

## 1 SUMMARY

To solve a sparse Hermitian system of linear equations. Given an  $n \times n$  sparse Hermitian matrix  $\mathbf{A} = \{a_{ij}\}$  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. The real version, MA27, is discussed by Duff and Reid, AERE R.10533 (1982).

**ATTRIBUTES** — **Version:** 1.0.0. **Types:** ME27A, ME27AD. **Original date:** July 1982. **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:

- (a) ME27A/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 ME27B/BD. The user may input his own pivot sequence in which case only the necessary information for ME27B/BD will be generated.
- (b) ME27B/BD factorizes a matrix  $\mathbf{A}$  using the information from a previous call to ME27A/AD. The actual pivot sequence used may differ slightly from that produced by ME27A/AD if  $\mathbf{A}$  is not definite.
- (c) ME27C/CD uses the factors generated by ME27B/BD to solve a system of equations  $\mathbf{Ax}=\mathbf{b}$ .

A call to ME27C/CD must be preceded by a call to ME27B/BD which in turn must be preceded by a call to ME27A/AD. Since the information passed from one subroutine to the next is not corrupted by the second, several calls to ME27B/BD for matrices with the same sparsity pattern but different values may follow a single call to ME27A/AD, and similarly ME27C/CD can be used repeatedly to solve for different right hand sides  $\mathbf{b}$ .

### To perform symbolic manipulations

*The single precision version*

```
CALL ME27A(N,NZ,IRN,ICN,IW,LIW,IKEEP,IW1,NSTEPS,IFLAG)
```

*The double precision version*

```
CALL ME27AD(N,NZ,IRN,ICN,IW,LIW,IKEEP,IW1,NSTEPS,IFLAG)
```

**N** is an INTEGER variable which must be set by the user to the order  $n$  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 non-zeros 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 non-zero  $a_{ij}$  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 percent greater than this (see NCMFA in 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 ME27A/AD and subsequent calls to ME27B/BD. If the user wishes to input his own 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.

NSTEPS is an INTEGER variable. It need not be set by the user on input and should be passed unchanged when later calling ME27B/BD.

IFLAG is an INTEGER variable which the user must set to zero if he wishes a suitable pivot order to be chosen automatically or to 1 if he wishes the pivot order he has set in IKEEP to be used. On exit from ME27A/AD, a value of zero indicates that the subroutine has performed successfully. For non-zero values, see section 2.3.

### To factorize a matrix

#### The single precision version

```
CALL ME27B(N,NZ,IRN,ICN,A,LA,IW,LIW,
* IKEEP,NSTEPS,MAXFRT,IW1,IFLAG)
```

#### The double precision version

```
CALL ME27BD(N,NZ,IRN,ICN,A,LA,IW,LIW,
* IKEEP,NSTEPS,MAXFRT,IW1,IFLAG)
```

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 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 COMPLEX array (COMPLEX\*16 in the D version) of length LA. These must be set by the user to hold the non-zeros. A diagonal non-zero  $a_{ii}$  is held as  $A(k)=a_{ii}$ ,  $IRN(k)=ICN(k)=i$  and a pair of off-diagonal non-zeros  $a_{ij}=\text{conj}\{a_{ji}\}$  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 non-zero 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 ME27C/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 of common block ME27E/ED (see section 2.2), set by ME27A/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 as output from ME27A/AD in common block ME27E/ED (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 ME27B/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 ME27C/CD.

LIW is an INTEGER variable which must be set by the user to the length of array IW. It is not altered by ME27B/BD.

IKEEP is an INTEGER array of length  $3*N$  which must be passed unchanged since the last call to ME27A/AD. It is not altered by ME27B/BD.

NSTEPS is an INTEGER variable which must be passed unchanged since the last call to ME27A/AD. It is not altered by

ME27B/BD.

MAXFRT is an INTEGER variable which need not be set by the user and should be passed unchanged to subsequent calls of ME27C/CD.

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

IFLAG is an INTEGER variable. On exit from ME27A/AD, a value of zero indicates that the subroutine has performed successfully. For non-zero values, see section 2.3.

