NAG Toolbox for MATLAB

f04ea

1 Purpose

f04ea calculates the approximate solution of a set of real tridiagonal linear equations with a single right-hand side.

2 Syntax

```
[d, du, dl, b, ifail] = f04ea(d, du, dl, b, 'n', n)
```

3 Description

f04ea is based on the LINPACK routine SGTSL (see Dongarra *et al.* 1979) and solves the equation

$$Tx = b,$$

where T is a real n by n tridiagonal matrix, by Gaussian elimination with partial pivoting. This routine should only be used if T is known not to be nearly singular since the function only tests for exactly zero pivots.

4 References

Dongarra J J, Moler C B, Bunch J R and Stewart G W 1979 *LINPACK Users' Guide* SIAM, Philadelphia

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

5 Parameters

5.1 Compulsory Input Parameters

1: **d**(**n**) – **double array**

The diagonal elements of T.

2: **du**(**n**) – **double array**

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

3: **dl**(**n**) – **double array**

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

4: **b**(**n**) – **double array**

The right-hand side vector b.

5.2 Optional Input Parameters

1: **n** – **int32 scalar**

Default: The dimension of the arrays **d**, **du**, **dl**, **b**. (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: **d**(**n**) – **double array**

The diagonal elements of the upper triangular matrix, U, of the factorization of T.

2: **du**(**n**) – **double array**

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

3: **dl**(**n**) – **double array**

The elements of the second superdiagonal of U, stored in **dl**(3) to **dl**(n).

4: **b**(**n**) – **double array**

The array contains the solution vector x.

5: **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, **n** < 1.

ifail > 1

The (**ifail** $- 1$)th diagonal element of U is zero. Unless **ifail** $=$ **n** $+ 1$, the factorization of T will not have been completed.

overflow

If overflow occurs during execution of this function then either a diagonal element of U, while not being zero, is very small, or an element of b is very large. In the former case T is probably nearly singular and it would be advisable to use f01le, while in the latter case the equations should be scaled so that the elements of b are of more moderate size.

underflow

Any underflows which occur during the execution of this function are harmless, but if you who wish to avoid underflow and your installation returns the value **true** from x02da you may prefer to use the combination f01le and f04le to solve your tridiagonal equations.

7 Accuracy

The computed solution, say $\bar{x}$, of the equations $Tx = b$ will satisfy the equation

$$(T + E)\bar{x} = b,$$

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

$$\|E\| \leq \alpha\epsilon\|T\|,$$

α being a modest constant and ϵ being the ***machine precision***. The above result implies that the relative error in $\bar{x}$ satisfies the bound

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

where $c(T)$ is the condition number of T with respect to inversion. Thus if T is nearly singular, x can be expected to have a high relative error.

8 Further Comments

The time taken by f04ea is approximately proportional to n.

The tridiagonal equations $T^{\mathrm{T}}x = b$ may be solved by interchanging the parameters **du** and **dl** in the call to this function.

If you have multiple right-hand sides or are uncertain as to whether or not your matrix T is nearly singular, you are advised that the combination f01le and f04le can also be used to solve tridiagonal equations, but that this combination requires more storage and will usually be slower than f04ea.

9 Example

```
d = [3;
     2.3;
     -5;
     -0.9;
     7.1];
du = [-5.333924895381413e-39;
     2.1;
     -1;
     1.9;
     8];
dl = [-0.1175022125244143;
     3.4;
     3.6;
     7;
     -6];
b = [2.7;
     -0.5;
     2.6;
     0.6;
     2.7];
[dOut, duOut, dlOut, bOut, ifail] = f04ea(d, du, dl, b)
```

```
dOut =
    3.4000
    3.6000
    7.0000
   -6.0000
   -1.0154
duOut =
   -0.0000
    2.3000
   -5.0000
   -0.9000
    7.1000
dlOut =
   -0.1175
   -1.0000
    1.9000
    8.0000
   -6.0000
bOut =
   -4.0000
    7.0000
    3.0000
```

```
   -4.0000
   -3.0000
ifail =
           0
```