

# NAG Library Routine Document

## F04CEF

**Note:** before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

### 1 Purpose

F04CEF computes the solution to a complex system of linear equations  $AX = B$ , where  $A$  is an  $n$  by  $n$  Hermitian positive definite matrix, stored in packed format, and  $X$  and  $B$  are  $n$  by  $r$  matrices. An estimate of the condition number of  $A$  and an error bound for the computed solution are also returned.

### 2 Specification

```
SUBROUTINE F04CEF (UPLO, N, NRHS, AP, B, LDB, RCOND, ERBND, IFAIL)
INTEGER          N, NRHS, LDB, IFAIL
REAL (KIND=nag_wp) RCOND, ERBND
COMPLEX (KIND=nag_wp) AP(*), B(LDB,*)
CHARACTER(1)    UPLO
```

### 3 Description

The Cholesky factorization is used to factor  $A$  as  $A = U^H U$ , if  $UPLO = 'U'$ , or  $A = LL^H$ , if  $UPLO = 'L'$ , where  $U$  is an upper triangular matrix and  $L$  is a lower triangular matrix. The factored form of  $A$  is then used to solve the system of equations  $AX = B$ .

### 4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia <http://www.netlib.org/lapack/lug>

Higham N J (2002) *Accuracy and Stability of Numerical Algorithms* (2nd Edition) SIAM, Philadelphia

### 5 Parameters

- 1: UPLO – CHARACTER(1) *Input*  
*On entry:* if  $UPLO = 'U'$ , the upper triangle of the matrix  $A$  is stored.  
 If  $UPLO = 'L'$ , the lower triangle of the matrix  $A$  is stored.  
*Constraint:*  $UPLO = 'U'$  or  $'L'$ .
- 2: N – INTEGER *Input*  
*On entry:* the number of linear equations  $n$ , i.e., the order of the matrix  $A$ .  
*Constraint:*  $N \geq 0$ .
- 3: NRHS – INTEGER *Input*  
*On entry:* the number of right-hand sides  $r$ , i.e., the number of columns of the matrix  $B$ .  
*Constraint:*  $NRHS \geq 0$ .

- 4: AP(\*) – COMPLEX (KIND=nag\_wp) array Input/Output

**Note:** the dimension of the array AP must be at least  $\max(1, N \times (N + 1)/2)$ .

*On entry:* the  $n$  by  $n$  Hermitian matrix  $A$ . The upper or lower triangular part of the Hermitian matrix is packed column-wise in a linear array. The  $j$ th column of  $A$  is stored in the array AP as follows:

More precisely,

if UPLO = 'U', the upper triangle of  $A$  must be stored with element  $A_{ij}$  in  $AP(i + j(j - 1)/2)$  for  $i \leq j$ ;

if UPLO = 'L', the lower triangle of  $A$  must be stored with element  $A_{ij}$  in  $AP(i + (2n - j)(j - 1)/2)$  for  $i \geq j$ .

*On exit:* if IFAIL = 0 or  $N + 1$ , the factor  $U$  or  $L$  from the Cholesky factorization  $A = U^H U$  or  $A = LL^H$ , in the same storage format as  $A$ .

- 5: B(LDB, \*) – COMPLEX (KIND=nag\_wp) array Input/Output

**Note:** the second dimension of the array B must be at least  $\max(1, NRHS)$ .

*On entry:* the  $n$  by  $r$  matrix of right-hand sides  $B$ .

*On exit:* if IFAIL = 0 or  $N + 1$ , the  $n$  by  $r$  solution matrix  $X$ .

- 6: LDB – INTEGER Input

*On entry:* the first dimension of the array B as declared in the (sub)program from which F04CEF is called.

*Constraint:*  $LDB \geq \max(1, N)$ .

- 7: RCOND – REAL (KIND=nag\_wp) Output

*On exit:* if IFAIL = 0 or  $N + 1$ , an estimate of the reciprocal of the condition number of the matrix  $A$ , computed as  $RCOND = 1 / (\|A\|_1 \|A^{-1}\|_1)$ .

