

1 SUMMARY

To solve a sparse system of linear equations whose non-zero structure is symmetric or nearly so. Given a sparse matrix $\mathbf{A} = \{a_{ij}\}_{n \times n}$ and an n -vector \mathbf{b} , this subroutine solves the system $\mathbf{Ax}=\mathbf{b}$ or $\mathbf{A}^T\mathbf{x}=\mathbf{b}$. The matrix \mathbf{A} need not be symmetric but the subroutine will perform badly if the sparsity pattern is not nearly symmetric.

The method used is a direct method based on a sparse variant of Gaussian elimination and is discussed further by Duff and Reid, SIAM J. Sci. Stat. Comput. **5** (1984), 633-641.

ATTRIBUTES — **Version:** 1.0.0. **Types:** MA37A, MA37AD. **Original date:** January 1983. **Origin:** I. S. Duff and J. K. Reid, Harwell.

2 HOW TO USE THE PACKAGE

2.1 Argument lists and calling sequences

There are three entries:

- MA37A/AD accepts the pattern of \mathbf{A} and chooses pivots for Gaussian elimination using a selection criterion to preserve sparsity and subsequently constructs subsidiary information for actual factorization by MA37B/BD. The user may input a given pivot sequence in which case only the necessary information for MA37B/BD will be generated.
- MA37B/BD factorizes a matrix \mathbf{A} using the information from a previous call to MA37A/AD. The actual pivot sequence used may differ slightly from that produced by MA37A/AD if \mathbf{A} is not diagonally dominant.
- MA37C/CD uses the factors generated by MA37B/BD to solve a system of equations $\mathbf{Ax}=\mathbf{b}$ or $\mathbf{A}^T\mathbf{x}=\mathbf{b}$.

A call to MA37C/CD must be preceded by a call to MA37B/BD, which in turn must be preceded by a call to MA37A/AD. Since the information passed from one subroutine to the next is not corrupted by the second, several calls to MA37B/BD for matrices with the same sparsity pattern but different values may follow a single call to MA37A/AD, and similarly MA37C/CD can be used repeatedly to solve for different right hand sides \mathbf{b} .

2.1.1 The argument list for MA37A/AD

To perform symbolic manipulations

The single precision version

```
CALL MA37A(N,NZ,IRN,ICN,IW,LIW,IKEEP,IW1,NSTEPS,IFLAG)
```

The double precision version

```
CALL MA37AD(N,NZ,IRN,ICN,IW,LIW,IKEEP,IW1,NTSEPS,IFLAG)
```

N is an INTEGER variable which must be set by the user to the order of the matrix \mathbf{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, ICN are INTEGER arrays of length **NZ**. The user must set them so that each entry a_{ij} is represented by $IRN(K)=i$ and $ICN(K)=j$, $K=1, \dots, NZ$. 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 array is used by the subroutine as workspace. 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 percent greater than this (see NCMPA in section 2.2).

LIW is an INTEGER variable. It must be set by the user to the length of the array IW. It 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 MA37A/AD and subsequent calls to MA37B/BD. If the user wishes to input a given permutation, 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. Note that the given order may be replaced by another that gives virtually identical numerical results.

IW1 is an INTEGER array of length $2*N$. It is used as workspace by the subroutine MA37A/AD.

NSTEPS is an INTEGER variable. It need not be set by the user on input and must be passed unchanged to MA37B/BD.

IFLAG is an INTEGER variable which the user must set to 0 if a suitable pivot order is to be chosen automatically or to 1 if the pivot order set in IKEEP is to be used. On exit from MA37A/AD, a value of zero indicates that the subroutine has performed successfully. For nonzero values, see section 2.3.

2.1.2 The argument list for MA37B/BD

To factorize a matrix

The single precision version

```
CALL MA37B(N,NZ,IRN,ICN,A,LA,IW,LIW,IKEEP,NSTEPS,MAXFRT,IFLAG)
```

The double precision version

```
CALL MA37BD(N,NZ,IRN,ICN,A,LA,IW,LIW,IKEEP,NSTEPS,MAXFRT,IFLAG)
```

N is an INTEGER variable which must be set by the user to the order 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 in the matrix **A**. It is not altered by the subroutine. **Restriction:** $NZ \geq 0$.

IRN, ICN, A. IRN and ICN are INTEGER arrays of length NZ and A is a REAL (DOUBLE PRECISION in the D version) array of length LA. These arrays must be set by the user to hold the entries of **A**. An entry a_{ij} is held as $A(K)=a_{ij}$, $IRN(K)=i$, $ICN(K)=j$, $K=1, \dots, NZ$. Multiple entries are summed and any entries with $IRN(K)$ or $ICN(K)$ out of range are ignored. On exit, the array A will hold the nonzero 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 MA37C/CD. The arrays 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 $IW(K)$, $K > NZ$. If this equivalence is used, note that the length of IW must be at least $2*NZ + 4*N$.

