves weighted sparse linear least-squares problems**. Given an $m \times n$ ($m \geq n$) sparse matrix $\mathbf{A} = \{a_{ij}\}$ of rank n , an $m \times m$ diagonal matrix \mathbf{W} of weights, and an m -vector \mathbf{b} , the routine calculates the solution vector \mathbf{x} that minimizes the Euclidean norm of the weighted residual vector $\mathbf{r} = \mathbf{W}(\mathbf{Ax} - \mathbf{b})$ by solving the normal equations $\mathbf{A}^T \mathbf{W}^2 \mathbf{Ax} = \mathbf{A}^T \mathbf{W}^2 \mathbf{b}$.

Three forms of data storage are permitted for the input matrix: storage by columns, where row indices and column pointers describe the matrix; storage by rows, where column indices and row pointers describe the matrix; and the coordinate scheme, where both row and column indices describe the position of entries in the matrix.

For the statistical analysis of the weighted least-squares problem, there are two entries: one to obtain a column and one to obtain the diagonal of the covariance matrix $(\mathbf{A}^T \mathbf{W}^2 \mathbf{A})^{-1}$.

ATTRIBUTES — **Version:** 1.1.2. (1 November 2023) **Types:** Real (single, double). **Calls:** MA27, MC26, MC46, MC59. **Original date:** November 2001. **Remark:** MA75 is a threadsafe version of MA45. **Origin:** I.S. Duff and J.A. Scott, Rutherford Appleton Laboratory, P.P.M. de Rijk, University of Amsterdam.

2 HOW TO USE THE PACKAGE

2.1 Argument Lists and Calling Sequences

There are five entries:

- (a) MA75I/ID initializes the control options.
- (b) MA75A/AD calculates the matrix $\mathbf{A}^T \mathbf{W}^2 \mathbf{A}$ and factorizes it into its Cholesky decomposition using a pivotal strategy designed to preserve sparsity.
- (c) MA75B/BD uses the factors generated by MA75A/AD to solve the normal equations $\mathbf{A}^T \mathbf{W}^2 \mathbf{Ax} = \mathbf{A}^T \mathbf{W}^2 \mathbf{b}$. Optionally the weighted residual vector $\mathbf{r} = \mathbf{W}(\mathbf{Ax} - \mathbf{b})$ is calculated.
- (d) MA75C/CD uses the factors from MA75A/AD to calculate a column of the covariance matrix $(\mathbf{A}^T \mathbf{W}^2 \mathbf{A})^{-1}$.
- (e) MA75D/DD uses the factors from MA75A/AD to calculate the diagonal of $(\mathbf{A}^T \mathbf{W}^2 \mathbf{A})^{-1}$.

A call to MA75B/BD, MA75C/CD, or MA75D/DD must be preceded by a call to MA75A/AD, but a single call to MA75A/AD may be followed by several calls to MA75B/BD to solve for different right-hand sides \mathbf{b} , and similarly MA75C/CD can be used repeatedly to calculate different columns of $(\mathbf{A}^T \mathbf{W}^2 \mathbf{A})^{-1}$.

The control option values described in Section 2.2 can be initialized to default values by a call to MA75I/ID. The user is advised to call MA75I/ID before any use is made of the other entries.

To initialize the control values

The single precision version

```
CALL MA75I (ICNTL, CNTL)
```

The double precision version

```
CALL MA75ID (ICNTL, CNTL)
```

ICNTL is the INTEGER array of length 5 that is used in the MA75A/AD and MA75C/CD entries.

CNTL is the REAL (DOUBLE PRECISION in the D version) array of length 5 that is used in the MA75A/AD entry.

To form $A^T W^2 A$ and calculate its decomposition*The single precision version*

```
CALL MA75A(M,N, ITYPE, NZAIN, NZA, YESWGT, WGT, A, IND, IPTR, ATA, LATA,
*          IATA, LIATA, IW, ICNTL, CNTL, INFO)
```

The double precision version

```
CALL MA75AD(M,N, ITYPE, NZAIN, NZA, YESWGT, WGT, A, IND, IPTR, ATA, LATA,
*          IATA, LIATA, IW, ICNTL, CNTL, INFO)
```

M is an INTEGER variable which must be set by the user to the number of rows in the matrix **A**. This argument is not altered by the routine. **Restriction:** $M \geq 1$.

