

NAG Fortran Library Routine Document

F04ARF

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

F04ARF calculates the approximate solution of a set of real linear equations with a single right-hand side, using an *LU* factorization with partial pivoting.

2 Specification

```
SUBROUTINE F04ARF(A, IA, B, N, C, WKSPCE, IFAIL)
INTEGER          IA, N, IFAIL
real           A(IA,*), B(*), C(*), WKSPCE(*)
```

3 Description

Given a set of 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. The approximate solution x is found by forward and backward substitution in $Ly = Pb$ and $Ux = y$, where b is the right-hand side.

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 $\max(1, N)$.
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 F04ARF 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: $N \geq 0$.

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: WKSPCE(*) – *real* array Workspace

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

7: 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 $IA < \max(1, N)$.

7 Accuracy

The accuracy of the computed solution depends on the conditioning of the original matrix. 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 vector will overwrite the right-hand side. However this is not standard Fortran 77, and may not work on all systems.

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.

```
*      F04ARF Example Program Text
*      Mark 15 Revised.  NAG Copyright 1991.
*      .. Parameters ..
      INTEGER          NMAX, IA
      PARAMETER        (NMAX=8,IA=NMAX)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*      .. Local Scalars ..
      INTEGER          I, IFAIL, J, N
*      .. Local Arrays ..
      real             A(IA,NMAX), B(NMAX), C(NMAX), WKSPACE(NMAX)
*      .. External Subroutines ..
      EXTERNAL         F04ARF
*      .. Executable Statements ..
      WRITE (NOUT,*) 'F04ARF 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 F04ARF(A,IA,B,N,C,WKSPACE,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

```
F04ARF Example Program Data
3
 33   16   72
-24  -10  -57
  -8   -4  -17
-359 281   85
```

9.3 Program Results

```
F04ARF Example Program Results

Solution
 1.0000
-2.0000
-5.0000
```