LA is an INTEGER variable which must be set by the user to the length of the array A. It must be at least as great as NRLNEC of common block MA37E/ED (see section 2.2), set by MA37A/AD. It is advisable to allow a slightly greater value because the use of numerical pivoting might increase storage requirements marginally. This argument is not altered by the subroutine.

IW is an INTEGER array of length at least as great as NIRNEC as output from MA37A/AD in common block MA37E/ED (see section 2.2). A slightly greater value is recommended because numerical pivoting may increase storage requirements marginally. If IRN/ICN have been equivalenced to IW, then the length of IW must be at least $2*NZ + 4*N$. The array IW, which need not be set by the user, is used as workspace by MA37B/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 MA37C/CD.

LIW is an INTEGER variable which must be set by the user to the length of the array IW. It is not altered by MA37B/BD.

IKEEP is an INTEGER array of length $3*N$ which must be passed unchanged since the last call to MA37A/AD. It is not

altered by MA37B/BD.

NSTEPS is an INTEGER variable which must be passed unchanged since the last call to MA37A/AD. It is not altered by MA37B/BD.

MAXFRT is an INTEGER variable which need not be set by the user and should be passed unchanged to subsequent calls of MA37C/CD.

IFLAG is an INTEGER variable. On exit from MA37A/AD, a value of zero indicates that the subroutine has performed successfully. For nonzero values, see section 2.3.

2.1.3 The argument list for MA37C/CD

To solve equations using the factors from MA37B/BD

The single precision version

```
CALL MA37C(N,A,LA,IW,LIW,W,MAXFRT,RHS,IW1,NSTEPS,W2,MTYPE)
```

The double precision version

```
CALL MA37CD(N,A,LA,IW,LIW,W,MAXFRT,RHS,IW1,NSTEPS,W2,MTYPE)
```

N is an INTEGER variable which must be set by the user to the order of the matrix **A**. It is not altered by the subroutine.

A is a REAL (DOUBLE PRECISION in the D version) array of length LA which must be unchanged since the last call to MA37B/BD. It is not altered by the subroutine.

LA is an INTEGER variable which must be set by the user to the length of the 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 MA37B/BD. It is not altered by the subroutine.

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

W is a REAL (DOUBLE PRECISION in the D version) array of length MAXFRT as output from MA37B/BD (see section 2.2). This length will be at most N and will often be substantially less. The array W is used as workspace by MA37C/CD.

MAXFRT is an INTEGER variable which must be passed unchanged since the last call to MA37B/BD. It is not altered by MA37C/CD.

RHS is a REAL (DOUBLE PRECISION in the D version) array of length N. On entry, RHS(I) must hold the I th component of the right hand side vector **b** of the equations being solved (I=1, . . . ,N). On exit it will be equal to the corresponding entry of the solution vector.

IW1 is an INTEGER array of length NSTEPS as output from MA37A/AD. This length will be at most N. The array IW1 is used as workspace by MA37C/CD.

NSTEPS is an INTEGER variable which must be passed unchanged since the last call to MA37A/AD. It is not altered by the subroutine.

W2 is a REAL (DOUBLE PRECISION in the D version) array of length N. This array is used as workspace by MA37C/CD.

MTYPE is an INTEGER variable. If it is set by the user to 1, the system $\mathbf{Ax}=\mathbf{b}$ will be solved. Other values will cause MA37C/CD to solve the system $\mathbf{A}^T\mathbf{x}=\mathbf{b}$.

2.2 Common blocks used

There are three common blocks used by the MA37 routines. The first one, MA37D/DD, allows the user to input parameters to control the solution process and diagnostic printing and the second, MA37E/ED, provides information on the decomposition. The third, MA37F/FD, allows control of certain system dependent parameters and is unlikely to be of concern to many users.

The first common block can be declared as:

The single precision version

```
COMMON/ MA37D /U,LP,MP,LDIAG
```

The double precision version

```
COMMON/ MA37DD /U,LP,MP,LDIAG
```

where all the values are given default values by a block data subprogram. None of these values is altered by the subroutines.

U is a REAL (DOUBLE PRECISION in the D version) variable which is used by the subroutine to control numerical pivoting. Its absolute value is used. Values greater than 1.0 are treated as 1.0. The default value is 0.1. If U is positive, numerical pivoting will be performed. If U is zero, no pivoting will be performed and the subroutine will fail if a zero pivot is encountered.