N is an INTEGER variable which must be set by the user to the number of columns in the matrix **A**. This argument is not altered by the routine. **Restriction:** $1 \leq N \leq M$.

ITYPE is an INTEGER variable which must be set by the user. The permitted values are:

- 1 for the input matrix being stored by columns,
- 2 for the input matrix being stored by rows,
- 3 for the input matrix being stored by the coordinate scheme.

This argument is not altered by the routine. **Restriction:** $1 \leq \text{ITYPE} \leq 3$.

NZAIN is an INTEGER variable which must be set by the user to the number of entries input by the user in the matrix **A**. This argument is not altered by the routine. **Restriction:** $\text{NZAIN} \geq 1$.

NZA is an INTEGER variable which need not be set by the user. On exit, **NZA** is set to the number of entries in the matrix **WA**. This will differ from **NZAIN** if the user included entries which were out of range and/or duplicate entries. It may also differ from **NZAIN** if **YESWGT** is set to **.TRUE.** and the user supplied one or more zero weights. The value of **NZA** must be preserved by the user between a call to this subroutine and subsequent calls to **MA75B/BD**.

YESWGT is a logical variable which must be set by the user. If **YESWGT** = **.TRUE.**, the problem will be weighted using the weights in **WGT**, otherwise the problem will not be weighted. This argument is not altered by the routine.

WGT is a REAL (DOUBLE PRECISION in the D version) array. If **YESWGT** = **.TRUE.**, the array must be of length **M** and must be set by the user so that the absolute value of **WGT(I)** is the weight for row **I** of **A** ($I = 1, \dots, M$). If **WGT(I)** is zero for any **I**, a positive flag is set (see §2.3) and row **I** of **A** is ignored. The number of zero weights must not exceed **M-N**. If **YESWGT** = **.FALSE.**, the array is not accessed. This argument is not altered by the routine.

A is a REAL (DOUBLE PRECISION in the D version) array of length **NZAIN**, which must be set by the user to hold the values of the entries in the matrix **A**. The order depends upon the value of **ITYPE**. For **ITYPE** = 1, the entries must be ordered by columns, with the entries in each column contiguous and those of column **J** preceding those of column **J+1** ($J = 1, \dots, N$); the ordering within each column is unimportant. For **ITYPE** = 2, the entries must be ordered by rows, with the entries in each row contiguous, and those of row **I** preceding those of row **I+1** ($I = 1, \dots, M$); the ordering within each row is unimportant. For **ITYPE** = 3 the entries may be in any order. In each case, any entries which are out of range are ignored and if duplicate entries are entered, one copy of each duplicated entry is kept and their numerical values are summed. On exit, the first **NZA** entries in the array hold the entries of **WA** stored by rows. These values in **A** must be preserved by the user between a call to this subroutine and subsequent calls to **MA75B/BD**.

IND is an INTEGER array of length **NZAIN**, which must be set by the user. For **ITYPE** = 1 and 3 it must be set to the row indices of the corresponding entries in **A**. For **ITYPE** = 2, the array must be set to the column indices of the corresponding entries in **A**. On exit, the first **NZA** entries of the array are set by the routine to the column indices of the corresponding entries in **A**. These values in **IND** must be preserved by the user between a call to this

subroutine and subsequent calls to MA75B/BD.

IPTR is an INTEGER array of length $M+1$, which need only be set by the user if $ITYPE = 1$ or 2 . If $ITYPE = 1$, it must be set so that $IPTR(J)$ points to the position in arrays A and IND of the first entry in column J ($J = 1, \dots, N$); $IPTR(N+1)$ must be set to $NZAIN+1$. If $ITYPE = 2$, it must be set so that $IPTR(I)$ points to the position in arrays A and IND of the first entry in row I ($I = 1, \dots, M$); $IPTR(M+1)$ must be set to $NZAIN+1$. On exit, for $ITYPE = 1, 2$, or 3 , it is set by the routine so that $IPTR(I)$ points to the position in arrays A and IND of the first entry in row I of WA and $IPTR(M+1)$ is set to $NZA+1$. These values in IPTR must be preserved by the user between a call to this subroutine and subsequent calls to MA75B/BD.

