1 SUMMARY

To **solve one or more sets of sparse linear complex equations, $\mathbf{Ax}=\mathbf{b}$, $\mathbf{A}^T\mathbf{x}=\mathbf{b}$, or $\mathbf{A}^H\mathbf{x}=\mathbf{b}$, by the frontal method,** optionally using direct access files for the matrix factors. ($\mathbf{A}^H$ denotes the conjugate transpose of $\mathbf{A}$). Use is made of high level BLAS kernels. The code has low in-core memory requirements. The complex matrix $\mathbf{A}$ may be input by the user in either of the following ways:

(i) by elements in a finite-element calculation,

(ii) by equations (matrix rows).

In both cases, the complex coefficient matrix and complex right-hand side(s) are of the form

$$\mathbf{A}=\sum_{k=1}^{m}\mathbf{A}^{(k)}, \quad \mathbf{b}=\sum_{k=1}^{m}\mathbf{b}^{(k)} .$$

In case (i), the summation is over finite elements. $\mathbf{A}^{(k)}$ is nonzero only in those rows and columns which correspond to variables in the k-th element. $\mathbf{b}^{(k)}$ is nonzero only in those rows which correspond to variables in element k.

In case (ii), the summation is over equations and $\mathbf{A}^{(k)}$ and $\mathbf{b}^{(k)}$ are nonzero only in row k.

In both cases, for each k, the user must supply a list specifying which columns of $\mathbf{A}$ are associated with $\mathbf{A}^{(k)}$, and arrays containing $\mathbf{A}^{(k)}$ and $\mathbf{b}^{(k)}$ in packed form. The nature of this packed form is defined more precisely in section 2.1.5 (arguments IVAR, AVAR, and RHS).

A principal feature of ME42 is that it can solve large problems in a predetermined and relatively small amount of in-core memory. At an intermediate stage of the solution, l say, the 'front' normally contains those variables associated with one or more of $\mathbf{A}^{(k)}$, $k=1,2,...,l$, which are also present in one or more of $\mathbf{A}^{(k)}$, $k=l+1,...,m$, although for stability reasons it may include a few more variables from $\mathbf{A}^{(k)}$, $k=1,2,...,l$. To keep the amount of in-core memory low, it is important the user orders the $\mathbf{A}^{(k)}$ so that the number of variables in the front (the 'frontwidth') is small. For example, a very rectangular grid should be ordered pagewise parallel to the short side of the rectangle. For finite-element calculations, we recommend using the Harwell Subroutine Library routine MC63 to obtain an efficient element ordering before ME42 is used. For equation entry, the equations should be ordered so that the matrix $\mathbf{A}$ is banded. MC62 may be used to obtain a good ordering.

ATTRIBUTES — Version: 1.2.1. (27 February 2023) **Types:** Real (single, double). **Remark:** ME42 is a complex version of MA42. **Helpful:** MC62, MC63. **Calls:** _AXPY,_GERU, _GEMV, _TPSV, _GEMM, _TRSM. **Language:** Fortran 77. **Original date:** March 1993. **Origin:** I.S. Duff and J.A. Scott, Rutherford Appleton Laboratory.

2 HOW TO USE THE PACKAGE

2.1 Argument lists and calling sequences

There are six entries:

(a) The initialization subroutine ME42I/ID must first be called. This subroutine need only be called once prior to calling ME42A/AD, ME42J/JD, ME42P/PD, ME42B/BD, and ME42C/CD.

(b) ME42A/AD must be called for each element or equation to specify which variables are associated with it.

(c) The use of ME42J/JD is optional. If the user wishes to perform a symbolic factorization, ME42J/JD must be called for each element or equation.

(d) The use of ME42P/PD is optional. If direct access files are to be used, ME42P/PD must be called once prior to calling ME42B/BD and ME42C/CD.

(e) ME42B/BD must be called for each element or equation to specify the nonzeros of $\mathbf{A}^{(k)}$ and, optionally, $\mathbf{b}^{(k)}$. ME42B/BD uses the data from ME42A/AD to factorize the matrix $\mathbf{A}=\sum_{k=1}^{m}\mathbf{A}^{(k)}$ and, if $\mathbf{b}^{(k)}$ are specified, ME42B/BD solves the equations $\mathbf{Ax}=\mathbf{b}$ with right-hand side(s) $\mathbf{b}=\sum_{k=1}^{m}\mathbf{b}^{(k)}$.

(f) The use of ME42C/CD is optional. ME42C/CD uses the factors produced by ME42B/BD to rapidly solve either further systems of the form $\mathbf{Ax}=\mathbf{b}$ or systems of the form $\mathbf{A}^{\mathrm{T}}\mathbf{x}=\mathbf{b}$ or $\mathbf{A}^{\mathrm{H}}\mathbf{x}=\mathbf{b}$.

2.1.1 The initialization subroutine

To initialise control parameters the user must first make a call of the following form:

The single precision version

```
CALL ME42I(ICNTL,CNTL,ISAVE)
```

The double precision version

```
CALL ME42ID(ICNTL,CNTL,ISAVE)
```

ICNTL is an INTEGER array of length 8 which need not be set by the user. This array is used to hold control parameters. On exit, ICNTL contains default values. If the user wishes to use values other than the defaults, the corresponding entries in ICNTL should be reset after the call to ME42I/ID. Details of the control parameters are given in section 2.2.1.

CNTL is an REAL (DOUBLE PRECISION in the D version) array of length 2 which need not be set by the user. This array is used to hold control parameters. On exit, CNTL contains default values. If the user wishes to use values other than the defaults, the corresponding entries in CNTL should be reset after the call to ME42I/ID. Details of the control parameters are given in section 2.2.1.

ISAVE is an INTEGER array of length 45 which need not be set by the user. This array is used to hold parameters which must be preserved between calls to routines in the ME42 package. After the call to ME42I/ID, the user may set ISAVE(19) to the minimum number of variables that are eliminated at once during the factorization (by default, this has value 1 but using a larger value may allow greater use to be made of Level 3 BLAS). In addition, the user should set ISAVE(20) to 2 if zeros within the frontal matrix are to be exploited (by default, ISAVE(20) = 1 and zeros are not exploited). Otherwise, ISAVE must not be changed by the user.

2.1.2 Specification of which variables belong in each element or equation

A call of the following form must be made for each element (case (i)) or each equation (case (ii)):

The single precision version

```
CALL ME42A(NVAR,IVAR,NDF,LAST,LENLST,ICNTL,ISAVE,INFO)
```

The double precision version

```
CALL ME42AD(NVAR,IVAR,NDF,LAST,LENLST,ICNTL,ISAVE,INFO)
```

NVAR is an INTEGER variable which must be set by the user to the number of variables in the element or equation. This argument is not altered by the routine. **Restriction:** NVAR $\geq$ 1.

IVAR is an INTEGER array of length at least NVAR which must be set by the user to contain the indices of the variables associated with the element or equation. These indices need not be in increasing order but must be distinct. This argument is not altered by the routine. **Restrictions:** 1 $\leq$ IVAR(I) $\leq$ LENLST and IVAR(I) $\neq$ IVAR(J), I, J = 1, 2, ..., NVAR.

NDF is an INTEGER variable which need not be set by the user. On each exit, it will be set to the largest integer so far used to index a variable. It must not be changed by the user between calls to ME42A/AD nor prior to subsequent calls to ME42J/JD and ME42B/BD. Note that if the variables are not numbered contiguously, NDF will exceed the number of variables in the problem (see INFO(3) in section 2.2.2).

LAST is an INTEGER array of length LENLST which need not be set by the user. On each exit, LAST(I) indicates the element or equation in which the variable with index I last appeared or, if it has not appeared, LAST(I) is zero , I = 1, 2,..., NDF. On exit from the final call, if I has been used to index a variable, LAST(I) is the element or equation at which variable I is fully summed and is zero otherwise (I = 1, 2,..., NDF). The first NDF entries of this array must not be changed between calls to ME42A/AD nor prior to subsequent calls to ME42J/JD and ME42B/BD.

LENLST is an INTEGER variable which must be set by the user to the dimension of array LAST. LENLST must be at least as large as the largest integer used to index a variable and must not be changed between calls to ME42A/AD. This argument is not altered by the routine. **Restriction:** LENLST ≥ 1.

ICNTL is an INTEGER array of length 8 which must be set by the user to hold control parameters. Default values are set by the call to ME42I/ID. Details of the control parameters are given in section 2.1.1. This argument is not altered by the routine.

ISAVE is an INTEGER array of length 45 which is used to hold parameters which must be preserved between calls to routines in the ME42 package. ISAVE must not be changed by the user.

INFO is an INTEGER array of length 23 which need not be set by the user. On successful exit, INFO(1) is set to 0. Negative values of INFO(1) indicate a fatal error has been detected (see section 2.3). INFO(I), I ≥ 2, is not accessed by the routine.