If the system is diagonally dominant, 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.

LP is an INTEGER variable 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.

MP is an INTEGER variable 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.

LDIAG is an INTEGER variable used by the subroutines to control diagnostic printing. If LDIAG 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 common blocks) and a few entries of array parameters on entry and exit from each subroutine, while LDIAG equal to 2 will print all parameter values on entry and exit.

The second common block is:

The single precision version

```
COMMON/ MA37E /OPSA, OPSB, IERROR, NRLTOT,
*NIRTOT, NRLNEC, NIRNEC, NRLB, NIRB, NRLADU,
*NIRADU, NRLBDU, NIRBDU, NCPA, NCMPBR,
*NCMPBI, NOFF
```

The double precision version

```
COMMON/ MA37ED /OPSA, OPSB, IERROR, NRLTOT,
*NIRTOT, NRLNEC, NIRNEC, NRLB, NIRB, NRLADU,
*NIRADU, NRLBDU, NIRBDU, NCPA, NCMPBR,
*NCMPBI, NOFF
```

where

OPSA is a REAL (DOUBLE PRECISION in the D version) variable. On output from MA37A/AD, OPSA will be set to the number of multiply-add pairs of operations required by the factorization if no pivoting is performed. Numerical pivoting in MA37B/BD may increase the number of operations slightly.

OPSB is a REAL (DOUBLE PRECISION in the D version) variable. On output from MA37B/BD, OPSB will be set to the number of multiply-add pairs of operations actually required by the factorization.

IERROR is an INTEGER variable. In the case of an error condition, extra information is placed here. For details, see section 2.3.

NRLTOT, NIRTOT are INTEGER variables. On exit from MA37A/AD, they give the total amount of REAL (or DOUBLE PRECISION for the D version) and INTEGER words, respectively, required for a successful completion of MA37B/BD without the need for data compression provided no numerical pivoting is performed. The actual amounts required may differ because of numerical pivoting.

NRLNEC, NIRNEC are INTEGER variables. On exit from MA37A/AD, they give the amount of REAL (or DOUBLE PRECISION for the D version) and INTEGER words required, respectively, for successful completion of MA37B/BD allowing data compression (see NCOMPBR), again provided no numerical pivoting is performed. Numerical pivoting may cause different values to be required. 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 percent although the tightness of the bound is very problem dependent. For example, a tight bound will generally be obtained if there are many more non-zeros in the factors than in the input matrix.

NRLB, NIRB are INTEGER variables. On exit from MA37B/BD, they give the total amount of REAL (or DOUBLE PRECISION for the D version) and INTEGER words, respectively, used during the factorization by MA37B/BD.

NRLADU, NIRADU are INTEGER variables. On exit from MA37A/AD, they give the number of REAL (or DOUBLE PRECISION for the D version) and INTEGER words required, respectively, to hold the matrix factors if no numerical pivoting is performed by MA37B/BD. Numerical pivoting may change this slightly.

NRLBDU, NIRBDU are INTEGER variables. On exit from MA37B/BD, they give the amount of REAL (or DOUBLE PRECISION for the D version) and INTEGER words, respectively, actually used to hold the factors of the matrix.

NCMPA is an INTEGER variable. On exit from MA37A/AD, NCMPA will hold the number of compresses of the internal data structure performed by MA37A/AD. If this is high (say > 10), the performance of MA37A/AD may be improved by increasing the length of array IW.

NCOMPBR, NCOMPBI are INTEGER variables. On exit from MA37B/BD, they will 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.

NOFF is an INTEGER variable. On exit from MA37B/BD, this gives the number of off-diagonal pivots used during the factorization.

The third common block is not of interest to the general user. It is discussed more fully by Duff and Reid (AERE R-10533, 1982). If the user is working in an environment where all common blocks must be declared in the calling program, a suitable declaration would be:

The single precision version

```
COMMON/ MA37F /IDUMMY(22)
```

The double precision version

```
COMMON/ MA37FD /IDUMMY(22)
```

where

IDUMMY is an INTEGER array of length 22.

2.3 Error diagnostics

A successful return from MA37A/AD or MA37B/BD is indicated by a value of IFLAG equal to zero. There are no error returns from MA37C/CD. Possible non-zero values for IFLAG are given below. In each case an identifying message is output on unit LP (errors) or MP (warnings).

