Warning: Subroutine ME28 has been superseded by subroutine ME48 which uses improved algorithms; the use of the latter routine is recommended. The superseded routine may be removed from later releases of the library.

1 SUMMARY

To **solve a sparse system of linear complex equations.** Given a sparse matrix $\mathbf{A}=\{a_{ij}\}_{n\times n}$ with complex elements, this subroutine **decomposes A into factors,** and given a complex vector **b**, solves $\mathbf{Ax}=\mathbf{b}$ (or optionally $\mathbf{A}^T\mathbf{x}=\mathbf{b}$). It will decompose a new matrix having the same sparsity pattern as a previous one by using the same pivotal sequence taking much less processing time than the original factorization. The matrix **A** is also allowed to be singular or rectangular.

The method is a variant of Gaussian elimination for sparse systems and further information is given in Duff, AERE R.8730, (1977), and Duff, CSS 78 (1980).

ATTRIBUTES — Version: 1.0.0. **Types:** ME28A, ME28AD. **Calls:** ME30, ME20, ME22, ME23. **Original date:** August 1979. **Origin:** I.S.Duff, Harwell.

2 HOW TO USE THE PACKAGE

2.1 Argument lists and calling sequences

There are three entries:

(a) ME28A/AD decomposes **A** into factors using a pivotal strategy designed to compromise between maintaining sparsity and controlling loss of accuracy through roundoff.

(b) ME28B/BD factorizes a new matrix **A** of the same pattern, using the pivotal sequence determined by an earlier entry to ME28A/AD.

(c) ME28C/CD uses the factors produced by ME28A/AD (or ME28B/BD) to solve **Ax=b** or $\mathbf{A}^T$**x=b**.

ME28B/BD is much faster than ME28A/AD. In some applications it is expected that there will be many calls to ME28B/BD for each call to ME28A/AD. Also, it is expected that ME28C/CD may be called with many different vectors for the same matrix **A**.

We first describe the argument list for ME28A/AD. Reference should be made to this description for information on parameters which are common to ME28A/AD, ME28B/BD and ME28C/CD.

To decompose a matrix

The single precision version:

```
CALL ME28A(N,NZ,A,LICN,IRN,LIRN,ICN,U,IKEEP,IW,IFLAG)
```

The double precision version:

```
CALL ME28AD(N,NZ,A,LICN,IRN,LIRN,ICN,U,IKEEP,IW,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:** $1 \le n$.

NZ is an INTEGER variable which must be set by the user to the number of non-zeros in the matrix **A**. It is not

altered by the subroutine. **Restriction:** NZ≥1.

A is a REAL (DOUBLE PRECISION in the D version) two-dimensional array fo first dimension 2 and second dimension of length LICN and A(K), K=1,NZ must be set by the user to hold the non-zero elements of the matrix **A**. (A(1,*), A(2,*) containing the real and imaginary parts respectively). On exit, A holds the non-zero elements in the factors of the matrix **A**. It should be preserved by the user between calls to this subroutine and ME28C/CD.

LICN is an INTEGER variable which must be set by the user to the length of arrays A and ICN. Since the decomposition is returned in A and ICN, LICN should be large enough to accommodate this and should ordinarily be 2 to 4 times as large as NZ (see section 2.4). It is not altered by the subroutine. **Restriction:** LICN≥NZ.

IRN is an INTEGER array of length LIRN. On entry to ME28A/AD, IRN(K) must hold the row index of the non-zero stored in A(K), K=1,NZ. It is used as workspace by ME28A/AD, is altered by ME28A/AD, and need not be preserved for any subsequent calls.

LIRN is an INTEGER variable which must be set by the user to the length of array IRN. LIRN need not be as large as LICN, normally it will not need to be very much greater than NZ. It is not altered by the subroutine. **Restriction:** LIRN≥NZ.

ICN is an INTEGER array of length LICN. On entry ICN(K) must hold the column index of the non-zero stored in A(K), K=1,NZ. On output, it holds the column indices of the factors of the matrix **A**. These entries should be unaltered by the user between a call to this subroutine and subsequent calls to ME28B/BD or ME28C/CD.

U is a REAL (DOUBLE PRECISION in the D version) variable On input to ME28A/AD, the user should set U to a value between zero and one to control the choice of pivots. A value of 0.10 has been found to work well on test examples. The subroutine will not fail if U is outside the above range; values of U less than zero are treated as zero and values of U greater than one are treated as one. It is unaltered by the subroutine.

IKEEP is an INTEGER array of length 5*N. It need never be referenced by the user and should be preserved between calls to this subroutine and ME28B/BD or ME28C/CD.

IW is an INTEGER array of length 5*N. It is used as workspace by the subroutine.

IFLAG is an INTEGER variable. On exit from ME28A/AD, a value of zero indicates that the subroutine has performed successfully. For non-zero values, see section 2.3.

To decompose a matrix which has a similar structure to that previously decomposed by ME28A/AD

The single precision version:

```
CALL ME28B(N,NZ,A,LICN,IVECT,JVECT,ICN,IKEEP,IW,W,IFLAG)
```

The double precision version:

```
CALL ME28BD(N,NZ,A,LICN,IVECT,JVECT,ICN,IKEEP,IW,W,IFLAG)
```

The user must input his matrix in the same way in which he input the original matrix to ME28A/AD. In this case, the parameters are as follows:

N INTEGER variable equal to the order of the matrix. It is not altered by the subroutine.

NZ INTEGER variable equal to number of non-zeros in the matrix. It is not altered by the subroutine.

A is a REAL (DOUBLE PRECISION in the D version) two-dimensional array fo first dimension 2 and second dimension of length LICN. The user must set A(K), K=1,NZ to hold the non-zero elements of the matrix **A**. (A(1,*), A(2,*) containing the real and imaginary parts respectively). On exit, A holds the non-zero elements of the factors of the matrix **A**. It must be preserved by the user between calls to this subroutine and ME28C/CD.

LICN INTEGER variable equal to length of arrays A and ICN. It is not altered by the subroutine.

IVECT,JVECT INTEGER arrays of length NZ. IVECT(K) and JVECT(K) must contain respectively the row and column index of the non-zero stored in A(K), K=1,NZ. They are not altered by ME28B/BD.

The other parameters are as follows:

ICN,IKEEP are the INTEGER arrays (of lengths LICN, and 5*N, respectively) of the same names as in the previous call to ME28A/AD. They should be unchanged since this earlier call and they are not altered by ME28B/BD.

IW is an INTEGER array of length 4*N used as workspace by ME28B/BD.

W is a REAL (DOUBLE PRECISION in the D version) array of length 2*N which is used as workspace.

IFLAG is an INTEGER variable which will be set to zero on successful exit from ME28B/BD, otherwise it will have a non-zero value (see section 2.3).

To solve equations Ax=b or A^Tx=b, using the factors of A from ME28A/AD or ME28B/BD

The single precision version:

```
CALL ME28C(N,A,LICN,ICN,IKEEP,RHS,W,MTYPE)
```

The double precision version:

```
CALL ME28CD(N,A,LICN,ICN,IKEEP,RHS,W,MTYPE)
```

Information about the factors of **A** is communicated to this subroutine via the parameters N, A, LICN, ICN and IKEEP where:

N INTEGER variable equal to the order of the matrix. It is not altered by the subroutine.

A REAL (DOUBLE PRECISION in the D version) two dimensional array of first dimension 2 and second dimension of length LICN. It must be unchanged since the last call to ME28A/AD or ME28B/BD. It is not altered by the subroutine.

LICN is an INTEGER **variable** equal to the length of arrays A and ICN. It is not altered by the subroutine.

ICN,IKEEP are the INTEGER arrays (of lengths LICN and 5*N, respectively) of the same names as in the previous call to ME28A/AD. They should be unchanged since this earlier call and they are not altered by ME28C/CD.

The other parameters are as follows:

RHS is a REAL (DOUBLE PRECISION in the D version) two-dimensional array of first dimension 2 and second dimension of length N. The user must set RHS(1,I) and RHS(2,I) to contain the real and imaginary parts of the Ith component of the right hand side ($\mathbf{b}_I$) I=1,N. On exit, RHS(1,I) and RHS(1,I) contains the real and imaginary parts of the Ith component of the solution vector ($\mathbf{x}_I$), I=1,N.

W is a REAL (DOUBLE PRECISION in the D version) array of of length 2*N. It is used as workspace by ME28C/CD.

MTYPE is an INTEGER variable which the user must set to determine whether ME28C/CD will solve **Ax=b** (MTYPE equal to 1) or $\mathbf{A}^T\mathbf{x}=\mathbf{b}$ (MTYPE≠1, zero say). It is not altered by ME28C/CD.

2.2 Parameter usage summary

ME28A
Input N, NZ, A(LICN), LICN, IRN(LIRN), LIRN, ICN(LICN), U
Unchanged by ME28A N, NZ, LICN, LIRN, U
Output A, ICN, IKEEP(5*N), IFLAG
Work-arrays IW(5*N).

ME28B
Input (by user) NZ, A(LICN), IVECT(NZ), JVECT(NZ)
Input (from ME28A) N, ICN(LICN), LICN, IKEEP(5*N)
Unchanged BY ME28B N, NZ, ICN, LICN, IKEEP, IVECT, JVECT.
Output A, IFLAG.
Work-arrays IW(4*N), W(N).

ME28C
Input (by user) RHS(N), MTYPE
Input (from ME28A) N, ICN(LICN), LICN, IKEEP(5*N)
Input (from ME28A or ME28B) A(LICN)
Unchanged BY ME28C N, ICN, LICN, IKEEP, A, MTYPE
Output RHS
Work-arrays W(N)

2.3 Error diagnostics

A successful return from ME28A/AD or ME28B/BD is indicated by a value of IFLAG equal to zero. There are no error returns from ME28C/CD. Possible non-zero values for IFLAG are given below. Unless otherwise stated error returns are for both ME28A/AD and ME28B/BD entries:

–15 Non-zeros removed from structure during ME28A/AD because they are smaller than drop tolerance (see common block ME28J/JD in section 2.4) as non-zero set for original factors may not be same as that for decomposition with ME28B/BD. (ME28B/BD entry only)

–14 to –8 Error in user's input matrix. The nature is specified in an output message.

–14 More than one non-zero in same position in matrix. Action taken is to proceed with value equal to sum of duplicate elements. (See common block variable MP in section 2.4).

–13 Non-zero was not present in factors after previous call to ME28A/AD. (ME28B/BD entry only).

–12 Row or column index out-of-range.

–11 1≤N violated.

–10 NZ≤0

–9 LICN<NZ

–8 LIRN<NZ (ME28A/AD entry only)

–7 Insufficient space for block triangularization phase. (ME28A/AD entry only)

–6 to –3 Storage allocation for decomposition is insufficient (see common block variables MINICN and MINIRN, section 2.4) (all ME28A/AD only).

–6 LIRN and LICN too small....information available from MINICN (see section 2.4).

–5 LICN too small....increase to at least value given by common block variable MINICN (see section 2.4).

–4 LICN far too small. No useful information in MINICN.

–3 LIRN too small.

–2 Matrix numerically singular.

–1 Matrix structurally singular. This means that the non-zero pattern is such that the matrix will be singular for all possible numerical values of the non-zeros (ME28A/AD only).

+1 Successful decomposition on a structurally singular matrix (ME28A/AD only).

+2 Successful decomposition on a numerically singular matrix (ME28A/AD only).

+I (I=1,2,..,N) Warning. Very small pivot in row I (ME28B/BD only).

2.4 Common blocks used

In single precision version:

```
COMMON/ME28E/LP,MP,LBLOCK
COMMON/ME28F/EPS,RMIN,RESID,IRNCP,ICNCP,MININRN,MINICN,IRANK,
* ABORT1,ABORT2
COMMON/ME28G/ IDISP
COMMON/ME28J/ TOL,THEMAX,BIG,IDROP,LBIG
```

In double precision version:

