ME57

PACKAGE SPECIFICATION HSL

1 SUMMARY

To **solve a sparse complex symmetric or Hermitian system of linear equations.** Given a sparse complex symmetric or Hermitian matrix $\mathbf{A}=\{a_{ij}\}_{n\times n}$ and an n-vector **b** (or an $n\times s$ matrix **B**), this subroutine solves the system **Ax**=**b** (**AX**=**B**). The matrix **A** can be either complex symmetric or Hermitian.

The multifrontal method is used. ME57 is a complex version of the code MA57. It is a direct method based on a sparse variant of Gaussian elimination and is discussed further by Duff and Reid, ACM Trans. Math. Software **9** (1983), 302-325. A detailed discussion on the MA57 strategy and performance is given by Duff, ACM Trans. Math. Software **30** (2004), 118-144. Relevant work on pivoting and scaling strategies is given by Duff and Pralet, SIAM Journal Matrix Analysis and Applications **27** (2005), 313-340. More recent work on static pivoting is given in the paper by Duff and Pralet, SIAM Journal Matrix Analysis and Applications **29** (2007), 1007-1024.

The ME57 package has a range of options including several sparsity orderings, multiple right-hand sides, partial solutions, error analysis, scaling, a matrix modification facility, an option for handling significantly rank-deficient systems, and a stop and restart facility. Although the default settings should work well in general, there are several parameters available to enable the user to tune the code for his or her problem class or computer architecture.

ATTRIBUTES — Version: 2.4.3. (1 November 2023) **Types:** Complex (single, double) **Calls:** FD15, MC34, MC64, MF71, MC47, BLAS routines _GEMM, _TPSV, and _GEMV, and (optionally) METIS_NODEND from the MeTiS package. **Remark:** Supersedes ME27. **Original date:** October 2001. **Origin:** I. S. Duff, Rutherford Appleton Laboratory.

2 HOW TO USE THE PACKAGE

2.1 Argument lists and calling sequences

There are six entries:

(a) ME57I/ID sets default values for the components of the arrays that hold control parameters. Normally the user will call ME57I/ID prior to any call to ME57A/AD. If non-default values for any of the control parameters are required, they should be set immediately after the call to ME57I/ID.

(b) ME57A/AD accepts the pattern of **A** and chooses pivots for Gaussian elimination using a selection criterion to preserve sparsity. It subsequently constructs subsidiary information for the actual factorization by ME57B/BD. The user may provide the pivot sequence; in which case only the necessary information for ME57B/BD will be generated.

(c) ME57B/BD factorizes a matrix **A** using the information from a previous call to ME57A/AD. The actual pivot sequence used may differ from that of ME57A/AD if **A** is not definite.

(d) ME57C/CD uses the factors generated by ME57B/BD to solve a system of equations **Ax**=**b** (**AX**=**B**).

(e) ME57D/DD uses the factors generated by ME57B/BD to solve a system of equations **Ax**=**b** using iterative refinement and (optionally) returning estimates of the error.

(f) ME57E/ED maps the data into new larger size arrays following a failure of ME57B/BD through insufficient storage so that the numerical factorization can be continued using these larger arrays.

A call to ME57C/CD or ME57D/DD must be preceded by a call to ME57B/BD which in turn must be preceded by a call to ME57A/AD. Since the information passed from one subroutine to the next is not corrupted by the second, several calls to ME57B/BD for matrices with the same sparsity pattern but different values may follow a single call to ME57A/AD, and similarly ME57C/CD or ME57D/DD can be used repeatedly to solve for different right-hand sides **b** (**B**). Note that it would be possible to use ME57A/AD on several matrices before calling ME57B/BD. When we state that parameters "must be unchanged since the call to *subroutine*", we mean a successful call of the routine on the same matrix.

In ME57I, ME57A, ME57B, ME57C, ME57D, and ME57E, all REALs (COMPLEXs) are default reals (complex); in ME57ID, ME57AD, ME57BD, ME57CD, ME57DD, and ME57ED, all REALs (COMPLEXs) are default double precision (double precision complex).

2.1.1 To set default values of controlling parameters

The single precision version

```
CALL ME57I(CNTL,ICNTL)
```

The double precision version

```
CALL ME57ID(CNTL,ICNTL)
```

CNTL is a REAL array of length 5 that need not be set by the user. On return it contains default values. For further information see Section 2.2.

ICNTL is an INTEGER array of length 20 that need not be set by the user. On return it contains default values. For further information see Section 2.2.

2.1.2 To perform symbolic manipulations

The single precision version

```
CALL ME57A(KIND,N,NE,IRN,JCN,LKEEP,KEEP,IWORK,ICNTL,INFO,RINFO)
```

The double precision version

```
CALL ME57AD(KIND,N,NE,IRN,JCN,LKEEP,KEEP,IWORK,ICNTL,INFO,RINFO)
```

KIND is an INTEGER variable that must be set by the user to indicate whether the matrix is complex symmetric (KIND=1) or complex Hermitian (KIND≠1).

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

NE is an INTEGER variable that must be set by the user to the number of entries input in IRN and JCN. It is not altered by the subroutine. **Restriction:** NE ≥ 0.

IRN and JCN are INTEGER arrays of length NE. The user must set them so that each diagonal entry a_{ii} is represented by IRN(k)=i and JCN(k)=i and each pair of off-diagonal entries a_{ij} and a_{ji} is represented by IRN(k)=i and JCN(k)=j or by IRN(k)=j and JCN(k)=i. If the matrix is Hermitian (KIND≠1), then all entries must be in the upper triangle (that is, JCN(k) ≥ IRN(k), k = 1, ... NE). If a diagonal entry is zero, it can be omitted. Multiple entries are permitted. If IRN(k) or JCN(k) are less than 1 or greater than N, the entry is ignored. These arrays are not altered by any of the calls to the ME57 package. They must be preserved by the user between this call and a call to ME57D/DD for the same matrix. **Restriction:** When KIND≠1, JCN(k) ≥ IRN(k), k = 1, ... NE.

LKEEP is an INTEGER variable that must be set by the user to the length of array KEEP. It might be more efficient to allocate more than the minimum required, say about N to 2*N more space. **Restriction:** LKEEP ≥ 5*N+NE+MAX(N,NE)+42.

KEEP is an INTEGER array of length LKEEP. It need not be set by the user and must be preserved between a call to ME57A/AD and subsequent calls to ME57B/BD. If the user wishes to input the pivot sequence, the position of variable i in the pivot order should be placed in KEEP(I), I = 1, 2,..., N and ICNTL(6) should be set to 1. The subroutine may replace the given order by another that gives the same fill-in pattern and virtually identical numerical results.

IWORK is an INTEGER array of length 5*N that is used as workspace.

ICNTL is an INTEGER array of length 20 that contains control parameters and must be set by the user. Default values for the components may be set by a call to ME57I/ID. Details of the control parameters are given in Section 2.2.

