NAG Fortran Library Routine Document

F04AAF

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

F04AAF calculates the approximate solution of a set of real linear equations with multiple right-hand sides, using an LU factorization with partial pivoting.

2 Specification

```
SUBROUTINE F04AAF(A, IA, B, IB, N, M, C, IC, WKSPCE, IFAIL)
INTEGER            IA, IB, N, M, IC, IFAIL
real               A(IA,*), B(IB,*), C(IC,*), WKSPCE(*)
```

3 Description

Given a set of real linear equations $AX = \mathrm{B}$, 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. The columns x of the solution X are found by forward and backward substitution in $Ly = Pb$ and $Ux = y$, where b is a column of the right-hand side matrix B.

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/Output*

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

On entry: the n by n matrix A.

On exit: A is overwritten by the lower triangular matrix L and the off-diagonal elements of the upper triangular matrix U. The unit diagonal elements of U are not stored.

2: IA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F04AAF is called.

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

3: B(IB,*) – ***real*** array *Input*

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

On entry: the n by m right-hand side matrix B. See also Section 8.

4: IB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F04AAF is called.

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

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. See also Section 8.

8: IC – INTEGER *Input*

On entry: the first dimension of the array C as declared in the (sub)program from which F04AAF 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: 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 $= 2$

On entry, $N < 0$,
or $M < 0$,
or $IA < \max(1, N)$,
or $IB < \max(1, N)$,
or $IC < \max(1, N)$.

7 Accuracy

The accuracy of the computed solutions depends on the stability of the original matrix of coefficients. 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.

Unless otherwise stated in the Users' Note for your implementation, the routine may be called with the same actual array supplied for parameters B and C, in which case the solution vectors will overwrite the right-hand sides. However this is not standard Fortran 77, and may not work on all systems.

If there is only one right-hand side, it is simpler to use F04ARF.

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.

```
*     F04AAF Example Program Text
*     Mark 15 Revised.  NAG Copyright 1991.
*     .. Parameters ..
      INTEGER          NMAX, IA, IB, IC
      PARAMETER        (NMAX=8,IA=NMAX,IB=NMAX,IC=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), C(IC,1), WKSPCE(NMAX)
*     .. External Subroutines ..
      EXTERNAL         F04AAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F04AAF 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,1),I=1,N)
         M = 1
         IFAIL = 0
*
         CALL F04AAF(A,IA,B,IB,N,M,C,IC,WKSPCE,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

```
F04AAF Example Program Data
  3
   33   16   72
  -24  -10  -57
   -8   -4  -17
 -359  281   85
```

9.3 Program Results

```
 F04AAF Example Program Results

  Solution
    1.0000
   -2.0000
   -5.0000
```