1 SUMMARY

These subroutines are for the solution of a general sparse $m \times n$ system of complex linear equations (the most usual case being square, $m = n$).

(a) ME50I/ID provides default values for two arrays that contain parameters to control the execution of the other subroutines.

(b) ME50A/AD is given a matrix **A** and finds permutations **P** and **Q** suitable for the triangular factorization **PAQ** = **LU**, where **L** is lower triangular and **U** is upper triangular. It aims to preserve sparsity and control numerical stability. Packed storage is used until a density threshold is reached, from which point full storage is used. There is an option for dropping small entries from the factorization. It may also be used for given permutations to find the switch point and estimate storage requirements.

(c) ME50B/BD performs the triangular factorization of the permutation **PAQ** of a given matrix **A**. The permutation matrices **P** and **Q** must be supplied. The matrix **P** is altered if necessary for numerical stability, but the matrix **Q** is not altered. An option exists for subsequent calls for matrices with the same sparsity pattern to be made faster on the assumption that the altered **P** is still suitable. Subsequent calls may also be made faster if certain columns of **A** are unchanged.

(d) ME50C/CD uses the factorization produced by ME50B/BD to solve the equation $\mathbf{Ax} = \mathbf{b}$ or $\mathbf{A}^T\mathbf{x} = \mathbf{b}$ or $\mathbf{A}^H\mathbf{x} = \mathbf{b}$ ($\mathbf{A}^H$ is the conjugate transpose of the matrix **A**).

If the user requires a more convenient data interface, the ME48 package should be used. The ME48 subroutines call the ME50 subroutines after checking and sorting the user's input data and optionally using MC21 and MC13 to permute the matrix to block triangular form.

ATTRIBUTES — **Version:** 1.1.0. (23 January 2023) **Types:** Real (single, double). **Remark:** This is normally called through the ME48 package. It supersedes ME30. **Original date:** May 1993. **Origin:** I. S. Duff and J. K. Reid, Rutherford Appleton Laboratory. **Calls:** _AXPY, _DOTU, _DOTC, _GEMM, _GEMV, _SCAL, _SWAP, _TRSM, _TRSV.

2 HOW TO USE THE PACKAGE

2.1 Argument lists

We describe the argument list for each of the four subroutines in turn. If they are being called in the same program, a parameter of the same name in different subroutines could be represented by the same array or variable.

2.1.1 Argument list for ME50I/ID

ME50I/ID sets default values for the elements of two arrays that control the execution of the other subroutines.

The single precision version

```
CALL ME50I(CNTL,ICNTL)
```

The double precision version

```
CALL ME50ID(CNTL,ICNTL)
```

CNTL is a REAL (DOUBLE PRECISION in the D version) array of length 4 that need not be set by the user. On exit from ME50I/ID, it contains default values (see Section 2.2 for details).

ICNTL is an INTEGER array of length 7 that need not be set by the user. On exit from ME50I/ID, it contains default values (see Section 2.2 for details).

2.1.2 Argument list for ME50A/AD

The single precision version

```
      CALL ME50A(M,N,NE,LA,A,IRN,JCN,IQ,CNTL,ICNTL,IP,NP,JFIRST,
     *           LENR,LASTR,NEXTR,IW,IFIRST,LENC,LASTC,NEXTC,INFO,RINFO)
```

The double precision version

```
      CALL ME50AD(M,N,NE,LA,A,IRN,JCN,IQ,CNTL,ICNTL,IP,NP,JFIRST,
     *           LENR,LASTR,NEXTR,IW,IFIRST,LENC,LASTC,NEXTC,INFO,RINFO)
```

`M` is an `INTEGER` variable that must be set by the user to the number of rows. `M` is not altered by `ME50A/AD`. **Restriction:** `M` > 0.

`N` is an `INTEGER` variable that must be set by the user to the number of columns. `N` is not altered by `ME50A/AD`. **Restriction:** `N` > 0.

`NE` is an `INTEGER` variable that must be set by the user to the number of entries in the matrix. `NE` is not altered by `ME50A/AD`. **Restriction:** `NE` > 0.

`LA` is an `INTEGER` variable that must be set by the user to the length of arrays `A`, `IRN`, and `JCN`. Usually a sufficient value is 2*`NE` although this value is very problem dependent. On return, guidance on the choice of `LA` is provided by `INFO(2)` and `INFO(3)` (see Section 2.2). `LA` is not altered by `ME50A/AD`. **Restriction:** `LA` ≥ `NE`.