INFO is an INTEGER array of length 40 that need not be set by the user. On return from ME57A/AD, a non-negative value for INFO(1) indicates that the subroutine has performed successfully. For nonzero values, see Section 2.3. For the meanings of the values of other components of INFO, see Section 2.2.

RINFO is a REAL array of length 20 that need not be set by the user. This array supplies information on the execution of the subroutine. For the meanings of the values of the components of RINFO, see Section 2.2.

2.1.3 To factorize a matrix (or continue factorization after restart)

The single precision version

```
      CALL ME57B(KIND,N,NE,A,FACT,LFACT,IFACT,LIFACT,LKEEP,KEEP,IWORK,WORK,ICNTL,CNTL,
     *           INFO,RINFO)
```

The double precision version

```
      CALL ME57BD(KIND,N,NE,A,FACT,LFACT,IFACT,LIFACT,LKEEP,KEEP,IWORK,WORK,ICNTL,CNTL,
     *            INFO,RINFO)
```

KIND is an INTEGER variable that must be set by the user to indicate whether the matrix is complex symmetric (KIND=1) or complex Hermitian (KIND≠1).

N is an INTEGER variable that must be set by the user to the order n of the matrix **A**. It must be unchanged since the call to ME57A/AD and is not altered by the subroutine. **Restriction:** N ≥ 1.

NE is an INTEGER variable that must be set by the user to the number of entries in the matrix **A**. It must be unchanged since the call to ME57A/AD and is not altered by the subroutine. **Restriction:** NE ≥ 0.

A is a COMPLEX array of length NE that must be set by the user so that A(k) holds the value of the diagonal entry or pair of off-diagonal entries whose indices were held in IRN(k) and JCN(k) on entry to ME57A/AD, for k = 1, 2,..., NE. Multiple entries are summed and any that correspond to an IRN(k) or JCN(k) value that was out of range are ignored. The array A is not altered by the subroutine and must be preserved by the user between this call and a call to ME57D/DD for the same matrix.

FACT is a COMPLEX array of length LFACT. It need not be set by the user and, on exit, will hold the entries of the factors of the matrix **A**. These entries in FACT must be preserved by the user between calls to this subroutine and subsequent calls to ME57C/CD or ME57D/DD. If ME57B/BD is being called after a call to ME57E/ED, then FACT must be the same array as the array NEWFAC as output from ME57E/ED.

LFACT is an INTEGER variable that must be set by the user to the length of array FACT. It must be at least as great as INFO(9) as output from ME57A/AD (see Section 2.2). It is advisable to allow a slightly greater value because numerical pivoting might increase storage requirements. In extreme cases, for example when there are many zeros on the diagonal, a value of more than twice INFO(9) might be needed. It is not altered by the subroutine.

IFACT is an INTEGER array of length LIFACT. It need not be set by the user and, on exit, holds integer indexing information on the matrix factors. It must be preserved by the user between calls to this subroutine and subsequent calls to ME57C/CD or ME57D/DD. If ME57B/BD is being called after a call to ME57E/ED, then IFACT must be the same array as the array NEWIFC as output from ME57E/ED.

LIFACT is an INTEGER variable that must be set by the user to the length of array IFACT. It must be at least as great as INFO(10) as output from ME57A/AD (see Section 2.2). A slightly greater value is recommended because numerical pivoting may increase storage requirements. It is not altered by the subroutine.

LKEEP is an INTEGER variable that must be set by the user to the length of array KEEP. **Restriction:** LKEEP ≥ 5*N+NE+MAX(N,NE)+42.

KEEP is an INTEGER array of length LKEEP that must be unchanged since the call to ME57A/AD. It is not altered by

ME57B/BD.

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

WORK is a REAL array of length at least 2*NE+3*N. It is used as workspace by the subroutine.

ICNTL is an INTEGER array of length 20 that contains control parameters and must be set by the user. Default values for the components may be set by a call to ME57I/ID. Details of the control parameters are given in Section 2.2.

CNTL is a REAL array of length 5 that contains control parameters and must be set by the user. Default values for the components may be set by a call to ME57I/ID. Details of the control parameters are given in Section 2.2.

INFO is an INTEGER array of length 40 that need not be set by the user. On return from ME57B/BD, a non-negative value for INFO(1) indicates that the subroutine has performed successfully. For nonzero values, see Section 2.3. For the meanings of the values of other components of INFO, see Section 2.2.

RINFO is a REAL array of length 20 that need not be set by the user. This array supplies information on the execution of the subroutine. For the meanings of the values of the components of RINFO, see Section 2.2.

2.1.4 To solve equations using the factors from ME57B/BD without iterative refinement

The single precision version

```
      CALL ME57C(KIND,JOB,N,FACT,LFACT,IFACT,LIFACT,NRHS,RHS,LRHS,WORK,
     *           LWORK,IWORK,ICNTL,INFO)
```

The double precision version

```
      CALL ME57CD(KIND,JOB,N,FACT,LFACT,IFACT,LIFACT,NRHS,RHS,LRHS,WORK,
     *           LWORK,IWORK,ICNTL,INFO)
```

KIND is an INTEGER variable that must be set by the user to indicate whether the matrix is complex symmetric (KIND=1) or complex Hermitian (KIND≠1).

JOB is an INTEGER variable that must be set by the user to determine the system of equations being solved. If JOB is less than than or equal to 1, then the system(s)

$$\mathbf{Ax}=\mathbf{b} \text{ or } \mathbf{AX}=\mathbf{B}$$

is solved.

For JOB equal to 2, 3, or 4, a partial solution is obtained. Specifically, if the factorization is expressed as

$$\mathbf{PSASP}^T = \mathbf{LDL}^T,$$

where **S** is the diagonal scaling matrix from the scaling when ICNTL(15) equals 1 and **P** the permutation matrix, then the systems solved are:

JOB = 2 $\quad \mathbf{LSx} = \mathbf{PSb} \quad$ or $\quad \mathbf{LSX} = \mathbf{PSB}$

JOB = 3 $\quad \mathbf{DS}^{-1}\mathbf{x} = \mathbf{Sb} \quad$ or $\quad \mathbf{DS}^{-1}\mathbf{X} = \mathbf{SB}$

JOB ≥ 4 $\quad \mathbf{L}^T\mathbf{P}^{-1}\mathbf{S}^{-1}\mathbf{x} = \mathbf{S}^{-1}\mathbf{b} \quad$ or $\quad \mathbf{L}^T\mathbf{P}^{-1}\mathbf{S}^{-1}\mathbf{X} = \mathbf{S}^{-1}\mathbf{B}$

where the right-hand side(s) **b** or **B** is input in array RHS.

N is an INTEGER variable that must be set by the user to the order *n* of the matrix **A**. It must be unchanged since the call to ME57B/BD and is not altered by the subroutine.

FACT is a COMPLEX array of length LFACT that must be unchanged since the call to ME57B/BD. It is not altered by the subroutine.

