NAG Library Routine Document

F04ATF

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

F04ATF calculates the accurate solution of a set of real linear equations with a single right-hand side, using an LU factorization with partial pivoting, and iterative refinement.

2 Specification

```
SUBROUTINE F04ATF(A, LDA, B, N, C, AA, LDAA, WKS1, WKS2, IFAIL)

INTEGER            LDA, N, LDAA, IFAIL
double precision   A(LDA,*), B(*), C(N), AA(LDAA,N), WKS1(N), WKS2(N)
```

3 Description

Given a set of real linear equations, $Ax = b$, the routine first computes an LU factorization of A with partial pivoting, $PA = LU$, where P is a permutation matrix, L is lower triangular and U is unit upper triangular. An approximation to x is found by forward and backward substitution in $Ly = Pb$ and $Ux = y$. The residual vector $r = b - Ax$ is then calculated using ***additional precision***, and a correction d to x is found by solving $LUd = r$. x is replaced by $x + d$, and this iterative refinement of the solution is repeated until full 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*

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

On entry: the n by n matrix A.

2: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F04ATF is called.

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

3: B($*$) – ***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(N) – ***double precision*** array *Output*

On exit: the solution vector x.

6: AA(LDAA,N) – ***double precision*** array *Output*

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

On exit: the triangular factors L and U, except that the unit diagonal elements of U are not stored.

7: LDAA – INTEGER *Input*

On entry: the first dimension of the array AA as declared in the (sub)program from which F04ATF is called.

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

8: WKS1(N) – ***double precision*** array *Workspace*

9: WKS2(N) – ***double precision*** array *Workspace*

10: 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: $\mathrm{IFAIL} = 0$ 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 $\mathrm{IFAIL} = 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

$\mathrm{IFAIL} = 1$

The matrix A is singular, possibly due to rounding errors.

$\mathrm{IFAIL} = 2$

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

$\mathrm{IFAIL} = 3$

On entry, $\mathrm{N} < 0$,
or $\mathrm{LDA} < \max(1, \mathrm{N})$,
or $\mathrm{LDAA} < \max(1, \mathrm{N})$.

7 Accuracy

The computed solutions should be correct to full machine accuracy. For a detailed error analysis see page 107 of Wilkinson and Reinsch (1971).

8 Further Comments

The time taken by F04ATF 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

$$A = \begin{pmatrix} 33 & 16 & 72 \\ -24 & -10 & -57 \\ -8 & -4 & -17 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} -359 \\ 281 \\ 85 \end{pmatrix}.$$

9.1 Program Text

```
*     F04ATF Example Program Text
*     Mark 15 Revised. NAG Copyright 1991.
*     .. Parameters ..
      INTEGER          NMAX, LDA, LDAA
      PARAMETER        (NMAX=8,LDA=NMAX,LDAA=NMAX)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, N
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), AA(LDAA,NMAX), B(NMAX), C(NMAX),
     +                 WKS1(NMAX), WKS2(NMAX)
*     .. External Subroutines ..
      EXTERNAL         F04ATF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F04ATF 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 F04ATF(A,LDA,B,N,C,AA,LDAA,WKS1,WKS2,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,' ** F04ATF returned with IFAIL = ',I5)
      END
```

9.2 Program Data

```
F04ATF Example Program Data
  3
   33   16   72
  -24  -10  -57
   -8   -4  -17
 -359  281   85
```

9.3 Program Results

```
 F04ATF Example Program Results

  Solution
    1.0000
   -2.0000
   -5.0000
```