ATA is a REAL (DOUBLE PRECISION in the D version) array of length LATA, which need not be set by the user. On exit, ATA contains the values of the entries in the factors of $A^T W^2 A$. These values in ATA must be preserved by the user between a call to this subroutine and subsequent calls to MA75B/BD, MA75C/CD, or MA75D/DD.

LATA is an INTEGER variable which must be set by the user to the length of array ATA. LATA must be at least $\text{MAX}(N+NZA, \text{INFO}(9))$, where $\text{INFO}(9)$ is the number of entries in the lower triangle of the matrix $A^T W^2 A$ (see §2.2). If it is not large enough, an error return is provoked and a suggested value is placed in $\text{INFO}(7)$ (see §2.3). This argument is not altered by the routine.

IATA is an INTEGER array of length LIATA, which need only be set by the user if $ITYPE = 3$. If so, the first NZAIN values must be set to the column indices of the entries of the matrix A , in the same order as the entries in the arrays A and IND . On exit, the array contains integer index information on the entries of the factors of $A^T W^2 A$. These values in IATA must be preserved by the user between a call to this subroutine and subsequent calls to MA75B/BD, MA75C/CD, or MA75D/DD.

LIATA is an INTEGER variable which must be set by the user to the length of array IATA. LIATA must be at least $\text{MAX}(N+NZA, 3 * \text{INFO}(9) + 3 * N + 1)$, where $\text{INFO}(9)$ is the number of entries in the lower triangle of the matrix $A^T W^2 A$ (see §2.2). If it is not large enough, an error return is provoked and a suggested value is placed in $\text{INFO}(8)$ (see §2.3). This argument is not altered by the routine. **Restrictions:** if $ITYPE=1$, $LIATA \geq \text{MAX}(M, 3 * N + 1)$; if $ITYPE=2$, $LIATA \geq 3 * N + 1$; if $ITYPE=3$, $LIATA \geq \text{MAX}(NZAIN, 3 * N + 1)$.

IW is an INTEGER array of length $M+5 * N$, which is used as workspace.

ICNTL is an INTEGER array of length 5 set to the optional integer control values described in Section 2.2.

CNTL is a REAL array of length 5 set to the optional real control values described in Section 2.2.

INFO is an INTEGER array of length 15 in which MA75 returns information to user, see Section 2.2.

To solve least-squares problems using the factors from MA75A/AD

The single precision version

```
CALL MA75B(M,N,NZA,YESWGT,WGT,A,IND,IPTR,ATA,LATA,
*         IATA,LIATA,RES,RHS,X,W,IW)
```

The double precision version

```
CALL MA75BD(M,N,NZA,YESWGT,WGT,A,IND,IPTR,ATA,LATA,
*         IATA,LIATA,RES,RHS,X,W,IW)
```

M is an INTEGER variable which must be set by the user to the number of rows in the matrix A . This argument is not altered by the routine.

N is an INTEGER variable which must be set by the user to the number of columns in the matrix A . This argument is not altered by the routine.

NZA is an INTEGER variable, which must be passed unchanged since the call to MA75A/AD. This argument is not altered by the routine.

YESWGT is a logical variable, which must be passed unchanged since the call to MA75A/AD. This argument is not altered by the routine.

WGT is an integer array of length M, which must be passed unchanged since the call to MA75A/AD. This argument is not altered by the routine.

A is a REAL (DOUBLE PRECISION in the D version) array of length NZA, which must be passed unchanged since the call to MA75A/AD. This argument is not altered by the routine.

IND is an INTEGER array of length NZA, which must be passed unchanged since the call to MA75A/AD. This argument is not altered by the routine.

IPTR is an INTEGER array of length M+1, which must be passed unchanged since the call to MA75A/AD. This argument is not altered by the routine.

