Warning: Subroutine ME05 performs functions which are adequately treated by routines in other standard subroutine libraries (for example, LAPACK). The use of this routine is not recommended, and it may be removed from future releases of this library.

1 SUMMARY

Given an n by n complex matrix $\mathbf{A}=\{a_{ij}\}_{n\times n}$ performs one or more of the following tasks.

(a) **solves** the system of **linear algebraic equations**

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \qquad i=1,2,...,n$$

given the right-hand sides b_i, i=1,2,...,n, and provides a re-entry facility for the rapid solution of further systems of equations which have the same elements a_{ij}. (a_{ij} , b_i and x_i are in general complex).

(b) computes the **inverse matrix** $\mathbf{A}^{-1}$ of **A**.

(c) computes the value of the **determinant** of **A**.

The subroutine will optionally perform iterative refinement in order to both improve the accuracy of the answer (solution or inverse) and provide error estimates based either on the precision of the computer or on user supplied accuracy information. An option for scaling the matrix is provided.

The method is basically Gaussian elimination with partial pivoting, implicit scaling and iterative refinement and applying small random perturbations in order to estimate errors, see S. Marlow and J.K. Reid, AERE R.6899.

ATTRIBUTES — **Version:** 1.0.0. **Types:** ME05A; ME05AD. **Calls:** FA01; FD05. **Language:** ME05AD uses COMPLEX*16. **Original date:** August 1971. **Origin:** S.Marlow and J.K.Reid, Harwell.

2 HOW TO USE THE PACKAGE

2.1 Entries and argument lists

The entry points

To solve systems of linear equations as in § 1(a).

The single precision version

```
CALL ME05A(A,IA,N,B,W,E,RW)
```

The double precision version

```
CALL ME05AD(A,IA,N,B,W,E,RW)
```

To find inverse of matrix **A** in § 1(b).

The single precision version

```
CALL ME05B(A,IA,N,W,E,RW)
```

The double precision version

```
CALL ME05BD(A,IA,N,W,E,RW)
```

To find determinant of the matrix **A** as in § 1(c).

The single precision version

```
CALL ME05C(A,IA,N,DET,IDET,W,RW)
```

The double precision version

```
CALL ME05CD(A,IA,N,DET,IDET,W,RW)
```

Provided arguments `A, IA, N, W` have not been altered since a previous entry to `ME05A/AD` a new system with the same coefficients a_{ij} but different right-hand side elements b_i may be solved without repeating any of the work involved in the factorisation.

N.B. Iterative refinement `E>0` must not be requested unless the previous `ME05A/AD` entry also requested iterative refinement.

The single precision version

```
CALL ME05D(A,IA,N,B,W,E)
```

The double precision version

```
CALL ME05DD(A,IA,N,B,W,E)
```

The arguments

`A` is an `COMPLEX` (`COMPLEX*16` in the D version) two-dimensional array which contains the coefficients a_{ij} on entry. It is overwritten by its triangular factorisation on `ME05A/AD, ME05C/CD` entries and its inverse on `ME05B/BD` entry.

`IA` is an `INTEGER` variable set by the user to the first dimension of the array `A` . It is not altered by the subroutine.

`N` is an `INTEGER` variable set to the number of equations (i.e. the order of the matrix **A**). It is not altered by the subroutine.

`B` is a `COMPLEX` (`COMPLEX*16` in the D version) array of length at least `N` which must be set to the numbers b_1,b_2,.....b_n and will contain the solution x_1,x_2,....x_n on exit.

`W` is a `COMPLEX` (`COMPLEX*16` in the D version) workspace array. The minimum length of this array is as follows.

(i) Iterative refinement required `N*(N+5)` (`ME05A/B/D, ME05AD/BD/DD).`

(ii) No iterative refinement, with scaling `5*N` (`ME05A/B/C/D, ME05AD/BD/CD/DD).`

(iii) No iterative refinement, no scaling `N` (`ME05A/C/D, ME05AD/CD/DD`) or `2*N` (`ME05B/BD).`

If iterative refinement is required then parts of `W` may return error estimates on the solutions (see § 2.2); also `W(4*N+I)` will be set to contain `B(I)` , `I=1,N` and `W(5*N+I), I=1,N*N` will be set to contain the matrix **A**. If scaling is in use then the subroutine sets `W(2*N+I), W(3*N+I), I=1,N` to the row and column scaling factors used.

