

NAG Toolbox

nag_matop_real_gen_tridiag_lu (f01le)

1 Purpose

nag_matop_real_gen_tridiag_lu (f01le) computes an LU factorization of a real tridiagonal matrix, using Gaussian elimination with partial pivoting.

2 Syntax

```
[a, b, c, d, ipiv, ifail] = nag_matop_real_gen_tridiag_lu(a, lambda, b, c, tol,
'n', n)
[a, b, c, d, ipiv, ifail] = f01le(a, lambda, b, c, tol, 'n', n)
```

3 Description

The matrix $T - \lambda I$, where T is a real n by n tridiagonal matrix, is factorized as

$$T - \lambda I = PLU,$$

where P is a permutation matrix, L is a unit lower triangular matrix with at most one nonzero subdiagonal element per column, and U is an upper triangular matrix with at most two nonzero superdiagonal elements per column.

The factorization is obtained by Gaussian elimination with partial pivoting and implicit row scaling.

An indication of whether or not the matrix $T - \lambda I$ is nearly singular is returned in the n th element of the array **ipiv**. If it is important that $T - \lambda I$ is nonsingular, as is usually the case when solving a system of tridiagonal equations, then it is strongly recommended that **ipiv**(n) is inspected on return from nag_matop_real_gen_tridiag_lu (f01le). (See the argument **ipiv** and Section 9 for further details.)

The argument λ is included in the function so that nag_matop_real_gen_tridiag_lu (f01le) may be used, in conjunction with nag_linsys_real_tridiag_fac_solve (f04le), to obtain eigenvectors of T by inverse iteration.

4 References

Wilkinson J H (1965) *The Algebraic Eigenvalue Problem* Oxford University Press, Oxford

Wilkinson J H and Reinsch C (1971) *Handbook for Automatic Computation II, Linear Algebra* Springer-Verlag

5 Parameters

5.1 Compulsory Input Parameters

- 1: **a**(**n**) – REAL (KIND=nag_wp) array
The diagonal elements of T .
- 2: **lambda** – REAL (KIND=nag_wp)
The scalar λ . nag_matop_real_gen_tridiag_lu (f01le) factorizes $T - \lambda I$.
- 3: **b**(**n**) – REAL (KIND=nag_wp) array
The superdiagonal elements of T , stored in **b**(2) to **b**(n); **b**(1) is not used.

4: **c**(**n**) – REAL (KIND=nag_wp) array

The subdiagonal elements of T , stored in **c**(2) to **c**(n); **c**(1) is not used.

5: **tol** – REAL (KIND=nag_wp)

A relative tolerance used to indicate whether or not the matrix $(T - \lambda I)$ is nearly singular. **tol** should normally be chosen as approximately the largest relative error in the elements of T . For example, if the elements of T are correct to about 4 significant figures, then **tol** should be set to about 5×10^{-4} . See Section 9 for further details on how **tol** is used. If **tol** is supplied as less than ϵ , where ϵ is the *machine precision*, then the value ϵ is used in place of **tol**.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the dimension of the arrays **a**, **b**, **c**. (An error is raised if these dimensions are not equal.)

n , the order of the matrix T .

Constraint: $n \geq 1$.

5.3 Output Parameters

1: **a**(**n**) – REAL (KIND=nag_wp) array

The diagonal elements of the upper triangular matrix U .

2: **b**(**n**) – REAL (KIND=nag_wp) array

The elements of the first superdiagonal of U , stored in **b**(2) to **b**(n).

3: **c**(**n**) – REAL (KIND=nag_wp) array

The subdiagonal elements of L , stored in **c**(2) to **c**(n).

4: **d**(**n**) – REAL (KIND=nag_wp) array

The elements of the second superdiagonal of U , stored in **d**(3) to **d**(n); **d**(1) and **d**(2) are not used.

5: **ipiv**(**n**) – INTEGER array

Details of the permutation matrix P . If an interchange occurred at the k th step of the elimination, then **ipiv**(k) = 1, otherwise **ipiv**(k) = 0. If a diagonal element of U is small, indicating that $(T - \lambda I)$ is nearly singular, then the element **ipiv**(n) is returned as positive. Otherwise **ipiv**(n) is returned as 0. See Section 9 for further details. If the application is such that it is important that $(T - \lambda I)$ is not nearly singular, then it is strongly recommended that **ipiv**(n) is inspected on return.