2.1.3 Symbolic factorization of A

The frontal method is a variant of Gaussian elimination and involves the factorization

$$\mathbf{A} = \mathbf{PLUQ},$$

where **P** and **Q** are permutation matrices, **L** is a lower triangular matrix, and **U** is an upper triangular matrix. In ME42, the row and column indices of the variables in **UQ** and **PL** are stored separately from the values of the entries in the factors **UQ** and **PL**. The indices require a file of length at most $(5+2maxf)n$ INTEGER words, where *maxf* is the maximum frontwidth and *n* is the order of the system. In practice, a file of length considerably less than $(5+2f)n$ INTEGER words, where *f* is the average frontwidth, will normally suffice. The factor **UQ** is stored with the corresponding right-hand sides and requires a file of length about $(f+nrhs)n$ COMPLEX (COMPLEX*16 in the D version) words, where *nrhs* is the number of right-hand sides input to ME42B/BD. If ME42C/CD is to be called, the factor **PL** requires a file of length about *fn* COMPLEX (COMPLEX*16 in the D version) words; otherwise **PL** need not be stored.

If the user wishes to compute lower bounds for the maximum frontwidth and for the lengths of the files required by **UQ** and **PL**, together with an estimate for the length of file required by the row and column indices, a symbolic factorization may be performed by making a call of the following form for each element (case (i)) or each equation (case (ii)). The elements/equations must have the same index lists and be in exactly the same order as when ME42A/AD was called.

Note that calling ME42J/JD is optional and that all the calls to ME42A/AD for a particular problem must be completed before ME42J/JD is called.

The single precision version

```
CALL ME42J(NVAR,IVAR,NDF,LAST,NMAXE,IFSIZE,ICNTL,ISAVE,INFO)
```

The double precision version

```
CALL ME42JD(NVAR,IVAR,NDF,LAST,NMAXE,IFSIZE,ICNTL,ISAVE,INFO)
```

NVAR and IVAR are as in the corresponding calls to ME42A/AD but ME42J/JD does not check IVAR for duplicate indices. NVAR and IVAR are not altered by the routine.

NDF is an INTEGER variable which must be unchanged since the final call to ME42A/AD. This argument is not altered by the routine.

LAST is an INTEGER array of length NDF which must be unchanged since the final call to ME42A/AD. This argument is not altered by the routine.

NMAXE is an INTEGER variable which must be set by the user to be greater than 1 if input is by elements and to 1 if input is by equations. NMAXE must be unchanged between calls to ME42J/JD. This argument is not altered by the routine. **Restriction:** NMAXE ≥ 1.

IFSIZE is an INTEGER array of length 5 which need not be set by the user. On exit from the final call to ME42J/JD, IFSIZE(I), I = 1, 2, hold lower bounds on the number of rows and columns required by the frontal matrix for a successful factorization. For element entry (NMAXE > 1), IFSIZE(1) = IFSIZE(2). In addition, IFSIZE(I), I = 3, 4, hold lower bounds on the length, in COMPLEX (COMPLEX*16) words, of the file required by the factors **UQ** and **PL**, respectively, and IFSIZE(5) holds an estimate of the length, in INTEGER words, of the file required by the row and column indices. For element entry, IFSIZE(3) = IFSIZE(4). Note that, because of the way singular matrices are treated, the bounds may not be reliable if the matrix is singular.

ICNTL, ISAVE, INFO are as in the calls to ME42A/AD.

2.1.4 To setup direct access files

In ME42, the user may choose whether to hold the files for **UQ** (including the corresponding right-hand sides), **PL**, and the row and column indices in-core or in direct access files. If direct access files are to be used, they are opened by ME42P/PD and during the matrix factorization (ME42B/BD), data is put into in-core buffers (workspace arrays) then, once a buffer is full, it is written to a direct access file. For efficiency, it is advisable that the COMPLEX (COMPLEX*16 in the D version) buffers are chosen to be significantly longer than the maximum frontwidth.

If the user wishes to use direct access files (which will keep in-core memory requirements small) a call of the following form must be made:

The single precision version

```
CALL ME42P(ISTRM,LENBUF,LENFLE,ICNTL,ISAVE,INFO)
```

The double precision version

```
CALL ME42PD(ISTRM,LENBUF,LENFLE,ICNTL,ISAVE,INFO)
```

ISTRM is an INTEGER array of length 3. ISTRM(1), ISTRM(2), and ISTRM(3) must be set by the user to specify the unit numbers of the direct access files for **UQ**, **PL**, and the row and column indices, respectively. If ME42C/CD is not going to be called, ISTRM(2) should be set to zero. This argument is not altered by the routine. **Restrictions:** ISTRM(1) and ISTRM(3) must lie in the range [1, 99], and ISTRM(2) must lie in the range [0, 99], ISTRM(I) ≠ 6, ICNTL(1), or ICNTL(2), and ISTRM(I) ≠ ISTRM(J) (I, J =1, 2, 3).

LENBUF is an INTEGER array of length 3. LENBUF(1) and LENBUF(2) must be set by the user to the lengths, in COMPLEX (COMPLEX*16) words, of the buffers associated with the direct access files for **UQ** (including the corresponding right-hand sides) and **PL**, respectively, and LENFLE(3) must be set by the user to the length, in INTEGER words, of the buffer associated with the direct access file for the row and column indices. LENBUF(I) (I =1, 2, 3) have a crucial effect on the amount of workspace required by ME42B/BD and ME42C/CD (see arguments LW and LIW in sections 2.1.5 and 2.1.6). If *maxf* is the maximum frontwidth and *nrhs* the number of right-hand sides to be input to ME42B/BD, for efficiency LENBUF(1) and LENBUF(3) should be at least 10(*maxf*+*nrhs*) and 10(2*maxf*+5), respectively. If ISTRM(2) = 0, LENBUF(2) need not be set by the user. Otherwise, a value for LENBUF(2) of at least 10*maxf* is recommended. LENBUF(1) and LENBUF(3) are not altered by the routine. LENBUF(2) is set to 0 by ME42P/PD if ISTRM(2) = 0 but is otherwise not altered by the routine. **Restriction:** LENBUF(I) ≥ 1, I = 1 and 3 and LENBUF(2) ≥ 1 if ISTRM(2) ≠ 0.

LENFLE is an INTEGER array of length 3. LENFLE(1) and LENFLE(2) must be set by the user to the lengths, in COMPLEX (COMPLEX*16) words, of the direct access files for **UQ** (including the corresponding right-hand sides) and **PL**, respectively, and LENFLE(3) must be set by the user to the length, in INTEGER words, of the direct access file for the row and column indices. If ME42J/JD has not been called, the user should follow the advice given at the beginning of section 2.1.3 on suitable lengths for these files. Otherwise, to allow pivots to be chosen to avoid numerical instability, LENFLE(1) should be set somewhat larger than IFSIZE(3) + NDF*NRHS, where NRHS is the number of right-hand sides to be input to ME42B/BD, and LENFLE(2) and LENFLE(3) should be set somewhat larger than IFSIZE(4) and IFSIZE(5), respectively (IFSIZE(I), I = 3, 4, 5 as output from the final call to ME42J/JD). If ISTRM(2) = 0, LENFLE(2) is not accessed. This argument is not altered by the routine. **Restriction:** LENFLE(I) ≥ LENBUF(I), I = 1 and 3, and LENFLE(2) ≥ LENBUF(2) if ISTRM(2) ≠ 0.

ICNTL, ISAVE, INFO are as in the calls to ME42A/AD with the restrictions ICNTL(3) > 0 and ICNTL(4) > 0.

2.1.5 To factorize A and optionally solve Ax = b

A call of the following form must be made for each element (case (i)) or each equation (case (ii)). The elements/equations must have the same index lists and be in exactly the same order as when ME42A/AD was called.

Note that all the calls to ME42A/AD (and to ME42J/JD if ME42J/JD is being used) for a particular problem must be completed before calling ME42B/BD.

The single precision version

```
      CALL ME42B(NVAR,IVAR,NDF,LAST,NMAXE,AVAR,NRHS,RHS,LRHS,
     *           LX,X,NFRONT,LENBUF,LW,W,LIW,IW,ICNTL,CNTL,
     *           ISAVE,INFO,RINFO)
```

The double precision version

```
      CALL ME42BD(NVAR,IVAR,NDF,LAST,NMAXE,AVAR,NRHS,RHS,LRHS,
     *            LX,X,NFRONT,LENBUF,LW,W,LIW,IW,ICNTL,CNTL,
     *            ISAVE,INFO,RINFO)
```

NVAR and IVAR are as in the corresponding calls to ME42A/AD but ME42B/BD does not check IVAR for duplicate indices. NDF and LAST are as in the corresponding calls to ME42J/JD. NVAR and NDF are not altered by the routine. IVAR is used locally by ME42B/BD as workspace but on exit holds the same data as on entry (although possibly reordered). Between calls to ME42B/BD, LAST is used as workspace and will be changed but on exit from the final call, LAST will be restored to its value on entry to the first call.

