1. Suppose $x$ and $y$ can be represented without error in double precision. Can the same be said for $x^2$ and $y^2$?

Which would be more accurate, $x^2 - y^2$ or $(x - y)(x + y)$?

2. A simulation is produced to compare two variants of a manufacturing process. Let $A$ be the (random) annual cost using the first variant and $B$ the annual cost using the second variant. We would like to estimate $\delta = EA - EB$.

In each case $n$ simulations of the annual cost were produced, denoted $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$. Suppose that the $a_i$ and $b_i$ are both around 1,000,000, that $n = 10,000$, and that the difference between $EA$ and $EB$ is around 1%. Working in double precision, how many decimal places of accuracy do you expect for $\hat{\delta}$?

3. Suppose $f: \mathbb{R}^d \rightarrow \mathbb{R}$. Since $\partial f(x)/\partial x_i = \lim_{\epsilon \to 0}(f(x + \epsilon e_i) - f(x))/\epsilon$, we have for small $\epsilon$

   $$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(x + \epsilon e_i) - f(x)}{\epsilon}.$$ 

In the same way, for $i \neq j$

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} \approx \frac{f(x + \epsilon e_i + \epsilon e_j) - f(x + \epsilon e_i) - f(x + \epsilon e_j) + f(x)}{\epsilon^2}$$

and

$$\frac{\partial^2 f(x)}{\partial x_i^2} \approx \frac{f(x + 2\epsilon e_i) - 2f(x + \epsilon e_i) + f(x)}{\epsilon^2}.$$

(a) Test the accuracy of these approximations using the function $f(x, y) = x^3 + xy^2$ at the point $(1, 1)$. That is, for a variety of $\epsilon$, calculate the approximate gradient and Hessian, and see by how much they differ from the true gradient and Hessian.

(b) Take $d = 1$ and suppose that at $x$ we have $f(x) \approx 10^9$, $f'(x) \approx 10^6$ and $f''(x) \approx 10^5$.

Show that using double precision $f(x + \epsilon) - f(x)$ has relative error $\approx 10^{-16-a+b}/\epsilon$ and $1/\epsilon$ has relative error $\approx 10^{-16}/\epsilon$, and thus that $(f(x + \epsilon) - f(x))/\epsilon$ has an absolute error of approximately

$$\frac{10^{a-16}}{\epsilon} \text{ or } 10^{b-16}.$$ 

Given that $(f(x + \epsilon) - f(x))/\epsilon - f'(x) \approx \epsilon f''(x)$, roughly how large should $\epsilon$ be to minimise the error in the approximation?

4. Horner’s algorithm for evaluating the polynomial $p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$ consists of re-expressing it as

$$a_0 + x(a_1 + x(a_2 + \cdots + x(a_{n-1} + xa_n)\cdots)).$$

How many operations are required to evaluate $p(x)$ in each form?

5. How many multiplications/divisions are performed by the following code, as a function of $n$? (We assume that $n$ is already defined.) Suggest a way(s) of making the code more efficient.

```r
x <- 2
s <- 1
for (k in 1:n) {
  t <- 1
  for (j in 1:k) {
    t <- t*j
  }
  s <- s + x^k/t
}
```
6. A picture is worth a thousand words.

The function `newtonraphson.show` is a modification of `newtonraphson` that plots intermediate results. Instead of using the variables `tol` and `max.iter` to determine when the algorithm stops, at each step you will be prompted to enter `y` at the keyboard if you want to continue. There are also two new inputs, `xmin` and `xmax`, which are used to determine the range of the plot. `xmin` and `xmax` have defaults `x0 - 1` and `x0 + 1`, respectively.

Use `newtonraphson.show` to investigate the roots of the following functions:

(a) $\cos(x) - x$ using $x_0 = 1, 3, 6$
(b) $\log(x) - \exp(-x)$ using $x_0 = 2$
(c) $x^3 - x - 3$ using $x_0 = 0$
(d) $x^3 - 7x^2 + 14x - 8$ using $x_0 = 1.1, 1.2, \ldots, 1.9$
(e) $\log(x) \exp(-x)$ using $x_0 = 2$.

7. A simple way of using local search techniques to find a global maximum is to consider several different starting points, and hope that for one of them its local maximum is in fact the global maximum. If you have no idea where to start, then randomisation can be used to choose the starting points.

Consider the function

$$f(x, y) = -(x^2 + y^2 - 2)(x^2 + y^2 - 1)(x^2 + y^2 + 1)(x^2 + y^2 + 2) \times (2 - \sin(x^2 - y^2) \cos(y - \exp(y))).$$

It has several local maxima in the region $[-1.5, 1.5] \times [-1.5, 1.5]$. Using several randomly chosen starting points, use the BFGS algorithm in `optim` to find all of the local maxima of $f$, and thus the global maximum. You can use the command `runif(2, -1.5, 1.5)` to generate a random point $(x, y)$ in the region $[-1.5, 1.5] \times [-1.5, 1.5]$.

A picture of $f$ is given in Figure 1. Note that $f$ has been truncated below at $-3$.

The script `spur12.8.6.incomplete.r` contains implementations of $f$ and $\nabla f$.

8. Suppose that we conduct $m$ independent trials, and in each trial there are $d$ possible outcomes with probabilities $p_1, \ldots, p_d$. Let $Y = (Y_1, \ldots, Y_d)$ be the number of times each outcome occurs, then we say that $Y \sim \text{multinomial}(m; p_1, \ldots, p_d)$, and

$$\mathbb{P}(Y = y) = \frac{m!}{y_1! \cdots y_d!} \prod_{i=1}^{d} p_i^{y_i}. \quad (a)$$

Let $p = (p_1, \ldots, p_{d-1})$, and put $p_d = 1 - p_1 - \cdots - p_{d-1}$. We will assume that $0 < p_i$ for all $i$ and that $\sum_{i=1}^{d-1} p_i < 1$, so $p_d > 0$ as well. Let $y(1), \ldots, y(n)$ be an independent sample of multinomial vectors, then obtain expressions for the observed information and the Fisher information, as functions of $p$.

(b) Consider the following sample:

- $y(1) = (1, 2, 0)$
- $y(2) = (1, 1, 1)$
- $y(3) = (2, 1, 0)$
- $y(4) = (0, 2, 1)$

For these data, use Newton’s method to estimate $p$, starting from $p(0) = (1/3, 1/3, 1/3)$.

(c) Fisher scoring is a variant of Newton’s method where we use the Fisher information matrix instead of the observed information matrix. The Fisher information is often easier/quicker to calculate, and is guaranteed to be positive definite (unlike the observed information).

Use Fisher scoring to estimate $p$, starting from the same point as before.

(d) In this case we can find the maximum likelihood estimator explicitly. Which method converges most quickly for the example above, Newton’s method or Fisher scoring?
Figure 1: A function with a number of local maxima.