NAG Library Routine Document

F04BFF

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

F04BFF computes the solution to a real system of linear equations $AX = B$, where A is an n by n symmetric 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 F04BFF (UPLO, N, KD, NRHS, AB, LDAB, B, LDB, RCOND, ERRBND,        &
                   IFAIL)
INTEGER            N, KD, NRHS, LDAB, LDB, IFAIL
REAL (KIND=nag_wp) AB(LDAB,*), B(LDB,*), RCOND, ERRBND
CHARACTER(1)       UPLO
```

3 Description

The Cholesky factorization is used to factor A as $A = U^{\mathrm{T}}U$, if UPLO = 'U', or $A = LL^{\mathrm{T}}$, 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 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 ≥ 0.

3: KD – INTEGER *Input*

On entry: the number of superdiagonals k (and the number of subdiagonals) of the band matrix A.

Constraint: KD ≥ 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: $\text{NRHS} \geq 0$.

5: AB(LDAB,$*$) – REAL (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array AB must be at least $\max(1, \text{N})$.

On entry: the n by n symmetric band matrix A. The upper or lower triangular part of the symmetric matrix is stored in the first $\text{KD} + 1$ rows of the array. The jth column of A is stored in the jth column of the array AB as follows:

if UPLO = 'U', $\text{AB}(k+1+i-j, j) = a_{ij}$ for $\max(1, j-k) \leq i \leq j$;

if UPLO = 'L', $\text{AB}(1+i-j, j) = a_{ij}$ for $j \leq i \leq \min(n, j+k)$.

See Section 8 below for further details.

On exit: if $\text{IFAIL} = 0$ or $\text{N} + 1$, the factor U or L from the Cholesky factorization $A = U^{\mathrm{T}}U$ or $A = LL^{\mathrm{T}}$, 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 F04BFF is called.

Constraint: $\text{LDAB} \geq \text{KD} + 1$.

7: B(LDB,$*$) – REAL (KIND=nag_wp) array *Input/Output*

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

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

On exit: if $\text{IFAIL} = 0$ or $\text{N} + 1$, 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 F04BFF is called.

Constraint: $\text{LDB} \geq \max(1, \text{N})$.

9: RCOND – REAL (KIND=nag_wp) *Output*

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

10: ERRBND – REAL (KIND=nag_wp) *Output*

On exit: if $\text{IFAIL} = 0$ or $\text{N} + 1$, an estimate of the forward error bound for a computed solution $\hat{x}$, such that $\|\hat{x} - x\|_1 / \|x\|_1 \leq \text{ERRBND}$, 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 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 ith argument had an illegal value.

IFAIL $= -999$

Allocation of memory failed. The integer allocatable memory required is N, and the real allocatable memory required is $3 \times$ N. Allocation failed before the solution could be computed.

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.

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 ϵ 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. F04BFF uses the approximation $\|E\|_1 = \epsilon\|A\|_1$ to estimate ERRBND. See Section 4.4 of Anderson *et al.* (1999) for further details.

8 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:

$$\begin{array}{cccccc} * & * & a_{13} & a_{24} & a_{35} & a_{46} \\ * & a_{12} & a_{23} & a_{34} & a_{45} & a_{56} \\ a_{11} & a_{22} & a_{33} & a_{44} & a_{55} & a_{66} \end{array}$$

On exit:

$$\begin{array}{cccccc} * & * & u_{13} & u_{24} & u_{35} & u_{46} \\ * & u_{12} & u_{23} & u_{34} & u_{45} & u_{56} \\ u_{11} & u_{22} & u_{33} & u_{44} & u_{55} & u_{66} \end{array}$$

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

On entry:

$$\begin{array}{cccccc} a_{11} & a_{22} & a_{33} & a_{44} & a_{55} & a_{66} \\ a_{21} & a_{32} & a_{43} & a_{54} & a_{65} & * \\ a_{31} & a_{42} & a_{53} & a_{64} & * & * \end{array}$$

On exit:

$$\begin{array}{cccccc} l_{11} & l_{22} & l_{33} & l_{44} & l_{55} & l_{66} \\ l_{21} & l_{32} & l_{43} & l_{54} & l_{65} & * \\ l_{31} & l_{42} & l_{53} & l_{64} & * & * \end{array}$$

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 complex analogue of F04BFF is F04CFF.

9 Example

This example solves the equations

$$AX = B,$$

where A is the symmetric positive definite band matrix

$$A = \begin{pmatrix} 5.49 & 2.68 & 0 & 0 \\ 2.68 & 5.63 & -2.39 & 0 \\ 0 & -2.39 & 2.60 & -2.22 \\ 0 & 0 & -2.22 & 5.17 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 22.09 & 5.10 \\ 9.31 & 30.81 \\ -5.24 & -25.82 \\ 11.83 & 22.90 \end{pmatrix}.$$

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

9.1 Program Text

```
    Program f04bffe

!     F04BFF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: f04bff, nag_wp, x04caf
!     .. 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 ..
      Real (Kind=nag_wp), Allocatable  :: ab(:,:), b(:,:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max, min
!     .. Executable Statements ..
      Write (nout,*) 'F04BFF 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 f04bff(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 x04caf('General',' ',n,nrhs,b,ldb,'Solution',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 x04caf('General',' ',n,nrhs,b,ldb,'Solution',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 (6X,1P,E9.1)
99998 Format (1X,A,I3,A)
99997 Format (1X,' ** F04BFF returned with IFAIL = ',I5)
    End Program f04bffe
```

9.2 Program Data

```
F04BFF Example Program Data

  4      1      2           : n, kd, nrhs

  5.49   2.68
         5.63  -2.39
                2.60  -2.22
                       5.17 : matrix A

 22.09   5.10
  9.31  30.81
 -5.24 -25.82
 11.83  22.90               : matrix B
```

9.3 Program Results

```
 F04BFF Example Program Results

 Solution
             1          2
 1      5.0000    -2.0000
 2     -2.0000     6.0000
 3     -3.0000    -1.0000
 4      1.0000     4.0000

 Estimate of condition number
        7.4E+01

 Estimate of error bound for computed solutions
        8.2E-15
```