NMAXE is an INTEGER variable which must be set by the user to the first dimension of the arrays AVAR and RHS. NMAXE must be greater than 1 if input is by elements and must be equal to 1 if input is by equations. This argument is not altered by the routine. **Restriction:** NMAXE ≥ 1 and, if NMAXE > 1, NMAXE ≥ NVAR.

AVAR is a COMPLEX (COMPLEX*16 in the D version) array of dimensions NMAXE by NVAR which must be set by the user to contain $\mathbf{A}^{(k)}$ in packed form. That is, for element entry, AVAR(I,J) must contain the contribution to entry IVAR(I), IVAR(J) in the matrix **A** from the current element (I, J = 1, 2,...,NVAR). Contributions to the same entry from different elements are summed. For equation entry (NMAXE = 1), AVAR(1,J) must contain the coefficient of variable IVAR(J) in the current equation (J = 1, 2,...,NVAR). On exit, AVAR will contain the same data as on entry but if the entries of IVAR have been permuted, AVAR will also have been permuted.

NRHS is an INTEGER variable which must be set by the user to the number of right-hand sides and must not be changed between calls to ME42B/BD. This argument is not altered by the routine. **Restriction:** NRHS ≥ 0.

RHS is a COMPLEX (COMPLEX*16 in the D version) array of dimensions NMAXE by LRHS which must be set by the user to contain $\mathbf{b}^{(k)}$ in packed form. That is, for element entry, RHS(I,J) must contain the contribution to component IVAR(I) of the J-th right-hand side from the current element (I = 1, 2,..., NVAR, J = 1, 2,..., NRHS). Contributions to the same component from different elements are summed. For equation entry, RHS(1,J) must contain the contribution to the J-th right-hand side for the current equation (J = 1, 2,..., NRHS). This argument is changed by the routine.

LRHS is an INTEGER variable which must be set by the user to the second dimension of arrays RHS and X, and must not be changed between calls to ME42B/BD. This argument is not altered by the routine. **Restriction:** LRHS $\geq$ max(1, NRHS).

LX is an INTEGER variable which must be set by the user to the first dimension of array X. This argument is not altered by the routine. **Restriction:** If NRHS $\geq$ 1, LX $\geq$ NDF.

X is a COMPLEX (COMPLEX*16 in the D version) array of dimension LX by LRHS which need not be set by the user. If NRHS = 0, the array X is not accessed. Otherwise, on successful exit from the final call to ME42B/BD, if I has been used to index a variable, X(I,J) holds the solution for variable I to system J and is set to zero otherwise (I=1,2,...,NDF, J=1,2,..., NRHS).

NFRONT is an INTEGER array of length 2. NFRONT(I), I = 1, 2 must be set by the user to the greatest number of rows and columns, respectively, allowed in the frontal matrix. If input is by elements (NMAXE > 1), NFRONT(2) will be set equal to NFRONT(1) by ME42B/BD. NFRONT(I), I = 1, 2 have a crucial effect on the amount of in-core memory required (see LW and LIW below). A suitable choice is problem dependent. If ME42J/JD has been used, to allow pivots to be chosen to avoid numerical instability, the user should set NFRONT(I) somewhat larger than IFSIZE(I), I = 1, 2. If there is insufficient space for the factorization (INFO(1) = −12), on exit from the final call to ME42B/BD, NFRONT(1) and NFRONT(2) hold lower bounds on the dimensions of the frontal matrix necessary for a successful factorization. (See also the error return INFO(1) = 4 in section 2.3). **Restriction:** NFRONT(1) $\geq$ 1 and if NMAXE = 1, NFRONT(2) $\geq$ 1.

LENBUF is an INTEGER array of length 3. If the user is using direct access files, LENBUF must be unchanged since the call to ME42P/PD. Otherwise, LENBUF(1) and LENBUF(2) must be set by the user to the lengths, in COMPLEX (COMPLEX*16) words, of the files for **UQ** (including the corresponding right-hand sides) and **PL**, respectively, and LENBUF(3) must be set by the user to the length, in INTEGER words, of the file for the row and column indices. If the user has called ME42J/JD, to allow pivots to be chosen to avoid numerical instability, LENBUF(1) and LENBUF(3) should be set somewhat larger than IFSIZE(3) + NDF*NRHS and IFSIZE(5), respectively, and if ME42C/CD is to be called, LENBUF(2) should be set somewhat larger than IFSIZE(4). If ME42J/JD has not been called, the user should refer to the advice given at the beginning of section 2.1.3 on suitable lengths for these files. If the user does not want to call ME42C/CD, LENBUF(2) should be set to 0. This array must not be changed between calls to ME42B/BD. This argument is not altered by the routine. **Restriction:** LENBUF(1) $\geq$ 1, LENBUF(2) $\geq$ 0, and LENBUF(3) $\geq$ 1.

LW is an INTEGER variable which must be set by the user to the dimension of array W. It must be unchanged between calls to ME42B/BD. This argument is not altered by the routine. **Restriction:** LW $\geq$ 1 + LENBUF(1) + LENBUF(2) + max{LRHS*NFRONT(1), NRHS*NFRONT(2)} + NFRONT(1)*NFRONT(2).

W is a COMPLEX (COMPLEX*16 in the D version) array of length LW which is used as workspace by ME42B/BD. This array must be unchanged between calls to ME42B/BD. If direct access files are not being used (ME42P/PD not called), the first LENBUF(1)+LENBUF(2) entries of W must be preserved between the last call to ME42B/BD and any subsequent calls to ME42C/CD.

LIW is an INTEGER variable which must be set by the user to the dimension of array IW. It must not be changed between calls to ME42B/BD. This argument is not altered by the routine. **Restriction:** LIW $\geq$ LENBUF(3) +2*NFRONT(1) + 4*NFRONT(2).

IW is an INTEGER array of length LIW which is used as workspace by ME42B/BD. This array must be unchanged between calls to ME42B/BD. If direct access files are not being used (ME42P/PD not called), the first LENBUF(3) entries of IW must be preserved between the final call to ME42B/BD and any subsequent calls to ME42C/CD.

ICNTL is an INTEGER array of length 8 which must be set by the user to hold control parameters. Default values are set by the call to ME42I/ID. Details of the control parameters are given in section 2.2.1. If, after calling ME42P/PD, the user sets ICNTL(J) equal to ISTRM(I) for some I and J (I = 1, 2, or 3, J = 1 or 2), ICNTL(J) will be reset by the routine to the default value of 6. Otherwise this argument is not altered by the routine.

CNTL is an REAL (DOUBLE PRECISION in the D version) array of length 2 which must be set by the user to hold control parameters. Default values are set by the call to ME42I/ID. Details of the control parameters are given in section 2.2.1.

ISAVE is as in the calls to ME42A/AD.

INFO is an INTEGER array of length 23 which need not be set by the user. On successful exit, INFO(1) is set to 0. Negative values indicate a fatal error. Values greater than 0 are associated with a warning or non-terminal error. For nonzero values of INFO(1), see section 2.3. For details of the information contained in the other components of INFO, see section 2.2.2. This array must not be altered by the user.

RINFO is an REAL (DOUBLE PRECISION in the D version) array of length 2 which need not be set by the user. For details of the information contained in RINFO on exit from the final call to ME42B/BD, see section 2.2.2. This array must not be altered by the user.

2.1.6 To solve further systems $\mathbf{Ax}=\mathbf{b}$ or systems $\mathbf{A}^T\mathbf{x}=\mathbf{b}$ or $\mathbf{A}^H\mathbf{x}=\mathbf{b}$.

The single precision version

```
CALL ME42C(KIND,NRHS,LX,B,X,LW,W,LIW,IW,ICNTL,ISAVE,INFO)
```

The double precision version

```
CALL ME42CD(KIND,NRHS,LX,B,X,LW,W,LIW,IW,ICNTL,ISAVE,INFO)
```

KIND is an INTEGER variable which must be set by the user to 1 if systems of the form $\mathbf{Ax}=\mathbf{b}$ are to be solved, to 2 if systems of the form $\mathbf{A}^T\mathbf{x}=\mathbf{b}$ are to be solved, and to 3 if systems of the form $\mathbf{A}^H\mathbf{x}=\mathbf{b}$ are to be solved. This argument is not altered by the routine.

NRHS is an INTEGER variable which must be set by the user to the number of systems which are to be solved. This argument is not altered by the routine. **Restriction:** NRHS $\geq$ 1.

LX is an INTEGER variable which must be set by the user to the first dimension of arrays B and X. This argument is not altered by the routine. **Restriction:** LX $\geq$ NDF (NDF as output from the final call to ME42A/AD).

B is a COMPLEX (COMPLEX*16 in the D version) array of dimensions LX by NRHS which must be set by the user so that if I has been used to index a variable, B(I,J) is the corresponding component of the right-hand side for the J-th system (J=1,2,..., NRHS). This argument is changed by the routine.

