NAG Library Routine Document

F04ASF

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

F04ASF calculates the accurate solution of a set of real symmetric positive-definite linear equations with a single right-hand side, $Ax = b$, using a Cholesky factorization and iterative refinement.

2 Specification

```
SUBROUTINE F04ASF(A, LDA, B, N, C, WK1, WK2, IFAIL)
INTEGER           LDA, N, IFAIL
double precision  A(LDA,*), B(max(1,N)), C(max(1,N)), WK1(max(1,N)),
1                 WK2(max(1,N))
```

3 Description

Given a set of real linear equations $Ax = b$, where A is a symmetric positive-definite matrix, F04ASF 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 vector $r = b - Ax$ is then calculated using ***additional precision*** and a correction d to x is found by solving $LL^{\mathrm{T}}d = r$. x is then replaced by $x + d$, and this iterative refinement of the solution is repeated until machine accuracy is obtained.

4 References

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

5 Parameters

1: A(LDA,$*$) – ***double precision*** array *Input/Output*

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

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 F04ASF is called.

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

3: B($\max(1, \mathrm{N})$) – ***double precision*** array *Input*

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

On entry: the right-hand side vector b.

4: N – INTEGER *Input*

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

Constraint: $\mathrm{N} \geq 0$.

5: C(max(1, N)) – ***double precision*** array *Output*

On exit: the solution vector x.

6: WK1(max(1, N)) – ***double precision*** array *Workspace*

7: WK2(max(1, N)) – ***double precision*** array *Workspace*

8: 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.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

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.**

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 LDA < max(1, N).

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 Further Comments

The time taken by F04ASF is approximately proportional to n^3.

The routine **must not** be called with the same name for parameters B and C.

9 Example

This example solves the set of linear equations $Ax = b$ where

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

9.1 Program Text

```
*     F04ASF Example Program Text
*     Mark 15 Revised. NAG Copyright 1991.
*     .. Parameters ..
      INTEGER          NMAX, LDA
      PARAMETER        (NMAX=8,LDA=NMAX)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, N
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), B(NMAX), C(NMAX), WK1(NMAX),
     +                 WK2(NMAX)
*     .. External Subroutines ..
      EXTERNAL         F04ASF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F04ASF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      WRITE (NOUT,*)
      IF (N.GE.0 .AND. N.LE.NMAX) THEN
         READ (NIN,*) ((A(I,J),J=1,N),I=1,N), (B(I),I=1,N)
         IFAIL = 1
*
         CALL F04ASF(A,LDA,B,N,C,WK1,WK2,IFAIL)
*
         IF (IFAIL.EQ.0) THEN
            WRITE (NOUT,*) ' Solution'
            WRITE (NOUT,99998) (C(I),I=1,N)
         ELSE
            WRITE (NOUT,99997) IFAIL
         END IF
      ELSE
         WRITE (NOUT,99999) 'N is out of range: N = ', N
      END IF
*
99999 FORMAT (1X,A,I5)
99998 FORMAT (1X,F9.4)
99997 FORMAT (1X,' ** F04ASF returned with IFAIL = ',I5)
      END
```

9.2 Program Data

```
F04ASF 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

```
 F04ASF Example Program Results

  Solution
    1.0000
    1.0000
    1.0000
    1.0000
```