### To solve equations using the factors from ME27B/BD

#### The single precision version

```
CALL ME27C(N,A,LA,IW,LIW,W,MAXFRT,RHS,IW1,NSTEPS)
```

#### The double precision version

```
CALL ME27CD(N,A,LA,IW,LIW,W,MAXFRT,RHS,IW1,NSTEPS)
```

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.

A is a COMPLEX array (COMPLEX\*16 in the D version) of length LA which must be unchanged since the last call to ME27B/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 ME27B/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 COMPLEX array (COMPLEX\*16 in the D version) of length MAXFRT as output from ME27B/BD (see section 2.2). This value will always be less than  $n$ . W is used as workspace by ME27C/CD.

MAXFRT is an INTEGER variable which must be passed unchanged since the last call to ME27B/BD. It is not altered by ME27C/CD.

RHS is a COMPLEX array (COMPLEX\*16 in the D version) of length  $n$ . On entry,  $RHS(i)$  must hold the  $i^{\text{th}}$  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 array of length NSTEPS as output from ME27A/AD. This value will be at most  $n$ . IW1 is used as workspace by ME27C/CD.

NSTEPS is an INTEGER variable which must be passed unchanged since the last call to ME27A/AD. It is not altered by the subroutine.

## 2.2 Common blocks used

There are three common blocks used by the ME27 subroutines. The first one, ME27D/DD, allows the user to input parameters to control the solution process and diagnostic printing and the second, ME27E/ED, provides information on the decomposition. The third, ME27F/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/ME27D/U,LP,MP,LDIAG
```

#### The double precision version

COMMON/ME27DD/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. 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 and the subroutine will fail if a zero pivot is encountered. If U is non-positive and not all pivots are of the same sign a flag (see section 2.3) will be set and the factorization will continue if U is zero and will exit immediately a sign change is detected 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.

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/ME27E/OPS, IERROR, NRLTOT, NIRTOT,
*NRLNEC, NIRNEC, NRLADU, NIRADU, NRLBDU,
*NIRBDU, NCMPA, NCMPBR, NCMPBI, NTWO, NEIG
```

*The double precision version*

```
COMMON/ME27ED/OPS, IERROR, NRLTOT, NIRTOT,
*NRLNEC, NIRNEC, NRLADU, NIRADU, NRLBDU,
*NIRBDU, NCMPA, NCMPBR, NCMPBI, NTWO, NEIG
```

OPS is a REAL (DOUBLE PRECISION in the D version) variable. On output from ME27A/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 ME27B/BD may increase the number of operations slightly.

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 ME27A/AD, they give the total amount of COMPLEX (or COMPLEX\*16 for the D version) and INTEGER words respectively required for a successful completion of ME27B/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 percent.

NRLNEC, NIRNEC are INTEGER variables. On exit from ME27A/AD, they give the amount of COMPLEX (or COMPLEX\*16 for the D version) and INTEGER words required respectively for successful completion of ME27B/BD allowing data compression (see NCMPBR), again provided no numerical pivoting is performed. Numerical pivoting may cause a higher value to be required, but probably not by more than 3 percent. 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.

NRLADU, NIRADU are INTEGER variables. On exit from ME27A/AD, they give the number of COMPLEX (or COMPLEX\*16 for the D version) and INTEGER words required to hold the matrix factors if no numerical pivoting is performed by ME27B/BD. Numerical pivoting may change this slightly.

NRLBDU, NIRBDU are INTEGER variables. On exit from ME27B/BD, they give the amount of COMPLEX (or COMPLEX\*16 for the D version) and INTEGER words actually used to hold the factorization.

NCMPA is an INTEGER variable. On exit from ME27A/AD, NCMPA will hold the number of compresses of the internal data structure performed by ME27A/AD. If this is high (say > 10), the performance of ME27A/AD may be improved by increasing the length of array IW.

NCMPBR, NCMPBI are INTEGER variables. On exit from ME27B/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.

NTWO is an INTEGER variable. On exit from ME27B/BD, this gives the number of 2x2 pivots used during the factorization.

NEIG is an INTEGER variable. On exit from ME27B/BD this gives the number of negative eigenvalues of the matrix A.

