F04ABF – NAG Fortran Library Routine Document

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, IA, B, IB, N, M, C, IC, WKSPCE, BB, IBB, IFAIL)
INTEGER            IA, IB, N, M, IC, IBB, IFAIL
real               A(IA,*), B(IB,*), C(IC,*), WKSPCE(*), BB(IBB,*)
```

3 Description

Given a set of real linear equations $AX = B$, where A is symmetric positive-definite, this routine first computes a Cholesky factorization of A as $A = LL^T$, 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 $LL^T D = R$. X is replaced by $X + D$, and this iterative refinement of the solution is repeated until full machine accuracy has been obtained.

4 References

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

5 Parameters

1: A(IA,∗) — ***real*** 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: IA — INTEGER *Input*

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

Constraint: IA ≥ max(1,N).

3: B(IB,∗) — ***real*** 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: IB — INTEGER *Input*

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

Constraint: IB ≥ max(1,N).

5: N — INTEGER *Input*

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

Constraint: N $\geq$ 0.

6: M — INTEGER *Input*

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

Constraint: M $\geq$ 0.

7: C(IC,$*$) — ***real*** array *Output*

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

On exit: the n by m solution matrix X.

8: IC — INTEGER *Input*

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

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

9: WKSPCE($*$) — ***real*** array *Workspace*

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

10: BB(IBB,$*$) — ***real*** array *Output*

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

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

11: IBB — INTEGER *Input*

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

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

12: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL = 0 unless the routine detects an error (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 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 M < 0,
or IA < max(1,N),
or IB < max(1,N),
or IC < max(1,N),
or IBB < max(1,N).

7 Accuracy

The computed solutions should be correct to full machine accuracy. For a detailed error analysis see Wilkinson and Reinsch [1] page 39.

8 Further Comments

The time taken by the routine is approximately proportional to n^3.

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

9 Example

6 To solve the set of linear equations $AX = B$ where

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

9.1 Program Text

Note. The listing of the example program presented below uses bold italicised terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     F04ABF Example Program Text
*     Mark 15 Revised.  NAG Copyright 1991.
*     .. Parameters ..
      INTEGER          NMAX, IA, IB, IC, IBB
      PARAMETER        (NMAX=8,IA=NMAX,IB=NMAX,IC=NMAX,IBB=NMAX)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, M, N
*     .. Local Arrays ..
      real             A(IA,NMAX), B(IB,1), BB(IBB,1), C(IC,1),
     +                 WKSPCE(NMAX)
*     .. External Subroutines ..
      EXTERNAL         F04ABF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F04ABF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      WRITE (NOUT,*)
      M = 1
      IF (N.GE.0 .AND. N.LE.NMAX) THEN
         READ (NIN,*) ((A(I,J),J=1,N),I=1,N), (B(I,1),I=1,N)
         IFAIL = 0
*
         CALL F04ABF(A,IA,B,IB,N,M,C,IC,WKSPCE,BB,IBB,IFAIL)
*
         WRITE (NOUT,*) ' Solution'
         WRITE (NOUT,99998) (C(I,1),I=1,N)
      ELSE
         WRITE (NOUT,99999) 'N is out of range: N = ', N
      END IF
      STOP
```

```
*
99999 FORMAT (1X,A,I5)
99998 FORMAT (1X,F9.4)
      END
```

9.2 Program Data

```
F04ABF Example Program Data
  4
    5    7    6    5
    7   10    8    7
    6    8   10    9
    5    7    9   10
   23   32   33   31
```

9.3 Program Results

```
 F04ABF Example Program Results

  Solution
    1.0000
    1.0000
    1.0000
    1.0000
```
