Quick and dirty exponential function

Anders Bak-Nyhus

The exponential function

An exponential function, is a function of the form

$$f(x) = ab^x \tag{1}$$

A more interesting case of exponential functions, is the one with the base e, where the exponential function is it's own derivative

$$\frac{d}{dt}e^x = e^x \tag{2}$$

The exponential function can be written as a power series

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} \tag{3}$$

Implementing the exponential function

There are different ways of how to implement the exponential function, in c code. In math.h there is an exponential function, where we use exp(x), however there could be a reason for not using this implementation, and might want to make your own. One implementation could be the so called quick and dirty (QAD) implementation described below.

```
double ex(double x)
if(x < 0) return 1/ex(-x);
if(x > 1./8) return pow(ex(x/2),2);
return 1+x*(1+x/2*(1+x/3*(1+x/4*(1+x/5*(1+x/6*(1+x/7*(1+x/8*(1+x/9*(1+x/10)))))))));
```

This implementation, is an approximation of the power series mentioned above, where the sum goes from $k = 0$ to $k = 10$, it is however a convoluted version so as to have less operations, and save time. As it is only an approximation, the implementation of the exponential function, might differ from the *exp* function you get through using math.h.

Now as for why this implementation might be usefulf, we need to look at the amount of operations, needed to run it. We need to determine how fast and how accurate we need our program to be, and if we want it to be faster with less accuracy, it is useful to have less terms, so that the program can run through it faster.

If the program only needs to handle numbers between less than one, then we can neglect more terms, as the terms in the power series becomes smaller and smaller.

In the implementation above, there is a line *if(x>1./8) return pow(ex(x/2),2);*, this line is implemented as it works best between zero and one. This is done in practice by using

$$x^k = (x^{k/2})^2$$

this will make sure our values put in to the exponential function (bottom line in the implementation) make it go to zero the larger the power.

Test of implementation

In the figure below, two different ways of implementing the exponential function, has been plotted. One being the *exp* function from math.h, the other being the quick and dirty implementation described above. As can be seen from the figure, the two plots are identical, so in this case the implementation is useful, especially if it saves time.

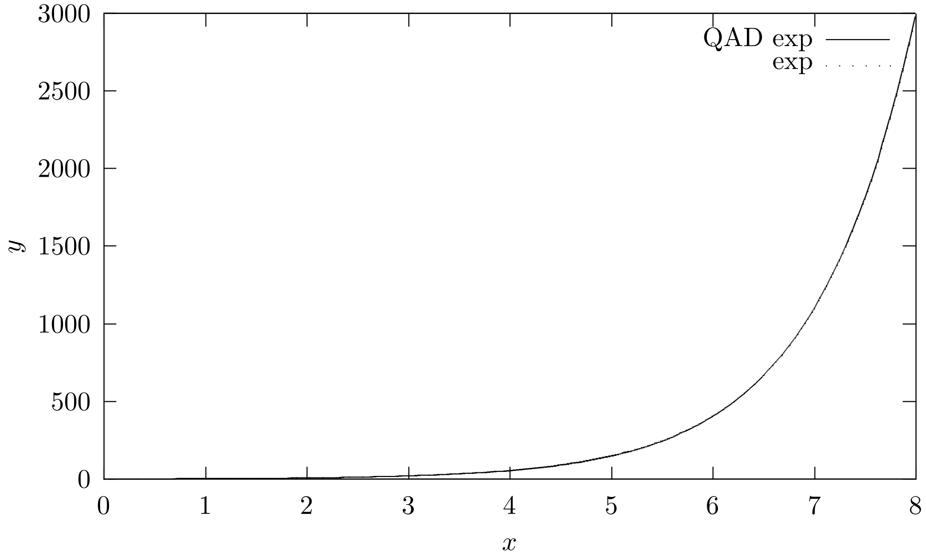

Figure 1: exponential function plot