`A` is a `COMPLEX` (`COMPLEX*16` in the D version) array of length `LA` whose entries 1 to `NE` must be set by the user to the values of the entries of the matrix. Those belonging to a single column must be contiguous, but the ordering within each column is unimportant. The entries of column j precede those of column j+1, j=1,..., n–1 and no wasted space is allowed between the columns. No column may have more than one entry with the same row index. This array is used as workspace by `ME50A/AD`.

`IRN` is an `INTEGER` array of length `LA`. `IRN(k)` must be set by the user to hold the row index of the entry stored in `A(k)`, `k=1,NE`. This array is used as workspace by `ME50A/AD`.

`JCN` is an `INTEGER` array of length `LA` that need not be set by the user and is used for workspace.

`IQ` is an `INTEGER` array of length `N`. `IQ(j)` must be set by the user to hold the position in `A` and `IRN` of the first entry of column `j`, `j=1,N`. On output from `ME50A/AD`, `IQ(j)` indicates the column of **A** that corresponds to column `j` of **PAQ**.

`CNTL` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length 4 that controls the execution of the subroutine and must be set by the user. Default values for the elements may be set by a call to `ME50I/ID.` Details of the effects are given in Section 2.2. This array is not altered by `ME50A/AD`.

`ICNTL` is an `INTEGER` array of length 7 that controls the execution of the subroutine and must be set by the user. Default values for the elements may be set by a call to `ME50I/ID.` Details of the effects are given in Section 2.2. This array is not altered by `ME50A/AD`.

`IP` is an `INTEGER` array of length `M` that need be set only by the user who is providing permutations (see `ICNTL(7)` in Section 2.2). On output from `ME50A/AD`, `IP(i)` indicates the row of **PAQ** that corresponds to row `i` of **A**.

`NP` is an `INTEGER` variable that need not be set by the user. On output from `ME50A/AD`, it holds the number of columns processed in packed storage using sparsity techniques.

`JFIRST,LENR,LASTR,NEXTR,IW` are all `INTEGER` arrays of length `M` that are used as workspace by `ME50A/AD`. Only if `ICNTL(4)` is set to the value 0 are the arrays `LASTR` and `NEXTR` referenced by `ME50A/AD`.

`IFIRST` is an `INTEGER` array of length `N` that need be set only by the user who is providing permutations (see `ICNTL(7)` in Section 2.2). In this case, it is not altered. If `ICNTL(4)` has the value 0, it is used as a workarray.

`LENC,LASTC,NEXTC` are all `INTEGER` arrays of length `N` that are used as workspace by `ME50A/AD`.

`INFO` is an `INTEGER` array of length 7 that need not be set by the user. On exit from `ME50A/AD`, it contains

information about the execution of the subroutine. After a normal successful call, INFO(1) will have value zero. Negative values of INFO(1) indicate errors and positive values indicate warnings (see Section 2.3.1). Details of the information in the other elements of INFO are given in Section 2.2.

RINFO is a REAL (DOUBLE PRECISION in the D version) variable that need not be set on entry. On exit, it holds a prediction for the number of floating-point operations needed for the factorization. It may be inaccurate if any entries are dropped by this subroutine or by ME50B/BD.

2.1.3 Argument list for ME50B/BD

The single precision version

```
      CALL ME50B(M,N,NE,JOB,AA,IRNA,IPTRA,CNTL,ICNTL,IP,IQ,NP,
     *           LFACT,FACT,IRNF,IPTRL,IPTRU,W,IW2,INFO,RINFO)
```

The double precision version

```
      CALL ME50BD(M,N,NE,JOB,AA,IRNA,IPTRA,CNTL,ICNTL,IP,IQ,NP,
     *            LFACT,FACT,IRNF,IPTRL,IPTRU,W,IW2,INFO,RINFO)
```

M is an INTEGER variable that must be set by the user to the number of rows. M is not altered by ME50B/BD. **Restriction:** M > 0.

N is an INTEGER variable that must be set by the user to the number of columns. N is not altered by ME50B/BD. **Restriction:** N > 0.

NE is an INTEGER variable that must be set by the user to the number of entries in the matrix. NE is not altered by ME50B/BD. **Restriction:** NE > 0.

