NAG Toolbox

nag_lapack_dpttrs (f07je)

1 Purpose

nag_lapack_dpttrs (f07je) computes the solution to a real system of linear equations $AX = B$, where A is an n by n symmetric positive definite tridiagonal matrix and X and B are n by r matrices, using the LDL^{T} factorization returned by nag_lapack_dpttrf (f07jd).

2 Syntax

```
[b, info] = nag_lapack_dpttrs(d, e, b, 'n', n, 'nrhs_p', nrhs_p)
[b, info] = f07je(d, e, b, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

nag_lapack_dpttrs (f07je) should be preceded by a call to nag_lapack_dpttrf (f07jd), which computes a modified Cholesky factorization of the matrix A as

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

where L is a unit lower bidiagonal matrix and D is a diagonal matrix, with positive diagonal elements. nag_lapack_dpttrs (f07je) then utilizes the factorization to solve the required equations. Note that the factorization may also be regarded as having the form $U^{\mathrm{T}}DU$, 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: **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^{T} factorization of A.

2: **e**(:) – REAL (KIND=nag_wp) array

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

Must contain the $(n - 1)$ subdiagonal elements of the unit lower bidiagonal matrix L. (**e** can also be regarded as the superdiagonal of the unit upper bidiagonal matrix U from the $U^{\mathrm{T}}DU$ factorization of A.)

3: **b**(ldb, :) – REAL (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**$(ldb, :)$ – REAL (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

$\mathbf{info} = 0$ 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.

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 $\kappa(A) = \|A^{-1}\|_1\|A\|_1$, 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_dptcon (f07jg) can be used to estimate the condition number of A and nag_lapack_dptrfs (f07jh) 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 complex analogue of this function is nag_lapack_zpttrs (f07js).

9 Example

This example solves the equations

$$AX = B,$$

where A is the symmetric positive definite tridiagonal matrix

$$A = \begin{pmatrix} 4.0 & -2.0 & 0 & 0 & 0 \\ -2.0 & 10.0 & -6.0 & 0 & 0 \\ 0 & -6.0 & 29.0 & 15.0 & 0 \\ 0 & 0 & 15.0 & 25.0 & 8.0 \\ 0 & 0 & 0 & 8.0 & 5.0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 6.0 & 10.0 \\ 9.0 & 4.0 \\ 2.0 & 9.0 \\ 14.0 & 65.0 \\ 7.0 & 23.0 \end{pmatrix}.$$

9.1 Program Text

```
function f07je_example

fprintf('f07je example results\n\n');

% Symmetric tridiagonal A stored as two diagonals
d = [ 4     10     29     25     5];
e = [-2     -6     15     8       ];

% RHS
b = [ 6, 10;
      9,  4;
      2,  9;
     14, 65;
      7, 23];

% Factorize
[df, ef, info] = f07jd( ...
                        d, e);

%Solve
[x, info] = f07je( ...
                   df, ef, b);

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

9.2 Program Results

```
f07je example results

Solution(s)
    2.5000    2.0000
    2.0000   -1.0000
    1.0000   -3.0000
   -1.0000    6.0000
    3.0000   -5.0000
```