- 8: ERRBND – REAL (KIND=nag\_wp) Output

*On exit:* if IFAIL = 0 or  $N + 1$ , an estimate of the forward error bound for a computed solution  $\hat{x}$ , such that  $\|\hat{x} - x\|_1 / \|x\|_1 \leq ERRBND$ , where  $\hat{x}$  is a column of the computed solution returned in the array B and  $x$  is the corresponding column of the exact solution  $X$ . If RCOND is less than **machine precision**, then ERRBND is returned as unity.

- 9: IFAIL – INTEGER Input/Output

*On entry:* IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

*On exit:* IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

## 6 Error Indicators and Warnings

If on entry  $IFAIL = 0$  or  $-1$ , explanatory error messages are output on the current error message unit (as defined by  $X04AAF$ ).

Errors or warnings detected by the routine:

$IFAIL < 0$  and  $IFAIL \neq -999$

If  $IFAIL = -i$ , the  $i$ th argument had an illegal value.

$IFAIL > 0$  and  $IFAIL \leq N$

If  $IFAIL = i$ , the leading minor of order  $i$  of  $A$  is not positive definite. The factorization could not be completed, and the solution has not been computed.

$IFAIL = N + 1$

$RCOND$  is less than *machine precision*, so that the matrix  $A$  is numerically singular. A solution to the equations  $AX = B$  has nevertheless been computed.

$IFAIL = -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

$IFAIL = -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

$IFAIL = -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

## 7 Accuracy

The computed solution for a single right-hand side,  $\hat{x}$ , satisfies an equation of the form

$$(A + E)\hat{x} = b,$$

where

$$\|E\|_1 = O(\epsilon)\|A\|_1$$

and  $\epsilon$  is the *machine precision*. An approximate error bound for the computed solution is given by

$$\frac{\|\hat{x} - x\|_1}{\|x\|_1} \leq \kappa(A) \frac{\|E\|_1}{\|A\|_1},$$

where  $\kappa(A) = \|A^{-1}\|_1 \|A\|_1$ , the condition number of  $A$  with respect to the solution of the linear equations. F04CEF uses the approximation  $\|E\|_1 = \epsilon \|A\|_1$  to estimate  $ERRBND$ . See Section 4.4 of Anderson *et al.* (1999) for further details.

## 8 Parallelism and Performance

F04CEF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F04CEF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

## 9 Further Comments

The packed storage scheme is illustrated by the following example when  $n = 4$  and UPLO = 'U'. Two-dimensional storage of the Hermitian matrix  $A$ :

$$\begin{array}{cccc} a_{11} & a_{12} & a_{13} & a_{14} \\ & a_{22} & a_{23} & a_{24} \\ & & a_{33} & a_{34} \\ & & & a_{44} \end{array} \quad (a_{ij} = \bar{a}_{ji})$$

Packed storage of the upper triangle of  $A$ :

$$AP = [a_{11}, a_{12}, a_{22}, a_{13}, a_{23}, a_{33}, a_{14}, a_{24}, a_{34}, a_{44}]$$

The total number of floating-point operations required to solve the equations  $AX = B$  is proportional to  $(\frac{1}{3}n^3 + n^2r)$ . The condition number estimation typically requires between four and five solves and never more than eleven solves, following the factorization.

In practice the condition number estimator is very reliable, but it can underestimate the true condition number; see Section 15.3 of Higham (2002) for further details.

The real analogue of F04CEF is F04BEF.

## 10 Example

This example solves the equations

$$AX = B,$$

where  $A$  is the Hermitian positive definite matrix

$$A = \begin{pmatrix} 3.23 & 1.51 - 1.92i & 1.90 + 0.84i & 0.42 + 2.50i \\ 1.51 + 1.92i & 3.58 & -0.23 + 1.11i & -1.18 + 1.37i \\ 1.90 - 0.84i & -0.23 - 1.11i & 4.09 & 2.33 - 0.14i \\ 0.42 - 2.50i & -1.18 - 1.37i & 2.33 + 0.14i & 4.29 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 3.93 - 6.14i & 1.48 + 6.58i \\ 6.17 + 9.42i & 4.65 - 4.75i \\ -7.17 - 21.83i & -4.91 + 2.29i \\ 1.99 - 14.38i & 7.64 - 10.79i \end{pmatrix}.$$

An estimate of the condition number of  $A$  and an approximate error bound for the computed solutions are also printed.