X is a COMPLEX (COMPLEX*16 in the D version) array of dimension LX by NRHS which need not be set by the user. On exit, if I has been used to index a variable, X(I,J) holds the solution for variable I to system J and is set to zero otherwise (J=1,2,..., NRHS).

LW is an INTEGER variable which must be set by the user to the dimension of array W. A sufficient value for LW is L1+L2, where L1 = NRHS* max{INFO(8), INFO(9)} (INFO(8) and INFO(9) as output from the last call to ME42B/BD). If direct access files are not being used (ME42P/PD was not called), L2 = LENBUF(1)+LENBUF(2), otherwise, L2 = max{LENBUF(1), LENBUF(2)} + INFO(22)* max{INFO(8), INFO(9) + *nrhsb*}, where *nrhsb* is the number of right-hand sides on the calls to ME42B/BD and INFO(22) as output from the last call to ME42B/BD. This argument is not altered by the routine. **Restriction:** LW $\geq$ L1 + L2.

W is an COMPLEX (COMPLEX*16 in the D version) array of length LW. If direct access files are not being used (ME42P/PD was not called), the first LENBUF(1)+LENBUF(2) entries of W must be unchanged since the last call to ME42B/BD and these entries are unchanged by ME42C/CD. Otherwise, W is used by ME42C/CD as workspace.

LIW is an INTEGER variable which must be set by the user to the dimension of array IW. If direct access files are not being used (ME42P/PD was not called), LIW must be at least L1 = LENBUF(3). Otherwise, LIW must be at least L1 = 5 + LENBUF(3) + INFO(8) + INFO(9) (INFO(8) and INFO(9) as output from the last call to ME42B/BD). This argument is not altered by the routine. **Restriction:** LIW $\geq$ L1.

LIW is an INTEGER variable which must be set by the user to the dimension of array IW. If direct access files are not being used (ME42P/PD was not called), LIW must be at least L1 = LENBUF(3). Otherwise, LIW must be at least

L1 = 5 + LENBUF(3) + NFRONT(1) + NFRONT(2). This argument is not altered by the routine. **Restriction:** LIW ≥ L1.

IW is a INTEGER array of length LIW. If direct access files are not being used (ME42P/PD was not called), the first LENBUF(3) entries of IW must be unchanged since the last call to ME42B/BD and these entries are unchanged by ME42C/CD. Otherwise, IW is used by ME42C/CD as workspace.

ICNTL, ISAVE, INFO are as in the calls to ME42A/AD.

2.2 Arrays for control and information

2.2.1 Control parameters

The elements of the arrays ICNTL and CNTL control the action of ME42A/AD, ME42J/JD, ME42P/PD, ME42B/BD, and ME42C/CD. Default values are set by ME42I/ID.

ICNTL(1) is the stream number for error messages and has the default value 6. Printing of error messages is suppressed if ICNTL(1) ≤ 0.

ICNTL(2) is the stream number for warning messages and has the default value 6. Printing of warning messages is suppressed if ICNTL(2) ≤ 0.

ICNTL(3) is the number of bytes for a complex word. ICNTL(3) has the default value 8 (16) for the single (double precision) version.

ICNTL(4) is the number of bytes for an integer word. ICNTL(4) has the default value 4.

ICNTL(5) has the default value 0. If ICNTL(5) is greater than 0, then, when the number of potential pivot columns is greater than or equal to ICNTL(5), an elimination will be performed even if the best pivot candidate does not satisfy the threshold criterion determined by CNTL(2).

ICNTL(6) has the default value 0. If ICNTL(5) and ICNTL(6) are greater than 0, then, when the number of potential pivot columns is greater than or equal to ICNTL(5), only ICNTL(6) of the potential pivot columns are searched for a pivot. The best pivot candidate (largest relative to the other nonzeros in its column) from these ICNTL(6) columns is then used as a pivot.

ICNTL(7) has the default value 0. If ICNTL(7) is nonzero, static condensations are not performed. Static condensations are eliminations performed within an individual element/equation when a variable appears only in that single element/equation.

ICNTL(8) has the default value 0. If the matrix is found to be singular during the decomposition and ICNTL(8) is equal to 0, an error flag is set and the computation terminates (see INFO(1) = –14 in section 2.3). If ICNTL(8) is nonzero, a warning is given, the computation continues and components of the solution vector X corresponding to zero pivots are set equal to zero (see also INFO(23) in section 2.2.2 and INFO(1) = +1 in section 2.3).

CNTL(1) has the default value zero. The matrix is declared singular if, during the factorization, the entry of largest absolute value in any column is less than or equal to CNTL(1).

CNTL(2) has the default value 0.1. An element of the frontal matrix is only considered suitable for use as a pivot if it is of absolute value at least as large as CNTL(2) times the entry of largest absolute value in its column.

2.2.2 Information arrays

The elements of the arrays INFO and RINFO provide information on the action of ME42A/AD, ME42J/JD, ME42P/PD, ME42B/BD, and ME42C/CD. INFO(1) is used by each of the routines ME42A/AD, ME42J/JD, ME42P/PD, ME42B/BD, and ME42C/CD, but INFO(I), I ≥ 2, and RINFO are only accessed by ME42B/BD.

INFO(1) is used as an error and a warning flag. If a call to a routine in the ME42 package is successful, on exit INFO(1) has value 0. A nonzero value of INFO(1) indicates an error has been detected or a warning issued (see section 2.3). If an error is detected during a call to ME42B/BD, the information contained in INFO(I), I $\geq$ 2 and in RINFO may be incomplete.

INFO(2) is set to 1 if the matrix is nonsingular and to 0 if the matrix is found to be singular. (See also RINFO(1)).

INFO(3) holds the total number of variables in the problem.

INFO(4) holds the number of nonzeros in the factor **UQ** (see section 2.1.3).

INFO(5) holds the total storage for the factor **UQ** and corresponding right-hand side vectors, in COMPLEX (COMPLEX*16 in the D version) words.

INFO(6) holds the number of nonzeros in the factor **PL** (see section 2.1.3). This is equal to the total storage for **PL** in COMPLEX (COMPLEX*16 in the D version) words.

INFO(7) holds the total storage for the row and column indices in INTEGER words.

INFO(8) holds the maximum number of rows in the frontal matrix.

INFO(9) holds the maximum number of columns in the frontal matrix. For an element entry, this value will equal INFO(8).

INFO(10) holds the number of buffers used for the factor **UQ** and corresponding right-hand side vectors.

INFO(11) holds the number of buffers used for the factor **PL**.

INFO(12) holds the number of buffers used for the row and column indices.

INFO(13) holds the number of columns searched during pivot searches.

INFO(14) holds the number of nonzeros tested for stability as pivots.

INFO(15) holds the number of nonzeros accessed during the pivot selection process.

INFO(16) holds the number of pivots chosen which did not satisfy the threshold criterion based on the value of CNTL(2).

INFO(17) holds the number of static condensations performed.

INFO(18) holds the number of potential static condensations. This may be greater than INFO(17) because numerical considerations may prevent immediate elimination of internal variables.

INFO(19) holds the maximum number of buffers required to hold a block of pivot rows.

INFO(20) holds the maximum number of buffers required to hold a block of pivot columns.

INFO(21) holds the maximum number of buffers required to hold the row and column indices for a block of pivot rows and columns.

INFO(22) holds the maximum number of rows and columns in a pivot block.

INFO(23) holds, on exit from the final call to ME42B/BD, with INFO(1) = +1 and ICNTL(8) nonzero, an estimate of the deficiency of the matrix. Otherwise, INFO(23) is set to 0.

RINFO(1) holds the natural logarithm of the modulus of the determinant of the matrix **A**. If the matrix is found to be singular, RINFO(1) is set to zero.

RINFO(2) holds the number of floating-point operations in the innermost loops. This count includes operations performed during static condensation.

2.3 Error diagnostics

On successful completion, the subroutines in the ME42 package will exit with the parameter INFO(1) set to 0. Other values for INFO(1) and the reasons for them are given below.

A negative value for INFO(1) is associated with a fatal error. If ICNTL(1) is greater than zero, a self-explanatory message is, in each case, output on unit ICNTL(1) (see section 2.2.1). The negative values for INFO(1) are:

–1 LENLST ≤ 0 on entry to ME42A/AD. (ME42A/AD entry only). Note that LENLST is only checked on the first entry to ME42A/AD.

–2 NVAR ≤ 0 in the current element or equation. (ME42A/AD, ME42J/JD, and ME42B/BD entries). This error is also returned if NVAR is greater than NMAXE (ME42B/BD element entry only i.e. NMAXE > 1).

–3 A variable index in the current element or equation is out of range. (ME42A/AD, ME42J/JD, and ME42B/BD entries).

–4 Duplicate occurrences of the same variable index found in the current element or equation. (ME42A/AD entry only).

–5 NRHS ≥ 1 and the defined first dimension LX of the array X (and the array B) is less than NDF as output from the final call to ME42A/AD. (ME42B/BD and ME42/CD entries). Note that LX is only checked on the first entry to ME42B/BD and on entry to ME42C/CD.