LFACT is an INTEGER variable that must be set by the user to the length of array FACT. It is not altered by the subroutine.

IFACT is an INTEGER array of length LIFACT that must be unchanged since the call to ME57B/BD. It is not altered by the subroutine.

LIFACT is an INTEGER variable that must be set by the user to the length of array IFACT. It is not altered by the subroutine.

NRHS is an INTEGER variable that must be set by the user to the number of right-hand sides, *s*, being solved. It is not altered by the subroutine. **Restriction:** NRHS > 0.

RHS is a COMPLEX array of dimensions LRHS by NRHS. On entry, RHS(I,J) must hold the Ith component of the Jth right-hand side of the equations being solved. On exit it will hold the corresponding entry of the Jth solution vector.

LRHS is an INTEGER variable that must be set by the user to the leading dimension of array RHS. It is not altered by the subroutine. **Restriction:** LRHS ≥ N.

WORK is a COMPLEX array of length LWORK that is used as workspace by ME57C/CD.

LWORK is an INTEGER variable that must be set by the user to the length of array WORK. It must be at least as large as N*NRHS. It is not altered by ME57C/CD. **Restriction:** LWORK ≥ N*NRHS.

IWORK is an INTEGER array of length N that is used as workspace by ME57C/CD.

ICNTL is an INTEGER array of length 20 that contains control parameters and must be set by the user. Default values for the components may be set by a call to ME57I/ID. Details of the control parameters are given in Section 2.2.

INFO is an INTEGER array of length 40 that need not be set by the user. On return from ME57E/ED, a non-negative value for INFO(1) indicates that the subroutine has performed successfully. For nonzero values, see Section 2.3.

2.1.5 To solve equations using iterative refinement

The single precision version

```
CALL ME57D(KIND,JOB,N,NE,A,IRN,JCN,FACT,LFACT,IFACT,LIFACT,RHS,X,RESID,
           WORKC,WORKR,IWORK,ICNTL,CNTL,INFO,RINFO)
```

The double precision version

```
CALL ME57DD(KIND,JOB,N,NE,A,IRN,JCN,FACT,LFACT,IFACT,LIFACT,RHS,X,RESID,
            WORKC,WORKR,IWORK,ICNTL,CNTL,INFO,RINFO)
```

KIND is an INTEGER variable that must be set by the user to indicate whether the matrix is complex symmetric (KIND=1) or complex Hermitian (KIND≠1).

JOB must be set by the user to determine what action is desired by the user. Values of JOB and their effect are:
If ICNTL(9)>1, then JOB=0 if no estimate of solution in X, and JOB=2 if estimate of solution in X.
If ICNTL(9)=1, then:

0: Solve **Ax**=**b**, calculate residual **r=b**–**Ax** and exit.
(Note that ME57C/CD should be used if solution without residual is required)

1: Solve **Ax=b**, calculate residual **r=b**–**Ax**, solve **A(dx)=r**, update solution and exit (that is, we perform one step of iterative refinement).

If JOB > 1, an estimate of the solution must be input in X.

2: Calculate residual **r=b**–**Ax**, solve **A(dx)=r**, update solution and exit.

If JOB > 2, the residual for this estimate must also be input.

3: Solve **A(dx)=r**, update solution and exit.

4: Solve **A(dx)=r**, update solution, calculate residual for new solution and exit.
Restriction: 0 ≤ JOB ≤ 4 and JOB = 0 or 2 if ICNTL(9) > 1.

N is an INTEGER variable that must be set by the user to the order *n* of the matrix **A**. It must be unchanged since the call to ME57B/BD and is not altered by the subroutine. **Restriction:** N ≥ 1.

NE is an INTEGER variable that must be set by the user to the number of entries in the matrix **A**. It must be unchanged since the call to ME57A/AD and is not altered by the subroutine. **Restriction:** NE ≥ 0.

A is a COMPLEX array of length NE that must be unchanged since the call to ME57B/BD. It is not altered by the subroutine.

IRN and JCN are INTEGER arrays of length NE that must be unchanged since the call to ME57A/AD. They are not altered by the subroutine.

FACT is a COMPLEX array of length LFACT that must be unchanged since the call to ME57B/BD. It is not altered by the subroutine.

LFACT is an INTEGER variable that must be set by the user to the length of array FACT. It is not altered by the subroutine.

IFACT is an INTEGER array of length LIFACT that must be unchanged since the call to ME57B/BD. It is not altered by the subroutine.

LIFACT is an INTEGER variable that must be set by the user to the length of array IFACT. It is not altered by the subroutine.

RHS is a COMPLEX array of of length N that must be set by the user to the right-hand side for the equation being solved. It is not altered by the subroutine.

X is a COMPLEX array of of length N. If JOB≥2, it must be set by the user to an estimated solution. Otherwise, it need not be set by the user. On exit, it holds the improved solution vector.

RESID is a COMPLEX array of length N. If JOB>2, it must be set on entry to the value of the residual for the current solution estimate held in X. Otherwise, it need not be set by the user on entry. If JOB=0 or 4 or if ICNTL(9)>1, on exit it will hold the residual vector for the equations being solved. If 1≤JOB≤3, then RESID will hold on exit the last correction vector added to the solution X.

WORKC is a COMPLEX two-dimensional work array of first dimension N. If ICNTL(9)=1, its second dimension must be at least 1. If ICNTL(9)>1 and ICNTL(10)>0, its second dimension must be at least 2.

WORKR is a REAL two-dimensional work array of first dimension N. If ICNTL(9)=1, it is not accessed. If ICNTL(9)>1, its second dimension must be at least 2.

IWORK is an INTEGER work array. If ICNTL(9)=1, it is not accessed. If ICNTL(9)>1, it must be of length at least N.

ICNTL is an INTEGER array of length 20 that contains control parameters and must be set by the user. Default values for the components may be set by a call to ME57I/ID. Details of the control parameters are given in Section 2.2.

CNTL is a REAL array of length 5 that contains control parameters and must be set by the user. Default values for the components may be set by a call to ME57I/ID. Details of the control parameters are given in Section 2.2.

INFO is an INTEGER array of length 40 that need not be set by the user. On return from ME57D/DD, a non-negative value for INFO(1) indicates that the subroutine has performed successfully. For nonzero values, see Section 2.3.

RINFO is a REAL array of length 20 that need not be set by the user. This array supplies information on the execution of ME57D/DD. For the meanings of the values of the components of RINFO, see Section 2.2.

2.1.6 To copy data into a larger array.

If the arrays FACT or IFACT are not of sufficient length for ME57B/BD to complete the factorization, the contents of these arrays can be moved to larger arrays (NEWFAC and NEWIFC, respectively) by calling ME57E/ED. ME57B/BD can then be recalled using these arrays in place of FACT and IFACT, and the factorization will start from where it finished during the previous factorization attempt without the need to re-perform the computations up to that point.

