

1 SUMMARY

To solve a sparse symmetric system of linear equations. Given a sparse symmetric matrix $\mathbf{A} = \{a_{ij}\}_{n \times n}$ and an n -vector \mathbf{b} , this subroutine solves the system $\mathbf{Ax}=\mathbf{b}$. The matrix \mathbf{A} need not be definite.

The method used is a direct method based on a sparse variant of Gaussian elimination and is discussed further by Duff and Reid, ACM Trans. Math. Software **9** (1983), 302-325.

ATTRIBUTES — **Version:** 1.0.0. **Types:** Real (single, double) **Original date:** June 1982. **Remark:** Sept. 2001: this version of MA27 is threadsafe. **Origin:** I. S. Duff and J. K. Reid, Harwell.

2 HOW TO USE THE PACKAGE

2.1 Argument lists and calling sequences

There are four entries:

- (a) MA27I/ID should be called before any other use of MA27 to initialize the user options and calculation parameters.
- (b) MA27A/AD accepts the pattern of \mathbf{A} and chooses pivots for Gaussian elimination using a selection criterion to preserve sparsity. It subsequently constructs subsidiary information for actual factorization by MA27B/BD. The user may input the pivot sequence in which case only the necessary information for MA27B/BD will be generated.
- (c) MA27B/BD factorizes a matrix \mathbf{A} using the information from a previous call to MA27A/AD. The actual pivot sequence used may differ slightly from that of MA27A/AD if \mathbf{A} is not definite.
- (d) MA27C/CD uses the factors generated by MA27B/BD to solve a system of equations $\mathbf{Ax}=\mathbf{b}$.

A call to MA27C/CD must be preceded by a call to MA27B/BD which in turn must be preceded by a call to MA27A/AD. Since the information passed from one subroutine to the next is not corrupted by the second, several calls to MA27B/BD for matrices with the same sparsity pattern but different values may follow a single call to MA27A/AD, and similarly MA27C/CD can be used repeatedly to solve for different right-hand sides \mathbf{b} .

Initialization

The single precision version

```
CALL MA27I(ICNTL,CNTL)
```

The double precision version

```
CALL MA27ID(ICNTL,CNTL)
```

ICNTL is an INTEGER array of length 30 which contains integer control values used by the other MA27 entries. It is initialized to default values which may be reset by the user, see Section 2.2.

CNTL is an REAL (DOUBLE PRECISION in the D version) array of length 5 which contains real control values used by the other MA27 entries. It is initialized to default values which may be reset by the user, see Section 2.2.

To perform symbolic manipulations*The single precision version*

```
CALL MA27A(N,NZ,IRN,ICN,IW,LIW,IKEEP,IW1,NSTEPS,IFLAG,ICNTL,CNTL,INFO,OPS)
```

The double precision version

```
CALL MA27AD(N,NZ,IRN,ICN,IW,LIW,IKEEP,IW1,NSTEPS,IFLAG,ICNTL,CNTL,INFO,OPS)
```

N is an INTEGER variable which must be set by the user to the order n of the matrix **A**. It is not altered by the subroutine. **Restriction:** $N \geq 1$.

NZ is an INTEGER variable which must be set by the user to the number of entries being input. It is not altered by the subroutine. **Restriction:** $NZ \geq 0$.

IRN and **ICN** are INTEGER arrays of length **NZ**. The user must set them so that each off-diagonal nonzero a_{ij} is represented by $IRN(k)=i$ and $ICN(k)=j$ or by $IRN(k)=j$ and $ICN(k)=i$. These arrays will be unaltered by the subroutine unless the user wishes to conserve storage which can be done by equivalencing $IRN(1)$ to $IW(1)$ and $ICN(1)$ to $IW(k)$, $k > NZ$.

IW is an INTEGER array of length **LIW**. This is used as workspace by the subroutine. Its length must be at least $2*NZ+3*N+1$ (or $NZ+3*N+1$ if the pivot order is specified in **IKEEP**), but we recommend that it should be at least 20% greater than this (see **NCMPA** returned in **INFO(11)**, section 2.2).

LIW is an INTEGER variable. It must be set by the user to the length of array **IW** and is not altered by the subroutine.

IKEEP is an INTEGER array of length $3*N$. It need not be set by the user and must be preserved between a call to **MA27A/AD** and subsequent calls to **MA27B/BD**. If the user wishes to input the pivot sequence, the position of variable i in the pivot order should be placed in $IKEEP(i)$, $i = 1, 2, \dots, n$ and **IFLAG** should be set to 1. The given order may be replaced by another that gives the same fill-in pattern and virtually identical numerical results.