ATA is a REAL (DOUBLE PRECISION in the D version) array of length LATA, which must be passed unchanged since the call to MA75A/AD. This argument is not altered by the routine.

LATA is an INTEGER variable which must be set by the user to the length of array ATA. This argument is not altered by the routine.

IATA is an INTEGER array of length LIATA, which must be passed unchanged since the call to MA75A/AD. This argument is not altered by the routine.

LIATA is an INTEGER variable which must be set by the user to the length of array IATA. This argument is not altered by the routine.

RES is a LOGICAL variable which must be set by the user to `.TRUE.` if the weighted residual vector \mathbf{r} is wanted and `.FALSE.` otherwise. This argument is not altered by the routine.

RHS is a REAL (DOUBLE PRECISION in the D version) array of length M. On entry, `RHS(I)` must hold the I-th component of the right-hand side \mathbf{b} of the least-squares problem being solved. If YESWGT is set to `.TRUE.` and RES is set to `.FALSE.`, RHS is overwritten by \mathbf{Wb} . If RES is set to `.TRUE.`, RHS is overwritten by the weighted residual vector \mathbf{r} . Otherwise it is not altered by the routine.

X is a REAL (DOUBLE PRECISION in the D version) array of length N, which need not be set by the user. On exit, `X(J)` holds the J-th component of the solution vector.

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

IW is an INTEGER array of length N, which is used as workspace.

To calculate a column of the covariance matrix using the factors from MA75A/AD

The single precision version

```
CALL MA75C(N,ATA,LATA,IATA,LIATA,JCOL,V,W,IW,ICNTL,INFO)
```

The double precision version

```
CALL MA75CD(N,ATA,LATA,IATA,LIATA,JCOL,V,W,IW,ICNTL,INFO)
```

N is an INTEGER variable which must be set by the user to the number of columns in the matrix \mathbf{A} . This argument is not altered by the routine.

ATA is a REAL (DOUBLE PRECISION in the D version) array of length LATA, which must be passed unchanged since the call to MA75A/AD. This argument is not altered by the routine.

LATA is an INTEGER variable which must be set by the user to the length of array ATA. This argument is not altered by the routine.

IATA is an INTEGER array of length LIATA, which must be passed unchanged since the call to MA75A/AD. This

argument is not altered by the routine.

LIATA is an INTEGER variable which must be set by the user to the length of array IATA. This argument is not altered by the routine.

JCOL is an INTEGER variable which must be set by the user to the index of the column of the covariance matrix that is required. This argument is not altered by the routine. **Restriction:** $1 \leq \text{JCOL} \leq N$.

V is a REAL (DOUBLE PRECISION in the D version) array of length N, which need not be set by the user. On exit, $V(I)$ holds the I-th component of the JCOL-th column of the covariance matrix.

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

IW is an INTEGER array of length N, which is used as workspace.

ICNTL is an INTEGER array of length 5 set to the optional integer control values described in Section 2.2.

INFO is an INTEGER array of length 15 in which MA75 returns information to user, see Section 2.2.

To calculate the diagonal of the covariance matrix using the factors from MA75A/AD

The single precision version

```
CALL MA75D(N,ATA,LATA,IATA,LIATA,VAR,W,IW)
```

The double precision version

```
CALL MA75DD(N,ATA,LATA,IATA,LIATA,VAR,W,IW)
```

N is an INTEGER variable which must be set by the user to the number of columns in the matrix **A**. This argument is not altered by the routine.

ATA is a REAL (DOUBLE PRECISION in the D version) array of length LATA, which must be passed unchanged since the call to MA75A/AD. The contents of this array are destroyed by the routine.

LATA is an INTEGER variable which must be set by the user to the length of array ATA. This argument is not altered by the routine.

IATA is an INTEGER array of length LIATA, which must be passed unchanged since the call to MA75A/AD. This argument is not altered by the routine.

LIATA is an INTEGER variable which must be set by the user to the length of array IATA. This argument is not altered by the routine.

VAR is a REAL (DOUBLE PRECISION in the D version) array of length N, which need not be set by the user. On exit, it holds the diagonal of the covariance matrix.

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

IW is an INTEGER array of length $4*N+4$, which is used as workspace.