The single precision version

```
CALL ME57E(N,IC,KEEP,FACT,LFACT,NEWFAC,LNEW,IFACT,LIFACT,NEWIFC,LINEW,INFO)
```

The double precision version

```
CALL ME57ED(N,IC,KEEP,FACT,LFACT,NEWFAC,LNEW,IFACT,LIFACT,NEWIFC,LINEW,INFO)
```

N is an INTEGER variable that must be set by the user to the order n of the matrix **A**. It must be unchanged since the failed call to ME57B/BD and is not altered by the subroutine.

IC is an INTEGER variable that must be set by the user to 0 if the real array is being copied, and to ≥1 if the integer array is being copied. Values less than 0 are treated as 0.

KEEP in an INTEGER array of length 5*N+NE+MAX(N,NE)+42. This array should be unchanged since the failed call to ME57B/BD and should be preserved by the user between this call and a subsequent call to ME57B/BD.

FACT is a COMPLEX array of length LFACT that must be unchanged since the failed call to ME57B/BD. It is unchanged by ME57E/ED.

LFACT is an INTEGER variable that must be set to the length of array FACT. It is unchanged by ME57E/ED.

NEWFAC is a COMPLEX array of length LNEW that need not be set by the user and is only accessed if IC < 1. On a successful exit from ME57E/ED, NEWFAC will hold the factors and other data generated by ME57B/BD up to the point at which it failed. ME57B/BD can now be called using this array in place of FACT.

LNEW is an INTEGER variable that must be set to the length of array NEWFAC. It is unchanged by ME57E/ED. **Restriction:** LNEW > LFACT.

IFACT is an INTEGER array of length LIFACT that must be unchanged since the failed call to ME57B/BD. It is unchanged by ME57E/ED.

LIFACT is an INTEGER variable that must be set to the length of array IFACT. It is unchanged by ME57E/ED.

NEWIFC is an INTEGER array of length LINEW that need not be set by the user and is only accessed if IC ≥ 1. On a successful exit from ME57E/ED, NEWIFC will hold the integer information generated by ME57B/BD up to the point at which it failed. ME57B/BD can now be called using this array in place of IFACT.

LINEW is an INTEGER variable that must be set to the length of array NEWIFC. It is unchanged by ME57E/ED. **Restriction:** LINEW > LIFACT.

INFO is an INTEGER array of length 40 that need not be set by the user. On return from ME57E/ED, a non-negative value for INFO(1) indicates that the subroutine has performed successfully. For nonzero values, see Section 2.3.

2.2 Arrays for control and information

The components of the arrays CNTL and ICNTL control the action of ME57. Default values are set by ME57I/ID. Only the first five components of CNTL and the first 16 components of ICNTL are presently used by ME57. The remaining components are set to zero by ME57I/ID and are not accessed in ME57. The components of the arrays RINFO and INFO provide information on the action of ME57.

CNTL(1) has default value 0.01 and is used for threshold pivoting. Values less than 0.0 will be treated as 0.0 and values greater than 0.5 as 0.5. Values near 0.0 may perhaps give faster factorization times and less entries in the factors but may result in a less stable factorization. This parameter is only accessed if ICNTL(7) is equal to 1.

CNTL(2) has default value 10^{-20}. ME57B/BD will treat any pivot whose modulus is less than CNTL(2) as zero. If ICNTL(16) = 1, then blocks of entries less than CNTL(2) can be discarded during the factorization and the corresponding pivots are placed at the end of the ordering. In this case, a normal value for CNTL(2) could be 10^{-12}.

CNTL(3) has default value 0.5. It is used by ME57D/DD to monitor the convergence of the iterative refinement. If the norm of the scaled residuals does not decrease by a factor of at least CNTL(3), convergence is deemed to be too slow and ME57D/DD terminates with INFO(1) set to -8.

CNTL(4) has default value 0.0. It is used by ME57B/BD to control the static pivoting (see Section 4). If CNTL(4)>0.0, then small pivots may be replaced by entries of value CNTL(4) so that the factorization could be inaccurate. If CNTL(5) is also greater than zero, then CNTL(4) is treated as zero (that is uneliminated variables are delayed) until CNTL(5)*N fully summed variables have been delayed. If static pivots are used, it is recommended that iterative refinement is used when computing the solution (see Section 2.1.5).

CNTL(5) has default value 0.0. Static pivoting is invoked if CNTL(4) is greater than zero and the accumulated number of delayed pivots exceeds CNTL(5)*N.

ICNTL(1) has default value 6. It is the output stream for error messages. If it is negative, these messages will be suppressed.

ICNTL(2) has default value 6. It is the output stream for warning messages. If it is negative, these messages are suppressed.

ICNTL(3) has default value 6. It is the output stream for monitoring printing. If it is negative, this printing is suppressed.

ICNTL(4) has default value -1. It is the output stream for the printing of statistics. If it is negative, the statistics are not printed. This control parameter is not operative in the present version.

ICNTL(5) is used by ME57 to control printing of error, warning, and monitoring messages. It has default value 2. Possible values are:

- < 1 No messages output.
- 1 Only error messages printed.
- 2 Errors and warnings printed.
- 3 Errors and warnings and terse monitoring (only first ten entries of arrays printed).
- ≥ 4 Errors and warnings and all information on input and output parameters printed.

ICNTL(6) has default value 5 and must be set by the user to 1 if the pivot order in KEEP is to be used by ME57A/AD. For any other value of ICNTL(6), a suitable pivot order will be chosen automatically. Our experience is that the choice of ordering can be quite crucial to performance so our default invokes logic that chooses automatically between the MeTiS ordering and a version of approximate minimum degree. In general, the MeTiS ordering gives the best results for large matrices and so we recommend that the user should ensure that this library is available. It can be obtained from:

http://www-users.cs.umn.edu/˜karypis/metis/metis/download.html

Currently the following choices are available:

- 0 AMD ordering using MC47 (without the detection of dense rows).
- 2 AMD ordering using MC47.
- 3 Minimum degree ordering as generated by the MA27 code.
- 4 MeTiS ordering is used. Note that the user needs to supply the MeTiS library. If it is not supplied and

this option is requested, the routine will return immediately with INFO(1) set to -18.

5 An automatic choice is made between the MeTiS ordering and MC47. If the MeTiS library is not supplied and this option is requested, the routine uses MC47.

≥ 6 Currently is equivalent to setting ICNTL(6)=5, but may be used for alternative orderings in later releases of ME57.

ICNTL(7) has default value 1 and is used to control numerical pivoting. It is used in ME57A/AD in the automatic selection of an ordering (if ICNTL(6) = 5) and by ME57B/BD to determine the pivoting strategy. Values out of range cause an error return with INFO(1) equal to -10. Possible values are:

1 Numerical pivoting is performed using the threshold value in CNTL(1).

2 No pivoting will be performed and an error exit will occur immediately a sign change or a zero is detected among the pivots. This is suitable for cases when **A** is thought to be definite and is likely to decrease the factorization time while still providing a stable decomposition.