Although 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/ME27F/IDUMMY(22)
```

*The double precision version*

```
COMMON/ME27FD/IDUMMY(22)
```

where

IDUMMY is an INTEGER array of length 22.

### 2.3 Error diagnostics

A successful return from ME27A/AD or ME27B/BD is indicated by a value of IFLAG equal to zero. There are no error returns from ME27C/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. Either  $N < 1$  (ME27A/AD and ME27B/BD entries).
- 2 Value of NZ out of range.  $NZ < 0$ . (ME27A/AD and ME27B/BD entries).
- 3 Failure due to insufficient space allocated to array IW (ME27A/AD and ME27B/BD entries). IERROR in COMMON is set to a value that may suffice.
- 4 Failure due to insufficient space allocated to array A (ME27B/BD entry only). IERROR in COMMON is set to a value that may suffice.
- 5 Matrix is singular (ME27B/BD entry only). IERROR in COMMON 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. IERROR in COMMON is set to the pivot step at which the change was detected. (ME27B/BD entry only).

-7 A complex diagonal entry has been found. IERROR in COMMON is set to k where the diagonal in error is the kth non-zero. (ME27B/BD entry only).

A positive flag value 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 (ME27A/AD and ME27B/BD entries). IERROR in COMMON is set to the number of faulty entries. Details of the first ten are printed on unit MP.

+2 Pivots have different signs when factorizing a supposedly definite matrix (input value of U is zero) (ME27B/BD entry only). IERROR in COMMON is set to the number of sign changes. Note that this warning will overwrite an IFLAG=1 warning. Details of the first ten are printed on unit MP.

+3 Matrix is rank deficient. In this case, a decomposition will still have been produced which will enable the subsequent solution of consistent equations (ME27B/BD entry only). IERROR in COMMON will be set to the rank of the matrix. Note that this warning will overwrite an IFLAG=1 or IFLAG=2 warning.

### 3 GENERAL INFORMATION

**Use of common:** The subroutines use common blocks ME27D/DD, ME27E/ED, ME27F/FD (see section 2.2).

**Workspace:**

ME27A/AD:  
IW (INTEGER) of length LIW  
IW1 (INTEGER) of length 2\*N

ME27B/BD:  
IW1 (INTEGER) of length N

ME27C/CD:  
W (COMPLEX COMPLEX\*16 for 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 ME27 package. They are called ME27G/GD, ME27H/HD, ME27I/ID, ME27J/JD, ME27K/KD, ME27L/LD, ME27M/MD, ME27N/ND, ME27O/OD, ME27P/PD, ME27Q/QD, ME27R/RD.

**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 ME27A/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 ME27B/BD entry factorizes the matrix by using the assembly and elimination ordering generated by ME27A/AD. At each stage in the multifrontal approach pivoting and elimination are performed on full submatrices

and, when diagonal  $1 \times 1$  pivots would be numerically unstable,  $2 \times 2$  diagonal blocks are used. Thus ME27B/BD can be used to factor indefinite systems and will perform well on machines capable of vectorization.

The ME27C/CD entry uses the factors from ME27B/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.

ME27 is a complex version of MA27, a full account of which is given 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

$$\begin{pmatrix} 2 & 3+2i & & & \\ 3-2i & 0 & 4+i & & 6+3i \\ & 4-i & 1 & 5+4i & \\ & & 5-4i & 0 & \\ 6-3i & & & & 1 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 8+4i \\ 45+16i \\ 31+14i \\ 15-12i \\ 17-6i \end{pmatrix}$$

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 ME27 PACKAGE
  INTEGER IRN(10),ICN(10),IW(40),IKEEP(15)
  INTEGER IW1(10)
  DOUBLE PRECISION U
  COMPLEX*16 A(30),W(5),RHS(5)
  COMMON /ME27DD/ U,LP,MP,LDIAG
C
C STORE ARRAY LENGTHS
  LIW=40
  LA=30
C
C ASK FOR FULL PRINTING FROM ME27 PACKAGE
  LDIAG=2
C
C SET IFLAG TO INDICATE PIVOT SEQUENCE IS TO BE FOUND BY ME27AD
  IFLAG=0
C
C READ MATRIX AND RIGHT-HAND SIDE
  READ(5,10)N,NZ
10  FORMAT(2I2,2F4.1)
  READ(5,10)(IRN(I),ICN(I),A(I),I=1,NZ)
  READ(5,20)(RHS(I),I=1,N)
20  FORMAT(10F4.1)
C
C ANALYZE SPARSITY PATTERN
  CALL ME27AD(N,NZ,IRN,ICN,IW,LIW,IKEEP,IW1,NSTEPS,IFLAG)
C
C FACTORIZE MATRIX
  CALL ME27BD(N,NZ,IRN,ICN,A,LA,IW,LIW,IKEEP,NSTEPS,MAXFRT,
*           IW1,IFLAG)
C
C SOLVE THE EQUATIONS
  CALL ME27CD(N,A,LA,IW,LIW,W,MAXFRT,RHS,IW1,NSTEPS)
  STOP
  END
```

