NAG Toolbox for MATLAB

e04hc

1 Purpose

e04hc checks that a (sub)program for evaluating an objective function and its first derivatives produces derivative values which are consistent with the function values calculated.

2 Syntax

```
[f, g, iw, w, ifail] = e04hc(funct, x, iw, w, 'n', n, 'liw', liw, 'lw',
lw)
```

3 Description

Routines for minimizing a function of several variables may require you to supply a (sub)program to evaluate the objective function $F(x_1, x_2, \ldots, x_n)$ and its first derivatives. e04hc is designed to check the derivatives calculated by such user-supplied (sub)programs . As well as the function to be checked (**funct**), you must supply a point $x = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ at which the check will be made. Note that e04hc checks functions of the form required for e04kd and e04lb.

e04hc first calls user-supplied (sub)program **funct** to evaluate F and its first derivatives $g_j = \dfrac{\partial F}{\partial x_j}$, for $j = 1, 2, \ldots, n$ at x. The components of the user-supplied derivatives along two orthogonal directions (defined by unit vectors p_1 and p_2, say) are then calculated; these will be $g^{\mathrm{T}} p_1$ and $g^{\mathrm{T}} p_2$ respectively. The same components are also estimated by finite differences, giving quantities

$$v_k = \frac{F(x + hp_k) - F(x)}{h}, \qquad k = 1, 2$$

where h is a small positive scalar. If the relative difference between v_1 and $g^{\mathrm{T}} p_1$ or between v_2 and $g^{\mathrm{T}} p_2$ is judged too large, an error indicator is set.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

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

funct must evaluate the function and its first derivatives at a given point. (The minimization functions mentioned in Section 3 gives you the option of resetting parameters of **funct** to cause the minimization process to terminate immediately. e04hc will also terminate immediately, without finishing the checking process, if the parameter in question is reset.)

Its specification is:

```
[iflag, fc, gc, iw, w] = funct(iflag, n, xc, iw, liw, w, lw)
```

Input Parameters

1: **iflag – int32 scalar**

Will be set to 2.

If you reset **iflag** to a negative number in **funct** and return control to e04hc, e04hc will terminate immediately with **ifail** set to your setting of **iflag**.

2: **n – int32 scalar**

The number n of variables.

3: **xc(n) – double array**

The point x at which F and its derivatives are required.

4: **iw(liw) – int32 array**
5: **liw – int32 scalar**
6: **w(lw) – double array**
7: **lw – int32 scalar**

These parameters are present so that **funct** will be of the form required by the minimization functions mentioned in Section 3. **funct** is called with e04hc's parameters **iw**, **liw**, **w**, **lw** as these parameters. If the advice given in the minimization function documents is being followed, you will have no reason to examine or change any elements of **iw** or **w**. In any case, **funct must not change** the first $3 \times$ **n** elements of **w**.

Output Parameters

1: **iflag – int32 scalar**

Will be set to 2.

If you reset **iflag** to a negative number in **funct** and return control to e04hc, e04hc will terminate immediately with **ifail** set to your setting of **iflag**.

2: **fc – double scalar**

Unless **funct** resets **iflag**, **fc** must be set to the value of the function F at the current point x.

3: **gc(n) – double array**

Unless **funct** resets **iflag**, **gc**(j) must be set to the value of the first derivative $\frac{\partial F}{\partial x_j}$ at the point x, for $j = 1, 2, \ldots, n$.

4: **iw(liw) – int32 array**
5: **w(lw) – double array**

These parameters are present so that **funct** will be of the form required by the minimization functions mentioned in Section 3. **funct** is called with e04hc's parameters **iw**, **liw**, **w**, **lw** as these parameters. If the advice given in the minimization function documents is being followed, you will have no reason to examine or change any elements of **iw** or **w**. In any case, **funct must not change** the first $3 \times$ **n** elements of **w**.

2: **x(n) – double array**

x(j), for $j = 1, 2, \ldots, n$, must be set to the co-ordinates of a suitable point at which to check the derivatives calculated by user-supplied (sub)program **funct**. 'Obvious' settings, such as 0.0 or 1.0, should not be used since, at such particular points, incorrect terms may take correct values (particularly zero), so that errors could go undetected. Similarly, it is preferable that no two elements of **x** should be the same.

3: **iw(liw) – int32 array**

This array is in the parameter list so that it can be used by other library functions for passing integer quantities to user-supplied (sub)program **funct**. It is not examined or changed by e04hc. Generally, you must provide an array **iw** but are advised not to use it.