IW1 is an INTEGER array of length $2*N$. It is used as workspace by the subroutine.

NSTEPS is an INTEGER variable. It need not be set by the user on input and should be passed unchanged when later calling **MA27B/BD**.

IFLAG is an INTEGER variable which the user must set to zero if a suitable pivot order is to be chosen automatically or to 1 if the pivot order set in **IKEEP** is to be used. It is not altered by the subroutine. Note that **IFLAG** is no longer used as an error flag, see **INFO(1)** in Section 2.2.

ICNTL is an INTEGER array of length 30 which contains integer control values, see Section 2.2.

CNTL is an REAL (DOUBLE PRECISION in the D version) array of length 5 which contains real control values, see Section 2.2.

INFO is an INTEGER array of length 20 in which **MA27** returns information to the user, see Section 2.2.

OPS is a REAL (DOUBLE PRECISION in the D version) variable. On return from **MA27A/AD**, **OPS** will be set to the number of multiply-add pairs of operations required by the factorization if no pivoting is performed. Numerical pivoting in **MA27B/BD** may increase the number of operations slightly.

To factorize a matrix*The single precision version*

```
CALL MA27B(N,NZ,IRN,ICN,A,LA,IW,LIW,IKEEP,NSTEPS,
*      MAXFRT,IW1,ICNTL,CNTL,INFO)
```

The double precision version

```
CALL MA27BD(N,NZ,IRN,ICN,A,LA,IW,LIW,IKEEP,NSTEPS,
*      MAXFRT,IW1,ICNTL,CNTL,INFO)
```

- N** is an INTEGER variable which must be set by the user to the order n of the matrix **A**. It must be unchanged since the last call to MA27A/AD and is not altered by the subroutine. **Restriction:** $N \geq 1$.
- NZ** is an INTEGER variable which must be set by the user to the number of entries in the matrix **A**. It is not altered by the subroutine. **Restriction:** $NZ \geq 0$.
- IRN** and **ICN** are INTEGER arrays of length **NZ**, see the next argument description.
- A** is a REAL (DOUBLE PRECISION in the D version) array of length **LA**.
- The arguments **IRN**, **ICN** and **A** must be set by the user to hold the nonzeros. A diagonal entry a_{ii} is held as $A(k)=a_{ii}$, $IRN(k)=ICN(k)=i$ and a pair of off-diagonal entries $a_{ij}=a_{ji}$ is held as $A(k)=a_{ij}$ and $IRN(k)=i$, $ICN(k)=j$ or vice-versa. Multiple entries are summed and any with $IRN(k)$ or $ICN(k)$ out of range are ignored. On exit, array **A** will hold the entries of the factors of the matrix **A**. These entries in **A** must be preserved by the user between calls to this subroutine and subsequent calls to MA27C/CD. **IRN** and **ICN** will be unaltered by the subroutine unless the user wishes to conserve storage by equivalencing $IRN(1)$ to $IW(1)$ and $ICN(1)$ to $IW(k)$, $k > NZ$.
- LA** is an INTEGER variable which must be set by the user to the length of array **A**. It must be at least as great as **NRLNEC** returned in **INFO(5)** (see section 2.2), set by MA27A/AD. It is advisable to allow a slightly greater value because the use of numerical pivoting might increase storage requirements marginally. It is not altered by the subroutine.
- IW** is an INTEGER array of length at least as great as **NIRNEC** returned by MA27A/AD in **INFO(6)** (see section 2.2). A slightly greater value is recommended because numerical pivoting may increase storage requirements marginally. **IW**, which need not be set by the user, is used as workspace by MA27B/BD and on exit holds integer indexing information on the matrix factors. It must be preserved by the user between calls to this subroutine and MA27C/CD.
- LIW** is an INTEGER variable which must be set by the user to the length of array **IW**. It is not altered by MA27B/BD.
- IKEEP** is an INTEGER array of length $3*N$ which must be passed unchanged since the last call to MA27A/AD. It is not altered by MA27B/BD.
- NSTEPS** is an INTEGER variable which must be passed unchanged since the last call to MA27A/AD. It is not altered by MA27B/BD.
- MAXFRT** is an INTEGER variable which need not be set by the user and should be passed unchanged to subsequent calls of MA27C/CD.
- IW1** is an INTEGER array of length **N**. It is used as workspace by the subroutine.
- ICNTL** is an INTEGER array of length 30 which contains integer control values, see Section 2.2.
- CNTL** is an REAL (DOUBLE PRECISION in the D version) array of length 5 which contains real control values, see Section 2.2.
- INFO** is an INTEGER array of length 20 in which MA27 returns information to the user, see Section 2.2. Note that **INFO(1)** is the error flag for MA27B/BD.