JOB is an INTEGER variable that must be set by the user to one of the values

1 First call.

2 Fast call for the case where only the numerical values in AA have changed since a previous call to ME50B/BD. No numerical pivoting is performed, which may be numerically unstable if the matrix entries are markedly different from those of the earlier call (see Section 4 for further explanation). This call is not available when any entries have been dropped (see CNTL(3) and CNTL(4) in Section 2.2).

3 Intermediate call for the case where only some of the columns of AA have changed since a previous call to ME50B/BD (see description of ICNTL(6) in Section 2.2). Pivoting is performed and the altered columns may have a changed structure as well as changed values. This call is not available when any entries have been dropped.

JOB is not altered by ME50B/BD. **Restriction:** $1 \leq$ JOB ≤ 3.

AA is a COMPLEX (COMPLEX*16 in the D version) array of length NE whose entries must be set by the user to the values of the entries of the matrix. Those belonging to a single column must be contiguous, but the ordering within each column is unimportant. The entries of column j precede those of column j+1, j=1,..., n–1 and no wasted space is allowed between the columns. No column may have more than one entry with the same row index. AA is not altered by ME50B/BD.

IRNA is an INTEGER array of length NE. IRNA(k) must be set by the user to hold the row index of the entry stored in AA(k), k=1,NE. IRNA is not altered by ME50B/BD.

IPTRA is an INTEGER array of length N. IPTRA(j) must be set by the user to hold the position in AA and IRNA of the first entry of column j, j=1,N. IPTRA is not altered by ME50B/BD.

CNTL is a REAL (DOUBLE PRECISION in the D version) array of length 4 that controls the execution of the subroutine and must be set by the user. Default values for the elements may be set by a call to ME50I/ID. Details of the effects are given in Section 2.2. This array is not altered by ME50B/BD.

ICNTL is an INTEGER array of length 7 that controls the execution of the subroutine and must be set by the user.

Default values for the elements may be set by a call to ME50I/ID. Details of the effects are given in Section 2.2. This array is not altered by ME50B/BD.

IP is an INTEGER array of length M. On a JOB=1 call, IP(i) must be set by the user so that IP(i) < IP(j) if row i is recommended to precede row j in the pivot sequence. If JOB=2 or JOB=3, it need not be set. If JOB=1 or JOB=3, IP(i) is used for workspace and is eventually reset to the format required to recommend the new pivot sequence to a subsequent JOB=1 call. If JOB=2, IP is not referenced.

IQ is an INTEGER array. On a JOB=1 call, it must be set by the user to hold the permutation **Q**. Either it must have length N and IQ(j) indicates the column of **A** that corresponds to column j of **PAQ**, j=1,N, or be of length 1 with IQ(1)=0 to indicate the identity permutation. On a JOB≥2 call, IQ must be unchanged since the previous call to ME50B/BD. IQ is not altered by ME50B/BD.

NP is an INTEGER variable that must set by the user to the number of columns to be processed in packed storage. NP is not altered by ME50B/BD. **Restriction:** $0 \leq$ NP $\leq$ N.

LFACT is an INTEGER variable that must be set by the user to the length of arrays FACT and IRNF. A suitable value will have been given by INFO(4) (see Section 2.2) on exit from ME50A/AD. LFACT is not altered by ME50B/BD. **Restriction:** LFACT ≥ NE+2.

FACT is a COMPLEX (COMPLEX*16 in the D version) array of length LFACT that need not be set on a JOB=1 call. If JOB ≥ 2, FACT must be unchanged since the previous call to ME50B/BD. On output, FACT(1) holds the drop tolerance provided in CNTL(3), FACT(2) holds the pivot tolerance provided in CNTL(4), positions 3 to IPTRL(N) hold the entries of the packed part of the factorization, and the full part of the factorization is held immediately afterwards.

IRNF is an INTEGER array of length LFACT that need not be set on a JOB=1 call. If JOB ≥ 2, IRNF must be unchanged since the previous call to ME50B/BD. On output, IRNF(1) holds the number of entries dropped; IRNF(2) holds the number of rows in full storage; for k=3,IPTRL(N), IRNF(k) holds the row index of the entry of the factorization that is stored in A(k); the row indices of the full part of the factorization are held from position IPTRL(N)+1; and the row indices are immediately followed by a vector of size N-NP containing pivoting information from the full part of the factorization.