3 No pivoting will be performed and an error exit will occur if a pivot of modulus less than or equal to CNTL(2) is detected. This is likely to decrease the factorization time, but may be unstable if there is a sign change among the pivots.

4 No pivoting will be performed but the matrix will be altered so that all pivots are of the same sign. The alterations to the pivots will be returned by ME57B/BD in entries LFACT-N to LFACT-1 of array FACT, if scaling is switched off (ICNTL(15) ≠ 1), and from positions LFACT-2*N to LFACT-N-1, otherwise. ICNTL(7) must also be set to 4 on entry to ME57A/AD so that the analysis can predict the space required when using this option. This option is only operative when the matrix is Hermitian (KIND≠1).

ICNTL(8) has default value 0. If ME57B/BD is called with ICNTL(8) = 0 and runs out of space, then the factorization will discard factors and try to continue the factorization to determine the amount of space needed for a successful factorization. In this case, a factorization will not have been produced. If ICNTL(8) ≠ 0, and the factorization stops because of lack of space, the user must allocate a larger array to replace FACT or IFACT and transfer data from FACT or IFACT by using ME57E/ED. ME57B/BD can then be called again using these larger arrays, and the factorization will continue from the point at which it failed.

ICNTL(9) has default value 10. It corresponds to the maximum permitted number of steps of iterative refinement and is only operational on a call to ME57D/DD. If ICNTL(9) = 1, one step of iterative refinement is performed but no error analysis is done. If ICNTL(9) > 1, iterative refinement is performed and an estimate of the backward error is returned in RINFO(6) and RINFO(7), and the infinity norm of the input matrix, the infinity norm of the computed solution, and the norm of the scaled residual in RINFO(8) to RINFO(10), respectively. If ICNTL(10) is also positive, an estimate of the condition number and error are also returned. If ICNTL(9) < 0, then ME57D/DD will return with INFO(1)=–13.

ICNTL(10) has default value 0. On a call to ME57D/DD, if ICNTL(9) > 1, a positive value for ICNTL(10) will return estimates of the condition number of the system in RINFO(11) and RINFO(12) and an estimate of the error in RINFO(13).

ICNTL(11) has default value 16 and is the block size used by the Level 3 BLAS in ME57B/BD.

ICNTL(12) has default value 16. Two nodes of the assembly tree are merged only if both involve less than ICNTL(12) eliminations.

ICNTL(13) has default value 10 and is a control on the number of columns in a block pivot above which the Level 2 and Level 3 BLAS are used in ME57C/CD and ME57D/DD.

ICNTL(14) has default value 100 and is the percentage density of a row that is regarded as dense during the pivot selection strategy in ME57A/AD that uses the MA27 minimum degree ordering (ICNTL(6) equal to 3). This is useful since the ordering strategy can be slow on matrices with dense rows if this option is not used.

ICNTL(15) has default value 1 and indicates that the matrix will be scaled using a symmetrized version of the HSL code MC64. Setting ICNTL(15) to any other value will suppress this option. Scaling factors will be returned by ME57B/BD in entries LFACT-N to LFACT-1 of array FACT. Although the scaling factors are computed in the call to ME57B/BD, this parameter should also be set for the call to ME57A/AD so that the correct amount of storage can be predicted.

ICNTL(16) has default value 0. If ICNTL(16) is set to 1, then when small entries (defined by CNTL(2)) are detected, they are removed and the corresponding pivots placed at the end of the factorization. This can be particularly efficient if the matrix is highly rank deficient.

ICNTL(17) to ICNTL(20) are set to zero by ME57I/ID but are not currently used by ME57.

INFO(1) is zero if the routine is successful, is negative if an error occurred, and is positive for a warning (see Section 2.3).

INFO(2) holds additional information concerning the error or warning (see Section 2.3).

*Statistics produced after analysis phase (*ME57A/AD*). The forecasts will be correct if the matrix is definite.*

INFO(3) Number of indices out of range.

INFO(4) Number of off-diagonal duplicates.

INFO(5) Forecast number of reals in the factors.

INFO(6) Forecast number of integers in the factors.

INFO(7) Forecast maximum frontal matrix size.

INFO(8) Number of nodes in the assembly tree.

INFO(9) and INFO(10) are lengths of arrays FACT and IFACT required for a successful completion of ME57B/BD without the need for data compression provided no numerical pivoting is performed. The actual amount required may be higher because of numerical pivoting.

INFO(11) and INFO(12) are lengths of arrays FACT and IFACT required for a successful completion of ME57B/BD allowing data compression, again provided no numerical pivoting is performed. Numerical pivoting may cause a higher value to be required.

INFO(13) Number of data compresses. If this is high (say > 10), the performance of ME57A/AD may be improved by increasing the length of array KEEP.

INFO(36) Ordering actually used (refer to numbering scheme for ICNTL(6)).

*Statistics produced after factorization (*ME57B/BD*)*

INFO(14) Number of entries in factors.

INFO(15) Storage for real data for factors.

INFO(16) Storage for integer data for factors.

INFO(17) and INFO(18). The minimum length of arrays FACT and IFACT, respectively, that would be sufficient for a successful factorization of this matrix. In the event of a failure with INFO(1) equal to -3 (-4), INFO(17)(INFO(18)) is set to the estimated real (integer) space required.

INFO(19) and INFO(20). The minimum length of arrays FACT and IFACT, respectively, that would be sufficient for a successful factorization of this matrix without any compresses on the data structures.

INFO(21) Order of largest frontal matrix.

INFO(22) Number of 2×2 pivots used in the factorization.

INFO(23) Total number of fully-summed variables that were passed to the father node because of pivoting considerations (called delayed pivots. If a variable is passed further up the tree, it will be counted again. This will be zero for a definite system, if `ICNTL(7)` > `1` or if static pivoting is used (`CNTL(4)` > `0.0` and `CNTL(5)` equal to zero).

`INFO(24)` Number of negative eigenvalues in the factorization of **A**.

`INFO(25)` Rank of the factorization of **A**. Note that, if static pivoting is invoked (see `CNTL(4)` in Section 2.2), then the rank of the factorization will always be *n*.

`INFO(26)` Number of sign changes of pivot when `ICNTL(7)` = `3`.

`INFO(27)` Pivot step at which matrix modification commences when `ICNTL(7)` = `4`. `INFO(27)` is set to `0` if no modification performed.

`INFO(28)` and `INFO(29)` Number of data compresses on real and integer data structures respectively. If either of these is high (say > 10), then the speed of the factorization may be increased by allocating more space to the arrays `IFACT` or `FACT` as appropriate.

`INFO(31)` Number of block pivots in factors.

`INFO(32)` Number of zeros in the triangle of the factors.

`INFO(33)` Number of zeros in the rectangle of the factors.

`INFO(34)` Number of zero columns in rectangle of the factors.

