

The exponential function

Freja Borgaard Nielsen

June 29, 2021

1 The exponential function

The real exponential function, $\exp(x) : \mathbb{R} \rightarrow \mathbb{R}$, is one of the first functions we all learn to use in throughout school years. For long it remains as merely yet another tool in our steadily growing collection of mathematical expertise. In this short report we will discuss the exponential function in greater detail, specifically we focus on its approximations.

The definition of the real exponential function originates from a power series, expressed as follows,

$$\exp(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}. \quad (1)$$

The convergence radius of this series is infinite, and thus the exponential function can be extended to the complex numbers \mathbb{C} .

2 A numerical approximation

In this exercise we use an approximation of the power series above as seen in the following code snippet

```
#include<math.h>
double exponential(double val){
    if( val < 0 ) {
        double result = 1.0 / exponential( -val );
        return result;
    }
    if( val > 1./8 ) {
        double result = pow( exponential( val / 2
            ), 2 );
        return result;
    }
    double result =
        1 + val*( 1 + val/2*( 1 + val/3*( 1 +
            val/4*(1 + val/5*( 1 + val/6*( 1 +
            val/7*( 1 + val/8*( 1 + val/9*( 1 +
            val/10) ) ) ) ) ) ) ) );
    return result;
}
```

In this implementation input values, the query points are given in the double valued parameter `val`. First we assure there are no negative inputs, in which case we recursively return the inverse of the function value, of the corresponding positive input value, i.e. $\exp(-\text{val})^{-1}$. This implementation, is known as the Taylor series around zero, and will be most accurate for small values, so we should hence next check for the size of the input value. Squaring the exponential function, due to the nature of exponents, will be the same as doubling the input value, hence the division by two. This ensures the input value can be made arbitrarily small. This is done recursively until the input value is a number of value smaller than $\frac{1}{8}$. The final computation of the Taylor series is then performed. Instead of using exponents, we simply multiply the value onto it self and arbitrary number of times. This operation is time efficient, and thus very fast. It has the added advantage of not computing factorials, that is, very large numbers, that will conflict with our precision very quickly - as we use a Taylor series. Hence the computer will not be forced to add up numbers of different orders of magnitude, and we retain precision.

3 Figures

An illustration of the numerical implementation described above can be seen in figure (1) which shows the "pdf" terminal of Pyxplot.

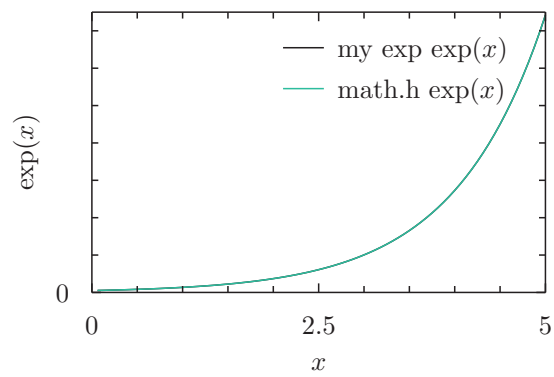


Figure 1: Plot of numerical implementation described above, versus the standard implementation found in the header `math.h`. Notice the two graphs are indistinguishable.