NAG Toolbox for MATLAB

e04bb

1 Purpose

e04bb searches for a minimum, in a given finite interval, of a continuous function of a single variable, using function and first derivative values. The method (based on cubic interpolation) is intended for functions which have a continuous first derivative (although it will usually work if the derivative has occasional discontinuities).

2 Syntax

```
[e1, e2, a, b, maxcal, x, f, g, user, ifail] = e04bb(funct, e1, e2, a,
b, maxcal, 'user', user)
```

3 Description

e04bb is applicable to problems of the form:

$$\text{Minimize}\, F(x) \qquad \text{subject to} \qquad a \leq x \leq b$$

when the first derivative $\frac{dF}{dx}$ can be calculated. The function normally computes a sequence of x values which tend in the limit to a minimum of $F(x)$ subject to the given bounds. It also progressively reduces the interval $[a, b]$ in which the minimum is known to lie. It uses the safeguarded cubic-interpolation method described in Gill and Murray 1973.

You must supply a user-supplied (sub)program **funct** to evaluate $F(x)$ and $\frac{dF}{dx}$. The parameters **e1** and **e2** together specify the accuracy

$$Tol(x) = \mathbf{e1} \times |x| + \mathbf{e2}$$

to which the position of the minimum is required. Note that **funct** is never called at a point which is closer than $Tol(x)$ to a previous point.

If the original interval $[a, b]$ contains more than one minimum, e04bb will normally find one of the minima.

4 References

Gill P E and Murray W 1973 Safeguarded steplength algorithms for optimization using descent methods *NPL Report NAC 37* National Physical Laboratory

5 Parameters

5.1 Compulsory Input Parameters

1: **funct – string containing name of m-file**

You must supply this function to calculate the values of $F(x)$ and $\frac{dF}{dx}$ at any point x in $[a, b]$.

It should be tested separately before being used in conjunction with e04bb.

```
[fc, gc, user] = funct(xc, user)
```

Input Parameters

1: **xc – double scalar**

The point x at which the values of F and $\frac{dF}{dx}$ are required.

2: **user – Any MATLAB object**

funct is called from e04bb with **user** as supplied to e04bb

Output Parameters

1: **fc – double scalar**

Must be set to the value of the function F at the current point x.

2: **gc – double scalar**

Must be set to the value of the first derivative $\frac{dF}{dx}$ at the current point x.

3: **user – Any MATLAB object**

funct is called from e04bb with **user** as supplied to e04bb

2: **e1 – double scalar**

The relative accuracy to which the position of a minimum is required. (Note that, since **e1** is a relative tolerance, the scaling of x is automatically taken into account.)

e1 should be no smaller than 2ϵ, and preferably not much less than $\sqrt{\epsilon}$, where ϵ is the ***machine precision***.

3: **e2 – double scalar**

The absolute accuracy to which the position of a minimum is required. **e2** should be no smaller than 2ϵ.

4: **a – double scalar**

The lower bound a of the interval containing a minimum.

5: **b – double scalar**

The upper bound b of the interval containing a minimum.

6: **maxcal – int32 scalar**

The maximum number of calls of user-supplied (sub)program **funct** to be allowed.

Constraint: **maxcal** ≥ 2. (Few problems will require more than 20.)

There will be an error exit (see Section 6) after **maxcal** calls of user-supplied (sub)program **funct**

5.2 Optional Input Parameters

1: **user – Any MATLAB object**

user is not used by e04bb, but is passed to **funct**. Note that for large objects it may be more efficient to use a global variable which is accessible from the m-files than to use **user**.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **e1 – double scalar**

If you set **e1** to 0.0 (or to any value less than ϵ), **e1** will be reset to the default value $\sqrt{\epsilon}$ before starting the minimization process.

2: **e2 – double scalar**

If you set **e2** to 0.0 (or to any value less than ϵ), **e2** will be reset to the default value $\sqrt{\epsilon}$.

3: **a – double scalar**

An improved lower bound on the position of the minimum.

4: **b – double scalar**

An improved upper bound on the position of the minimum.

5: **maxcal – int32 scalar**

The total number of times that user-supplied (sub)program **funct** was actually called.

6: **x – double scalar**

The estimated position of the minimum.

7: **f – double scalar**

The function value at the final point given in **x**.

8: **g – double scalar**

The value of the first derivative at the final point in **x**.

9: **user – Any MATLAB object**

user is not used by e04bb, but is passed to **funct**. Note that for large objects it may be more efficient to use a global variable which is accessible from the m-files than to use **user**.

10: **ifail – int32 scalar**

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

6 Error Indicators and Warnings

Note: e04bb may return useful information for one or more of the following detected errors or warnings.

ifail $= 1$

On entry, $(\mathbf{a} + \mathbf{e2}) \geq \mathbf{b}$,
or $\mathbf{maxcal} < 2$.

ifail $= 2$

The number of calls of user-supplied (sub)program **funct** has exceeded **maxcal**. This may have happened simply because **maxcal** was set too small for a particular problem, or may be due to a mistake in **funct**. If no mistake can be found in **funct**, restart e04bb (preferably with the values of **a** and **b** given on exit from the previous call of e04bb).

7 Accuracy

If $F(x)$ is δ-unimodal for some $\delta < Tol(x)$, where $Tol(x) = \mathbf{e1} \times |x| + \mathbf{e2}$, then, on exit, x approximates the minimum of $F(x)$ in the original interval $[a, b]$ with an error less than $3 \times Tol(x)$.

8 Further Comments

Timing depends on the behaviour of $F(x)$, the accuracy demanded and the length of the interval $[a, b]$. Unless $F(x)$ and $\frac{dF}{dx}$ can be evaluated very quickly, the run time will usually be dominated by the time spent in user-supplied (sub)program **funct**.

If $F(x)$ has more than one minimum in the original interval $[a, b]$, e04bb will determine an approximation x (and improved bounds a and b) for one of the minima.

If e04bb finds an x such that $F(x - \delta_1) > F(x) < F(x + \delta_2)$ for some $\delta_1, \delta_2 \geq Tol(x)$, the interval $[x - \delta_1, x + \delta_2]$ will be regarded as containing a minimum, even if $F(x)$ is less than $F(x - \delta_1)$ and $F(x + \delta_2)$ only due to rounding errors in the (sub)program. Therefore user-supplied (sub)program **funct** should be programmed to calculate $F(x)$ as accurately as possible, so that e04bb will not be liable to find a spurious minimum. (For similar reasons, $\frac{dF}{dx}$ should be evaluated as accurately as possible.)

9 Example

```
e04bb_funct.m

function [fc,gc,user] = e04bb_funct(xc,user)
 fc=sin(xc)/xc;
 gc=(cos(xc)-fc)/xc;
```

```
e1 = 0;
e2 = 0;
a = 3.5;
b = 5;
maxcal = int32(30);
[e1Out, e2Out, aOut, bOut, maxcalOut, x, f, g, user, ifail] = ...
    e04bb('e04bb_funct', e1, e2, a, b, maxcal)
```

```
e1Out =
   1.0542e-08
e2Out =
   1.0542e-08
aOut =
    4.4934
bOut =
    4.4934
maxcalOut =
           6
x =
    4.4934
f =
   -0.2172
g =
  -3.7679e-16
user =
     0
ifail =
           0
```