IPTRL,IPTRU are INTEGER arrays of length N that need not be set on a JOB=1 call. If JOB≥2, they must be unchanged since the previous call to ME50B/BD.

W is a COMPLEX (COMPLEX*16 in the D version) array of length M that is used as workspace by ME50B/BD.

IW2 is an INTEGER array of length M+2*N that is used as workspace by ME50B/BD.

INFO is an INTEGER array of length 7 that need not be set by the user. On exit from ME50B/BD, it contains information on the execution of the subroutine. After a normal successful call, INFO(1) will have a non-negative value. Negative values of INFO(1) indicate error conditions while positive values indicate that the matrix has been successfully factorized but is rank deficient (see Section 2.3.2). Details of the information in the other elements of INFO are given in Section 2.2.

RINFO is a REAL (DOUBLE PRECISION in the D version) variable that need not be set on entry. On exit, it holds the number of floating-point operations needed for the factorization.

2.1.4 Argument list for ME50C/CD

ME50C/CD uses the factorization produced by ME50B/BD to solve the system **Ax** = **b** or the system $\mathbf{A}^T\mathbf{x} = \mathbf{b}$ or the system $\mathbf{A}^H\mathbf{x} = \mathbf{b}$.

The single precision version

```
      CALL ME50C(M,N,ICNTL,IQ,NP,KIND,LFACT,FACT,IRNF,IPTRL,IPTRU,
     *           B,X,W,INFO)
```

The double precision version

```
CALL ME50CD(M,N,ICNTL,IQ,NP,KIND,LFACT,FACT,IRNF,IPTRL,IPTRU,
*           B,X,W,INFO)
```

`M` is an `INTEGER` variable that must be set by the user to the number of rows. `M` is not altered by `ME50C/CD`. **Restriction:** `M` > 0.

`N` is an `INTEGER` variable that must be set by the user to the number of columns. `N` is not altered by `ME50C/CD`. **Restriction:** `N` > 0.

`ICNTL` is an `INTEGER` array of length 7 that controls the execution of the subroutine and must be set by the user. Default values for the elements may be set by a call to `ME50I/ID`. Details of the effects are given in Section 2.2. This array is not altered by `ME50C/CD`.

`IQ` is an `INTEGER` array that must be unchanged since the previous call to `ME50B/BD`. `IQ` is not altered by `ME50C/CD`.

`NP` is an `INTEGER` variable that must be unchanged since the previous call to `ME50B/BD`. `NP` is not altered by `ME50C/CD`.

`KIND` is an `INTEGER` variable that must be set to `1` if $\mathbf{Ax} = \mathbf{b}$ is to be solved, to `2` if $\mathbf{A}^T\mathbf{x} = \mathbf{b}$ is to be solved, and to `3` if $\mathbf{A}^H\mathbf{x} = \mathbf{b}$ is to be solved. `KIND` is not altered by `ME50C/CD`.

`LFACT` is an `INTEGER` variable that must be unchanged since the previous call to `ME50B/BD`. `LFACT` is not altered by `ME50C/CD`.

`FACT` is a `COMPLEX` (`COMPLEX*16` in the D version) array that must be unchanged since the previous call to `ME50B/BD`. `FACT` is not altered by `ME50C/CD`.

`IRNF` is an `INTEGER` array that must be unchanged since the previous call to `ME50B/BD`. `IRNF` is not altered by `ME50C/CD`.

`IPTRL,IPTRU` are `INTEGER` arrays of length `N` that must be unchanged since the previous call to `ME50B/BD`. They are not altered by `ME50C/CD`.

`B` is a `COMPLEX` (`COMPLEX*16` in the D version) array that must be set by the user to the value of the right-hand side vector **b**. Its size is `M` if $\mathbf{Ax} = \mathbf{b}$ is to be solved and is `N` if $\mathbf{A}^T\mathbf{x} = \mathbf{b}$ is to be solved. `B` is not altered by `ME50C/CD`.

`X` is a `COMPLEX` (`COMPLEX*16` in the D version) array of size `N` if $\mathbf{Ax} = \mathbf{b}$ is to be solved and of size `M` if $\mathbf{A}^T\mathbf{x} = \mathbf{b}$ is to be solved. It need not be set on entry. On exit from `ME50C/CD`, `X` will hold the computed solution **x**.

`W` is a `COMPLEX` (`COMPLEX*16` in the D version) array of length max(`M`,`N`) that is used as workspace by `ME50C/CD`.