2.2 The control and information arrays

The INTEGER array ICNTL contains integer control values. It is initialized to default values by MA75I/ID and may subsequently be reset by the user.

ICNTL(1) (was LP) is used by the subroutines as the output stream for error messages. If it is set to zero these messages will be suppressed. The default value is 6.

ICNTL(2) (was MP) is used by the subroutines as the output stream for diagnostic printing and for warning messages. If it is set to zero then messages are suppressed. The default value is 6.

ICNTL(3) to ICNTL(5) are not used.

The REAL (DOUBLE PRECISION in the D version) array CNTL contains real control values. It is initialized to default values by MA27I/ID and may subsequently be reset by the user.

CNTL(1) (was U) is used by MA75A/AD to control numerical pivoting when factorizing $\mathbf{A}^T \mathbf{W}^2 \mathbf{A}$ using MA27. Its default value is zero and then no pivoting is performed. In that case, if some of the pivots have different signs, INFO(1) will be set to +4 (see §2.3) and if a zero pivot is encountered INFO(1) is set to -9 and the subroutine fails. If the user wants numerical pivoting, CNTL(1) should be set to a positive value between zero and 0.5, with the numerical control being greater the larger CNTL(1) is. Values greater than 0.5 are treated as 0.5. Negative values are treated as zero. This parameter is not altered by the routine.

CNTL(2) to CNTL(5) are not used.

On return from MA75A/AD and MA75C/CD the INTEGER array INFO contains information for user.

INFO(1) (was IFLAG) is an error flag. A value of zero indicates that the subroutine has performed successfully. For nonzero values, see section 2.3.

INFO(2) (was IWARN) provides supplementary information when there is a warning or an error. If INFO(1) = +4 or INFO(1) = +5 is returned, extra information is placed in INFO(2) (see § 2.3).

INFO(3) (was IOUT). If the matrix \mathbf{A} contains entries which are out of range the number of such entries is placed in INFO(3).

INFO(4) (was IDUP). When MA75A/AD is called, INFO(4) is set to zero. For each entry in the matrix \mathbf{A} that appears k times, INFO(4) is incremented by $k-1$.

INFO(5) (was IZWGT). If YESWGT is set to .TRUE. and the user includes zero weights, the number of zero weights is placed in INFO(5).

INFO(6) (was ISTEP). If the matrix $\mathbf{A}^T \mathbf{W}^2 \mathbf{A}$ is found to be singular, INFO(6) is set to the pivot step at which the singularity was detected.

INFO(7) and INFO(8) (was MINRLS and MINIRS). If insufficient space is allocated to the array ATA, INFO(7) is set to a value that may suffice and INFO(8) is set to a value that may suffice for LIATA. If insufficient space is allocated to the array IATA, INFO(8) is set to a value that may suffice.

INFO(9) (was NZATA). On exit from MA75A/AD, it gives the number of entries in the lower triangle of the matrix $\mathbf{A}^T \mathbf{W}^2 \mathbf{A}$; a value of -1 indicates that this number could not be calculated.

INFO(10) and INFO(11) (was NRLS and NIRS). On exit from MA75A/AD, they give the amount of storage in the arrays ATA and IATA actually used to hold the factors of the matrix $\mathbf{A}^T \mathbf{W}^2 \mathbf{A}$.

INFO(12) to INFO(15) are not used.

2.3 Error diagnostics

A successful return from MA75A/AD or MA75C/CD is indicated by a value of INFO(1) equal to 0. There are no error returns from MA75B/BD and MA75D/DD. Of the possible nonzero values for INFO(1) given below, only the value -10 can be generated by MA75C/CD; the other nonzero values can only be returned by MA75A/AD. A negative value of INFO(1) is associated with a fatal error. The error is output on unit ICNTL(1).