6: **ifail** – INTEGER

ifail = 0 unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

On entry, $n < 1$.

ifail = –99

An unexpected error has been triggered by this routine. Please contact NAG.

ifail = –399

Your licence key may have expired or may not have been installed correctly.

ifail = –999

Dynamic memory allocation failed.

7 Accuracy

The computed factorization will satisfy the equation

$$PLU = (T - \lambda I) + E,$$

where

$$\|E\|_1 \leq 9 \times \max_{i \geq j} (|l_{ij}|, |l_{ij}|^2) \epsilon \|T - \lambda I\|_1$$

where ϵ is the *machine precision*.

8 Further Comments

The time taken by `nag_matop_real_gen_tridiag_lu` (f01le) is approximately proportional to n .

The factorization of a tridiagonal matrix proceeds in $(n - 1)$ steps, each step eliminating one subdiagonal element of the tridiagonal matrix. In order to avoid small pivot elements and to prevent growth in the size of the elements of L , rows k and $(k + 1)$ will, if necessary, be interchanged at the k th step prior to the elimination.

The element `ipiv`(n) returns the smallest integer, j , for which

$$|u_{jj}| \leq \left\| (T - \lambda I)_j \right\|_1 \times \mathbf{tol},$$

where $\left\| (T - \lambda I)_j \right\|_1$ denotes the sum of the absolute values of the j th row of the matrix $(T - \lambda I)$. If no such j exists, then `ipiv`(n) is returned as zero. If such a j exists, then $|u_{jj}|$ is small and hence $(T - \lambda I)$ is singular or nearly singular.

This function may be followed by `nag_linsys_real_tridiag_fac_solve` (f04le) to solve systems of tridiagonal equations. If you wish to solve single systems of tridiagonal equations you should be aware of `nag_lapack_dgtsv` (f07ca), which solves tridiagonal systems with a single call. `nag_lapack_dgtsv` (f07ca) requires less storage and will generally be faster than the combination of `nag_matop_real_gen_tridiag_lu` (f01le) and `nag_linsys_real_tridiag_fac_solve` (f04le), but no test for near singularity is included in `nag_lapack_dgtsv` (f07ca) and so it should only be used when the equations are known to be nonsingular.

9 Example

This example factorizes the tridiagonal matrix T where

$$T = \begin{pmatrix} 3.0 & 2.1 & 0 & 0 & 0 \\ 3.4 & 2.3 & -1.0 & 0 & 0 \\ 0 & 3.6 & -5.0 & 1.9 & 0 \\ 0 & 0 & 7.0 & -0.9 & 8.0 \\ 0 & 0 & 0 & -6.0 & 7.1 \end{pmatrix}$$

and then to solve the equations $Tx = y$, where

$$y = \begin{pmatrix} 2.7 \\ -0.5 \\ 2.6 \\ 0.6 \\ 2.7 \end{pmatrix}$$

by a call to `nag_linsys_real_tridiag_fac_solve` (f04le). The example program sets `tol` = 5×10^{-5} and, of course, sets `lambda` = 0.

9.1 Program Text

```
function f01le_example

fprintf('f01le example results\n\n');

% Tridiagonal matrix T stored by diagonals
b = [0 2.1 -1 1.9 8];
a = [3 2.3 -5 -0.9 7.1];
c = [0 3.4 3.6 7 -6];

% LU factorize T
lambda = 0;
tol = 5e-05;
[D, U1, L, U2, ipiv, ifail] = f01le( ...
    a, lambda, b, c, tol);

fprintf('Details of factorization\n\n');
disp(' Main diagonal of U');
disp(D);
disp(' First super-diagonal of U');
disp(U1(2:end));
disp(' Second super-diagonal of U');
disp(U2(3:end)');
disp(' Multipliers');
disp(L(2:end));
disp(' Vector of interchanges');
disp(double(ipiv(1:end-1))');

% Solve Tx = y, where
y = [2.7 -0.5 2.6 0.6 2.7 ];
% using factorized form of T
job = nag_int(1);

[x, ~, ifail] = f04le( ...
    job, D, U1, L, U2, ipiv, y, tol);

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

9.2 Program Results

```
f01le example results

Details of factorization

Main diagonal of U
  3.0000   3.6000   7.0000  -6.0000   1.1508

First super-diagonal of U
  2.1000  -5.0000  -0.9000   7.1000

Second super-diagonal of U
    0   1.9000   8.0000

Multipliers
  1.1333  -0.0222  -0.1587   0.0168
```

Vector of interchanges
0 1 1 1

Solution vector:
-4.0000 7.0000 3.0000 -4.0000 -3.0000