`INFO` is an `INTEGER` array of length 7 that need not be set by the user. On exit from `ME50C/CD, INFO(1)` has the value zero if the call was successful and a negative value in the event of an error (see Section 2.3.3). The other elements of `INFO` are not altered.

2.2 Arrays for control and information

The elements of the array `CNTL` control the execution of `ME50A/AD` and `ME50B/BD`, and the elements of the array `ICNTL` control the execution of `ME50A/AD`, `ME50B/BD`, and `ME50C/CD`. `CNTL` and `ICNTL` are not altered by any of these subroutines. Default values for their elements are set by `ME50I/ID`. The elements of the array `INFO` provide information on the execution of `ME50A/AD`, `ME50B/BD`, and `ME50C/CD`. `INFO` need not be set by the user.

`CNTL(1)` has default value 0.5. `ME50A/AD` switches to full matrix processing if the ratio of number of entries in the reduced matrix to the number that it would have as a full matrix is `CNTL(1)` or more. A value greater than 1.0 is treated as 1.0.

`CNTL(2)` has default value 0.1 and determines the balance in `ME50A/AD` and `ME50B/BD` between pivoting for sparsity and for stability, values near zero emphasizing sparsity and values near one emphasizing stability. If `CNTL(2)`

< 0.0, it is regarded as having the value 0.0; if CNTL(2) > 1.0, it is regarded as having the value 1.0.

CNTL(3) has default value zero. If it is set to a positive value, ME50A/AD or ME50B/BD will drop from the factors any entry whose modulus is less than CNTL(3). The factorization will then require less storage but will be inaccurate.

CNTL(4) has default value zero. If ME50A/AD or ME50B/BD finds a column of the reduced matrix with entries all of modulus less than or equal to CNTL(4), all such entries are dropped from the factorization (and contribute to the count in INFO(6)). They also require every pivot to have absolute value greater than CNTL(4).

ICNTL(1) has default value 6 and holds the unit number to which the error messages are sent. A non-positive value suppresses all messages.

ICNTL(2) has default value 6 and holds the unit number to which diagnostic printing is sent. A non-positive value suppresses all such printing.

ICNTL(3) is used by the subroutines to control printing. It has default value 1. Possible values are:

- ≤0 No printing.
- 1 Error messages only.
- 2 Error messages and warnings only.
- 3 Also scalar parameters and a few entries of array parameters on entry and exit from each subroutine.
- 4 Also all parameters on entry and exit from each subroutine.

ICNTL(4) has default value 3. If ICNTL(4) has a positive value, each pivot search in ME50A/AD is limited to a maximum of ICNTL(4) columns and the workspace arrays IFIRST, LASTR, and NEXTR (see Section 2.1.1) are not referenced by ME50A/AD. If ICNTL(4) is set to the value 0, a special search technique is used to find the best pivot and the workspace arrays are needed. This is usually only a little slower, but can occasionally be very slow. It may result in reduced fill-in.

ICNTL(5) is used by ME50B/BD and ME50C/CD to control the full-matrix processing. It has default value 32. Possible values are:

- 0 Level 1 BLAS used.
- 1 Level 2 BLAS used.
- ≥2 Level 3 BLAS used by ME50B/BD, with block column size ICNTL(5). Level 2 BLAS used by ME50C/CD.

ICNTL(6) has default value 0 and any value outside the range N > ICNTL(6) > 0 is treated as 0. The last ICNTL(6) columns of **A** will be chosen by ME50A/AD as the last ICNTL(6) columns of **PAQ**. On a JOB = 1 call to ME50B/BD, ICNTL(6) is ignored. On a JOB = 2 or JOB = 3 call to ME50B/BD with ICNTL(6) outside the range N > ICNTL(6) > 0, any entry of **A** may have changed, but if ICNTL(6) is within this range, only the last ICNTL(6) columns may have changed since the corresponding JOB = 1 call. This allows work to be limited to these columns. If JOB = 2, the structure must be unchanged because no numerical pivoting is performed. If JOB = 3, numerical pivoting is performed, which means that it is permissible for the structure to have changed.