- 1 Value of M out of range. $M < 1$. Immediate return with input values unchanged.
- 2 Value of N out of range. Either $N < 1$ or $N > M$. Immediate return with input values unchanged.
- 3 Value of NZAIN out of range. $NZAIN < 1$. Immediate return with input values unchanged.
- 4 Value of ITYPE out of range. Immediate return with input values unchanged.
- 5 YESWGT set to .TRUE. but the number of zero weights exceeds $M-N$. Immediate return with input values unchanged.
- 6 Value of LIATA out of range. If ITYPE = 1, $LIATA < \text{MAX}(M, 3*N+1)$; if ITYPE = 2, $LIATA < 3*N+1$; if ITYPE = 3,

$LIATA < \text{MAX}(NZAIN, 3*N+1)$. Immediate return with input values unchanged.

- 7 Failure due to insufficient space allocated to array ATA. $\text{INFO}(7)$ is set to a value that may suffice. Since there is a possibility that $LIATA$ will also be too small, $\text{INFO}(8)$ is set to a value that may suffice. Routine returns with the variable NZA and the arrays A , IND , and $IPTR$ computed to their output form, but the arrays ATA and $IATA$ are corrupted.
- 8 Failure due to insufficient space allocated to array IATA. $\text{INFO}(8)$ is set to a value that may suffice. Routine returns with the variable NZA and the arrays A , IND , and $IPTR$ computed to their output form, but the arrays ATA and $IATA$ are corrupted.
- 9 Matrix $A^T W^2 A$ is singular; $\text{INFO}(6)$ is set to the pivot step at which singularity was detected.
- 10 Value of $JCOL$ in MA75C/CD out of range. Either $JCOL < 1$ or $JCOL > N$. Immediate return from MA75C/CD with input values unchanged.

A positive value of $\text{INFO}(1)$ is associated with a warning. The warning message is output on unit $\text{ICNTL}(2)$. Note that a warning flag may be overwritten by a warning flag of a higher value.

- +1 The input data contains entries in A which are out of range. Entries which are out of range are removed by the routine. $\text{INFO}(3)$ is set to the number of such entries and NZA is set to the revised number of entries in the matrix. If all the entries in A are out of range, the computation is stopped and a warning is output on unit $\text{ICNTL}(2)$.
- +2 The input data contains entries in A which are duplicated. One copy of each duplicated entry is kept and the values of the duplicates are summed. If an entry appears k times, $\text{INFO}(4)$ is incremented by $k-1$ and NZA is set to the revised number of entries in the matrix.
- +3 One or more of the weights is zero (but the number of zero weights does not exceed $M-N$). $\text{INFO}(5)$ is set to the number of zero weights. The rows of A corresponding to the zero weights are ignored and NZA is set to the revised number of entries in the matrix. This warning can only be issued if YESWGT is set to `.TRUE.`
- +4 Pivots have different signs when factorizing $A^T W^2 A$ (input value of $\text{CNTL}(1)$ is non-positive). $\text{INFO}(2)$ is set to the number of sign changes.
- +5 Matrix $A^T W^2 A$ is rank deficient. In this case a decomposition is produced and $\text{INFO}(2)$ is set to the rank of the matrix.

3 GENERAL INFORMATION

Workspace:

MA75A/AD:

IW , an INTEGER array of length $M+5*N$.

MA75B/BD and MA75C/CD:

W , a REAL (DOUBLE PRECISION in the D version) array of length N .

IW , an INTEGER array of length N .

MA75D/DD:

W , a REAL (DOUBLE PRECISION in the D version) array of length N .

IW , an INTEGER array of length $4*N+4$.

Use of common: None.

Other routines called directly: The following subroutines are called by the routines in this package: MC26A/AD, MA27A/AD, MA27B/BD, MA27C/CD, MC46A/AD, MC59A/AD, and MA75G/GD. MC26A/AD calls further subroutines MC26B/BD and MC20A/AD, whereas the MA27 routines only call routines from the MA27 package itself, and MC59A/AD only calls internal subroutines MC59B/BD, MC59C/CD, and MC59D/DD. Common variables are initialized by the block data subprogram MA75F/FD.

Input/output: Error messages on unit ICNTL(1) ($\text{ICNTL}(1) \leq 0$ suppresses them) and warning messages on unit ICNTL(2) ($\text{ICNTL}(2) \leq 0$ suppresses them).