–6 Defined length LW of the complex workspace array W violates the restrictions on LW. (ME42B/BD and ME42C/CD entries). Note that LW is only checked on the first entry to ME42B/BD and on entry to ME42C/CD.

–7 Defined length LIW of the integer workspace array IW violates the restrictions on LIW. (ME42B/BD and ME42C/CD entries). Note that LIW is only checked on the first entry to ME42B/BD, and on entry to ME42C/CD.

–8 Either NMAXE ≤ 0 or NMAXE has been changed from being equal to 1 to a value greater than 1 (or visa versa) during a sequence of calls to ME42J/JD or ME42B/BD. (ME42J/JD and ME42B/BD entries).

–9 The user has changed the number of right-hand sides (NRHS) between calls to ME42B/BD. (ME42B/BD entry only).

–10 LRHS < max(1, NRHS). (ME42B/BD entry only). Note that LRHS is only checked on the first entry to ME42B/BD.

–11 If NMAXE > 1, either NFRONT(1) ≤ 0 or NFRONT(1) has been changed between calls to ME42B/BD. If NMAXE = 1, either NFRONT(I) ≤ 0 or NFRONT(I) has been changed between calls to ME42B/BD for some I, I = 1, 2 (ME42B/BD entry only).

–12 Not enough space has been allocated to the in-core frontal matrix to permit the factorization. However, a symbolic factorization has been performed and a lower bound on the space required is given in the output message and in NFRONT(I), I = 1, 2. See also error +4. (ME42B/BD entry only).

–13 A variable appears again after it has been fully summed (either an index list for an element or equation has been altered since ME42A/AD was called or the order of the elements or equations has been changed). (ME42J/JD and ME42B/BD entries).

–14 Singularity detected in the matrix during the factorization with the control parameter ICNTL(8) equal to zero (see section 2.2.1). (ME42B/BD entry only).

–15 On entry to ME42J/JD or ME42B/BD the value of NDF is not equal to the value output from the final call to ME42A/AD (ME42J/JD and ME42B/BD entries). Note that this is only checked on the first entry to ME42J/JD and on the first entry to ME42B/BD.

–16 Direct access files were not requested and one or more of the buffer lengths (LENBUF(1), LENBUF(2), or LENBUF(3)) was too small. The message also gives information on the buffer lengths necessary to prevent this failure on a subsequent run with identical data. For subsequent runs it may be better to use direct access files. See also error +5. (ME42B/BD entry only).

–17 Insufficient space has been allocated to one or more of the direct access files to permit a successful factorization. The message gives the space necessary for subsequent success, which is also given by INFO(10), INFO(11), and INFO(12) (section 2.2.2). To avoid this error with a subsequent run on identical data, the user must set LENFLE(I) to be at least as great as LENBUF(I) times these values, I = 1, 2, 3. See also error +6. (ME42B/BD entry only).

–18 The number of right-hand sides NRHS is out of range. Either NRHS < 0 (ME42B/BD entry only) or NRHS < 1. (ME42C/CD entry only).

–19 LENBUF(1) ≤ 0 or LENBUF(3) ≤ 0 or ISTRM(2) ≠ 0 and LENBUF(2) ≤ 0. (ME42P/PD entry only). This error is also returned by ME42B/BD if ME42P/PD has not been called and ME42B/BD is called with LENBUF(1) ≤ 0, or LENBUF(2) < 0, or LENBUF(3) ≤ 0. It is also returned by ME42B/BD if, for some I = 1, 2, or 3, LENBUF(I) has been changed between the call to ME42P/PD and the first call to ME42B/BD. (ME42B/BD first entry only).

–20 Either LENBUF(1) > LENFLE(1), or LENBUF(3) > LENFLE(3), or ISTRM ≠ 0 and LENBUF(2) > LENFLE(2). (ME42P/PD entry only).

–21 At least two of the stream numbers ISTRM(I), I = 1, 2, 3 are the same. The error message indicates which stream numbers are the same. (ME42P/PD entry only).

–22 Error in Fortran OPEN statement. . The iostat parameter is printed (the iostat parameter is a parameter which, after an input/output operation is completed, is set to zero if no error was detected and to a positive integer otherwise). (ME42P/PD entry only).

–23 ICNTL(I) ≤ 0 for I = 3 or 4. (ME42P/PD entry only).

–24 For some I = 1, 2, or 3, either ISTRM(I) lies out of range or is equal to 6, ICNTL(1), or ICNTL(2). (ME42P/PD entry only).

–25 Error detected when reading a direct access file. The iostat parameter is printed. (ME42B/BD and ME42C/CD entries).

–26 Error detected when writing to a direct access file. The iostat parameter is printed. (ME42B/BD entry only).

–27 ME42C/CD has been called after calling ME42P/PD with ISTRM(2) = 0 or after calling ME42B/BD with LENBUF(2) = 0. (ME42C/CD entry only).

There are two warning messages which are associated with INFO(1) = +1 and +2. These can only be returned by ME42B/BD. In each case, a self-explanatory message is output on unit ICNTL(2). The warnings are:

+1 The matrix **A** has been found to be singular and the control parameter ICNTL(8) was nonzero (see section 2.2.1). If the sequence of calls to ME42B/BD is completed, on exit from the final call INFO(23) (see section 2.2.2) will hold an estimate of the deficiency of the matrix.

+2 Pivots have been chosen which do not satisfy the threshold criterion determined by CNTL(2) (see section 2.1.1). If the sequence of calls to ME42B/BD is completed, on exit from the final call INFO(16) (see section 2.2.2) will hold the number of such pivots. INFO(1) = +2 will overwrite INFO(1) = +1.

There are three error returns associated with a positive value for INFO(1). These can only be returned by ME42B/BD. In each case, a message is output on unit ICNTL(2). The user is encouraged to continue the sequence of calls to ME42B/BD, at the end of which a negative value of INFO(1) will be returned, an error message will be output on unit ICNTL(1), and information to enable success on a subsequent run with identical data will be output, although the factorization will not have been completed and the information contained in the arrays INFO and RINFO will not be complete. Note that, if the user does not complete the sequence of calls, the array LAST (see section 2.1.3) may be corrupted. The positive values for INFO(1) associated with an error are:

+4 Insufficient space has been allocated to the in-core frontal matrix. If the user continues to call ME42B/BD, on exit from the final call a lower bound will be obtained for the front size required (see error –12). INFO(1) = +4 will overwrite INFO(1) = +1, +2, +5, and +6.

+5 Direct access files were not requested and one or more of the buffer lengths (LENBUF(1), LENBUF(2), or LENBUF(3)) was too small. If the user completes the sequence of calls to ME42B/BD, the buffer lengths required for a successful run will be output (error –16). INFO(1) = +5 will overwrite INFO(1) = +1 and +2.

+6 Insufficient space has been allocated to one or more of the direct access files. If the user completes the sequence of calls to ME42B/BD, the amount of space required for subsequent success will be given (see error –17).

INFO(1) = +6 will overwrite INFO(1) = +1 and +2.

3 GENERAL INFORMATION

3.1 Summary of information.

Use of common: The subroutines do not use common blocks.

Other routines called directly: The BLAS routines CAXPY/ZAXPY, CGERU/ZGERU, CGEMV/ZGEMV, CTPSV/ZTPSV, CGEMM/ZGEMM, CTRSM/ZTRSM. Subroutines internal to the package are ME42D/DD, ME42E/ED, ME42F/FD, ME42G/GD, ME42H/HD, ME42K/KD, ME42L/LD, ME42M/MD, ME42N/ND, ME42O/OD.

Workspace: Workspace is provided by the arrays:
W(LW) (ME42B/BD and ME42C/CD).
IW(LIW) (ME42B/BD, and ME42C/CD).
IVAR(NVAR) and LAST(NDF) are used locally as workspace (ME42B/BD only).
ISAVE(45) is a work array which must be preserved between calls to routines in the ME42 package.

Input/output: In the event of errors, diagnostic messages are printed. The output streams for these messages are controlled by the variables ICNTL(1) and ICNTL(2) (see section 2.2.1). Stream ICNTL(1) is used for error messages (INFO(1) $<$ 0) and stream ICNTL(2) for warnings (INFO(1) $>$ 0).

Restrictions:

ISTRM(1) and ISTRM(3) lie in the range [1, 99] and do not equal 6, ICNTL(1), or ICNTL(2) (ME42P/PD entry only).

ISTRM(2) lies in the range [0, 99] and does not equal 6, ICNTL(1), or ICNTL(2) (ME42P/PD entry only).

ISTRM(I) $\neq$ ISTRM(J), I, J = 1, 2, 3 (ME42P/PD entry only).

ICNTL(I) $>$ 0, I = 3, 4 (ME42P/PD entry only).

LENBUF(1) $\geq$ 1 and LENBUF(3) $\geq$ 1 (ME42P/PD and ME42B/BD entries).

LENBUF(2) $\geq$ 0 (ME42B/BD entry only).

