

# NAG Library Routine Document

## F04CFF

**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

F04CFF computes the solution to a complex system of linear equations  $AX = B$ , where  $A$  is an  $n$  by  $n$  Hermitian positive definite band matrix of band width  $2k + 1$ , 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 F04CFF (UPLO, N, KD, NRHS, AB, LDAB, B, LDB, RCOND, ERBND, &  
IFAIL)

INTEGER N, KD, NRHS, LDAB, LDB, IFAIL  
REAL (KIND=nag\_wp) RCOND, ERBND  
COMPLEX (KIND=nag\_wp) AB(LDAB,\*), 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 band matrix with  $k$  superdiagonals, and  $L$  is a lower triangular band matrix with  $k$  subdiagonals. 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 Arguments

- 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: KD – INTEGER *Input*  
*On entry:* the number of superdiagonals  $k$  (and the number of subdiagonals) of the band matrix  $A$ .  
*Constraint:*  $KD \geq 0$ .

- 4: NRHS – INTEGER *Input*  
*On entry:* the number of right-hand sides  $r$ , i.e., the number of columns of the matrix  $B$ .  
*Constraint:* NRHS  $\geq 0$ .
- 5: AB(LDAB,\*) – COMPLEX (KIND=nag\_wp) array *Input/Output*  
**Note:** the second dimension of the array AB must be at least  $\max(1, N)$ .  
*On entry:* the  $n$  by  $n$  Hermitian band matrix  $A$ . The upper or lower triangular part of the Hermitian matrix is stored in the first  $KD + 1$  rows of the array. The  $j$ th column of  $A$  is stored in the  $j$ th column of the array AB as follows:  
The matrix is stored in rows 1 to  $k + 1$ , more precisely,  
if UPLO = 'U', the elements of the upper triangle of  $A$  within the band must be stored with element  $A_{ij}$  in  $AB(k + 1 + i - j, j)$  for  $\max(1, j - k) \leq i \leq j$ ;  
if UPLO = 'L', the elements of the lower triangle of  $A$  within the band must be stored with element  $A_{ij}$  in  $AB(1 + i - j, j)$  for  $j \leq i \leq \min(n, j + k)$ .  
See Section 9 below for further details.  
*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  $A$ .
- 6: LDAB – INTEGER *Input*  
*On entry:* the first dimension of the array AB as declared in the (sub)program from which F04CFF is called.  
*Constraint:* LDAB  $\geq KD + 1$ .
- 7: 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  $X$ .
- 8: LDB – INTEGER *Input*  
*On entry:* the first dimension of the array B as declared in the (sub)program from which F04CFF is called.  
*Constraint:* LDB  $\geq \max(1, N)$ .
- 9: 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  $RCOND = 1 / (\|A\|_1 \|A^{-1}\|_1)$ .
- 10: 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.
- 11: IFAIL – INTEGER *Input/Output*  
*On entry:* IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL =  $-399$

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL =  $-999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation 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. F04CFF 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

F04CFF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F04CFF 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 band storage scheme for the array AB is illustrated by the following example, when  $n = 6$ ,  $k = 2$ , and UPLO = 'U':

On entry:

```

      *   *   a13  a24  a35  a46
      *  a12  a23  a34  a45  a56
a11  a22  a33  a44  a55  a66

```

On exit:

```

      *   *   u13  u24  u35  u46
      *  u12  u23  u34  u45  u56
u11  u22  u33  u44  u55  u66

```

Similarly, if UPLO = 'L' the format of AB is as follows:

On entry:

```

a11  a22  a33  a44  a55  a66
a21  a32  a43  a54  a65  *
a31  a42  a53  a64  *   *

```

On exit:

```

      l11  l22  l33  l44  l55  l66
      l21  l32  l43  l54  l65  *
      l31  l42  l53  l64  *   *

```

Array elements marked \* need not be set and are not referenced by the routine.

Assuming that  $n \gg k$ , the total number of floating-point operations required to solve the equations  $AX = B$  is approximately  $n(k+1)^2$  for the factorization and  $4nkr$  for the solution following the factorization. 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 F04CFF is F04BFF.

## 10 Example

This example solves the equations

$$AX = B,$$

where  $A$  is the Hermitian positive definite band matrix

$$A = \begin{pmatrix} 9.39 & 1.08 - 1.73i & 0 & 0 \\ 1.08 + 1.73i & 1.69 & -0.04 + 0.29i & 0 \\ 0 & -0.04 - 0.29i & 2.65 & -0.33 + 2.24i \\ 0 & 0 & -0.33 - 2.24i & 2.17 \end{pmatrix}$$

and

$$B = \begin{pmatrix} -12.42 + 68.42i & 54.30 - 56.56i \\ -9.93 + 0.88i & 18.32 + 4.76i \\ -27.30 - 0.01i & -4.40 + 9.97i \\ 5.31 + 23.63i & 9.43 + 1.41i \end{pmatrix}.$$

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 f04cffe

!      F04CFF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
Use nag_library, Only: f04cff, 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, kd, ldab, ldb, n, &
                          nrhs
!      .. Local Arrays ..
Complex (Kind=nag_wp), Allocatable :: ab(:,,:), b(:,:)
Character (1)                :: clabs(1), rlabs(1)
!      .. Intrinsic Procedures ..
Intrinsic                    :: max, min
!      .. Executable Statements ..
Write (nout,*) 'F04CFF Example Program Results'
Write (nout,*)
Flush (nout)
!      Skip heading in data file
Read (nin,*)
Read (nin,*) n, kd, nrhs
ldab = kd + 1
ldb = n
Allocate (ab(ldab,n),b(ldb,nrhs))
!      Read the upper or lower triangular part of the band matrix A
!      from data file
If (uplo=='U') Then
  Do i = 1, n
    Read (nin,*)(ab(kd+1+i-j,j),j=i,min(n,i+kd))
  End Do
Else If (uplo=='L') Then
  Do i = 1, n
    Read (nin,*)(ab(1+i-j,j),j=max(1,i-kd),i)
  End Do
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 f04cff(uplo,n,kd,nrhs,ab,ldab,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,' ** F04CFF returned with IFAIL = ',I5)
End Program f04cffe

```

## 10.2 Program Data

F04CFF Example Program Data

```

         4           1           2                               : n, kd, nrhs
(  9.39,  0.00) (  1.08, -1.73)
              (  1.69,  0.00) ( -0.04,  0.29)
                                (  2.65,  0.00) ( -0.33,  2.24)
                                                (  2.17,  0.00) : matrix A

(-12.42, 68.42) ( 54.30,-56.56)
(- 9.93,  0.88) ( 18.32,  4.76)
(-27.30, -0.01) ( -4.40,  9.97)
(  5.31, 23.63) (  9.43,  1.41)                               : matrix B

```

## 10.3 Program Results

F04CFF Example Program Results

```

Solution
              1           2
1 (-1.0000, 8.0000) ( 5.0000,-6.0000)
2 ( 2.0000,-3.0000) ( 2.0000, 3.0000)
3 (-4.0000,-5.0000) (-8.0000, 4.0000)
4 ( 7.0000, 6.0000) (-1.0000,-7.0000)

```

Estimate of condition number  
1.3E+02

Estimate of error bound for computed solutions  
1.5E-14

---