NAG Library Routine Document

F04ABF

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

F04ABF calculates the accurate solution of a set of real symmetric positive definite linear equations with multiple right-hand sides, using a Cholesky factorization and iterative refinement.

2 Specification

```
SUBROUTINE F04ABF (A, LDA, B, LDB, N, M, C, LDC, WKSPCE, BB, LDBB,          &
                   IFAIL)

INTEGER             LDA, LDB, N, M, LDC, LDBB, IFAIL
REAL (KIND=nag_wp) A(LDA,*), B(LDB,*), C(LDC,M), WKSPCE(N), BB(LDBB,M)
```

3 Description

Given a set of real linear equations $AX = B$, where A is symmetric positive definite, F04ABF first computes a Cholesky factorization of A as $A = LL^{\mathrm{T}}$, where L is lower triangular. An approximation to X is found by forward and backward substitution. The residual matrix $R = B - AX$ is then calculated using ***additional precision***, and a correction D to X is found by solving $LL^{\mathrm{T}}D = R$. X is replaced by $X + D$, and this iterative refinement of the solution is repeated until full machine accuracy has been obtained.

4 References

Wilkinson J H and Reinsch C (1971) *Handbook for Automatic Computation II, Linear Algebra* Springer–Verlag

5 Parameters

1: A(LDA, $*$) – REAL (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array A must be at least max(1, N).

On entry: the upper triangle of the n by n positive definite symmetric matrix A. The elements of the array below the diagonal need not be set.

On exit: the elements of the array below the diagonal are overwritten; the upper triangle of A is unchanged.

2: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F04ABF is called.

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

3: B(LDB, $*$) – REAL (KIND=nag_wp) array *Input*

Note: the second dimension of the array B must be at least max(1, M).

On entry: the n by m right-hand side matrix B.

4: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F04ABF is called.

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

5: N – INTEGER *Input*

On entry: n, the order of the matrix A.

Constraint: N ≥ 0.

6: M – INTEGER *Input*

On entry: m, the number of right-hand sides.

Constraint: M ≥ 0.

7: C(LDC, M) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array C must be at least $\max(1, \mathrm{M})$.

On exit: the n by m solution matrix X.

8: LDC – INTEGER *Input*

On entry: the first dimension of the array C as declared in the (sub)program from which F04ABF is called.

Constraint: LDC $\geq \max(1, \mathrm{N})$.

9: WKSPCE(N) – REAL (KIND=nag_wp) array *Workspace*

10: BB(LDBB, M) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array BB must be at least $\max(1, \mathrm{M})$.

On exit: the final n by m residual matrix $R = B - AX$.

11: LDBB – INTEGER *Input*

On entry: the first dimension of the array BB as declared in the (sub)program from which F04ABF is called.

Constraint: LDBB $\geq \max(1, \mathrm{N})$.

12: 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 $= 1$

The matrix A is not positive definite, possibly due to rounding errors.

IFAIL $= 2$

Iterative refinement fails to improve the solution, i.e., the matrix A is too ill-conditioned.

IFAIL $= 3$

On entry, N < 0,
or N < 0,
or LDA $<$ max(1, N),
or LDB $<$ max(1, N),
or LDC $<$ max(1, N),
or LDBB $<$ max(1, N).

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 solutions should be correct to full machine accuracy. For a detailed error analysis see page 39 of Wilkinson and Reinsch (1971).

8 Parallelism and Performance

F04ABF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F04ABF 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 time taken by F04ABF is approximately proportional to n^3.

If there is only one right-hand side, it is simpler to use F04ASF.

10 Example

This example solves the set of linear equations $AX = B$ where

$$A = \begin{pmatrix} 5 & 7 & 6 & 5 \\ 7 & 10 & 8 & 7 \\ 6 & 8 & 10 & 9 \\ 5 & 7 & 9 & 10 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 23 \\ 32 \\ 33 \\ 31 \end{pmatrix}.$$

10.1 Program Text

```
    Program f04abfe

!     F04ABF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: f04abf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, lda, ldb, ldbb, ldc, m, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), b(:,:), bb(:,:), c(:,:),     &
                                          wkspce(:)
!     .. Executable Statements ..
      Write (nout,*) 'F04ABF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, m
      lda = n
      ldb = n
      ldbb = n
      ldc = n
      Allocate (a(lda,n),b(ldb,m),bb(ldbb,m),c(ldc,m),wkspce(n))
      Read (nin,*)(a(i,1:n),i=1,n), (b(i,1:m),i=1,n)

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call f04abf(a,lda,b,ldb,n,m,c,ldc,wkspce,bb,ldbb,ifail)

      Write (nout,*) ' Solution'
      Write (nout,99999)(c(i,1:m),i=1,n)

99999 Format (1X,F9.4)
    End Program f04abfe
```

10.2 Program Data

```
F04ABF Example Program Data
    4    1             : n, m
    5    7    6    5
    7   10    8    7
    6    8   10    9
    5    7    9   10
   23   32   33   31   : matrices A and B
```

10.3 Program Results

```
 F04ABF Example Program Results

  Solution
    1.0000
    1.0000
    1.0000
    1.0000
```