ICNTL(7) has default value 0. Other possible values are

- 1 ME50A/AD chooses pivots only from the main diagonal while processing the packed columns. This may lead to more fill-in, and a premature switch to full processing (with a diagnostic warning) is made if no suitable diagonal entries are available. The subsequent ME50B/BD call will use these diagonal entries unless they fail the stability test, in which case other pivots are used without any diagnostic warning.
- 2 ME50A/AD uses a column permutation provided by the user and a recommendation for the row permutation. ICNTL(4) must have the value 1, IFIRST must be set so that IFIRST(i) is the column in

position i of the permuted matrix and IP must be set so that IP(i) < IP(j) if row i is recommended to precede row j in the pivot sequence.

INFO(1) has a non-negative value if the call was successful and a negative value in the event of an error (see Section 2.3).

INFO(2) is used by ME50A/AD to monitor the adequacy of the allocated space in arrays A, IRN, and JCN by counting the total number of garbage collections performed on these arrays. If INFO(2) is fairly large (say greater than 10), it may be advantageous to increase the size of the arrays. INFO(2) is initialized to zero on entry to ME50A/AD and is incremented each time the garbage collection routine ME50D/DD is called.

INFO(3) indicates, after a successful call to ME50A/AD, the minimum value to which LA could be reduced while still permitting a successful call for the given matrix. If, however, the user were to decrease LA to that size, the number of compresses (INFO(2)) may be very high. In the event of failure, INFO(3) gives a recommended value for LA.

INFO(4) indicates, after a successful call to ME50B/BD, the minimum size of LFACT required to enable a successful decomposition by ME50B/BD assuming the same pivot sequence and set of dropped entries can be used. After a successful call to ME50A/AD, INFO(4) approximates this quantity with an overestimate. If ME50B/BD fails because LFACT is too small, INFO(4) gives a recommended value for LFACT.

INFO(5) is used by ME50A/AD to give an estimate of the rank of the matrix. It may be an overestimate since no processing is performed on the full block. INFO(5) is used by ME50B/BD to give the rank that it has computed. On an exit from either subroutine with INFO(1) equal to 0, INFO(5) will be equal to min(M,N).

INFO(6) holds, on exit from ME50A/AD or ME50B/BD, the number of entries dropped from the data structure.

INFO(7) holds, on exit from ME50A/AD or ME50B/BD, the number of rows processed in full storage.

2.3 Error diagnostics

2.3.1 Error diagnostics for ME50A/AD

If the subroutine encounters no complications or errors, the value of INFO(1) will be zero on exit from ME50A/AD. Negative values indicate errors and positive values indicate warnings. Error messages are output on stream ICNTL(1). Possible negative values for INFO(1) are as follows:

–1 M or N has a value less than 1.

–2 NE has a value less than 1.

–3 LA is too small. INFO(3) gives a recommended value.

–4 There are duplicated entries.

–5 Faulty column permutation in IFIRST when ICNTL(7)=2.

–6 ICNTL(4)≠1 when ICNTL(7)=2.

Positive values are accompanied by output on stream ICNTL(2) and are as follows:

+1 The matrix is rank deficient with rank no bigger than the value of INFO(5).

+2 Premature switch to full processing because of failure to find a stable diagonal pivot (ICNTL(7)=1 case only).

+3 Combination of warnings 1 and 2.

2.3.2 Error diagnostics for ME50B/BD

If ME50B/BD performs a successful decomposition, INFO(1) will have a non-negative value on exit. Negative values indicate errors and messages are output on stream ICNTL(1). Possible negative values for INFO(1) are as follows:

–1 M or N has a value less than 1.

–2 NE has a value less than 1.

–3 LFACT is too small. INFO(4) gives a recommended value.

–4 There are duplicated entries.

–5 JOB has a value less than 1 or greater than 3.

–6 JOB has the value 2 or 3, but some entries were dropped in the corresponding JOB=1 call.

–7 NP is less than 0 or greater than N.

–(7+J) The routine has terminated during a call with JOB=2 because a small pivot was found in column J of the permuted matrix.

Positive values are accompanied by output on stream ICNTL(2) and are as follows:

+1 The matrix is rank deficient with estimated rank INFO(5).

2.3.3 Error diagnostics for ME50C/CD

If ME50C/CD performs a successful solution, INFO(1) will have a zero value on exit. Negative values indicate errors and messages are output on stream ICNTL(1). Possible negative values are as follows:

–1 M or N has a value less than 1.

2.4 Singular or rectangular matrices

2.4.1 ME50A/AD

ME50A/AD can treat a matrix that is singular or rectangular. Since no processing is performed on the full block, the rank may be overestimated.

