NAG Toolbox for MATLAB

f01le

1 Purpose

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

2 Syntax

```
[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 nth 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 f01le. (See the parameter **ipiv** and Section 8 for further details.)

The parameter λ is included in the function so that f01le may be used, in conjunction with 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**) – **double array**

The diagonal elements of T.

2: **lambda – double scalar**

The scalar λ. f01le factorizes $T - \lambda I$.

3: **b**(**n**) – **double array**

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

4: **c**(**n**) – **double array**

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

5: **tol – double scalar**

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 8 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 – int32 scalar**

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

n, n, the order of the matrix T.

Constraint: $\mathbf{n} \geq 1$.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **a(n) – double array**

The diagonal elements of the upper triangular matrix U.

2: **b(n) – double array**

The elements of the first superdiagonal of U, stored in $\mathbf{b}(2)$ to $\mathbf{b}(n)$.

3: **c(n) – double array**

The subdiagonal elements of L, stored in $\mathbf{c}(2)$ to $\mathbf{c}(n)$.

4: **d(n) – double array**

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

5: **ipiv(n) – int32 array**

Details of the permutation matrix P. If an interchange occurred at the kth step of the elimination, then $\mathbf{ipiv}(k) = 1$, otherwise $\mathbf{ipiv}(k) = 0$. If a diagonal element of U is small, indicating that $(T - \lambda I)$ is nearly singular, then the element $\mathbf{ipiv}(n)$ is returned as positive. Otherwise $\mathbf{ipiv}(n)$ is returned as 0. See Section 8 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 $\mathbf{ipiv}(n)$ is inspected on return.

6: **ifail – int32 scalar**

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, $\mathbf{n} < 1$.

7 Accuracy

The computed factorization will satisfy the equation

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

where

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

where ϵ is the ***machine precision***.

8 Further Comments

The time taken by 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 kth 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 jth 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 f04le to solve systems of tridiagonal equations. If you wish to solve single systems of tridiagonal equations you should be aware of f07ca, which solves tridiagonal systems with a single call. f07ca requires less storage and will generally be faster than the combination of f01le and f04le, but no test for near singularity is included in f07ca and so it should only be used when the equations are known to be nonsingular.

9 Example

```
a = [3;
     2.3;
     -5;
     -0.9;
     7.1];
lambda = 0;
b = [0;
     2.1;
     -1;
     1.9;
     8];
c = [0;
     3.4;
     3.6;
     7;
     -6];
tol = 5e-05;
[aOut, bOut, cOut, d, ipiv, ifail] = f01le(a, lambda, b, c, tol)
```

```
aOut =
    3.0000
    3.6000
    7.0000
   -6.0000
    1.1508
bOut =
```

```
         0
    2.1000
   -5.0000
   -0.9000
    7.1000
cOut =
         0
    1.1333
   -0.0222
   -0.1587
    0.0168
d =
         0
         0
         0
    1.9000
    8.0000
ipiv =
           0
           1
           1
           1
           0
ifail =
           0
```