`INFO(35)` Number of pivots chosen in static pivoting phase.

Statistics produced after solution using `ME57D/DD` *with* `ICNTL(9)` > `1`

`INFO(30)` Number of solution steps in iterative refinement.

Not currently used by `ME57`.

`INFO(37)` to `INFO(40)`.

Statistics produced after analysis phase

`RINFO(1)` The forecast number of floating-point additions for the assembly processes.

`RINFO(2)` The forecast number of floating-point operations to perform the elimination operations (counting a multiply-add pair as two operations).

Statistics produced after factorization

`RINFO(3)` Number of floating-point additions for the assembly process.

`RINFO(4)` Number of floating-point operations for the elimination.

`RINFO(5)` Additional number of floating-point operations when using `_GEMM` for the elimination.

`RINFO(14)` Maximum change to pivot when `ICNTL(7)` = `4`.

`RINFO(15)` Smallest pivot in modified matrix when `ICNTL(7)` = `4`.

`RINFO(16)` Minimum value of scaling factor when `ICNTL(15)` = `1`.

`RINFO(17)` Maximum value of scaling factor when `ICNTL(15)` = `1`.

`RINFO(18)` Maximum modulus of matrix entry (held in `FACT(LFACT)`).

Statistics produced after solve using `ME57D/DD` *with* `ICNTL(9)` > `1`.

`RINFO(6)` and `RINFO(7)` are used to hold information on the backward error. See Section 2.4.

`RINFO(8)` Infinity norm of the input matrix, where

$$\|\mathbf{A}\|_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}| \ .$$

RINFO(9) Infinity norm of the computed solution, where

$$\|\mathbf{x}\|_\infty = \max_{1\le i\le n} |x_i| \ .$$

RINFO(10) Norm of scaled residuals

$$\frac{\max_{1\le i\le n} \left| \sum_{j=1}^{n} a_{ij} x_j - b_i \right|}{\|\mathbf{A}\|_\infty \ \|\mathbf{x}\|_\infty} \ .$$

The following are output after a call to ME57D/DD with ICNTL(9) > 1 and ICNTL(10) > 0.

RINFO(11) and RINFO(12) are the condition number for the system as used by Arioli, Demmel, and Duff.

RINFO(13) is an upper bound for the infinity norm of the error in the solution.

Not currently used by ME57.

RINFO(19) to RINFO(20).

2.3 Error diagnostics

A successful return is indicated by a value of INFO(1) equal to zero. Possible nonzero values for INFO(1) are given below. In each case an identifying message is output on unit ICNTL(1) (errors) or ICNTL(2) (warnings).

A negative value of INFO(1) is associated with an error message that will be output on unit ICNTL(2).

-1 Value of N is out of range on a call to ME57A/AD, ME57B/BD, ME57C/CD, or ME57D/DD. Value given is held in INFO(2).

-2 Value of NE is out of range on a call to ME57A/AD, ME57B/BD, or ME57D/DD. Value given is held in INFO(2).

-3 Failure due to insufficient REAL space on a call to ME57B/BD. INFO(17) is set to a value that may suffice for LFACT. INFO(2) is set to value of LFACT. The user can allocate a larger array and copy the contents of FACT into it using ME57E/ED, and recall ME57B/BD.

-4 Failure due to insufficient INTEGER space on a call to ME57B/BD. INFO(18) is set to a value that may suffice for LIFACT. INFO(2) is set to value of LIFACT. The user can allocate a larger array and copy the contents of IFACT into it using ME57E/ED, and recall ME57B/BD.

-5 A pivot with magnitude less than or equal to CNTL(2) was found at pivot step INFO(2) when calling ME57B/BD with ICNTL(7) = 2 or 3, or the correction obtained when using matrix modification does not give a pivot greater than CNTL(2) when ICNTL(7) = 4.

-6 A change of sign of pivots has been detected when ICNTL(7) = 2. INFO(2) is set to the pivot step at which the change was detected on a call to ME57B/BD.

-7 Either LNEW < LFACT or LINEW < LIFACT on a call to ME57E/ED. INFO(2) is set to LNEW or LINEW as appropriate.

-8 Iterative refinement fails to converge in specified number of iterations on a call to ME57D/DD.

-9 Error in permutation array when ICNTL(6)=1 on a call to ME57A/AD. INFO(2) holds first component at which error was detected.

-10 Value of ICNTL(7) out of range on a call to ME57B/BD. Value given held in INFO(2).

-11 LRHS < N on a call to ME57C/CD. INFO(2) holds value of LRHS.

-12 Invalid value for JOB on a call to ME57D/DD. Value given held in INFO(2).

-13 Invalid value of ICNTL(9) on a call to ME57D/DD. Value given held in INFO(2).

-14 Failure of MC71A/AD on a call to ME57D/DD with ICNTL(10)> 0.

-15 LKEEP less than 5*N+NE+MAX(N,NE)+42 on a call to ME57A/AD or ME57B/BD. INFO(2) holds value of LKEEP.

-16 NRHS less than 1 on call to ME57C/CD. INFO(2) holds value of NRHS.

-17 LWORK too small on entry to ME57C/CD. INFO(2) holds minimum length required.

-18 An ordering using MeTiS was requested in the call to ME57A/AD but the MeTiS package was not linked in.

-19 The matrix is Hermitian, but in a call to ME57A/AD an entry is in the lower triangle. The position of the first entry so encountered is returned in INFO(2).

-20 The matrix is Hermitian, but in a call to ME57B/BD with scaling on, a diagonal entry is not real. A diagonal entry with a nonzero imaginary part is given in INFO(2).

A positive value of INFO(1) is associated with a warning message that will be output on unit ICNTL(2).

+1 Index (in IRN or JCN) out of range on call to ME57A/AD or ME57D/DD. Action taken by subroutine is to ignore any such entries and continue. INFO(3) is set to the number of out-of-range entries. Details of the first ten are printed on unit ICNTL(2).

+2 Duplicate indices on call to ME57A/AD or ME57D/DD. Action taken by subroutine is to keep the duplicates and then to sum corresponding reals when ME57B/BD is called. INFO(4) is set to the number of duplicate entries. Details of the first ten are printed on unit ICNTL(2).

+3 Both out-of-range indices and duplicates exist.

+4 Matrix is rank deficient on exit from ME57B/BD. In this case, a decomposition will still have been produced that will enable the subsequent solution of consistent equations. INFO(25) will be set to the rank of the factorized matrix.

+5 Pivots have different signs when factorizing a supposedly definite matrix (ICNTL(7) = 3) on call to ME57B/BD. INFO(26) is set to the number of sign changes.

+8 During error analysis the infinity norm of the computed solution was found to be zero.

+10 Insufficient real space to complete factorization when ME57B/BD called with ICNTL(8) ≠ 0. User can copy real values to a longer array using ME57E/ED and recall ME57B/BD using this longer array to continue the factorization.