2.4.2 ME50B/BD

For a call with JOB=1 or JOB=3, ME50B/BD will factorize a singular or rectangular matrix.

On a call with JOB=2, ME50B/BD will factorise a singular or rectangular matrix provided that the previous JOB=1 matrix was successfully factorized and the same singularities are found.

2.4.3 ME50C/CD

If ME50B/BD has been used to decompose a singular or rectangular matrix, ME50C/CD will still perform the solution process. Components corresponding to a zero pivot are set to zero.

3 GENERAL INFORMATION

Use of common: None.

Restrictions: $M > 0$, $N > 0$, $NE > 0$, $LA \geq NE$, $1 \leq JOB \leq 3$, $0 \leq NP \leq N$.

Other routines called directly: ME50 calls ME50D/DD, ME50E/ED, ME50F/FD, ME50G/GD, and ME50H/HD, none of which need ever be called directly by the user. ME50B/BD and ME50C/CD call the BLAS routines _AXPY, _DOTU, _DOTC, _GEMM, _GEMV, _SCAL, _SWAP, _TRSM, and _TRSV.

Input/output: Error messages and diagnostic printing are output on Fortran streams ICNTL(1) and ICNTL(2).

4 METHOD

ME50A/AD uses a sparse variant of Gaussian elimination to compute a pivot ordering for the decomposition of **A** into its **LU** factors. It holds the reduced matrix by columns and its pattern by rows. It does not keep the factors themselves, which makes it less likely to fail through lack of storage. Each pivot a_{pj} is required to satisfy the stability test

$$|a_{pj}| \geq u \max_i |a_{ij}|$$

within the reduced matrix, where u is the threshold held in CNTL(2), with default value 0.1. For sparsity, the default strategy is that of Zlatev (*SIAM J. Numer. Anal.* **17**, 1990, 18-30). A small number of columns with fewest entries is searched for an entry with least Markowitz cost (product of the number of other entries in the row of the reduced matrix and the number of other entries in the column). Alternatively, the strategy of Markowitz (*Management Sci.* **3**, 1957, 255-269) is available. This minimizes the Markowitz cost over all entries.

ME50B/BD performs the actual factorization. It uses the technique of Gilbert and Peierls (*SIAM J. Sci. Stat. Comput.* **9**, 1988, 862-874) to allow row interchanges with a total computational workload proportional to the number of floating-point operations. The same stability test is used as in ME50A/AD. On some examples, this technique is so successful that it is little slower than the 'fast' (JOB = 2) call, which uses the pivotal sequence of the previous call and does not perform the stability test. For the full matrix processing, we have included options for the use of level 1, 2 or 3 BLAS. Which is the fastest will depend on the machine and the availability of optimized versions. Where speed is important, each should be tried.

ME50C/CD performs simple forward elimination and back-substitution operations using the factors from ME50B/BD. For the full matrix processing, we have included options for the use of level 1 or 2 BLAS. Where speed is important, each should be tried.