LENFLE(1) $\geq$ LENBUF(1) and LENFLE(3) $\geq$ LENBUF(3) (ME42P/PD entry only).

LENFLE(2) $\geq$ LENBUF(2) if ISTRM(2) $\neq$ 0 (ME42P/PD entry only).

NVAR $\geq$ 1 (ME42A/AD, ME42J/JD, and ME42B/BD entries),

LENLST $\geq$ 1 (ME42A/AD entry only).

1 $\leq$ IVAR(I) $\leq$ LENLST and IVAR(I) $\neq$ IVAR(J), I, J, = 1, 2, ..., NVAR (ME42A/AD entry only).

1 $\leq$ IVAR(I) $\leq$ NDF, I = 1, 2, ..., NVAR (ME42J/JD and ME42B/BD entries).

NMAXE $\geq$ 1 (ME42J/JD and ME42B/BD entries).

If NMAXE $>$ 1, NMAXE $\geq$ NVAR (ME42B/BD entry only).

NRHS $\geq$ 0 (ME42B/BD entry only), NRHS $\geq$ 1 (ME42C/CD entry only).

If NRHS $\geq$ 1, LX $\geq$ NDF (ME42B/BD and ME42C/CD entries).

LRHS $\geq$ max(1, NRHS) (ME42B/BD entry only).

NFRONT(1) $>$ 0 and if NMAXE = 1, NFRONT(2) $>$ 0 (ME42B/BD entry only).

LW $\geq$ 1+LENBUF(1)+LENBUF(2)+NFRONT(1)*NFRONT(2)+
max{NFRONT(1)*LRHS, NFRONT(2)*NRHS} (ME42B/BD entry only).

If ME42P/PD is not called,
LW $\geq$ LENBUF(1) + LENBUF(2) + NRHS* max{NFRONT(1), NFRONT(2)}.

Otherwise, LW ≥ max{LENBUF(1), LENBUF(2)} + INFO(22)* max{INFO(8), INFO(9) + *nrhsb*}
NRHS* max{INFO(8), INFO(9)} (ME42C/CD entry only).

LIW ≥ LENBUF(3)+2*NFRONT(1)+4*NFRONT(2) (ME42B/BD entry only).

If ME42P/PD is not called, LIW ≥ LENBUF(3).
Otherwise, LIW ≥ LENBUF(3)+INFO(8)+INFO(9)+5 (ME42C/CD entry only).

Portability: Fortran 77.

4 METHOD

The method used is a modification of the unsymmetric frontal scheme of Hood (1976). ME42 has been developed from the work by Cliffe, Jackson, Rae, and Winters (1978) and from the earlier Harwell Subroutine Library routine MA32 (see Duff 1981, 1983).

The elements or equations are assembled into an in-core frontal matrix one at a time. A variable which has appeared for the last time (i.e. does not occur in future elements or equations) is fully summed and is available for use as a pivot in the Gaussian elimination.

Eliminations are performed whenever a variable in the frontal matrix is fully summed and satisfies a numerical tolerance. Once all possible eliminations for the current element or equation have been performed, the pivot rows and, optionally, the pivot columns are written to in-core buffers and thence, if requested, to direct access files. In order to prevent the amount of in-core memory required becoming too large, the user should order the elements (or equations) so that the same variable does not occur in elements (or equations) which are widely apart in the ordering. Thus, for example, in a finite-element problem with a narrow pipe geometry, elements should be ordered across the cross-section of the pipe rather than along its length. For finite-element calculations, an efficient element ordering can be obtained using the Harwell Subroutine Library routine MC43 (see Duff, Reid, and Scott 1989). An estimate of the in-core memory required may be obtained by calling ME42J/JD, which performs a symbolic factorization.

ME42 is a complex version of the code MA42. Further details are given in Duff and Scott (1993).

References.

Cliffe, K.A., Jackson, C.P., Rae, J., and Winters, K.H. (1978) Finite element flow modelling using velocity and pressure variables. Report AERE R.9202, HMSO London.

Duff, I.S. (1981) MA32 – A package for solving sparse unsymmetric systems using the frontal method. Report AERE R.10079, HMSO London.

Duff, I.S. (1983) Enhancements to the MA32 package for solving sparse unsymmetric equations. Report AERE R.11009, HMSO London.

Duff, I.S., Reid, J.K., and Scott, J.A. (1989) The use of profile reduction algorithms with a frontal code. Int. J. Numer. Meth. Engng **28**, 2555-2568.

Duff, I.S. and Scott, J.A. (1993) MA42 – A new frontal code for solving sparse unsymmetric systems. Rutherford Appleton Laboratory Report, to appear.

Hood, P. (1976) Frontal solution program for unsymmetric matrices. Int. J. Numer. Meth. Engng **10**, 379-399.

5 EXAMPLE OF USE

We give an example of the code required to solve a set of equations using the ME42 package when input is by elements (example 5.1) and when input is by equations (example 5.2).

Example 5.1 illustrates the simplest calling sequence for ME42. Direct access files are not used and no symbolic factorization is performed. In this example, we wish to solve for one right-hand side at the same time as the factorization and we do not wish to retain the factors for solving further systems.

Example 5.2 illustrates the full calling sequence for the ME42 package. In this example, we wish to solve $\mathbf{Ax}=\mathbf{b}$, $\mathbf{A}^T\mathbf{x}=\mathbf{b}$, and $\mathbf{A}^H\mathbf{x}=\mathbf{b}$. We do not supply right-hand sides with the equations but thereafter we want to solve each system for two right-hand sides. Direct access files are used to hold the factors and ME42J/JD is used to perform a symbolic factorization.

5.1 Example of element input

Consider a 5×5 symmetric matrix

$$\begin{pmatrix} 3+6i & 0 & 0 & 0 & 2 \\ 0 & 5+2i & 3+3i & 6 & 0 \\ 0 & 3+3i & 3+8i & 6+6i & 0 \\ 0 & 6 & 6+6i & 9+7i & 2+i \\ 2 & 0 & 0 & 2+i & 5+5i \end{pmatrix}$$

generated from four elemental matrices

$$\begin{matrix} 4 \\ 5 \end{matrix}\begin{pmatrix} 4+3i & 2+i \\ 2+i & 4+4i \end{pmatrix} \qquad \begin{matrix} 1 \\ 5 \end{matrix}\begin{pmatrix} 3+6i & 2 \\ 2 & 1+i \end{pmatrix} \qquad \begin{matrix} 2 \\ 3 \\ 4 \end{matrix}\begin{pmatrix} 5+2i & 3+3i & 6 \\ 3+3i & 1+5i & 1+i \\ 6 & 1+i & 2+4i \end{pmatrix} \qquad \begin{matrix} 3 \\ 4 \end{matrix}\begin{pmatrix} 2+3i & 5+5i \\ 5+5i & 3 \end{pmatrix},$$

where the variable indices are indicated by the integers before each matrix (columns are identified symmetrically to rows). The corresponding elemental vectors $\mathbf{b}^{(k)}$ $(1\le k\le 4)$ are

$$\begin{pmatrix} 6+4i \\ 6+5i \end{pmatrix} \qquad \begin{pmatrix} 5+6i \\ 3+i \end{pmatrix} \qquad \begin{pmatrix} 14+5i \\ 5+9i \\ 8+5i \end{pmatrix} \qquad \begin{pmatrix} 7+8i \\ 8+5i \end{pmatrix}.$$