Restrictions:

$M \geq 1$,
 $1 \leq N \leq M$,
 $1 \leq \text{ITYPE} \leq 3$,
 $\text{NZAIN} \geq 1$,
 $1 \leq \text{JCOL} \leq N$.
 If $\text{ITYPE} = 1$, $\text{LIATA} \geq \text{MAX}(M, 3*N+1)$;
 if $\text{ITYPE} = 2$, $\text{LIATA} \geq 3*N+1$;
 if $\text{ITYPE} = 3$, $\text{LIATA} \geq \text{MAX}(\text{NZAIN}, 3*N+1)$.

4 METHOD

MA75A/AD

MA75A/AD first checks the input data. Control is immediately returned to the calling program if a fatal error is found. Otherwise, a check is made for duplicate entries and entries which are out of range. If the input matrix is stored using the coordinate scheme, MC49A/AD performs this check and sorts the matrix to an ordering by rows. Any entries which are out of range are ignored and if duplicate entries are found, one copy of each duplicated entry is kept and their numerical values are summed. In either case a positive flag is set. If the input matrix is stored by columns, MC46A/AD is used to reorder the matrix by rows. Then, if a weighted least-squares solution is required, \mathbf{WA} is computed. If any of the weights supplied by the user are zero, the corresponding rows in \mathbf{A} are removed, and, provided the number of zero weights does not exceed $M-N$, where M and N are, respectively, the number of rows and columns of \mathbf{A} , a positive flag is set. If the number of zero weights does exceed $M-N$, a negative flag is set and control is returned to the calling program.

Once the checks on the input data are complete and the problem has been weighted, MC26A/AD is used to calculate $\mathbf{A}^T \mathbf{W}^2 \mathbf{A}$, storing the lower triangle entries in the arrays ATA and IATA. If insufficient space has been allocated to either of these arrays, the calculation is stopped, the minimum required sizes are accumulated in INFO(7) and/or INFO(8), a negative flag is set and control is returned to the calling program. Otherwise, MA27A/AD is called to compute a minimum degree ordering and its associated assembly and elimination information. Finally, MA27B/BD is called to compute the factorization of $\mathbf{A}^T \mathbf{W}^2 \mathbf{A}$, overwriting the arrays ATA and IATA with the entries of the factors of $\mathbf{A}^T \mathbf{W}^2 \mathbf{A}$. Again, if insufficient space has been allocated to either of these arrays, or if the matrix $\mathbf{A}^T \mathbf{W}^2 \mathbf{A}$ is found to be singular, the calculation is stopped and an error return is made.

MA75B/BD

MA75B/BD first computes $\mathbf{A}^T \mathbf{W}^2 \mathbf{b}$, using the permuted matrix \mathbf{WA} as output by MA75A/AD. It then uses the factors of $\mathbf{A}^T \mathbf{W}^2 \mathbf{A}$ computed by MA75A/AD to call MA27C/CD to solve the system of normal equations. Optionally, it computes the weighted residual vector $\mathbf{r} = \mathbf{W}(\mathbf{Ax} - \mathbf{b})$.

MA75C/CD

In order to calculate the j -th column \mathbf{v}_j of the covariance matrix, MA75C/CD generates the j -th unit vector \mathbf{e}_j and solves the system $\mathbf{A}^T \mathbf{W}^2 \mathbf{A} \mathbf{v}_j = \mathbf{e}_j$ by calling MA27C/CD, using the factors from MA75A/AD.

MA75D/DD

Starting in the left-hand bottom corner, entries of $(\mathbf{A}^T \mathbf{W}^2 \mathbf{A})^{-1}$, which are lying in the sparsity pattern of the factorization of $\mathbf{A}^T \mathbf{W}^2 \mathbf{A}$, are successively calculated using only entries that have already been calculated and the entries in the factorization. The symmetry of $\mathbf{A}^T \mathbf{W}^2 \mathbf{A}$ is exploited. Afterwards, the diagonal elements of the covariance matrix are collected and placed in the appropriate order.