`RW` is a `REAL` (`DOUBLE PRECISION` in the D version) two-dimensional array of dimension `(N,7)` which is used by the routine as workspace.

`E` is a `REAL` (`DOUBLE PRECISION` in the D version) variable which must be set positive if iterative refinement is required and non-positive otherwise. On exit it will have one of the following values.

–2 Error condition in which execution could not continue. Any results should be treated as rubbish (see § 2.3).

–1 Error condition found. Execution continued but results may be unreliable. (see § 2.3).

0 Successful entry without iterative refinement.

≥0 Successful entry with iterative refinement. It is set to an estimate of the size of the largest error in the solution (x or $\mathbf{A}^{-1}$)based on the assumption that all data was accurate to the full word length (see § 2.5 if this is not so).

DET, IDET are COMPLEX (COMPLEX*16 in the D version) and INTEGER variables respectively, and are set so that the determinant is held as DET*16**DET. Usually IDET=0 but other values will be used if necessary to avoid overflow or underflow in DET.

For additional facilities see § 2.5.

2.2 Error estimates with iterative refinement in use.

If the rows and columns of **A** have maximal elements which vary widely in size, then the single number E may grossly over-estimate the errors in some of the components. In this case the user should examine the numbers W(N+1), ... W(2*N) which give separate estimates for the errors in $x_1, x_2, ...x_n$ (ME05A/D) or the numbers W(J)*W(I+N) which give estimates for the errors in $\mathbf{A}_{ij}^{-1}$ (ME05B).

N.B. If scaling has been inhibited by the use of JSCALE (see § 2.5) then no estimates additional to that contained in E are available and these components of W are not set.

2.3 Error messages.

ME05A/AD If it is found that a row or column (of **A** or a pivot is small or zero then a diagnostic is printed (limited to one message per entry, later messages being suppressed). Execution is continued but iterative refinement is not attempted and E is set to –1. In such cases the solution obtained is likely to be unreliable.

ME05B If a row or column is found to be zero an error message is printed, E set to –2 and a return is made to the calling program. If a pivot is found to be small the procedure for ME05A/D is adopted.

ME05A/B/C/D If N ≤ 0 then E is set to –2 before returning to calling program.

2.4 Method.

Unless requested otherwise (see § 2.5) the rows and columns of the matrix are implicitly scaled so that the pivots used are more likely to lead to low growth of roundoff errors. Gaussian elimination with row interchanges is used to factorise the given matrix.

When iterative refinement is requested, this is implicitly performed on the scaled system and continued until the largest change (in x or $\mathbf{A}^{-1}$) is greater than half its value on the previous iteration or is as small as the word-length allows. During this refinement pseudo-rendom changes are made to the data in accord with its specified accuracy (see § 2.5) and the size of the last maximal change is taken as an accuracy estimate. Finally the column scaling factors are multiplied by this estimate to give individual error estimates (see § 2.5) and E is set to the largest of them.

2.5 Use of Comnon

The subroutine contains a common block called ME05E/ED of the form:

The single precision version

```
COMMON/ME05E/ LP,JSCALE,EA,EB
```

The double precision version

```
COMMON/ME05ED/ LP,JSCALE,EA,EB
```

LP is INTEGER variable set initially to 6. It specifies the Fortran unit number for messages.

JSCALE is INTEGER variable set initially to 1. This indicates that scaling is done. If no scaling is required then the value of JSCALE should be zero or negative. This option should only be used where speed is important.

EA is REAL (DOUBLE PRECISION in the D version) variable. If EA ≥ 0 then it gives the fractional accuracy in the elements a_{ij} of **A** . If EA < 0 then |EA| gives the absolute accuracy of the elements. This is defaulted to zero. EA=0 indicates exact data.

EB is REAL (DOUBLE PRECISION in the D version) variable. This is specified similarly to EA and applies to the numbers $b_1, b_2, ...b_n$.

3 GENERAL INFORMATION

Use of common: uses a common block called ME05E/ED, see §2.5.

Portability: ME05AD, ME05BD, ME05CD, ME05DD use COMPLEX*16 facility.

Workspace: passed as arguments, W see § 2.1.

Other routines called directly: calls FA01 and FD05.

Input/output: error messages, see LP in §2.5.