NAG Toolbox

nag_lapack_zpttrs (f07js)

1 Purpose

nag_lapack_zpttrs (f07js) computes the solution to a complex system of linear equations $AX = B$, where A is an n by n Hermitian positive definite tridiagonal matrix and X and B are n by r matrices, using the LDL^{H} factorization returned by nag_lapack_zpttrf (f07jr).

2 Syntax

```
[b, info] = nag_lapack_zpttrs(uplo, d, e, b, 'n', n, 'nrhs_p', nrhs_p)
[b, info] = f07js(uplo, d, e, b, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

nag_lapack_zpttrs (f07js) should be preceded by a call to nag_lapack_zpttrf (f07jr), which computes a modified Cholesky factorization of the matrix A as

$$A = LDL^{\mathrm{H}},$$

where L is a unit lower bidiagonal matrix and D is a diagonal matrix, with positive diagonal elements. nag_lapack_zpttrs (f07js) then utilizes the factorization to solve the required equations. Note that the factorization may also be regarded as having the form $U^{\mathrm{H}}DU$, where U is a unit upper bidiagonal matrix.

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: **uplo** – CHARACTER(1)

Specifies the form of the factorization as follows:

uplo = 'U'
$A = U^{\mathrm{H}}DU$.

uplo = 'L'
$A = LDL^{\mathrm{H}}$.

Constraint: **uplo** = 'U' or 'L'.

2: **d**(:) – REAL (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 diagonal matrix D from the LDL^{H} or $U^{\mathrm{H}}DU$ factorization of A.

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

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

If **uplo** = 'U', **e** must contain the $(n-1)$ superdiagonal elements of the unit upper bidiagonal matrix U from the $U^{\mathrm{H}}DU$ factorization of A.

If **uplo** = 'L', **e** must contain the $(n-1)$ subdiagonal elements of the unit lower bidiagonal matrix L from the LDL^{H} factorization of 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 $\max(1, \mathbf{nrhs\_p})$.

The n by r matrix of right-hand sides 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 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: **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 $\max(1, \mathbf{nrhs\_p})$.

The n by r solution matrix X.

2: **info** – INTEGER

info = 0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

info < 0

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

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.

Following the use of this function nag_lapack_zptcon (f07ju) can be used to estimate the condition number of A and nag_lapack_zptrfs (f07jv) can be used to obtain approximate error bounds.

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_dpttrs (f07je).

9 Example

This example solves the equations

$$AX = B,$$

where A is the Hermitian positive definite tridiagonal matrix

$$A = \begin{pmatrix} 16.0 & 16.0-16.0i & 0 & 0 \\ 16.0+16.0i & 41.0 & 18.0+9.0i & 0.0 \\ 0 & 18.0-9.0i & 46.0 & 1.0+4.0i \\ 0 & 0 & 1.0-4.0i & 21.0 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 64.0+16.0i & -16.0-32.0i \\ 93.0+62.0i & 61.0-66.0i \\ 78.0-80.0i & 71.0-74.0i \\ 14.0-27.0i & 35.0+15.0i \end{pmatrix}.$$

9.1 Program Text

```
function f07js_example

fprintf('f07js example results\n\n');

% Hermitian tridiagonal A stored as two diagonals
d = [ 16            41          46            21];
e = [ 16 + 16i      18 - 9i      1 - 4i         ];

% Factorize
[df, ef, info] = f07jr( ...
                        d, e);

% RHS
b = [ 64 + 16i,  -16 - 32i;
      93 + 62i,   61 - 66i;
      78 - 80i,   71 - 74i;
      14 - 27i,   35 + 15i];

% Solve
uplo = 'L';
[x, info] = f07js( ...
                   uplo, df, ef, b);

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

9.2 Program Results

```
     f07js example results

Solution(s)
   2.0000 + 1.0000i  -3.0000 - 2.0000i
   1.0000 + 1.0000i   1.0000 + 1.0000i
   1.0000 - 2.0000i   1.0000 - 2.0000i
   1.0000 - 1.0000i   2.0000 + 1.0000i
```