## Data

```

5 7
1 1 2.0 0.0
1 2 3.0 2.0
2 3 4.0 1.0
2 5 6.0 3.0
3 3 1.0 0.0
3 4 5.0 4.0
5 5 1.0 0.0
8. 4. 45. 16. 31. 14. 15. -12. 17. -6.

```

## Output

```

ENTERING ME27AD 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 ME27AD WITH NSTEPS  IFLAG      OPS IERROR  NRLTOT  NIRTOT
                        4      0      4.     0     19     25
                        NRLNEC  NIRNEC  NRLADU  NIRADU  NCMPA
                        14     20      9     14     0
IKEEP(.,1)=      5      4      3     2     1
IKEEP(.,2)=      1      1      1     2
IKEEP(.,3)=      0      0      2     1

ENTERING ME27BD WITH      N      NZ      LA      LIW  NSTEPS      U
                        5      7      30     40      4     0.10
MATRIX NON-ZEROS      0.2000D+01      0.0000D+00      1     1
                        0.3000D+01      0.2000D+01      1     2
                        0.4000D+01      0.1000D+01      2     3
                        0.6000D+01      0.3000D+01      2     5
                        0.1000D+01      0.0000D+00      3     3
                        0.5000D+01      0.4000D+01      3     4
                        0.1000D+01      0.0000D+00      5     5
IKEEP(.,1)=      5      4      3     2     1
IKEEP(.,2)=      1      1      1     2
IKEEP(.,3)=      0      0      2     1

LEAVING ME27BD WITH MAXFRT  IFLAG  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  0.000D+00 -6.000D+00  3.000D+00
BLOCK PIVOT =      2 NROWS =      2 NCOLS =      3
COLUMN INDICES =     -4      3      2
REAL ENTRIES .. EACH ROW STARTS ON A NEW LINE
 -2.439D-02  0.000D+00  1.220D-01 -9.756D-02 -3.902D-01  5.122D-01
  0.000D+00  0.000D+00  0.000D+00  0.000D+00
BLOCK PIVOT =      3 NROWS =      2 NCOLS =      2
COLUMN INDICES =      2      1
REAL ENTRIES .. EACH ROW STARTS ON A NEW LINE
 -2.222D-02  0.000D+00  6.667D-02 -4.444D-02

```

4.369D-01 0.000D+00

```
ENTERING ME27CD 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  0.000D+00 -6.000D+00  3.000D+00
BLOCK PIVOT =          2 NROWS =          2 NCOLS =          3
COLUMN INDICES =       -4      3      2
REAL ENTRIES .. EACH ROW STARTS ON A NEW LINE
 -2.439D-02  0.000D+00  1.220D-01 -9.756D-02 -3.902D-01  5.122D-01
  0.000D+00  0.000D+00  0.000D+00  0.000D+00
BLOCK PIVOT =          3 NROWS =          2 NCOLS =          2
COLUMN INDICES =        2      1
REAL ENTRIES .. EACH ROW STARTS ON A NEW LINE
 -2.222D-02  0.000D+00  6.667D-02 -4.444D-02
  4.369D-01  0.000D+00
RHS   8.000D+00  4.000D+00  4.500D+01  1.600D+01  3.100D+01
      1.400D+01  1.500D+01 -1.200D+01  1.700D+01 -6.000D+00
```

```
LEAVING ME27CD WITH
RHS   1.000D+00  0.000D+00  2.000D+00  0.000D+00  3.000D+00
      0.000D+00  4.000D+00  0.000D+00  5.000D+00  0.000D+00
```