The following program is used to solve this problem. In this program we read the element data into arrays ELTPTR (location of first entry of element), ELTVAR (variable indices), VALUE (numerical values), and RHSVAL (right-hand sides). This method of storing the element data is used here for illustrative purposes; the user may prefer, for example, to read in the element data from a direct access file.

```
C Example to illustrate the use of ME42: element entry.
C
C      .. Parameters ..
       INTEGER MAXE,LIWMAX,LRHS,NZMAX,MELT,LENLST,LWMAX,MAXVL,MAXRVL,
     +         NFMAX,NDFMAX
       PARAMETER (MAXE=3,LIWMAX=120,LRHS=1,NZMAX=24,MELT=4,LENLST=6,
     +           LWMAX=80,MAXVL=30,MAXRVL=10,NFMAX=5,NDFMAX=5)
C      ..
C      .. Local Scalars ..
       INTEGER I,IELT,J,JSTRT,K,KSTRT,LIW,LW,NDF,NELT,NMAXE,NRHS,
     +         NVAR,NZ,RHSCRD,VALCRD
C      ..
C      .. Local Arrays ..
       COMPLEX AVAR(MAXE,MAXE),RHS(MAXE,LRHS),RHSVAL(MAXRVL),
     +         VALUE(MAXVL),W(LWMAX),X(NDFMAX,LRHS)
       REAL CNTL(2),RINFO(2)
       INTEGER ELTPTR(MELT+1),ELTVAR(NZMAX),ICNTL(8),INFO(23),
     +         ISAVE(45),IVAR(MAXE),IW(LIWMAX),LAST(LENLST),
     +         LENBUF(3),NFRONT(2)
```

```
C     ..
C     .. External Subroutines ..
      EXTERNAL ME42A,ME42B,ME42I
C     ..
C
C*** Call to ME42I
      CALL ME42I(ICNTL,CNTL,ISAVE)
C
C Read in the element data.
C NELT is number of elements.
C ELTVAR contains lists of the variables belonging to the finite
C elements, with those for element 1 preceding those for element
C 2, and so on. ELTPTR points to the position in ELTVAR
C of the first variable in element I. NZ is the total number
C of entries in the element lists.
C
      READ (*,FMT=*) NELT
      READ (*,FMT=*) (ELTPTR(I),I=1,NELT+1)
      NZ = ELTPTR(NELT+1) - 1
      READ (*,FMT=*) (ELTVAR(I),I=1,NZ)
C
C Calls to ME42A to establish when each variable is fully assembled
      DO 20 IELT = 1,NELT
         NVAR = ELTPTR(IELT+1) - ELTPTR(IELT)
         JSTRT = ELTPTR(IELT)
         DO 10 J = 1,NVAR
            IVAR(J) = ELTVAR(JSTRT+J-1)
   10    CONTINUE
C*** Call to ME42A
         CALL ME42A(NVAR,IVAR,NDF,LAST,LENLST,ICNTL,ISAVE,
     +               INFO)
         IF (INFO(1).LT.0) GO TO 60
   20 CONTINUE
C
C Input elemental matrices and right-hand sides.
C VALCRD is the number of numerical values to be input.
C VALUE contains lists of the numerical values in the elemental
C matrices, with element 1 preceding element 2, and so on.
C Since the elemental matrices are symmetric only the upper
C triangular part is stored.
C
      READ (*,FMT=*) VALCRD
      READ (*,FMT=*) (VALUE(I),I=1,VALCRD)
C
C RHSCRD is the number of right-hand side numerical values to
C be input.
C RHSVAL contains lists of the right-hand side numerical values
C corresponding to each of the elements in order.
C
      READ (*,FMT=*) RHSCRD
      READ (*,FMT=*) (RHSVAL(I),I=1,RHSCRD)
C
C Prepare to call ME42B. Note the L-factor is not stored.
      NMAXE = MAXE
      NFRONT(1) = NFMAX
      NFRONT(2) = NFMAX
      LENBUF(1) = 30
      LENBUF(2) = 0
      LENBUF(3) = 70
      NRHS = 1
      LW = 1 + LENBUF(1) + LENBUF(2) + NFRONT(1)*NFRONT(2) +
```

```
     +     LRHS*NFRONT(1)
      LIW = LENBUF(3) + 2*NFRONT(1) + 4*NFRONT(2)
      IF (LW.GT.LWMAX .OR. LIW.GT.LIWMAX) GO TO 70
C
      KSTRT = 1
      DO 50 IELT = 1,NELT
         NVAR = ELTPTR(IELT+1) - ELTPTR(IELT)
         JSTRT = ELTPTR(IELT)
         DO 40 J = 1,NVAR
            IVAR(J) = ELTVAR(JSTRT+J-1)
            DO 30 K = J,NVAR
               AVAR(K,J) = VALUE(KSTRT)
               IF (J.NE.K) AVAR(J,K) = AVAR(K,J)
               KSTRT = KSTRT + 1
   30       CONTINUE
            RHS(J,1) = RHSVAL(JSTRT+J-1)
   40    CONTINUE
C*** Call to ME42B
         CALL ME42B(NVAR,IVAR,NDF,LAST,NMAXE,AVAR,NRHS,RHS,LRHS,
     +               NDFMAX,X,NFRONT,LENBUF,LW,W,LIW,IW,ICNTL,CNTL,
     +               ISAVE,INFO,RINFO)
C
C Only trap fatal errors
         IF (INFO(1).LT.0) GO TO 60
   50 CONTINUE
C
C  Solution is in first NDF locations of X
      WRITE (*,FMT=9000)
      WRITE (*,FMT=9010) (X(I,1),I=1,NDF)
      WRITE (*,FMT=9040) (INFO(I),I=1,23)
      WRITE (*,FMT=9050) (RINFO(I),I=1,2)
      GO TO 70
C Print appropriate fatal error diagnostic
   60 WRITE (*,FMT=9020)
      WRITE (*,FMT=9030) INFO(1)
   70 STOP
C
 9000 FORMAT (/3X,'The solution is:'/)
 9010 FORMAT (2G12.4)
 9020 FORMAT (/3X,'Error return')
 9030 FORMAT (3X,'INFO(1) = ',I3)
 9040 FORMAT (/' INFO    = ',8I5,/'           ',8I5,
     +        /'           ',7I5)
 9050 FORMAT (/' RINFO = ',2G10.4)
      END
```

The input data used for this problem is:

```
4
1   3   5   8  10
4   5   1   5   2   3   4   3   4
15
(4.,  3.)
(2.,  1.)
(4.,  4.)
(3.,  6.)
(2.,  0.)
(1.,  1.)
(5.,  2.)
(3.,  3.)
(6.,  0.)
(1.,  5.)
(1.,  1.)
(2.,  4.)
(2.,  3.)
(5.,  5.)
(3.,  0.)
9
(2.,  10.)
(1.,  11.)
(-1., 11.)
(2.,   4.)
(9.,  19.)
(-4., 14.)
(4.,  14.)
(-1., 15.)
(3.,  13.)
```

This produces the following output:

```
   The solution is:

   1.000      1.000
   1.000      1.000
   1.000      1.000
   1.000      1.000
   1.000      1.000

   The solution is:

   1.000      1.000
   1.000      1.000
   1.000      1.000
   1.000      1.000
   1.000      1.000

 INFO    =     0    1    5   11   16   11   41    3
               3    1    0    1    5    2   10    0
               2    2    1    0    1    2    0

 RINFO =  9.009     56.00
```

5.2 Example of equation input

We wish to factorize the matrix **A** given by

$$\mathbf{A} = \begin{pmatrix} 3+i & 0. & 5+5i \\ 1+3i & 3+2i & 0 \\ 0 & 1 & 8+i \end{pmatrix} .$$

We then want to solve $\mathbf{Ax}=\mathbf{b}$ for the two right-hand sides

$$\begin{pmatrix} 6+2i \\ 1-3i \\ 10-8i \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 2-16i \\ 14+19i \\ -8-17i \end{pmatrix} ,$$

then to solve $\mathbf{A}^{\mathrm{T}}\mathbf{x}=\mathbf{b}$ for the two right-hand sides

$$\begin{pmatrix} 1+15i \\ -1+4i \\ -20+15i \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 13+5i \\ 14+i \\ 17+9i \end{pmatrix} ,$$

and finally to solve $\mathbf{A}^{\mathrm{H}}\mathbf{x}=\mathbf{b}$ for the two right-hand sides

$$\begin{pmatrix} 8 \\ 6+2i \\ 19+7i \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 12-4i \\ 10 \\ 32+i \end{pmatrix} .$$