5 EXAMPLE OF USE

The following program gives an example of the use of MA75. Suppose we are interested in the vector \mathbf{x} that minimizes $\|\mathbf{W}(\mathbf{Ax} - \mathbf{b})\|_2$, where

$$\mathbf{A} = \begin{pmatrix} 2.0 & 3.0 & 0.0 \\ 0.0 & 0.0 & 4.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 5.0 \end{pmatrix}, \quad \mathbf{W} = \begin{pmatrix} 2.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 2.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{pmatrix},$$

and

$$\mathbf{b}^T = (8.0 \ 12.0 \ 2.0 \ 15.0).$$

If we input matrix \mathbf{A} in coordinate form in arrays

```
A = 2.0, 1.0, 5.0, 3.0, 4.0
IND = 1, 3, 4, 1, 2
IATA = 1, 2, 3, 2, 3
```

then the program

```
INTEGER M, N, NZAIN, LATA, LIATA
PARAMETER( M=4, N=3, NZAIN=5, LATA=30, LIATA=50 )
DOUBLE PRECISION A(NZAIN), ATA(LATA), RHS(M), X(N), W(N),
* COVAR(N), VAR(N), WGT(M)
DOUBLE PRECISION CNTL(5)
INTEGER ITYPE, JCOL, NZA, K
INTEGER IPTR(M+1), IW(M+5*N)
INTEGER IND(NZAIN), IATA(LIATA)
INTEGER ICNTL(5), INFO(15)
LOGICAL YESWGT, RES
DATA IND / 1, 3, 4, 1, 2 /
DATA( IATA(K), K=1,NZAIN) / 1, 2, 3, 2, 3 /
DATA A / 2.0D0,1.0D0,5.0D0,3.0D0,4.0D0/
DATA WGT / 2.0D0,1.0D0,2.0D0,1.0D0/
DATA RHS / 8.0D0,12.0D0,2.0D0,15.0D0/
C
CALL MA75ID( ICNTL,CNTL )
ITYPE = 3
YESWGT = .TRUE.
CALL MA75AD( M,N,ITYPE,NZAIN,NZA,YESWGT,WGT,A,IND,IPTR,ATA,LATA,
+ IATA,LIATA,IW,ICNTL,CNTL,INFO )
RES = .FALSE.
CALL MA75BD( M,N,NZA,YESWGT,WGT,A,IND,IPTR,ATA,LATA,IATA,LIATA,
+ RES,RHS,X,W,IW )
JCOL = 2
CALL MA75CD( N,ATA,LATA,IATA,LIATA,JCOL,COVAR,W,IW,ICNTL,INFO )
CALL MA75DD( N,ATA,LATA,IATA,LIATA,VAR,W,IW )
```

```

C
  WRITE( 6, 100 ) A, IND, IPTR, WGT, RHS, X, COVAR, VAR
100 FORMAT(' The weighted least squares problem whose coefficient' /
*         ' matrix, weights, and',
*         ' RHS are given by the following arrays:' //
*         '   A = ',5F5.1/'   IND = ',5(I3,2X)/
*         '   IPTR = ',5(I3,2X)/
*         '   WGT = ',4F5.1/
*         '   RHS = ',4F5.1///
*         ' has the solution vector:' //
*         '   X = ',3(1PD13.5)///
*         ' The second column of the covariance',
*         ' matrix is' // ' COVAR = ',3(1PD13.5)///
*         ' The diagonal of the covariance',
*         ' matrix is' // '   VAR = ',3(1PD13.5)/// )
  STOP
  END

```

produces the following output:

The weighted least squares problem whose coefficient matrix, weights, and RHS are given by the following arrays:

```

  A =   4.0  6.0  4.0  2.0  5.0
  IND =  1   2   3   2   3
  IPTR =  1   3   4   5   6
  WGT =  2.0  1.0  2.0  1.0
  RHS = 16.0 12.0  4.0 15.0

```

has the solution vector:

```

  X =  1.00000D+00  2.00000D+00  3.00000D+00

```

The second column of the covariance matrix is

```

  COVAR = -3.75000D-01  2.50000D-01  0.00000D+00

```

The diagonal of the covariance matrix is

```

  VAR =  6.25000D-01  2.50000D-01  2.43902D-02

```