NAG Fortran 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, IA, B, N, C, AA, IAA, WKS1, WKS2, IFAIL)
INTEGER            IA, N, IAA, IFAIL
real               A(IA,*), B(*), C(*), AA(IAA,*), WKS1(*), WKS2(*)
```

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(IA,*) – ***real*** array *Input*

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

On entry: the n by n matrix A.

2: IA – INTEGER *Input*

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

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

3: B(*) – ***real*** array *Input*

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

On entry: the right-hand side vector b.

4: N – INTEGER *Input*

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

Constraint: N ≥ 0.

5: C(∗) – ***real*** array *Output*

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

On exit: the solution vector x.

6: AA(IAA,*) – ***real*** array *Output*

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

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

7: IAA – INTEGER *Input*

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

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

8: WKS1(∗) – ***real*** array *Workspace*

Note: the dimension of the array WKS1 must be at least max(1, N).

9: WKS2(∗) – ***real*** array *Workspace*

Note: the dimension of the array WKS2 must be at least max(1, N).

10: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 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, for users 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 singular, 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 IA $<$ max(1, N),
or IAA $<$ max(1, 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 the routine is approximately proportional to n^3.

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

9 Example

To solve 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

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.

```
*     F04ATF Example Program Text
*     Mark 15 Revised.  NAG Copyright 1991.
*     .. Parameters ..
      INTEGER          NMAX, IA, IAA
      PARAMETER        (NMAX=8,IA=NMAX,IAA=NMAX)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, N
*     .. Local Arrays ..
      real             A(IA,NMAX), AA(IAA,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 = 0
*
         CALL F04ATF(A,IA,B,N,C,AA,IAA,WKS1,WKS2,IFAIL)
*
         WRITE (NOUT,*) ' Solution'
         WRITE (NOUT,99998) (C(I),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

```
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
```