### 10.1 Program Text

```

Program f04cefe

!      F04CEF Example Program Text

!      Mark 25 Release. NAG Copyright 2014.

!      .. Use Statements ..
      Use nag_library, Only: f04cef, nag_wp, x04dbf
!      .. Implicit None Statement ..
      Implicit None
!      .. Parameters ..
      Integer, Parameter          :: nin = 5, nout = 6
      Character (1), Parameter   :: uplo = 'U'
!      .. Local Scalars ..

```

```

      Real (Kind=nag_wp)                :: errbnd, rcond
      Integer                          :: i, ierr, ifail, j, ldb, n, nrhs
!    .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: ap(:), b(:, :)
      Character (1)                    :: clabs(1), rlabs(1)
!    .. Executable Statements ..
      Write (nout,*) 'F04CEF Example Program Results'
      Write (nout,*)
      Flush (nout)
!    Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, nrhs
      ldb = n
      Allocate (ap((n*(n+1))/2),b(ldb,nrhs))
!    Read the upper or lower triangular part of the matrix A from
!    data file
      If (uplo=='U') Then
        Read (nin,*)((ap(i+(j*(j-1))/2),j=i,n),i=1,n)
      Else If (uplo=='L') Then
        Read (nin,*)((ap(i+((2*n-j)*(j-1))/2),j=1,i),i=1,n)
      End If

!    Read B from data file
      Read (nin,*)(b(i,1:nrhs),i=1,n)

!    Solve the equations AX = B for X

!    ifail: behaviour on error exit
!           =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 1
      Call f04cef(uplo,n,nrhs,ap,b,ldb,rcond,errbnd,ifail)

      If (ifail==0) Then
!        Print solution, estimate of condition number and approximate
!        error bound

          ierr = 0
          Call x04dbf('General',' ',n,nrhs,b,ldb,'Bracketed','F7.4','Solution', &
            'Integer',rlabs,'Integer',clabs,80,0,ierr)

          Write (nout,*)
          Write (nout,*) 'Estimate of condition number'
          Write (nout,99999) 1.0E0_nag_wp/rcond
          Write (nout,*)
          Write (nout,*) 'Estimate of error bound for computed solutions'
          Write (nout,99999) errbnd
      Else If (ifail==n+1) Then
!        Matrix A is numerically singular. Print estimate of
!        reciprocal of condition number and solution
          Write (nout,*)
          Write (nout,*) 'Estimate of reciprocal of condition number'
          Write (nout,99999) rcond
          Write (nout,*)
          Flush (nout)

          ierr = 0
          Call x04dbf('General',' ',n,nrhs,b,ldb,'Bracketed','F7.4','Solution', &
            'Integer',rlabs,'Integer',clabs,80,0,ierr)

      Else If (ifail>0 .And. ifail<=n) Then
!        The matrix A is not positive definite to working precision
          Write (nout,99998) 'The leading minor of order ', ifail, &
            ' is not positive definite'
      Else
          Write (nout,99997) ifail
      End If

99999 Format (4X,1P,E9.1)
99998 Format (1X,A,I3,A)
99997 Format (1X,' ** F04CEF returned with IFAIL = ',I5)
      End Program f04cefe

```

## 10.2 Program Data

F04CEF Example Program Data

```

      4              2              : n, nrhs
( 3.23,  0.00) ( 1.51, -1.92) ( 1.90,  0.84) ( 0.42,  2.50)
              ( 3.58,  0.00) (-0.23,  1.11) (-1.18,  1.37)
              ( 4.09,  0.00) ( 2.33, -0.14)
                              ( 4.29,  0.00) : matrix A

( 3.93, -6.14) ( 1.48,  6.58)
( 6.17,  9.42) ( 4.65, -4.75)
(-7.17,-21.83) (-4.91,  2.29)
( 1.99,-14.38) ( 7.64,-10.79)              : matrix B

```

## 10.3 Program Results

F04CEF Example Program Results

Solution

```

              1              2
1 ( 1.0000,-1.0000) (-1.0000, 2.0000)
2 (-0.0000, 3.0000) ( 3.0000,-4.0000)
3 (-4.0000,-5.0000) (-2.0000, 3.0000)
4 ( 2.0000, 1.0000) ( 4.0000,-5.0000)

```

Estimate of condition number

1.5E+02

Estimate of error bound for computed solutions

1.7E-14

---