Implementation of exponential function in C

Martin Aagaard

Abstract

In this short report an implementation of the exponential function will be investigated.

1 The exponential function

The exponential function is widely known and used and can be defined through the property that $\frac{dy(x)}{dx} = y$ if $y(x) = e^x$. The exponential function appears many places e.g. in physics which necessitates an implementation in computer languages. One way to do this is through the taylor series of the exponential function,

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \, . \tag{1}$$

2 Implementation

From the taylor series it is seen that one can take factor out factors of x and n leading to an expression on the following form:

$$1 + x \cdot (1 + \frac{x}{2} \cdot (1 + \frac{x}{3} \cdot (...))) \, . \tag{2}$$

This odd form of writing the series minimizes the number of time a program would need to multiply and through that minimizing the operation time. In C a final implementation could look like:

```
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)))))))));
} .
```

Here the if statements make sure that the value of x is significantly small to use double precision and not negative since summing alternating values will make the routine lose precision from subtracting and adding small numbers. As for

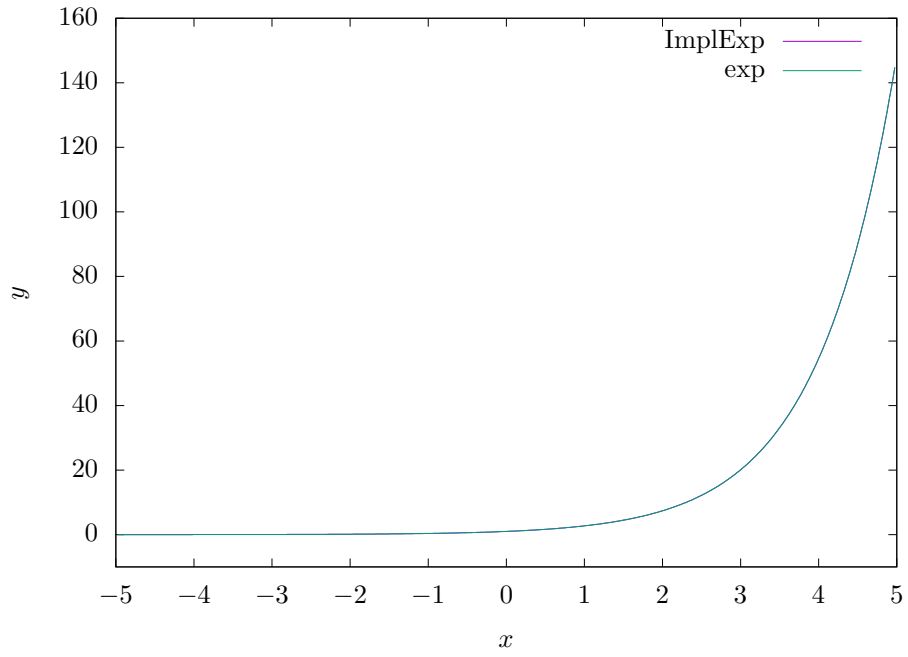

Figure 1: Implementated exponential function (Implexp) compared with the math.h implemantation(exp) via gnuplot "latex" terminal.

significantly small values of x, it is here calculated that for 10^{-15} precision with 10 terms, a value of x smaller than $\frac{1}{8}$ is needed. This allows us to include fewer terms of the taylor series.

3 Result

The implementation is now compared to the exp-function from math.h. The result is seen in figure (1)