4: **w(lw) – double array**

Constraint: $\mathbf{lw} \geq 3 \times \mathbf{n}$.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The dimension of the arrays **x**, **g**. (An error is raised if these dimensions are not equal.)

the number n of independent variables in the objective function.

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

2: **liw – int32 scalar**

Default: The dimension of the array **iw**.

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

3: **lw – int32 scalar**

Default: The dimension of the array **w**.

Constraint: $\mathbf{lw} \geq 3 \times \mathbf{n}$.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **f – double scalar**

Unless you set **iflag** negative in the first call of user-supplied (sub)program **funct**, **f** contains the value of the objective function $F(x)$ at the point given by you in **x**.

2: **g(n) – double array**

Unless you set **iflag** negative in the first call of user-supplied (sub)program **funct**, $\mathbf{g}(j)$ contains the value of the derivative $\frac{\partial F}{\partial x_j}$ at the point given in **x**, as calculated by **funct**, for $j = 1, 2, \ldots, n$.

3: **iw(liw) – int32 array**

This array is in the parameter list so that it can be used by other library functions for passing integer quantities to user-supplied (sub)program **funct**. It is not examined or changed by e04hc. Generally, you must provide an array **iw** but are advised not to use it.

4: **w(lw) – double array**

5: **ifail – int32 scalar**

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

6 Error Indicators and Warnings

Note: e04hc may return useful information for one or more of the following detected errors or warnings.

ifail < 0

A negative value of **ifail** indicates an exit from e04hc because you have set **iflag** negative in user-supplied (sub)program **funct**. The setting of **ifail** will be the same as your setting of **iflag**. The check on **funct** will not have been completed.

ifail = 1

On entry, **n** < 1,
or **liw** < 1,
or **lw** < 3 × **n**.

ifail = 2

You should check carefully the derivation and programming of expressions for the derivatives of $F(x)$, because it is very unlikely that user-supplied (sub)program **funct** is calculating them correctly.

7 Accuracy

ifail is set to 2 if

$$\left(v_k - g^{\mathrm{T}} p_k\right)^2 \geq h \times \left(\left(g^{\mathrm{T}} p_k\right)^2 + 1\right)$$

for $k = 1$ or 2. (See Section 3 for definitions of the quantities involved.) The scalar h is set equal to $\sqrt{\epsilon}$, where ϵ is the ***machine precision*** as given by x02aj.

8 Further Comments

The user-supplied (sub)program **funct** is called 3 times.

Before using e04hc to check the calculation of first derivatives, you should be confident that user-supplied (sub)program **funct** is calculating F correctly. The usual way of checking the calculation of the function is to compare values of $F(x)$ calculated by **funct** at nontrivial points x with values calculated independently. ('Non-trivial' means that, as when setting x before calling e04hc, co-ordinates such as 0.0 or 1.0 should be avoided.)

e04hc only checks the derivatives calculated by a when **iflag** = 2. So, if user-supplied (sub)program **funct** is intended for use in conjunction with a minimization function which may set **iflag** to 1, you must check that, for given settings of the **xc**(j), **funct** produces the same values for the **gc**(j) when **iflag** is set to 1 as when **iflag** is set to 2.

9 Example

```
e04hc_funct.m
```

```
function [iflag, fc, gc, iw, w] = funct(iflag, n, xc, iw, liw, w, lw)
  gc = zeros(n, 1);
  fc = 0;
  if (iflag ~= 1)
     fc = (xc(1)+10*xc(2))^2 + 5*(xc(3)-xc(4))^2 + (xc(2)-2*xc(3))^4 +
10*(xc(1)-xc(4))^4;
  end
  if (iflag ~= 0)
    gc(1) = 2*(xc(1)+10*xc(2)) + 40*(xc(1)-xc(4))^3;
    gc(2) = 20*(xc(1)+10*xc(2)) + 4*(xc(2)-2*xc(3))^3;
    gc(3) = 10*(xc(3)-xc(4)) - 8*(xc(2)-2*xc(3))^3;
    gc(4) = 10*(xc(4)-xc(3)) - 40*(xc(1)-xc(4))^3;
  end
```

```
x = [1.46;
```

```
     -0.82;
     0.57;
     1.21];
iw = [int32(0)];
w = zeros(12,1);
[f, g, iwOut, wOut, ifail] = e04hc('e04hc_funct', x, iw, w)
```

```
f =
   62.2726
g =
  -12.8550
 -164.9181
   53.8363
    5.7750
iwOut =
           0
wOut =
    1.4600
   -0.8200
    0.5700
    1.2100
    1.4600
   -0.8200
    0.5700
    1.2100
  -12.8550
 -164.9181
   53.8363
    5.7750
ifail =
           0
```