

NAG Library Routine Document

F04ADF

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

F04ADF calculates the approximate solution of a set of complex linear equations with multiple right-hand sides, using an *LU* factorization with partial pivoting.

2 Specification

```
SUBROUTINE F04ADF(A, LDA, B, LDB, N, M, C, LDC, WKSPCE, IFAIL)
INTEGER          LDA, LDB, N, M, LDC, IFAIL
double precision WKSPCE(N)
complex*16      A(LDA,*), B(LDB,*), C(LDC,M)
```

3 Description

Given a set of complex 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 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: $A(LDA,*)$ – ***complex*16*** 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: LDA – INTEGER *Input*
On entry: the first dimension of the array A as declared in the (sub)program from which F04ADF is called.
Constraint: $LDA \geq \max(1, N)$.
- 3: $B(LDB,*)$ – ***complex*16*** array *Input*
Note: the second dimension of the array B must be at least $\max(1, M)$.
On entry: the n by m right-hand side matrix B . See also Section 8.

- 4: LDB – INTEGER *Input*
On entry: the first dimension of the array B as declared in the (sub)program from which F04ADF is called.
Constraint: $LDB \geq \max(1, N)$.
- 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(LDC,M) – **complex*16** 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: LDC – INTEGER *Input*
On entry: the first dimension of the array C as declared in the (sub)program from which F04ADF is called.
Constraint: $LDC \geq \max(1, N)$.
- 9: WKSPCE(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: 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 singular, possibly due to rounding errors.

IFAIL = 2

On entry, $N < 0$,
 or $M < 0$,
 or $LDA < \max(1, N)$,
 or $LDB < \max(1, N)$,
 or $LDC < \max(1, N)$.

7 Accuracy

The accuracy of the computed solution depends on the conditioning of the original matrix. For a detailed error analysis see page 106 of Wilkinson and Reinsch (1971).

8 Further Comments

The time taken by F04ADF 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.

9 Example

This example solves the set of linear equations $AX = B$ where

$$A = \begin{pmatrix} 1 & 1+2i & 2+10i \\ 1+i & 3i & -5+14i \\ 1+i & 5i & -8+20i \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

9.1 Program Text

```
*      F04ADF Example Program Text
*      Mark 15 Revised. NAG Copyright 1991.
*      .. Parameters ..
      INTEGER          NMAX, LDA, LDB, LDC
      PARAMETER        (NMAX=4,LDA=NMAX,LDB=NMAX,LDC=NMAX)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*      .. Local Scalars ..
      INTEGER          I, IFAIL, J, M, N
*      .. Local Arrays ..
      COMPLEX *16      A(LDA,NMAX), B(LDB,1), C(LDC,1)
      DOUBLE PRECISION WKSPCE(NMAX)
*      .. External Subroutines ..
      EXTERNAL         F04ADF
*      .. Executable Statements ..
      WRITE (NOUT,*) 'F04ADF Example Program Results'
*      Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      WRITE (NOUT,*)
      M = 1
      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)
        IFAIL = 1
*
        CALL F04ADF(A,LDA,B,LDB,N,M,C,LDC,WKSPCE,IFAIL)
*
        IF (IFAIL.EQ.0) THEN
          WRITE (NOUT,*) ' Solution'
          WRITE (NOUT,99998) (C(I,1),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,'(',F7.4,',',F7.4,') ',:)
99997 FORMAT (1X,' ** F04ADF returned with IFAIL = ',I5)
      END
```

9.2 Program Data

F04ADF Example Program Data

```
3
( 1.0, 0.0 ) ( 1.0, 2.0 ) ( 2.0,10.0 )
( 1.0, 1.0 ) ( 0.0, 3.0 ) (-5.0,14.0 )
( 1.0, 1.0 ) ( 0.0, 5.0 ) (-8.0,20.0 )
( 1.0, 0.0 ) ( 0.0, 0.0 ) ( 0.0, 0.0 )
```

9.3 Program Results

F04ADF Example Program Results

```
Solution
(10.0000, 1.0000)
( 9.0000,-3.0000)
(-2.0000, 2.0000)
```