+11 Insufficient integer space to complete factorization when ME57B/BD called with ICNTL(8) ≠ 0. User can copy integer values to a longer array using ME57E/ED and recall ME57B/BD using this longer array to continue the factorization.

2.4 Error estimates

We calculate an estimate of the sparse backward error using the theory and measure developed by Arioli, Demmel, and Duff (1989). We use the notation $\bar{\mathbf{x}}$ for the computed solution and a modulus sign on a vector or matrix to indicate the vector or matrix obtained by replacing all entries by their moduli. The scaled residual

$$\frac{|\mathbf{b}-\mathbf{A}\bar{\mathbf{x}}|_i}{(|\mathbf{b}|+|\mathbf{A}||\bar{\mathbf{x}}|)_i} \tag{1}$$

is calculated for all equations except those for which the numerator is nonzero and the denominator is small. For the exceptional equations,

$$\frac{|\mathbf{b}-\mathbf{A}\bar{\mathbf{x}}|_i}{(|\mathbf{A}||\bar{\mathbf{x}}|)_i+\|\mathbf{A}_i\|_\infty\,\|\bar{\mathbf{x}}\|_\infty} \tag{2}$$

is used instead, where $\mathbf{A}_i$ is row i of $\mathbf{A}$. The largest scaled residual (1) is returned in RINFO(6) and the largest scaled residual (2) is returned in RINFO(7). If all equations are in category (1), zero is returned in RINFO(7). The computed solution $\bar{\mathbf{x}}$ is the exact solution of the equation

$$(\mathbf{A}+\delta\mathbf{A})\mathbf{x}=(\mathbf{b}+\delta\mathbf{b})$$

where $\delta\mathbf{A}_{ij} \le \max(\texttt{RINFO(6)},\texttt{RINFO(7)})|\mathbf{A}|_{ij}$ and $\delta\mathbf{b}_i \le \max(\texttt{RINFO(6)}|\mathbf{b}|_i,\texttt{RINFO(7)}\|\mathbf{A}_i\|_\infty\,\|\bar{\mathbf{x}}\|_\infty)$. Note that $\delta\mathbf{A}$ respects the sparsity in $\mathbf{A}$. An upper bound for the forward error is returned in RINFO(13) that is computed as RINFO(6)*RINFO(11) + RINFO(7)*RINFO(12) where RINFO(11) and RINFO(12) are condition numbers corresponding to κ_{ω_1} and κ_{ω_2}, respectively as defined in equation (15) of Arioli, Demmel, and Duff (1989).

Reference

Arioli, M. Demmel, J. W., and Duff, I. S. (1989). Solving sparse linear systems with sparse backward error. *SIAM J. Matrix Anal. and Applics.* **10**, 165-190.

3 GENERAL INFORMATION

Use of common: None.

Other routines called directly:

BLAS **routines:** CTPSV/ZTPSV, CGEMM/ZGEMM, and CGEMV/ZGEMV.

Routine from MeTiS package (optional): METIS_NODEND.

HSL routines: FD15A/AD, MC34A/AD, MC64W/WD, MF71A/AD, and MC47B/BD.

All the other subroutines called by the principal subroutines are in the ME57 package. They are called ME57F/FD, ME57G/GD, ME57H/HD, ME57J/JD, ME57K/KD, ME57L/LD, ME57M/MD, ME57N/ND, ME57O/OD, ME57P/PD, ME57Q/QD, ME57R/RD, ME57S/SD, ME57T/TD, ME57U/UD, ME57V/VD, ME57W/WD, ME57X/XD, ME57Y/YD, and ME57Z/ZD.

Input/output: Error, warning, monitoring, and statistical messages only. Error messages on unit ICNTL(1), which has default value 6. Warning and monitoring messages on unit ICNTL(2) and ICNTL(3), respectively, which have default value 6. Printing of these messages is controlled by ICNTL(5) and all messages are suppressed if ICNTL(5) is less than one or if the appropriate unit number is set to a negative value.

Restrictions: N ≥ 1, NE ≥ 0 on entry to ME57A/AD, ME57B/BD, ME57D/DD.
0 ≤ JOB ≤ 4 and JOB = 0 or 2 if ICNTL(9) > 1 on entry to ME57D/DD.
LNEW > LFACT and LINEW > LIFACT on entry to ME57E/ED.
NRHS > 0 and LRHS ≥ N and LWORK ≥ NRHS*N on entry to ME57C/CD.

Changes between Version 1.0.0 and Version 2.0.0:

Version 2.0.0 incorporates several additional features to those of Version 1.0.0. We give information on these in this section and indicate where default values of control parameters have changed from the earlier version.

Built-in scaling is now available through a symmetrized version of MC64. This is controlled by the new control parameter ICNTL(15) whose default is set so that scaling is performed. The matrix is explicitly scaled internally to the package as are the right-hand side and the solution so that the user need not be concerned with this. Iterative refinement, if requested, is based on the original unscaled matrix. The minimum and maximum scale factors are returned in RINFO(16) and RINFO(17), respectively.

The user can now use an ordering from the MeTiS package. This option is invoked by setting ICNTL(6) equal to 4 or 5. Since the user must load the MeTiS library separately from HSL, a dummy routine has been included that will return the error INFO(1)=-18 if the MeTiS library is not available and ICNTL(6) is equal to 4. If the MeTiS library is not available and ICNTL(6) is equal to 5, then action is taken as if ICNTL(6) had been set to 2.

Static pivoting is now used so that the factorization can be performed using the same storage as predicted by the analysis even if the matrix is not positive definite. This is invoked by setting CNTL(4) to a nonzero value. A certain amount of additional storage can be allowed by setting CNTL(5) to a nonzero value.

Further testing on large problems has determined that a better value for ICNTL(12) is 16 so that this value has now been set as the default for this parameter.

An additional option has been added to the choice of ordering strategies. If ICNTL(6) is set to 5 (now the default), then ME57A/AD will choose automatically between using MeTiS or MC47; first based on matrix order and density and then, if necessary, by running both ordering strategies. The actual ordering used is returned in INFO(36).

An additional option has been added (ICNTL(16)) to remove small entries from the factorization and place corresponding pivots at the end of the factorization. This can result in far more efficient code (time and memory) if the matrix is highly rank deficient.

4 METHOD

Technical Report RAL-TR-2002-024 “MA57 – a new code for the solution of sparse symmetric indefinite systems” by Duff (published as ACM Trans. Math. Software **30** (2004), 118-144) describes the code and illustrates its performance. Some of the features that are new to Version 2.0.0 are described in the report RAL-TR-2004-020 “Strategies for scaling and pivoting for sparse symmetric indefinite problems” by Duff and Pralet (published as SIAM Journal Matrix Analysis and Applications **27** (2005), 313-340). RAL reports can be obtained from the web site

http://www.numerical.rl.ac.uk/reports/reports.html

A version of sparse Gaussian elimination is used.

