

# A brief introduction to exponential function

Akgash Sundaralingam

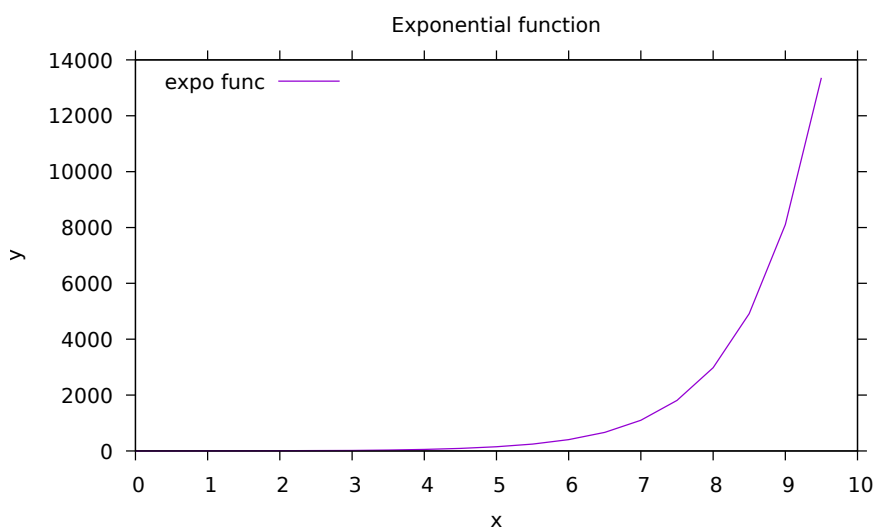


Figure 1: a nice plot

The exponential function is one of the most important functions in mathematics (though it would have to admit that the linear function ranks even higher in importance). To form an exponential function, we let the independent variable be the exponent. A simple example is the function  $y = e^x$ . The exponential function is one of the most important functions in mathematics (though it would have to admit that the linear function ranks even higher in importance). To form an exponential function, we let the independent variable be the exponent. A simple example is the function:  $y = e^x$ . The function seen in figure 1 here is something that looks like:

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

This function is widely used in both maths and physics to describe diverse functions and natural behaviours.

## 1 Thoughts and analysis in C

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:

$$1 + x \cdot \left(1 + \frac{x}{2} \cdot \left(1 + \frac{x}{3} \cdot (\dots)\right)\right). \quad (2)$$

This odd form of writing the series minimizes the number of time a program would need to use and multiply and through that minimizing the run-time for the algorithm. In C the 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)))))))));  
}
```

The first if-statement sends negative arguments into positive arguments. This has the advantage that one avoids alternating sums that are prone to large round-off errors. The second if-statement ensures that we never apply the truncated series directly to large arguments.