5 EXAMPLE OF USE

In the example code shown below, we read in the entries of a square sparse matrix, which is then sorted by columns. Note that the entries within each column may be in any order. We then find the permutations **P** and **Q**, decompose the matrix, and finally solve the equations.

```
C     .. Parameters ..
      INTEGER MAXN,MAXNE,LA,LFACT
      PARAMETER (MAXN=50,MAXNE=MAXN*MAXN,LA=500,LFACT=500)
C     ..
C     .. Local Scalars ..
      DOUBLE PRECISION RINFO
      INTEGER I,JDISP,JOB,KIND,N,NE,NP
C     ..
C     .. Local Arrays ..
      COMPLEX*16 A(LA),AA(MAXNE),B(MAXN),FACT(LFACT),W(MAXN),X(MAXN)
      DOUBLE PRECISION CNTL(4)
      INTEGER ICNTL(7),IFIRST(MAXN),INFO(7),IP(MAXN),IPTRA(MAXN),
     +        IPTRL(MAXN),IPTRU(MAXN),IQ(MAXN),IRN(LA),IRNA(MAXNE),
     +        IRNF(LFACT),IW(MAXN),IW2(2*MAXN),JCN(LA),JFIRST(MAXN),
     +        LASTC(MAXN),LASTR(MAXN),LENC(MAXN),LENR(MAXN),NEXTC(MAXN),
     +        NEXTR(MAXN),ICT59(10),INF59(10)
C     ..
C     .. External Subroutines ..
      EXTERNAL ME50AD,ME50BD,ME50CD,ME50ID,MC59AZ
C     ..
C Set CNTL and ICNTL
      CALL ME50ID(CNTL,ICNTL)
      READ (5,FMT=*) N,NE
C     Read in data in any order
      READ (5,FMT=*) (IRN(I),JCN(I),A(I),I=1,NE)
```

```
      JDISP = 0
C     Sort the data by columns
      ICT59(1) = 0
      ICT59(2) = 0
      ICT59(3) = 0
      ICT59(4) = 0
      ICT59(5) = 0
      ICT59(6) = 0
      CALL MC59AZ(ICT59,N,N,NE,IRN,NE,JCN,LA,A,MAXN,IQ,2*MAXN,IW2,
     +            INF59)
C     Preserve the data arrays
      DO 10 I = 1,NE
         IRNA(I) = IRN(I)
         AA(I) = A(I)
   10 CONTINUE
      DO 20 I = 1,N
         IPTRA(I) = IQ(I)
   20 CONTINUE
C Compute the pivot ordering.
      CALL ME50AD(N,N,NE,LA,A,IRN,JCN,IQ,CNTL,ICNTL,IP,NP,JFIRST,LENR,
     +            LASTR,NEXTR,IW,IFIRST,LENC,LASTC,NEXTC,INFO,RINFO)
      IF (ICNTL(1).LT.0) THEN
         WRITE (6,FMT=9030) ICNTL(1)
         STOP
      END IF
C Compute the factorization.
      JOB = 1
      CALL ME50BD(N,N,NE,JOB,AA,IRNA,IPTRA,CNTL,ICNTL,IP,IQ,NP,LFACT,
     +            FACT,IRNF,IPTRL,IPTRU,W,IW2,INFO,RINFO)
      IF (ICNTL(1).LT.0) THEN
         WRITE (6,FMT=9040) ICNTL(1)
         STOP
      END IF
C Read right-hand side vector.
      READ (5,FMT=*) (B(I),I=1,N)
      KIND = 1
C Solve  A x = b.
      CALL ME50CD(N,N,ICNTL,IQ,NP,KIND,LFACT,FACT,IRNF,IPTRL,IPTRU,B,X,
     +            W,INFO)
      IF (ICNTL(1).LT.0) THEN
         WRITE (6,FMT=9050) ICNTL(1)
         STOP
      END IF
      WRITE (6,FMT=9000)
      WRITE (6,FMT=9010) (I,X(I),I=1,N)
 9000 FORMAT (/,' The solution vector is:',/)
 9010 FORMAT ('  X(',I2,')= ',2D13.5)
 9030 FORMAT (' Error return from ME50AD with ICNTL(1) =',I2)
 9040 FORMAT (' Error return from ME50BD with ICNTL(1) =',I2)
 9050 FORMAT (' Error return from ME50CD with ICNTL(1) =',I2)
      END
```

If we wish to solve the following problem:

$$\begin{pmatrix} 1+2i & i & 4i & 4+2i \\ & 6+3i & 7 & 8+2i \\ 9+i & & 11+i & 12+2i \\ 2i & 14+5i & & 16+2i \end{pmatrix} \mathbf{x} = \begin{pmatrix} -4+14i \\ 16+26i \\ 28+36i \\ 21+39i \end{pmatrix}$$

we might have as input

```
        4      13
        4       4           (16.,2.)
        3       3           (11.,1.)
        2       2           (6. ,3.)
        1       1           (1. ,2.)
        4       2           (14.,5.)
        2       3           (7. ,0.)
        3       1           (9. ,1.)
        2       4           (8. ,2.)
        3       4           (12.,2.)
        1       4           (4. ,2.)
        1       2           (0. ,1.)
        1       3           (0. ,4.)
        4       1           (0. ,2.)
    (-4.,14)
    (16.,26)
    (28.,36)
    (21.,39)
```

and we would get the following output

```
   The solution vector is:

    X( 1)=   0.10000D+01  0.10000D+01
    X( 2)=   0.10000D+01  0.10000D+01
    X( 3)=   0.10000D+01  0.10000D+01
    X( 4)=   0.10000D+01  0.10000D+01
```