To solve equations using the factors from MA27B/BD

The single precision version

```
CALL MA27C(N,A,LA,IW,LIW,W,MAXFRT,RHS,IW1,NSTEPS,ICNTL,INFO)
```

The double precision version

```
CALL MA27CD(N,A,LA,IW,LIW,W,MAXFRT,RHS,IW1,NSTEPS,ICNTL,INFO)
```

- N** is an INTEGER variable which must be set by the user to the order n of the matrix **A**. It must be unchanged since the last call to MA27B/BD and is not altered by the subroutine. **Restriction:** $N \geq 1$.

- A is a REAL array (DOUBLE PRECISION in the D version) of length LA which must be unchanged since the last call to MA27B/BD. It is not altered by the subroutine.
- LA is an INTEGER variable which must be set by the user to the length of array A. It is not altered by the subroutine.
- IW is an INTEGER array of length LIW which must be unchanged since the last call to MA27B/BD. It is not altered by the subroutine.
- LIW is an INTEGER variable which must be set by the user to the length of array IW. It is not altered by the subroutine.
- W is a REAL (DOUBLE PRECISION in the D version) workspace array of length MAXFRT.
- MAXFRT is an INTEGER variable which must be passed unchanged since the last call to MA27B/BD. This value will not be greater than N. It is not altered by MA27C/CD.
- RHS is a REAL array (DOUBLE PRECISION in the D version) of length N. On entry, $RHS(i)$ must hold the i -th component of the right-hand side of the equations being solved. On exit it will be equal to the corresponding entry of the solution vector.
- IW1 is an INTEGER workspace array of length NSTEPS.
- NSTEPS is an INTEGER variable which must be passed unchanged since the last call to MA27A/AD. This value will not be greater than N. It is not altered by the subroutine.
- ICNTL is an INTEGER array of length 30 which contains integer control values, see Section 2.2.
- INFO is an INTEGER array of length 20 in which MA27 returns information to the user, see Section 2.2.

2.2 The control and information arrays

The INTEGER array ICNTL contains integer control values used by the other MA27 entries. It is initialized to default values by MA27I/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) (was LDIAG) is used by the subroutines to control diagnostic printing. If ICNTL(3) is equal to zero (the default), no diagnostic printing will be produced, a value of 1 will print scalar parameters (both in argument lists and in the control and information arrays) and a few entries of array parameters on entry and successful exit from each subroutine while ICNTL(3) equal to 2 will print all parameter values on entry and successful exit.

The entries ICNTL(4) to ICNTL(25) are not of interest to the general user and are discussed more fully by Duff and Reid (AERE R-10533, 1982) under the internal names IOVFLO, NEMIN and IFRLVL.

ICNTL(4) holds the value of IOVFLO.

ICNTL(5) holds the value of NEMIN.

ICNTL(6) to ICNTL(25) hold the 20 values in the array IFRLVL.

ICNTL(26) to ICNTL(30) 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 the subroutine to control numerical pivoting. Values greater than 0.5 are treated as 0.5 and less than -0.5 as -0.5. Its default value is 0.1. If U is positive, numerical pivoting will be performed. If U is non-positive, no pivoting will be performed, the subroutine will fail if a zero pivot is encountered, and a flag

(see section 2.3) will be set if not all pivots are of the same sign; the factorization will continue after a sign change is detected if U is zero but will exit immediately if U is less than zero.

If the system is definite, then setting U to zero will decrease the factorization time while still providing a stable decomposition. For problems requiring greater than average numerical care a higher value than the default would be advisable.

CNTL(2) and CNTL(3) are only altered in special circumstances and hold the values of FRATIO and PIVTOL described in Section 2.4.

CNTL(4) and CNTL(5) are not used.

On return from the three principal MA27 entries 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 IERROR) provides supplementary information when there is an error. For details, see section 2.3.

INFO(3) and INFO(4) (was NRLTOT and NIRTOT) give the total amount of REAL (DOUBLE PRECISION in the D version) and INTEGER words respectively required for a successful completion of MA27B/BD without the need for data compression provided no numerical pivoting is performed. The actual amount required may be higher because of numerical pivoting, but probably not by more than 3%.