```
COMMON/ME28ED/LP,MP,LBLOCK
COMMON/ME28FD/EPS,RMIN,RESID,IRNCP,ICNCP,MININRN,MINICN,IRANK,
* ABORT1,ABORT2
COMMON/ME28GD/ IDISP
COMMON/ME28JD/ TOL,THEMAX,BIG,IDROP,LBIG
```

LP,MP are INTEGER variables used by the subroutine as the unit numbers for its warning and diagnostic messages. Default value for both is 6 (for line printer output). The user can either reset them to a different stream number or suppress the output by setting them to zero. While LP directs the output of error diagnostics from the subroutines themselves and internally called subroutines, MP controls only the output of a message which warns the user that he has input two or more non-zeros A(I), . . ,A(K) with the same row and column indices. The action taken in this case is to proceed using a numerical value of A(I)+...+A(K). In the absence of other errors, IFLAG will equal –14 on exit (see section 2.3).

LBLOCK is a LOGICAL which controls an option of first preordering the matrix to block lower triangular form (using Harwell subroutine ME23A) (see section 2.6). The preordering is performed if LBLOCK is equal to its default value of .TRUE. If LBLOCK is set to .FALSE. , the option is not invoked and the space allocated to IKEEP can be reduced to 4*N+1.

EPS,RMIN are REAL (DOUBLE PRECISION in the D version) variables. If, on entry to ME28B/BD, EPS is less than one, then RMIN will give the smallest ratio of the pivot to the largest element in the corresponding row of the upper triangular factor thus monitoring the stability of successive factorizations. If RMIN becomes very small and BIG from ME28B/BD is also very large, it may be advisable to perform a new decomposition using ME28A/AD.

RESID is a REAL (DOUBLE PRECISION in the D version) variable which on exit from ME28C/CD gives the value of the maximum residual

$$\max_i |b_i - \sum_j a_{ij} x_j|$$

over all the equations unsatisfied because of dependency (zero pivots) (see section 2.5).

IRNCP,ICNCP are INTEGER variables which monitor the adequacy of 'elbow room' in IRN and A/ICN respectively. If either is quite large (say greater than N/10), it will probably pay to increase the size of the corresponding array for subsequent runs. If either is very low or zero then one can perhaps save storage by reducing the size of the corresponding array.

MINIRN,MINICN are INTEGER variables which, in the event of a successful return (IFLAG ≥ 0 or IFLAG=–14) give the minimum size of IRN and A/ICN respectively which would enable a successful run on an identical matrix. On an exit with IFLAG equal to –5, MINICN gives the minimum value of ICN for success on subsequent runs on an identical matrix. In the event of failure with IFLAG= –6, –4, –3, –2, or –1, then MINICN and MINIRN give the minimum value of LICN and LIRN respectively which would be required for a successful decomposition up to the point at which the failure occurred.

IRANK is an INTEGER variable which gives an upper bound on the rank of the matrix.

ABORT1 is a LOGICAL variable with default value .TRUE. If ABORT1 is set to .FALSE. then ME28A/AD will decompose structurally singular matrices (including rectangular ones).

ABORT2 is a LOGICAL variable with default value .TRUE. If ABORT2 is set to .FALSE. then ME28A/AD will decompose numerically singular matrices.

IDISP is an INTEGER array of length 2. It is used to communicate information between ME28A/AD and ME28B/BD and ME28C/CD. It must not be altered by the user after a call to ME28A/AD. In particular, another call to ME28A/AD for a different matrix will normally alter it.

TOL is a REAL (DOUBLE PRECISION in the D version) variable with default value zero. If it is set to a positive value before entry to ME28A/AD, then non-zero values less than TOL in modulus will be dropped from the factors during decomposition.

THEMAX is a REAL (DOUBLE PRECISION in the D version) variable. On exit from ME28A/AD or ME28B/BD, it will equal the modulus of the largest element in the input matrix.

BIG is a REAL (DOUBLE PRECISION in the D version) variable which on exit from ME28A/AD or ME28B/BD will (see LBIG) yield the value of the largest element encountered during decomposition. If the matrix is well scaled (see section 2.7), then a high value for BIG indicates that the **LU** decomposition may be inaccurate and the user should be wary of his results and perhaps increase U for subsequent runs.

IDROP is an INTEGER variable. If TOL>0 on entry to ME28A/AD, then, on exit from ME28A/AD, IDROP will equal the number of non-zeros dropped from the sparsity structure because they were less than TOL.

LBIG is a LOGICAL variable with default value .TRUE. If it is set by the user to .FALSE. then the element of growth will not be returned in BIG. This will cause ME28B/BD to run about 10 to 15 percent faster and ME28A/AD to run about 5 percent faster. (see Duff, CSS 78 (1980)).

2.5 Singular or rectangular matrices

Singular and rectangular matrices can be handled by resetting common block variables ABORT1 and/or ABORT2 (see section 2.4). Additionally, the user must set N to the largest dimension of his system. RESID (see section 2.4) will give useful information on the consistency of the equations.

2.6 Block lower triangular form

Many large unsymmetric matrices can be permuted to the form

$$\mathbf{PAQ} = \begin{pmatrix} \mathbf{A}_{1,1} & & & & & \\ . & \mathbf{A}_{2,2} & & & & \\ . & . & \mathbf{A}_{3,3} & & & \\ . & . & . & . & & \\ . & . & . & . & . & \\ \mathbf{A}_{k,1} & . & . & . & . & \mathbf{A}_{k,k,} \end{pmatrix}$$

whereupon the system

$$\mathbf{Ax=b} \qquad (\mathbf{A}^T\mathbf{x=b})$$

can be solved by block-forward–(back–) substitution giving a saving in storage and execution time if the matrices $\mathbf{A}_{ii}$ are much smaller than **A**.

This facility is invoked as a default and information on the reordering is given by the variables in

Single precision version:

```
COMMON/ME23B/LP,NUMNZ,NUM,LARGE,ABORT
```

Double precision version:

```
COMMON/ME23BD/LP,NUMNZ,NUM,LARGE,ABORT
```

where the INTEGER variables NUMNZ, NUM, LARGE give the structural rank, number of diagonal blocks (K), and the order of the largest block respectively. ABORT is a logical variable set by ME28A/AD to the value of ABORT1 (see section 2.4).

If the user wishes to suppress this option he may do so by setting common block variable LBLOCK to .FALSE. He can then also reduce the length of IKEEP to 4N+1.

2.7 Badly-scaled systems

If the user's input matrix has elements differing widely in magnitude, then an inaccurate solution may be obtained and the increase in element size given by ME28A/AD or ME28B/BD (BIG, section 2.4) will not be very meaningful. In such cases, the user is advised to obtain scaling factors for his matrix and then explicitly scale it prior to calling ME28A/AD. Thereafter, both left and right hand sides should be correspondingly scaled.

3 GENERAL INFORMATION

Use of common: the subroutine uses common blocks ME28E/ED, ME28F/FD, ME28G/GD, ME28J/JD, see sections 2.4 and 2.6.

Workspace: W of length 2N (in ME28B/BD and ME28C/CD entries only).

Other subprograms: The following subroutines are called by the subroutines in this package. ME20A/AD, ME22A/AD, ME23A/AD, ME30A/AD.

Input/output: Errors and warning messages only. Error messages on unit LP, warning messages on unit MP. Both have default value 6, and output is suppressed if they are set to zero.

Restrictions:

$1 \leq n$,

$0 \leq$ NZ,

LICN $\geq$ NZ,

LIRN $\geq$ NZ.

4 METHOD

These subroutines are really only data management subroutines. A description of the initial subroutines called is given by Duff (AERE Report R.8730, 1977 and CSS 78, 1980). The method used is a sparse variant of Gaussian elimination.