- 1 Value of N out of range. $N < 1$. (MA37A/AD and MA37B/BD entries).
- 2 Value of NZ out of range. $NZ < 0$. (MA37A/AD and MA37B/BD entries).
- 3 Failure due to insufficient space allocated to the array IW (MA37A/AD and MA37B/BD entries). IERROR in COMMON is set to a value that may suffice.
- 4 Failure due to insufficient space allocated to the array A (MA37B/BD entry only). IERROR in COMMON is set to a value that may suffice.
- 5 Matrix is singular when user claimed that it was diagonally dominant by setting U (see section 2.2) to zero. (MA37B/BD entry only). IERROR in COMMON is set to the pivot step at which singularity was detected.

A positive value of IFLAG is associated with a warning message which will be output on unit MP.

- +1 Index (in IRN or ICN) out of range. Action taken by subroutine is to ignore any such entries and continue (MA37A/AD and MA37B/BD entries). IERROR in COMMON is set to the number of faulty entries. Details of the first ten are printed on unit MP.
- +2 Matrix is rank deficient. In this case, a decomposition will still have been produced which will enable the subsequent solution of consistent equations (MA37B/BD entry only). IERROR in COMMON will be set to the rank of the matrix. Note that this warning will overwrite an IFLAG=1 warning.

3 GENERAL INFORMATION

Use of common: The subroutines use common blocks MA37D/DD, MA37E/ED, MA37F/FD (see section 2.2).

Workspace:

MA37A/AD:
IW INTEGER array of length LIW
IW1 INTEGER array of length 2*N

MA37C/CD:
W REAL (DOUBLE PRECISION in the D version) array of length at most N
W2 REAL (DOUBLE PRECISION in the D version) array of length N
IW1 INTEGER array of length at most N

Other routines called directly: All the subroutines called by the principal subroutines are in the MA37 package. They are called MA37G/GD, MA37H/HD, MA37I/ID, MA37J/JD, MA37K/KD, MA37L/LD, MA37M/MD, MA37N/ND, MA37O/OD, MA37P/PD, MA37Q/QD, MA37R/RD, MA37S/SD, MA37T/TD, MA37U/UD.

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

Restrictions:

$N \geq 1$,
 $NZ \geq 0$.

4 METHOD

A version of sparse Gaussian elimination is used.

The MA37A/AD entry (with IFLAG=0) chooses pivots from the diagonal using the minimum degree criterion on the sparsity pattern of $\mathbf{A} + \mathbf{A}^T$, 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 MA37B/BD entry factorizes the matrix by using the assembly and elimination ordering generated by MA37A/AD. At each stage in the multifrontal approach, pivoting and elimination are performed on full submatrices. Thus MA37B/BD can be used to factor general systems and will perform well on machines capable of vectorization.

The MA37C/CD entry uses the factors from MA37B/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 discussion of the design of this package is given by Duff and Reid (SIAM J. Sci. Stat. Comput. **5**, 1984, 633-641). The software is organized similarly to the MA27 package discussed by Duff and Reid (AERE R-10533, 1982).

5 EXAMPLE OF USE

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

$$\begin{pmatrix} 2 & 3 & 4 & & \\ 3 & & -3 & 6 & \\ & -1 & 1 & 2 & \\ & & 2 & & \\ 4 & & & & 1 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 20 \\ 24 \\ 9 \\ 6 \\ 13 \end{pmatrix}$$

We have set LDIAG 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 MA37 PACKAGE
  INTEGER IRN(12),ICN(12),IW(60),IKEEP(15),IW1(12),
  *      LP,MP,LDIAG,LIW,LA,MAXFRT,N,NZ,I,IFLAG
  DOUBLE PRECISION U,A(50),W(5),RHS(5),W2(5)
  COMMON /MA37DD/ U,LP,MP,LDIAG
C
C STORE ARRAY LENGTHS
  LIW=60
  LA=50
C
C ASK FOR FULL PRINTING FROM MA37 PACKAGE
  LDIAG=2
C
C SET IFLAG TO INDICATE PIVOT SEQUENCE IS TO BE FOUND BY MA37AD
  IFLAG=0
C
C READ MATRIX AND RIGHT-HAND SIDE
  READ(5,10)N,NZ
10  FORMAT(2I3,F5.1)
  READ(5,10)(IRN(I),ICN(I),A(I),I=1,NZ)
  READ(5,20)(RHS(I),I=1,N)
20  FORMAT(5F5.1)
C
C ANALYZE SPARSITY PATTERN
  CALL MA37AD(N,NZ,IRN,ICN,IW,LIW,IKEEP,IW1,NSTEPS,IFLAG)
C JUMP IF ERROR RETURN
  IF (IFLAG.LT.0) GO TO 50
C
C FACTORIZE MATRIX
  CALL MA37BD(N,NZ,IRN,ICN,A,LA,IW,LIW,IKEEP,NSTEPS,MAXFRT,
  *      IFLAG)
C JUMP IF ERROR RETURN
  IF (IFLAG.LT.0) GO TO 50
C
C SOLVE THE EQUATIONS
  CALL MA37CD(N,A,LA,IW,LIW,W,MAXFRT,RHS,IW1,NSTEPS,W2,1)
50  CONTINUE
  STOP
  END

```

Data

