# NAG Toolbox

# nag_lapack_zgesv (f07an)

## 1 Purpose

nag_lapack_zgesv (f07an) 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 Syntax

```
[a, ipiv, b, info] = nag_lapack_zgesv(a, b, 'n', n, 'nrhs_p', nrhs_p)
[a, ipiv, b, info] = f07an(a, b, 'n', n, 'nrhs_p', nrhs_p)
```

## 3 Description

nag_lapack_zgesv (f07an) 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

### 5.1 Compulsory Input Parameters

1: **a**($lda$, :) – COMPLEX (KIND=nag_wp) array

The first dimension of the array **a** must be at least max(1, **n**).

The second dimension of the array **a** must be at least max(1, **n**).

The $n$ by $n$ coefficient matrix $A$.

2: **b**($ldb$, :) – COMPLEX (KIND=nag_wp) array

The first dimension of the array **b** must be at least max(1, **n**).

The second dimension of the array **b** must be at least max(1, **nrhs_p**).

The $n$ by $r$ right-hand side matrix $B$.

### 5.2 Optional Input Parameters

1: **n** – INTEGER

*Default*: the first dimension of the array **b**.

$n$, the number of linear equations, i.e., the order of the matrix $A$.

*Constraint*: $\mathbf{n} \geq 0$.

2: **nrhs_p** – INTEGER

*Default*: the second dimension of the array **b**.

$r$, the number of right-hand sides, i.e., the number of columns of the matrix $B$.

*Constraint*: $\mathbf{nrhs\_p} \geq 0$.

### 5.3 Output Parameters

1: **a**$(lda,:)$ – COMPLEX (KIND=nag_wp) array

The first dimension of the array **a** will be $\max(1,\mathbf{n})$.

The second dimension of the array **a** will be $\max(1,\mathbf{n})$.

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

2: **ipiv**$(\mathbf{n})$ – INTEGER array

If no constraints are violated, the pivot indices that define the permutation matrix $P$; at the $i$th step row $i$ of the matrix was interchanged with row $\mathbf{ipiv}(i)$. $\mathbf{ipiv}(i) = i$ indicates a row interchange was not required.

3: **b**$(ldb,:)$ – COMPLEX (KIND=nag_wp) array

The first dimension of the array **b** will be $\max(1,\mathbf{n})$.

The second dimension of the array **b** will be $\max(1,\mathbf{nrhs\_p})$.

If $\mathbf{info} = 0$, the $n$ by $r$ solution matrix $X$.

4: **info** – INTEGER

$\mathbf{info} = 0$ unless the function detects an error (see Section 6).

## 6 Error Indicators and Warnings

**info** $< 0$

If $\mathbf{info} = -i$, argument $i$ had an illegal value. An explanatory message is output, and execution of the program is terminated.

**info** $> 0$ (*warning*)

Element $\langle value \rangle$ of the diagonal 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 $\epsilon$ 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 nag_lapack_zgesv (f07an), nag_lapack_zgecon (f07au) can be used to estimate the condition number of $A$ and nag_lapack_zgerfs (f07av) can be used to obtain approximate error bounds. Alternatives to nag_lapack_zgesv (f07an), which return condition and error estimates directly are nag_linsys_complex_square_solve (f04ca) and nag_lapack_zgesvx (f07ap).

## 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 function is nag_lapack_dgesv (f07aa).

## 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

```
function f07an_example

fprintf('f07an example results\n\n');

% Linear system Ax = B
a = [ -1.34 + 2.55i,  0.28 + 3.17i,  -6.39 - 2.2i,   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];
b = [ 26.26 + 51.78i;
       6.43 -  8.68i;
      -5.75 + 25.31i;
       1.16 +  2.57i];

[LU, ipiv, x, info] = f07an(a, b);

disp('Solution');
disp(x');
disp('Details of factorization');
disp(LU);
disp('Pivot indices');
disp(double(ipiv'));
```

## 9.2 Program Results

```
     f07an example results

Solution
   1.0000 - 1.0000i   2.0000 + 3.0000i  -4.0000 + 5.0000i   0.0000 - 6.0000i

Details of factorization
  -3.2900 - 2.3900i  -1.9100 + 4.4200i  -0.1400 - 1.3500i   1.7200 + 1.3500i
   0.2376 + 0.2560i   4.8952 - 0.7114i  -0.4623 + 1.6966i   1.2269 + 0.6190i
  -0.1020 - 0.7010i  -0.6691 + 0.3689i  -5.1414 - 1.1300i   0.9983 + 0.3850i
  -0.5359 + 0.2707i  -0.2040 + 0.8601i   0.0082 + 0.1211i   0.1482 - 0.1252i

Pivot indices
     3     2     3     4
```