NAG Toolbox for MATLAB

f04le

1 Purpose

f04le solves a system of tridiagonal equations following the factorization by f01le. This function is intended for applications such as inverse iteration as well as straightforward linear equation applications.

2 Syntax

```
[y, tol, ifail] = f04le(job, a, b, c, d, ipiv, y, tol, 'n', n)
```

3 Description

Following the factorization of the n by n tridiagonal matrix $(T - \lambda I)$ as

$$T - \lambda I = PLU$$

by f01le, f04le may be used to solve any of the equations

$$(T - \lambda I)x = y, \qquad (T - \lambda I)^{\mathrm{T}}x = y, \qquad Ux = y$$

for x, the choice of equation being controlled by the parameter **job**. In each case there is an option to perturb zero or very small diagonal elements of U, this option being intended for use in applications such as 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: **job – int32 scalar**

Must specify the equations to be solved.

job $= 1$

The equations $(T - \lambda I)x = y$ are to be solved, but diagonal elements of U are not to be perturbed.

job $= -1$

The equations $(T - \lambda I)x = y$ are to be solved and, if overflow would otherwise occur, diagonal elements of U are to be perturbed. See parameter **tol**.

job $= 2$

The equations $(T - \lambda I)^{\mathrm{T}}x = y$ are to be solved, but diagonal elements of U are not to be perturbed.

job $= -2$

The equations $(T - \lambda I)^{\mathrm{T}}x = y$ are to be solved and, if overflow would otherwise occur, diagonal elements of U are to be perturbed. See parameter **tol**.

job $= 3$

The equations $Ux = y$ are to be solved, but diagonal elements of U are not to be perturbed.

job $= -3$

The equations $Ux = y$ are to be solved and, if overflow would otherwise occur, diagonal elements of U are to be perturbed. See parameter **tol**.

2: **a**(**n**) – **double array**

The diagonal elements of U as returned by f01le.

3: **b**(**n**) – **double array**

The elements of the first superdiagonal of U as returned by f01le.

4: **c**(**n**) – **double array**

The subdiagonal elements of L as returned by f01le.

5: **d**(**n**) – **double array**

The elements of the second superdiagonal of U as returned by f01le.

6: **ipiv**(**n**) – **int32 array**

Details of the matrix P as returned by f01le.

7: **y**(**n**) – **double array**

The right-hand side vector y.

8: **tol** – **double scalar**

The minimum perturbation to be made to very small diagonal elements of U. **tol** is only referenced when **job** is negative. **tol** should normally be chosen as about $\epsilon\|U\|$, where ϵ is the ***machine precision***, but if **tol** is supplied as nonpositive, then it is reset to $\epsilon\max|u_{ij}|$.

5.2 Optional Input Parameters

1: **n** – **int32 scalar**

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

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: **y(n) – double array**

The array contains the solution vector x.

2: **tol – double scalar**

If on entry **tol** is nonpositive, it is reset as just described. Otherwise **tol** is unchanged.

3: **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, $\mathbf{n} < 1$,
or $\mathbf{job} = 0$,
or $\mathbf{job} < -3$ or $\mathbf{job} > 3$.

ifail > 1

Overflow would occur when computing the (**ifail** $- 1$)th element of the solution vector x. This can only occur when **job** is supplied as positive and either means that a diagonal element of U is very small or that elements of the right-hand side vector y are very large.

7 Accuracy

The computed solution of the equations $(T - \lambda I)x = y$, say $\bar{x}$, will satisfy an equation of the form

$$(T - \lambda I + E)\bar{x} = y,$$

where E can be expected to satisfy a bound of the form

$$\|E\| \leq \alpha\epsilon\|T - \lambda I\|,$$

α being a modest constant and ϵ being the ***machine precision***. The computed solution of the equations $(T - \lambda I)^{\mathrm{T}}x = y$ and $Ux = y$ will satisfy similar results. The above result implies that the relative error in $\bar{x}$ satisfies

$$\frac{\|\bar{x} - x\|}{\|\bar{x}\|} \leq c(T - \lambda I)\alpha\epsilon,$$

where $c(T - \lambda I)$ is the condition number of $(T - \lambda I)$ with respect to inversion. Thus if $(T - \lambda I)$ is nearly singular, $\bar{x}$ can be expected to have a large relative error. Note that f01le incorporates a test for near singularity.

8 Further Comments

The time taken by f04le is approximately proportional to n.

If you have single systems of tridiagonal equations to solve you are advised that f07ca requires less storage and will normally be faster than the combination of f01le and f04le, but f07ca does not incorporate a test for near singularity.

9 Example

```
job = int32(1);
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;
ipiv = [int32(0);
     int32(1);
     int32(1);
     int32(1);
     int32(0)];
y = [2.7;
     -0.5;
     2.6;
     0.6;
     2.7];
tol = 5e-05;
[a, b, c, d, ipiv, ifail] = f01le(a, lambda, b, c, tol);
[yOut, tolOut, ifail] = f04le(job, a, b, c, d, ipiv, y, tol)
```

```
yOut =
   -4.0000
    7.0000
    3.0000
   -4.0000
   -3.0000
tolOut =
   5.0000e-05
ifail =
           0
```