```

5 12
1 1 2.0
1 2 3.0
1 3 4.0
2 1 3.0
2 3 -3.0
2 5 6.0
3 2 -1.0
3 3 1.0
3 4 2.0
5 5 1.0
4 3 2.0
5 2 4.0
20. 24. 9. 6. 13.

```

Output

```

ENTERING MA37AD WITH      N      NZ      LIW  IFLAG
                          5      12      60    0
MATRIX NON-ZEROS         1      1      1      2      1      3
                          2      1      2      3      2      5
                          3      2      3      3      3      4
                          5      5      4      3      5      2

LEAVING MA37AD WITH NSTEPS  IFLAG  IERROR  NRLTOT  NIRTOT      OPSA
                          3      0      0      35      56      7.
                          NRLNEC  NIRNEC  NRLADU  NIRADU  NCMPE
                          25      46      15      21      0
IKEEP(.,1)=      5      4      3      2      1
IKEEP(.,2)=      1      1      3
IKEEP(.,3)=      0      0      2

ENTERING MA37BD WITH      N      NZ      LA      LIW  NSTEPS      U
                          5      12      50      60    3      0.100
MATRIX NON-ZEROS      2.000D+00      1      1      3.000D+00      1      2
                          4.000D+00      1      3      3.000D+00      2      1
                          -3.000D+00      2      3      6.000D+00      2      5
                          -1.000D+00      3      2      1.000D+00      3      3
                          2.000D+00      3      4      1.000D+00      5      5
                          2.000D+00      4      3      4.000D+00      5      2
IKEEP(.,1)=      5      4      3      2      1
IKEEP(.,2)=      1      1      3
IKEEP(.,3)=      0      0      2
LEAVING MA37BD WITH MAXFRT  IFLAG  NRLBDU  NIRBDU  NCMPEB  NCMPEI      OPSB
                          4      0      19      17      0      0      15.
                          NRLB   NIRB  IERROR  NOFF
                          32      42      0      1
BLOCK PIVOT =      1  NPIV =      1  NFRONT =      2
ROW INDICES   =      5      2
COLUMN INDICES =      5      2
REAL ENTRIES .. EACH R AND C STARTS ON A NEW LINE
  1.000D+00  -6.000D+00
  4.000D+00
BLOCK PIVOT =      2  NPIV =      4  NFRONT =      4
ROW INDICES   =      4      3      2      1
COLUMN INDICES =      3      4      2      1
REAL ENTRIES .. EACH R AND C STARTS ON A NEW LINE
  5.000D-01  -5.000D-01  1.500D+00  -2.000D+00
  0.000D+00  0.000D+00  0.000D+00
  5.000D-01  0.000D+00  0.000D+00
 -1.000D+00  0.000D+00
 -4.167D-02  1.250D-01
  3.000D+00
  4.211D-01

ENTERING MA37CD WITH      N      LA      LIW  MAXFRT  NSTEPS
                          5      50      60    4      3
BLOCK PIVOT =      1  NPIV =      1  NFRONT =      2
ROW INDICES   =      5      2
COLUMN INDICES =      5      2
REAL ENTRIES .. EACH R/C STARTS ON A NEW LINE
  1.000D+00  -6.000D+00

```

```
4.000D+00
BLOCK PIVOT =      2  NPIV =      4  NFRONT =      4
ROW INDICES   =      4      3      2      1
COLUMN INDICES =      3      4      2      1
REAL ENTRIES .. EACH R/C STARTS ON A NEW LINE
  5.000D-01  -5.000D-01   1.500D+00  -2.000D+00
  0.000D+00   0.000D+00   0.000D+00
  5.000D-01   0.000D+00   0.000D+00
 -1.000D+00   0.000D+00
 -4.167D-02   1.250D-01
  3.000D+00
  4.211D-01
RHS
  2.000D+01   2.400D+01   9.000D+00   6.000D+00   1.300D+01

LEAVING MA37CD WITH
RHS
  1.000D+00   2.000D+00   3.000D+00   4.000D+00   5.000D+00
```