The ME57 package supersedes ME27 for solving complex symmetric systems. In addition to being more efficient largely through its use of the Level 3 BLAS, it has many added features. Among these are: a fast mapping of data prior to a numerical factorization, the ability to modify pivots if the matrix is not definite, the efficient solution of several right-hand sides, a routine implementing iterative refinement, and the possibility of restarting the factorization should it run out of space. There is also an option that is much more efficient if the matrix is highly rank deficient.

The ME57A/AD entry (with ICNTL(6)≠1) chooses a pivot ordering based on either nested dissection or minimum degree using a generalized element model of the elimination to avoid storing the filled-in pattern explicitly. The elimination is represented as an assembly tree with the order of elimination determined by a depth-first search of the tree.

The ME57B/BD entry factorizes the matrix by using the assembly and elimination ordering generated by ME57A/AD. By default, the input matrix is first scaled using a symmetrized version of the HSL code MC64. At each stage in the multifrontal approach pivoting and elimination are performed on full submatrices and, when diagonal 1×1 pivots would be numerically unstable, 2×2 diagonal blocks are used. Thus ME57B/BD can be used to factor indefinite systems and will perform well on machines with caches.

If CNTL(4) is set to a value greater than zero, static pivoting is invoked (RAL-TR-2005-007 “Towards a stable static pivoting strategy for the sequential and parallel solution of sparse symmetric indefinite systems” by Duff and Pralet, to appear in SIAM J Matrix Analysis and Applications). This means that if, at any stage of the multifrontal elimination, there are fully summed rows and columns from which it is not possible to choose pivots because of the threshold criterion defined by CNTL(1), then we first try to get pivots using a weaker threshold by successively trying values one tenth of the previous until a threshold of √(CNTL(1)*CNTL(4)) is reached. If there are still fully summed rows and columns left, then the diagonal entries are replaced by CNTL(4) times the largest modulus of an entry in the scaled matrix and are used as 1×1 pivots in the factorization. If, furthermore, CNTL(5) is greater than zero, then CNTL(4) is treated as 0 (uneliminated variables are delayed) until CNTL(5)*N fully-summed variables have been delayed.

The ME57C/CD entry uses the factors from ME57B/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, depending on the value of ICNTL(13) and the size of the frontal matrix.

The ME57D/DD entry uses ME57C/CD to solve the system using either one step of iterative refinement or iterating until a measure of the backward error is reduced sufficiently according to the work of Arioli, Demmel, and Duff.

The ME57E/ED entry just copies the contents of arrays and to larger arrays and resets some pointers so that the factorization can continue using these larger arrays.

This multifrontal technique is described in detail by Duff and Reid (AERE-R.10533, 1982) and Duff and Reid, ACM Trans. Math. Software **9** (1983), 302-325.

5 EXAMPLE OF USE

We illustrate the use of the package on two sets of equations. The first is for a complex symmetric coefficient matrix viz.

$$\begin{pmatrix} 2+i & 3-i & & & \\ 3-i & 0 & 4 & & 6i \\ & 4 & i & 5+2i & \\ & & 5+2i & 0 & \\ & 6i & & & 1-3i \end{pmatrix} \mathbf{x} = \begin{pmatrix} 13+14i \\ -35+83i \\ 25+75i \\ 13+40i \\ 15+1i \end{pmatrix}$$

and the second for a complex Hermitian coefficient matrix viz.

$$\begin{pmatrix} 2 & 3-i & & & \\ 3+i & 0 & 4 & & 6i \\ & 4 & 1 & 5+2i & \\ & & 5-2i & 0 & \\ & -6i & & & -3 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 15+13i \\ -39+85i \\ 36+76i \\ 37+20i \\ -3-48i \end{pmatrix}$$

In both cases we print the solution. Note that these examples do not illustrate all the facilities.

Program

```
C Simple example of use of ME57 package
      INTEGER LRHS,LFACT,LKEEP,LIFACT,LWORK
      PARAMETER (LKEEP=100,LRHS=10,LWORK=12,LFACT=1000,LIFACT=500)
      INTEGER IRN(10),JCN(10),IWORK(40),KEEP(LKEEP),IFACT(LIFACT)
      INTEGER ICNTL(40),INFO(40),N,NE,I,JOB,KIND,IRUN
      COMPLEX*16 A(30),WORK(LWORK),FACT(LFACT),RHS(LRHS,1)
      DOUBLE PRECISION CNTL(5),RINFO(20),RWORK(LFACT)
      EXTERNAL ME57ID,ME57AD,ME57BD,ME57CD

C Set default values for control parameters.
      CALL ME57ID(CNTL,ICNTL)
      ICNTL(3) = 6

      DO 10 IRUN = 1,2

C Matrix is complex symmetric (KIND = 1)
C Matrix is Hermitian (KIND = 2)
        KIND = IRUN

C Read matrix and right-hand side
        READ (5,'(2I2)') N,NE
        READ (5,'(2I2,2F5.1)') (IRN(I),JCN(I),A(I),I=1,NE)
        READ (5,'(10F6.1)') (RHS(I,1),I=1,N)

C Analyse sparsity pattern
```

```
      CALL ME57AD(KIND,N,NE,IRN,JCN,LKEEP,KEEP,IWORK,ICNTL,INFO,RINFO)

C Factorize matrix
        CALL ME57BD(KIND,N,NE,A,FACT,LFACT,IFACT,LIFACT,LKEEP,KEEP,
     +              IWORK,RWORK,ICNTL,CNTL,INFO,RINFO)

C Solve the equations
        JOB = 1
        CALL ME57CD(KIND,JOB,N,FACT,LFACT,IFACT,LIFACT,1,RHS,LRHS,
     +              WORK,LWORK,IWORK,ICNTL,INFO)

C     Print the solution
        WRITE(6,'(//A,I1,A/(6F13.5))') 'Solution ',IRUN,' is ',
     +                          (RHS(I,1),I=1,N)


   10 CONTINUE

      END
```

Data

```
 5 7
 1 1  2.0  1.0
 1 2  3.0 -1.0
 2 3  4.0  0.0
 2 5  0.0  6.0
 3 3  0.0  1.0
 3 4  5.0  2.0
 5 5  1.0 -3.0
  13.0  14.0 -35.0  83.0  25.0  75.0  13.0  40.0  15.0   1.0
 5 7
 1 1  2.0  0.0
 1 2  3.0 -1.0
 2 3  4.0  0.0
 2 5  0.0  6.0
 3 3  1.0  0.0
 3 4  5.0  2.0
 5 5 -3.0  0.0
  15.0  13.0 -39.0  85.0  36.0  76.0  37.0  20.0  -3.0 -48.0
```

Output

```
 Solution 1 is
      1.00000      2.00000      3.00000      4.00000      5.00000      6.00000
      7.00000      8.00000      9.00000     10.00000


 Solution 2 is
      1.00000      2.00000      3.00000      4.00000      5.00000      6.00000
      7.00000      8.00000      9.00000     10.00000
```