An implementation of the exponential function in the C-language

Kasper Larsen

Abstract

An implementation of the exponential function.

1 A: The exponential function

The exponential function $\exp x$ is defined by the following power series:

$$\exp x \equiv \sum_{k=0}^{\infty} \frac{x^k}{k!} \tag{1}$$

It has numerous uses in physics.

2 A: Implementation

We can approximate (1) by its 10th partial sum; i.e.,

$$\exp x = \sum_{k=0}^{\infty} \frac{x^k}{k!} \simeq \sum_{k=0}^{10} \frac{x^k}{k!} \; .$$

For very large x, the accuracy may not be that good. We can expand the above into the following:

$$\begin{aligned} \exp x &\simeq \sum_{k=0}^{10} \frac{x^k}{k!} \qquad\qquad (2) \\ &= 1 + x\left(1 + \frac{x}{2}\left(1 + \frac{x}{3}\left(1 + \frac{x}{4}\left(1 + \frac{x}{5}\left(1 + \frac{x}{6}\left(1 + \frac{x}{7}\left(1 + \frac{x}{8}\left(1 + \frac{x}{9}\left(1 + \frac{x}{10}\right)\right)\right)\right)\right)\right)\right)\right)\right) \end{aligned}$$

On figure 1, both the implementation above and the built-in exponential function from the math.h library are plotted alongside eachother. The two plots are nearly indistinguishable. Here is the C-code:

```
#include <stdio.h>
#include <math.h>
```

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

int main(){
        int datapoints = 200;
        double xmax = 5;
        for (int i=0;i<datapoints;i++) {
                printf("%10g %10g %10g\n",
                xmax*i/datapoints,ex(xmax*i/
                datapoints),exp(xmax*i/datapoints));
        }
        return 0;
}
```

In the code above, simple identites for the exponential function are used for purposes explained in the next section.

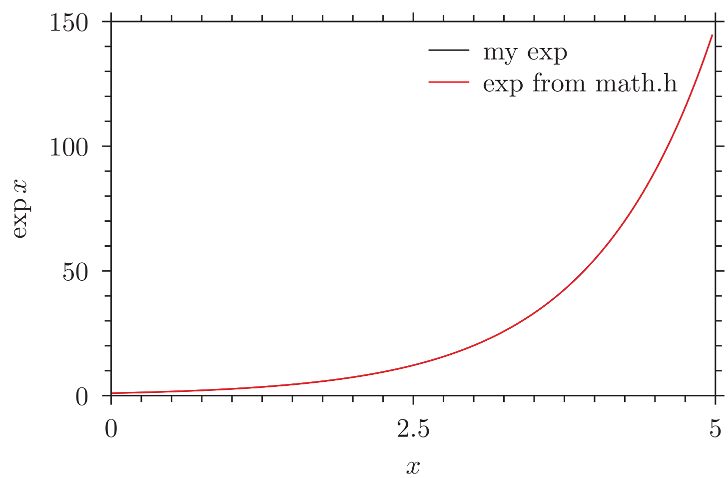

Figure 1: This is a plot of the exponential function from two implementations.

3 B: Numerical advantage

There is a numerical advantage of using the above implementation rather than simply using the Taylor series. The indentity

$$\exp x = \frac{1}{\exp(-x)} \tag{3}$$

is used for negative values of x to ensures that only positive numbers are added together, while the convolution ensures that numbers of the same orders are added together. Since the 10th partial sum only works well for relatively small x, the identity

$$\exp x = \exp(x/2) \cdot \exp(x/2) \tag{4}$$

is used iteratively until the argument of the exponential function is small enough.