NAG Toolbox

nag_lapack_zgtsv (f07cn)

1 Purpose

nag_lapack_zgtsv (f07cn) computes the solution to a complex system of linear equations

$$AX = B,$$

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

2 Syntax

```
[dl, d, du, b, info] = nag_lapack_zgtsv(dl, d, du, b, 'n', n, 'nrhs_p', nrhs_p)
[dl, d, du, b, info] = f07cn(dl, d, du, b, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

nag_lapack_zgtsv (f07cn) uses Gaussian elimination with partial pivoting and row interchanges to solve the equations $AX = B$. The matrix A is factorized as $A = PLU$, where P is a permutation matrix, L is unit lower triangular with at most one nonzero subdiagonal element per column, and U is an upper triangular band matrix, with two superdiagonals.

Note that the equations $A^{\mathrm{T}}X = B$ may be solved by interchanging the order of the arguments **du** and **dl**.

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

5 Parameters

5.1 Compulsory Input Parameters

1: **dl**(:) – COMPLEX (KIND=nag_wp) array

The dimension of the array **dl** must be at least $\max(1, \mathbf{n} - 1)$

Must contain the $(n - 1)$ subdiagonal elements of the matrix A.

2: **d**(:) – COMPLEX (KIND=nag_wp) array

The dimension of the array **d** must be at least $\max(1, \mathbf{n})$

Must contain the n diagonal elements of the matrix A.

3: **du**(:) – COMPLEX (KIND=nag_wp) array

The dimension of the array **du** must be at least $\max(1, \mathbf{n} - 1)$

Must contain the $(n - 1)$ superdiagonal elements of the matrix A.

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

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

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

Note: to solve the equations $Ax = b$, where b is a single right-hand side, **b** may be supplied as a one-dimensional array with length $ldb = \max(1, \mathbf{n})$.

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** and the dimension of the array **d**.

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: **dl**(:) – COMPLEX (KIND=nag_wp) array

The dimension of the array **dl** will be $\max(1, \mathbf{n} - 1)$

If no constraints are violated, **dl** stores the $(n - 2)$ elements of the second superdiagonal of the upper triangular matrix U from the LU factorization of A, in $\mathbf{dl}(1), \mathbf{dl}(2), \ldots, \mathbf{dl}(n - 2)$.

2: **d**(:) – COMPLEX (KIND=nag_wp) array

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

If no constraints are violated, **d** stores the n diagonal elements of the upper triangular matrix U from the LU factorization of A.

3: **du**(:) – COMPLEX (KIND=nag_wp) array

The dimension of the array **du** will be $\max(1, \mathbf{n} - 1)$

If no constraints are violated, **du** stores the $(n - 1)$ elements of the first superdiagonal of U.

4: **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 $\max(1, \mathbf{nrhs\_p})$.

Note: to solve the equations $Ax = b$, where b is a single right-hand side, **b** may be supplied as a one-dimensional array with length $ldb = \max(1, \mathbf{n})$.

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

5: **info** – INTEGER

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

6 Error Indicators and Warnings

$\mathbf{info} < 0$

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, and the solution has not been computed. The factorization has not been completed unless $\mathbf{n} = \langle value \rangle$.

7 Accuracy

The computed solution for a single right-hand side, $\hat{x}$, satisfies an 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.

Alternatives to nag_lapack_zgtsv (f07cn), which return condition and error estimates are nag_linsys_complex_tridiag_solve (f04cc) and nag_lapack_zgtsvx (f07cp).

8 Further Comments

The total number of floating-point operations required to solve the equations $AX = B$ is proportional to nr.

The real analogue of this function is nag_lapack_dgtsv (f07ca).

9 Example

This example solves the equations

$$Ax = b,$$

where A is the tridiagonal matrix

$$A = \begin{pmatrix} -1.3+1.3i & 2.0-1.0i & 0 & 0 & 0 \\ 1.0-2.0i & -1.3+1.3i & 2.0+1.0i & 0 & 0 \\ 0 & 1.0+1.0i & -1.3+3.3i & -1.0+1.0i & 0 \\ 0 & 0 & 2.0-3.0i & -0.3+4.3i & 1.0-1.0i \\ 0 & 0 & 0 & 1.0+1.0i & -3.3+1.3i \end{pmatrix}$$

and

$$b = \begin{pmatrix} 2.4 - 5.0i \\ 3.4 + 18.2i \\ -14.7 + 9.7i \\ 31.9 - 7.7i \\ -1.0 + 1.6i \end{pmatrix}.$$

9.1 Program Text

```
     function f07cn_example

fprintf('f07cn example results\n\n');

% Tridiagonal matrix stored by diagonals
du = [              2   - 1i     2   + 1i    -1   + 1i     1   - 1i  ];
d  = [-1.3 + 1.3i  -1.3 + 1.3i  -1.3 + 3.3i  -0.3 + 4.3i  -3.3 + 1.3i];
dl = [ 1   - 2i     1   + 1i     2   - 3i     1   + 1i               ];
```

```
% Rhs B
b = [  2.4 -  5.0i;
       3.4 + 18.2i;
     -14.7 +  9.7i;
      31.9 -  7.7i;
      -1   +  1.6i];

% Solve for x
[dl, d, du, x, info] = f07cn( ...
                              dl, d, du, b);

disp('Solution');
disp(x);
```

9.2 Program Results

```
     f07cn example results

Solution
   1.0000 + 1.0000i
   3.0000 - 1.0000i
   4.0000 + 5.0000i
  -1.0000 - 2.0000i
   1.0000 - 1.0000i
```