INFO(5) and INFO(6) (was NRLNEC and NIRNEC) give the amount of REAL (DOUBLE PRECISION in the D version) and INTEGER words required respectively for successful completion of MA27B/BD allowing data compression (see NCMPBR returned in INFO(12)), again provided no numerical pivoting is performed. Numerical pivoting may cause a higher value to be required, but probably not by more than 3%. If storage was conserved by equivalencing IW(1) with IRN(1), NRLNEC and NIRNEC cannot be calculated exactly but instead an upper bound will be returned. Experience has shown that this can overestimate the exact values by 50% although the tightness of the bound is very problem dependent. For example, a tight bound will generally be obtained if there are many more entries in the factors than in the input matrix.

INFO(7) and INFO(8) (was NRLADU and NIRADU) give the number of REAL (DOUBLE PRECISION in the D version) and INTEGER words required to hold the matrix factors if no numerical pivoting is performed by MA27B/BD. Numerical pivoting may change this slightly.

INFO(9) and INFO(10) (was NRLBDU and NIRBDU) give the amount of REAL (DOUBLE PRECISION in the D version) and INTEGER words actually used to hold the factorization.

INFO(11) (was NCMPA) holds the number of compresses of the internal data structure performed by MA27A/AD. If this is high (say > 10), the performance of MA27A/AD may be improved by increasing the length of array IW.

INFO(12) and INFO(13) (was NCMPBR and NCMPBI) hold the number of compresses of the real and integer data structure respectively required by the factorization. If either of these is high (say > 10), then the speed of the factorization may be increased by allocating more space to the arrays A or IW as appropriate.

INFO(14) (was NTWO) gives the number of 2×2 pivots used during the factorization.

INFO(15) (was NEIG) gives the number of negative eigenvalues of A .

INFO(16) to INFO(20) are not used.

2.3 Error diagnostics

A successful return from MA27A/AD or MA27B/BD is indicated by a value of INFO(1) equal to zero. There are no error returns from MA27C/CD. Possible nonzero values for INFO(1) are given below. In each case an identifying message is output on unit ICNTL(1) (errors) or ICNTL(2) (warnings).

-1 Value of N out of range. $N < 1$. (MA27A/AD and MA27B/BD entries).

- 2 Value of NZ out of range. $NZ < 0$. (MA27A/AD and MA27B/BD entries).
- 3 Failure due to insufficient space allocated to array IW (MA27A/AD and MA27B/BD entries). INFO(2) is set to a value that may suffice.
- 4 Failure due to insufficient space allocated to array A (MA27B/BD entry only). INFO(2) is set to a value that may suffice.
- 5 Matrix is singular (MA27B/BD entry only). INFO(2) is set to the pivot step at which singularity was detected.
- 6 A change of sign of pivots has been detected when U was negative. INFO(2) is set to the pivot step at which the change was detected. (MA27B/BD entry only).
- 7 Value of NSTEPS outside the range $1 \leq NSTEPS \leq N$ (MA27B/BD entry).

A positive flag value is associated with a warning message which will be output on unit ICNTL(2).

- +1 Index (in IRN or ICN) out of range. Action taken by subroutine is to ignore any such entries and continue (MA27A/AD and MA27B/BD entries). INFO(2) is set to the number of faulty entries. Details of the first ten are printed on unit ICNTL(2).
- +2 Pivots have different signs when factorizing a supposedly definite matrix (when the value of U in CNTL(1) is zero) (MA27B/BD entry only). INFO(2) is set to the number of sign changes. Note that this warning will overwrite an INFO(1)=1 warning. Details of the first ten are printed on unit ICNTL(2).
- +3 Matrix is rank deficient. In this case, a decomposition will still have been produced which will enable the subsequent solution of consistent equations (MA27B/BD entry only). INFO(2) will be set to the rank of the matrix. Note that this warning will overwrite an INFO(1)=1 or INFO(1)=2 warning.

2.4 Additional features

Two additional features are supported through the control elements CNTL(2) and CNTL(3).

CNTL(2) holds the value of FRATIO which is given the default value of 1.0 by MA27I/ID. If MA27A/AD encounters a row of the reduced matrix with a proportion of entries greater than FRATIO, the row is treated as full. FRATIO is not altered by MA27.

CNTL(3) holds the value of PIVTOL which is given the default value of 0.0 by MA27I/ID. MA27B/BD will not accept an entry with absolute value less than PIVTOL as a 1×1 pivot or the off-diagonal entry of a 2×2 pivot. PIVTOL is not altered by MA27.