The following program is used to solve this problem. Note that to illustrate the full calling sequence for the ME42 package, we call ME42J to obtain a bound on the front size, then call ME42P to open direct access files. ME42B is called with the number of right-hand sides set equal to 0 and finally ME42C is called to solve $\mathbf{Ax}=\mathbf{b}$, $\mathbf{A}^{\mathrm{T}}\mathbf{x}=\mathbf{b}$, and $\mathbf{A}^{\mathrm{H}}\mathbf{x}=\mathbf{b}$.

```
C Example to illustrate the use of ME42: equation entry.
C
C     .. Parameters ..
      INTEGER LIWMAX,LRHS,NZMAX,MVAR,MEQ,LWMAX,NDFMAX
      PARAMETER (LIWMAX=50,LRHS=2,NZMAX=9,MVAR=3,MEQ=3,LWMAX=130,
     +          NDFMAX=3)
C     ..
C     .. Local Scalars ..
      INTEGER I,IEQ,J,JSTRT,KIND,LIW,LW,NDF,NEQ,NMAXE,NRHS,NRHSC,NVAR,NZ
C     ..
C     .. Local Arrays ..
      COMPLEX AVAR(1,MVAR),B(NDFMAX,LRHS),VALUE(NZMAX),W(LWMAX),
     +        RHS(1,LRHS),X(NDFMAX,LRHS)
      REAL CNTL(2),RINFO(2)
      INTEGER ICNTL(8),IFSIZE(5),INFO(23),IRN(NZMAX),ISAVE(45),ISTRM(3),
     +        IVAR(MVAR),IW(LIWMAX),JP(MEQ+1),LAST(NDFMAX),LENBUF(3),
     +        LENFLE(3),NFRONT(2)
C     ..
C     .. External Subroutines ..
      EXTERNAL ME42A,ME42B,ME42C,ME42I,ME42J,ME42P
C     ..
C     .. Intrinsic Functions ..
      INTRINSIC MAX,MIN
C     ..
C*** Call to ME42I
      CALL ME42I(ICNTL,CNTL,ISAVE)
C
C Read in the data.
C NEQ is number of equations.
C IRN contains lists of the variables belonging to the
C equations, with those for equation 1 preceding those for equation
C 2, and so on. JP(I) points to the position in IRN
C of the first variable in equation I.
C
```

```
      READ (*,FMT=*) NEQ
      READ (*,FMT=*) (JP(I),I=1,NEQ+1)
C NZ is the total number of entries in the variable lists.
      NZ = JP(NEQ+1) - 1
      READ (*,FMT=*) (IRN(I),I=1,NZ)
C
C Calls to ME42A to establish when each variable is fully assembled
      DO 20 IEQ = 1,NEQ
         NVAR = JP(IEQ+1) - JP(IEQ)
         JSTRT = JP(IEQ)
         DO 10 J = 1,NVAR
            IVAR(J) = IRN(JSTRT+J-1)
   10    CONTINUE
C*** Call to ME42A
         CALL ME42A(NVAR,IVAR,NDF,LAST,NDFMAX,ICNTL,ISAVE,INFO)
         IF (INFO(1).LT.0) GO TO 110
   20 CONTINUE
C
C Calls to ME42J to perform symbolic factorization.
      NMAXE = 1
      DO 40 IEQ = 1,NEQ
         NVAR = JP(IEQ+1) - JP(IEQ)
         JSTRT = JP(IEQ)
         DO 30 J = 1,NVAR
            IVAR(J) = IRN(JSTRT+J-1)
   30    CONTINUE
C*** Call to ME42J
         CALL ME42J(NVAR,IVAR,NDF,LAST,NMAXE,IFSIZE,ICNTL,ISAVE,INFO)
         IF (INFO(1).LT.0) GO TO 110
   40 CONTINUE
      WRITE (*,FMT=9030) (IFSIZE(I),I=1,5)
C
C Call to ME42P to open direct access data sets.
C Choose stream numbers and file sizes (allow for pivoting)
      ISTRM(1) = 8
      ISTRM(2) = 9
      ISTRM(3) = 10
      DO 50 I = 1,3
         LENBUF(I) = 30
         LENFLE(I) = IFSIZE(2+I) + IFSIZE(2+I)/5
         LENFLE(I) = MAX(LENFLE(I),LENBUF(I))
   50 CONTINUE
C*** Call to ME42P
      CALL ME42P(ISTRM,LENBUF,LENFLE,ICNTL,ISAVE,INFO)
      IF (INFO(1).LT.0) GO TO 110
C
C Input the numerical values for each row.
C Number of numerical values to be input is NZ.
C VALUE contains lists of the numerical values
C with equation 1 preceding equation 2, and so on.
C
      READ (*,FMT=*) (VALUE(I),I=1,NZ)
C
C Initialize bounds on arrays and problem size.
C Perform decompostion and solve for 2 right-hand sides.
      NRHS = 2
      DO 55 J = 1,NRHS
         READ (*,FMT=*) (B(I,J),I=1,NDF)
   55 CONTINUE
      NFRONT(1) = IFSIZE(1) + IFSIZE(1)/5
      NFRONT(2) = IFSIZE(2) + IFSIZE(2)/5
```

```
      NFRONT(1) = MIN(NDFMAX,NFRONT(1))
      NFRONT(2) = MIN(NDFMAX,NFRONT(2))
      LW = 1 + LENBUF(1) + LENBUF(2) + NFRONT(1)*NFRONT(2) +
     +     MAX(LRHS*NFRONT(1),NRHS*NFRONT(2))
      LIW = LENBUF(3) + 2*NFRONT(1) + 4*NFRONT(2)
      IF (LW.GT.LWMAX .OR. LIW.GT.LIWMAX) GO TO 120
C
      DO 70 IEQ = 1,NEQ
         NVAR = JP(IEQ+1) - JP(IEQ)
         JSTRT = JP(IEQ)
         DO 60 J = 1,NVAR
            IVAR(J) = IRN(JSTRT+J-1)
            AVAR(1,J) = VALUE(JSTRT+J-1)
   60    CONTINUE
         DO 65 J = 1,NRHS
            RHS(1,J) = B(IEQ,J)
   65    CONTINUE
C*** Call to ME42B
         CALL ME42B(NVAR,IVAR,NDF,LAST,NMAXE,AVAR,NRHS,RHS,LRHS,NDFMAX,
     +              X,NFRONT,LENBUF,LW,W,LIW,IW,ICNTL,CNTL,ISAVE,INFO,
     +              RINFO)
C Only trap fatal errors.
         IF (INFO(1).LT.0) GO TO 110
   70 CONTINUE
      WRITE (*,FMT=9020) (INFO(I),I=1,23)
      WRITE (*,FMT=9090) (RINFO(I),I=1,2)
      WRITE (6,FMT=9070)
      DO 75 J = 1,NRHS
            WRITE (6,FMT=9040) J
            WRITE (6,FMT=9050) (X(I,J),I=1,NDF)
   75 CONTINUE

C  We wish to solve A(T)X = B and A(H)X = B each
C  for two right-hand sides
      NRHSC = 2
      LW = MAX(LENBUF(1),LENBUF(2)) + INFO(22)*MAX(INFO(8),INFO(9)+NRHS)
     +     + NRHSC*MAX(INFO(8),INFO(9))
      LIW = LENBUF(3) + INFO(8) + INFO(9) + 5
      IF (LW.GT.LWMAX .OR. LIW.GT.LIWMAX) GO TO 120
C
      DO 100 KIND = 2,3
C  Read J-th right-hand side into B(.,J), J=1,NRHS
         DO 80 J = 1,NRHSC
            READ (*,FMT=*) (B(I,J),I=1,NDF)
   80    CONTINUE
C*** Call to ME42C
         CALL ME42C(KIND,NRHSC,NDFMAX,B,X,LW,W,LIW,IW,ICNTL,ISAVE,INFO)
         IF (INFO(1).LT.0) GO TO 110
C
C  Solution for J-th right-hand side is in X(.,J), J=1,NRHS
         IF (KIND.EQ.2) WRITE (6,FMT=9060)
         IF (KIND.EQ.3) WRITE (6,FMT=9080)
         DO 90 J = 1,NRHSC
            WRITE (6,FMT=9040) J
            WRITE (6,FMT=9050) (X(I,J),I=1,NDF)
   90    CONTINUE
  100 CONTINUE
      GO TO 120
C Print appropriate fatal error diagnostic
  110 WRITE (*,FMT=9000)
      WRITE (*,FMT=9010) INFO(1)
```

```
   120 STOP
  C
   9000 FORMAT (/3X,'Error return')
   9010 FORMAT (3X,'INFO(1) = ',I3)
   9020 FORMAT (/' INFO    = ',8I5,/'           ',8I5,/'           ',7I5)
   9030 FORMAT (/' IFSIZE = ',5I5)
   9040 FORMAT (/3X,'The solution for right-hand side number',I2,' is:')
   9050 FORMAT (2G12.4)
   9060 FORMAT (/3X,'*** A(T)x = b ***')
   9070 FORMAT (/3X,'*** Ax    = b ***')
   9080 FORMAT (/3X,'*** A(H)x = b ***')
   9090 FORMAT (/' RINFO   = ',2G10.4)
        END
```

The input data used for this problem is:

```
  3
  1    3    5   7
  1    3    1    2    2    3
  (3.,    1.)
  (5.,    5.)
  (1.,    3.)
  (3.,    2.)
  (1.,    0.)
  (8.,    1.)
  (6.,    2.)
  (1.,   -3.)
  (10.,  -8.)
  (2.,  -16.)
  (14.,  19.)
  (-8., -17.)
  (1.,   15.)
  (-1.,   4.)
  (-20., 15.)
  (13.,   5.)
  (14.,   1.)
  (17.,   9.)
  (8.,    0.)
  (6.,    2.)
  (19.,   7.)
  (12.,  -4.)
  (10.,   0.)
  (32.,   1.)
```

This produces the following output:

```
 IFSIZE =     2    3    6    5   20

 INFO    =     0    1    3    6   12    5   20    2
               3    1    1    1    3    0    5    0
               0    0    1    1    1    2    0

 RINFO   =  4.660     16.00

   *** Ax    = b ***

   The solution for right-hand side number 1 is:
  -1.000       1.000
   1.000      -1.000
   1.000      -1.000

   The solution for right-hand side number 2 is:
   1.000       1.000
   6.000       1.000
  -2.000      -2.000

   *** A(T)x = b ***

   The solution for right-hand side number 1 is:
   2.000       3.000
   1.000       1.000
  -2.000      -1.000

   The solution for right-hand side number 2 is:
   1.000      -1.000
   3.000      -2.000
   1.000       1.000

   *** A(H)x = b ***

   The solution for right-hand side number 1 is:
   1.000       1.000
   1.000       1.000
   1.000       1.000

   The solution for right-hand side number 2 is:
   2.000       1.000
   2.000       1.000
   2.000       1.000
```