NAG Library Routine Document

F07ANF (ZGESV)

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

F07ANF (ZGESV) computes the solution to a complex system of linear equations

$$AX = B,$$

where A is an n by n matrix and X and B are n by r matrices.

2 Specification

```
SUBROUTINE F07ANF (N, NRHS, A, LDA, IPIV, B, LDB, INFO)

INTEGER              N, NRHS, LDA, IPIV(N), LDB, INFO
COMPLEX (KIND=nag_wp) A(LDA,*), B(LDB,*)
```

The routine may be called by its LAPACK name ***zgesv***.

3 Description

F07ANF (ZGESV) uses the LU decomposition with partial pivoting and row interchanges to factor A as

$$A = PLU,$$

where P is a permutation matrix, L is unit lower triangular, and U is upper triangular. The factored form of A is then used to solve the system of equations $AX = B$.

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia http://www.netlib.org/lapack/lug

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Parameters

1: N – INTEGER *Input*

On entry: n, the number of linear equations, i.e., the order of the matrix A.

Constraint: $N \geq 0$.

2: NRHS – INTEGER *Input*

On entry: r, the number of right-hand sides, i.e., the number of columns of the matrix B.

Constraint: $NRHS \geq 0$.

3: A(LDA,$*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

On entry: the n by n coefficient matrix A.

On exit: the factors L and U from the factorization $A = PLU$; the unit diagonal elements of L are not stored.

4: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F07ANF (ZGESV) is called.

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

5: IPIV(N) – INTEGER array *Output*

On exit: if no constraints are violated, the pivot indices that define the permutation matrix P; at the ith step row i of the matrix was interchanged with row IPIV(i). IPIV$(i) = i$ indicates a row interchange was not required.

6: B(LDB,$*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

On entry: the n by r right-hand side matrix B.

On exit: if INFO $= 0$, the n by r solution matrix X.

7: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F07ANF (ZGESV) is called.

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

8: INFO – INTEGER *Output*

On exit: INFO $= 0$ unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

If INFO $= -i$, the ith argument had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO > 0

If INFO $= i$, u_{ii} is exactly zero. The factorization has been completed, but the factor U is exactly singular, so the solution could not be computed.

7 Accuracy

The computed solution for a single right-hand side, $\hat{x}$, satisfies the equation of the form

$$(A + E)\hat{x} = b,$$

where

$$\|E\|_1 = O(\epsilon)\|A\|_1$$

and ϵ is the ***machine precision***. An approximate error bound for the computed solution is given by

$$\frac{\|\hat{x} - x\|_1}{\|x\|_1} \leq \kappa(A)\frac{\|E\|_1}{\|A\|_1}$$

where $\kappa(A) = \left\|A^{-1}\right\|_1 \|A\|_1$, the condition number of A with respect to the solution of the linear equations. See Section 4.4 of Anderson *et al.* (1999) for further details.

Following the use of F07ANF (ZGESV), F07AUF (ZGECON) can be used to estimate the condition number of A and F07AVF (ZGERFS) can be used to obtain approximate error bounds. Alternatives to F07ANF (ZGESV), which return condition and error estimates directly are F04CAF and F07APF (ZGESVX).

8 Further Comments

The total number of floating point operations is approximately $\frac{8}{3}n^3 + 8n^2r$, where r is the number of right-hand sides.

The real analogue of this routine is F07AAF (DGESV).

9 Example

This example solves the equations

$$Ax = b,$$

where A is the general matrix

$$A = \begin{pmatrix} -1.34 + 2.55i & 0.28 + 3.17i & -6.39 - 2.20i & 0.72 - 0.92i \\ -0.17 - 1.41i & 3.31 - 0.15i & -0.15 + 1.34i & 1.29 + 1.38i \\ -3.29 - 2.39i & -1.91 + 4.42i & -0.14 - 1.35i & 1.72 + 1.35i \\ 2.41 + 0.39i & -0.56 + 1.47i & -0.83 - 0.69i & -1.96 + 0.67i \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 26.26 + 51.78i \\ 6.43 - 8.68i \\ -5.75 + 25.31i \\ 1.16 + 2.57i \end{pmatrix}.$$

Details of the LU factorization of A are also output.

9.1 Program Text

```
    Program f07anfe

!     F07ANF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04dbf, zgesv
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter              :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                         :: i, ifail, info, lda, ldb, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), b(:)
      Integer, Allocatable            :: ipiv(:)
      Character (1)                   :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F07ANF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      ldb = n
      Allocate (a(lda,n),b(ldb),ipiv(n))

!     Read A and B from data file

      Read (nin,*)(a(i,1:n),i=1,n)
      Read (nin,*) b(1:n)

!     Solve the equations Ax = b for x

!     The NAG name equivalent of zgesv is f07anf
      Call zgesv(n,1,a,lda,ipiv,b,ldb,info)
```

```
      If (info==0) Then

!       Print solution

        Write (nout,*) 'Solution'
        Write (nout,99999) b(1:n)

!       Print details of factorization

        Write (nout,*)
        Flush (nout)

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04dbf('General',' ',n,n,a,lda,'Bracketed','F7.4', &
          'Details of factorization','Integer',rlabs,'Integer',clabs,80,0, &
          ifail)

!       Print pivot indices

        Write (nout,*)
        Write (nout,*) 'Pivot indices'
        Write (nout,99998) ipiv(1:n)
      Else
        Write (nout,99997) 'The (', info, ',', info, ')', &
          ' element of the factor U is zero'
      End If

99999 Format ((3X,4(' (',F7.4,',',F7.4,')':)))
99998 Format (1X,7I11)
99997 Format (1X,A,I3,A,I3,A,A)
    End Program f07anfe
```

9.2 Program Data

```
F07ANF Example Program Data

   4                                                         :Value of N

 (-1.34, 2.55) ( 0.28, 3.17) (-6.39,-2.20) ( 0.72,-0.92)
 (-0.17,-1.41) ( 3.31,-0.15) (-0.15, 1.34) ( 1.29, 1.38)
 (-3.29,-2.39) (-1.91, 4.42) (-0.14,-1.35) ( 1.72, 1.35)
 ( 2.41, 0.39) (-0.56, 1.47) (-0.83,-0.69) (-1.96, 0.67)  :End of matrix A

 (26.26,51.78) ( 6.43,-8.68) (-5.75,25.31) ( 1.16, 2.57)  :End of vector b
```

9.3 Program Results

```
 F07ANF Example Program Results

 Solution
    ( 1.0000, 1.0000) ( 2.0000,-3.0000) (-4.0000,-5.0000) ( 0.0000, 6.0000)

 Details of factorization
                   1                  2                  3                  4
 1  (-3.2900,-2.3900) (-1.9100, 4.4200) (-0.1400,-1.3500) ( 1.7200, 1.3500)
 2  ( 0.2376, 0.2560) ( 4.8952,-0.7114) (-0.4623, 1.6966) ( 1.2269, 0.6190)
 3  (-0.1020,-0.7010) (-0.6691, 0.3689) (-5.1414,-1.1300) ( 0.9983, 0.3850)
 4  (-0.5359, 0.2707) (-0.2040, 0.8601) ( 0.0082, 0.1211) ( 0.1482,-0.1252)

 Pivot indices
           3          2          3          4
```