The reason for introducing FRATIO is that MA27A/AD may be rather slow if there are a few rows that are full or nearly full. Code to circumvent this difficulty was added in 1999. The default value of 1.0 for FRATIO means that the code implements the minimum-degree algorithm, but tie-breaking may lead to a pivot sequence that is different from that of earlier versions of MA27. If the same sequence is desired, a value greater than 2.0 should be used. Our recommendation is for a smaller value such as 0.5, which is likely to execute faster in the presence of near-full rows without significant deterioration of the quality of the ordering. The code treats all the rows that it selects as near-full as if they had the sparsity pattern of their union; no attempt is made to follow later fill-in within this merged pattern.

The reason for introducing PIVTOL is to recognize matrices that are obviously near singular. Code was added in 2000.

3 GENERAL INFORMATION

Use of common: None.

Workspace:

MA27A/AD:
IW INTEGER of length LIW

IW1 INTEGER of length 2*N

MA27B/BD:

IW1 INTEGER of length N

MA27C/CD:

W REAL (DOUBLE PRECISION in the D version) of length at most N

IW1 INTEGER of length at most N

Other routines called directly: All the subroutines called by the principal subroutines are in the MA27 package. They are called MA27G/GD, MA27H/HD, MA27I/ID, MA27J/JD, MA27K/KD, MA27L/LD, MA27M/MD, MA27N/ND, MA27O/OD, MA27P/PD, MA27Q/QD, MA27R/RD, MA27U/UD.

Input/output: Error, warning and diagnostic messages only. Error messages on unit ICNTL(1) and warning and diagnostic messages on unit ICNTL(2). These have default value 6, and printing of these messages is suppressed if ICNTL(1) or ICNTL(2) is set to 0.

Restrictions:

$N \geq 1$,

$NZ \geq 0$.

4 METHOD

A version of sparse Gaussian elimination is used.

The MA27A/AD entry (with IFLAG=0) chooses pivots from the diagonal using the minimum degree criterion employing a generalized element model of the elimination thus avoiding the need to store the filled-in pattern explicitly. The elimination is represented as an assembly and elimination tree with the order of elimination determined by a depth-first search of the tree.

The MA27B/BD entry factorizes the matrix by using the assembly and elimination ordering generated by MA27A/AD. At each stage in the multifrontal approach pivoting and elimination are performed on full submatrices and, when diagonal 1×1 pivots would be numerically unstable, 2×2 diagonal blocks are used. Thus MA27B/BD can be used to factor indefinite systems and will perform well on machines capable of vectorization.

The MA27C/CD entry uses the factors from MA27B/BD to solve systems of equations either by loading the appropriate parts of the vectors into an array of the current front-size and using full matrix code or by indirect addressing at each stage, whichever performs better.

A fuller account of this method is given by Duff and Reid (AERE-R.10533, 1982) and Duff and Reid, ACM Trans. Math. Software **9** (1983), 302-325.

5 EXAMPLE OF USE

We illustrate the use of the package on the solution of the single set of equations

$$\begin{pmatrix} 2 & 3 & & & \\ 3 & 0 & 4 & & 6 \\ & 4 & 1 & 5 & \\ & & 5 & 0 & \\ 6 & & & & 1 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 8 \\ 45 \\ 31 \\ 15 \\ 17 \end{pmatrix}$$

We have set ICNTL(3) to 2 so that all the information passed to and from the package is displayed in this small case. Note that this example does not illustrate all the facilities.

Program

```

C SIMPLE EXAMPLE OF USE OF MA27 PACKAGE
  INTEGER IRN(10),ICN(10),IW(40),IKEEP(15)
  INTEGER IW1(10),ICNTL(30),INFO(20)
  INTEGER LIW,LA,IFLAG,N,NZ,I,NSTEPS,MAXFRT
  DOUBLE PRECISION A(30),W(5),RHS(5),CNTL(5),OPS
C
C INITIALIZE MA27
  CALL MA27ID(ICNTL,CNTL)
C
C STORE ARRAY LENGTHS
  LIW = 40
  LA = 30
C
C ASK FOR FULL PRINTING FROM MA27 PACKAGE
  ICNTL(3) = 2
C
C SET IFLAG TO INDICATE PIVOT SEQUENCE IS TO BE FOUND BY MA27AD
  IFLAG = 0
C
C READ MATRIX AND RIGHT-HAND SIDE
  READ (5,*) N,NZ
  READ (5,*) (IRN(I),ICN(I),A(I),I=1,NZ)
  READ (5,*) (RHS(I),I=1,N)
C
C ANALYSE SPARSITY PATTERN
  CALL MA27AD(N,NZ,IRN,ICN,IW,LIW,IKEEP,IW1,NSTEPS,IFLAG,
+           ICNTL,CNTL,INFO,OPS)
C
C FACTORIZE MATRIX
  CALL MA27BD(N,NZ,IRN,ICN,A,LA,IW,LIW,IKEEP,NSTEPS,MAXFRT,
+           IW1,ICNTL,CNTL,INFO)
C
C SOLVE THE EQUATIONS
  CALL MA27CD(N,A,LA,IW,LIW,W,MAXFRT,RHS,IW1,NSTEPS,ICNTL,
+           INFO)
  STOP
  END

```

Data

```

5 7
1 1 2.0
1 2 3.0
2 3 4.0
2 5 6.0
3 3 1.0
3 4 5.0
5 5 1.0
8. 45. 31. 15. 17.

```

Output

```

ENTERING MA27AD WITH      N      NZ      LIW  IFLAG
                        5       7      40     0

MATRIX NON-ZEROS
      1      1      1      2      2      3      2      5
      3      3      3      4      5      5

LEAVING MA27AD WITH NSTEPS  INFO(1)  OPS IERROR NRLTOT NIRTOT
                        4          0    4.    0     19     25

```

	NRLNEC	NIRNEC	NRLADU	NIRADU	NCMPA
	14	20	9	14	0
IKEEP(.,1)=	4	5	3	2	1
IKEEP(.,2)=	1	1	1	2	
IKEEP(.,3)=	0	0	2	1	

ENTERING MA27BD WITH

	N	NZ	LA	LIW	NSTEPS	U
	5	7	30	40	4	1.00D-01

MATRIX NON-ZEROS

2.000D+00	1	1	3.000D+00	1	2
4.000D+00	2	3	6.000D+00	2	5
1.000D+00	3	3	5.000D+00	3	4
1.000D+00	5	5			

IKEEP(.,1)= 4 5 3 2 1
IKEEP(.,2)= 1 1 1 2
IKEEP(.,3)= 0 0 2 1

LEAVING MA27BD WITH

MAXFRT	INFO(1)	NRLBDU	NIRBDU	NCMPBR	NCMPBI	NTWO	IERROR
3	0	10	13	0	0	1	0

BLOCK PIVOT = 1 NROWS = 1 NCOLS = 2
COLUMN INDICES = 5 2
REAL ENTRIES .. EACH ROW STARTS ON A NEW LINE
1.000D+00 -6.000D+00

BLOCK PIVOT = 2 NROWS = 2 NCOLS = 3
COLUMN INDICES = -4 3 2
REAL ENTRIES .. EACH ROW STARTS ON A NEW LINE
-4.000D-02 2.000D-01 -8.000D-01
0.000D+00 0.000D+00

BLOCK PIVOT = 3 NROWS = 2 NCOLS = 2
COLUMN INDICES = 1 2
REAL ENTRIES .. EACH ROW STARTS ON A NEW LINE
5.000D-01 -1.500D+00
-2.469D-02

ENTERING MA27CD WITH

	N	LA	LIW	MAXFRT	NSTEPS
	5	30	40	3	4

BLOCK PIVOT = 1 NROWS = 1 NCOLS = 2
COLUMN INDICES = 5 2
REAL ENTRIES .. EACH ROW STARTS ON A NEW LINE
1.000D+00 -6.000D+00

BLOCK PIVOT = 2 NROWS = 2 NCOLS = 3
COLUMN INDICES = -4 3 2
REAL ENTRIES .. EACH ROW STARTS ON A NEW LINE
-4.000D-02 2.000D-01 -8.000D-01
0.000D+00 0.000D+00

BLOCK PIVOT = 3 NROWS = 2 NCOLS = 2
COLUMN INDICES = 1 2
REAL ENTRIES .. EACH ROW STARTS ON A NEW LINE
5.000D-01 -1.500D+00
-2.469D-02

RHS 8.000D+00 4.500D+01 3.100D+01 1.500D+01 1.700D+01

LEAVING MA27CD WITH

RHS	1.000D+00	2.000D+00	3.000D+00	4.000D+00	5.000D+00
-----	-----------	-----------	-----------	-----------	-----------