Assignment 2 Writeup

Nhan Nguyen

January 2022

1 Introduction

- For this writeup, I'll be describing how I implemented the integrate() function using C code. In addition, I'll analyse how different partitions in integrate() affect the graph and discuss some important things that I learned after analysing such graphs

2 integrate() description and code

2.1 Description

- For this particular function the method used to do the integration is Simpson's 1/3 rule, more specifically the Composite Simpson's 1/3 rule.

$$\int_a^b f(x)dx \approx \frac{h}{3}\sum_{j=1}^{n/2}[f(x_{2j-2}) + f(x_{2j-1}) + f(x_{2j})]$$

with $j = a + jh$ for $j = 0, 1, 2, ..., n-1, n$ and $h = \frac{b-a}{n}$. The number **n** in the equation is the number of partitions, and this number should be even
The important aspect to note about integration is the number of partitions, which will be discuss later in this document

2.2 Code

```
double integrate(double (*f)(double), double a, double b, uint32_t
    n){
  double term1 = 0.0;
  double term2 = 0.0;
  double term3 = 0.0;
  double h = (b - a) / n;
  for(double j = 1.0; j <= n / 2; j++){
    term1 += (*f)(a + ((2 * j - 2) * h));
    term2 += (*f)(a + ((2 * j - 1) * h));
    term3 += (*f)(a + (2 * j * h));
  }
  term2 *= 4.0;
```

```
  return (h / 3) * (term1 + term2 + term3);
}
```

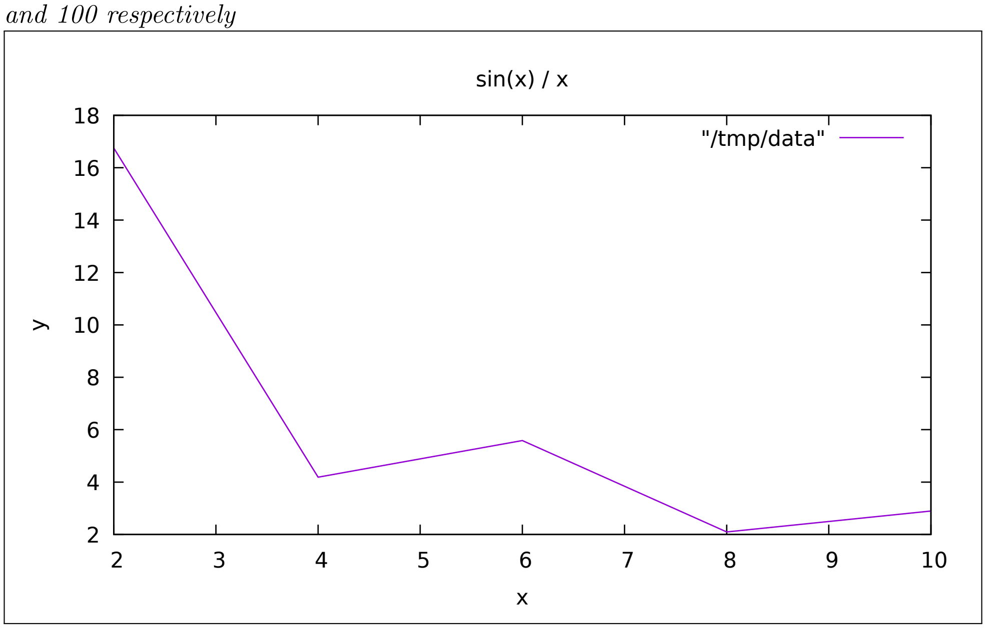
3 Graph and Analysis

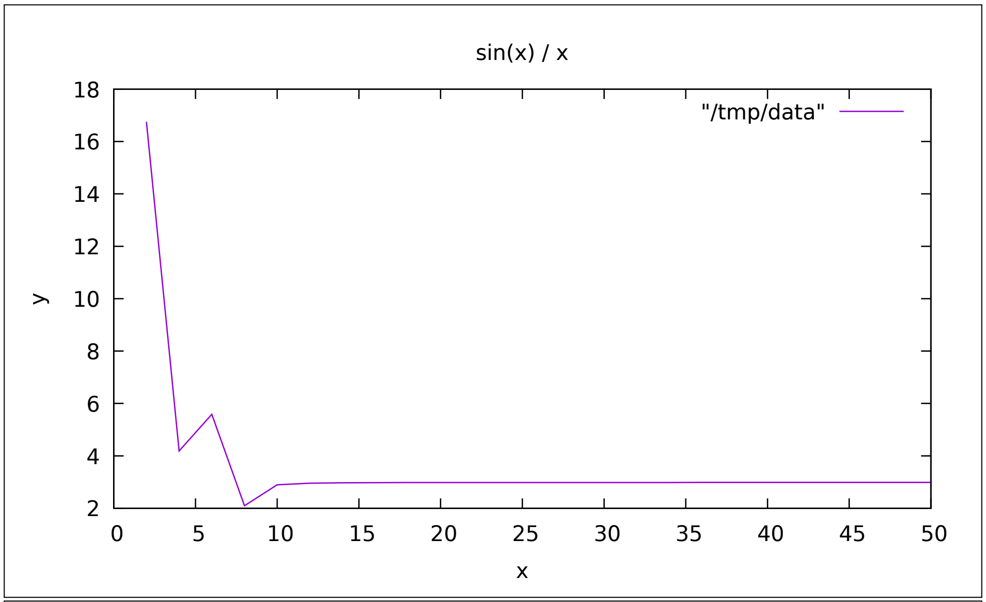
- Like I mentioned in Sections 2, the important aspect in an integration is the number of partitions, which is the defining factor to how accurate the result of the integration will be. To demonstrate this, I'll be generating 6 graphs (3 graphs for each equation) using **gnuplot** in order to show how the accuracy changes overtime, and the difference in these 3 graphs will be the number of partitions. Note that the equation used to generate the graphs is:

$$\int_{-4\pi}^{4\pi} \frac{sin(x)}{x} dx$$

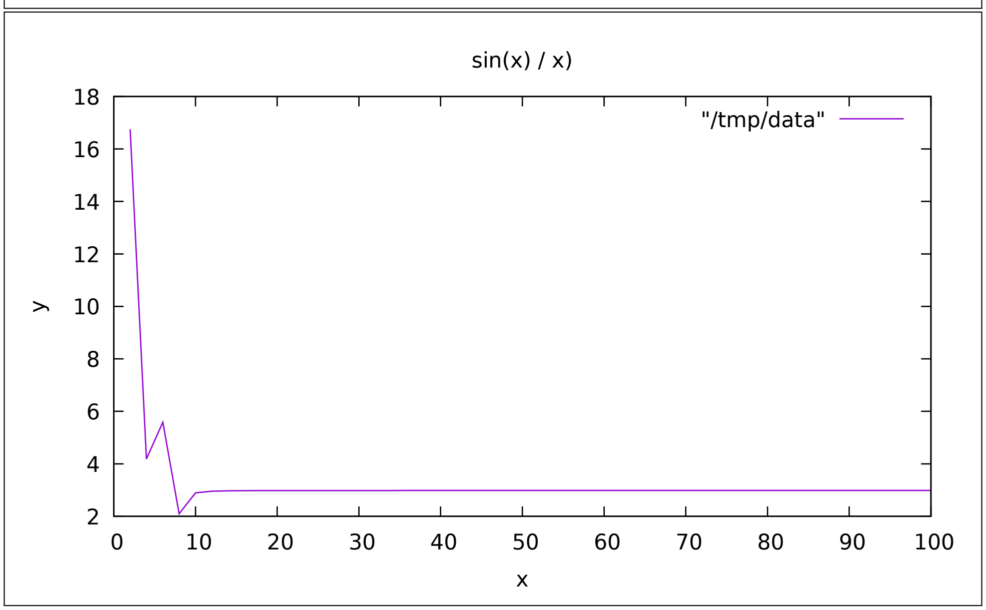
$$\int_{2}^{10} log(log(x)) dx$$

The value of equation $\int_{-4\pi}^{4\pi} \frac{sin(x)}{x} dx$ is **2.984322451168924**
The value of equation $\int_{2}^{10} log(log(x)) dx$ is **3.952914142858876**
The number of partitions shown in these 3 graphs for each functions are 10, 50, and 100 respectively

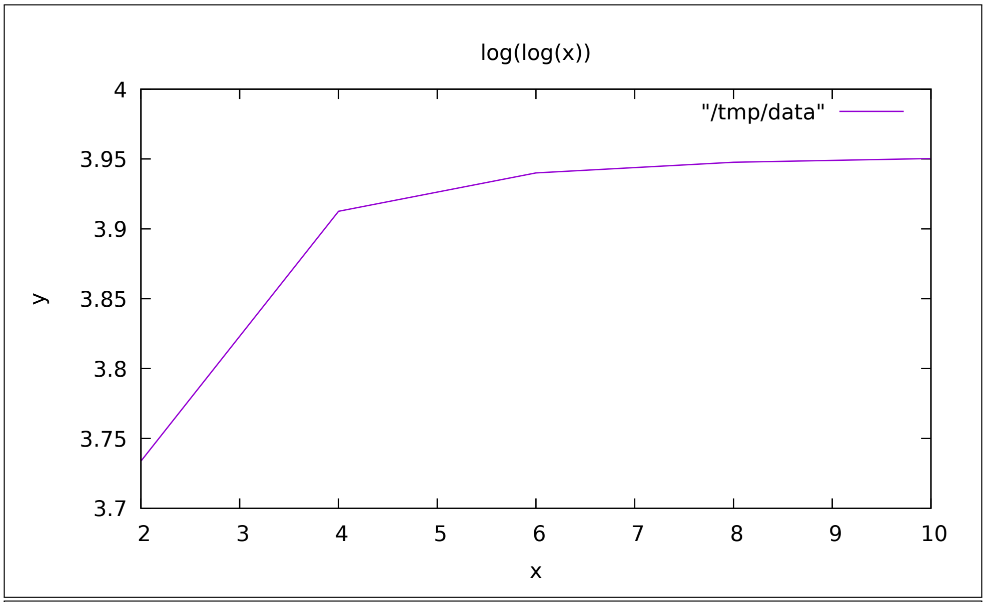


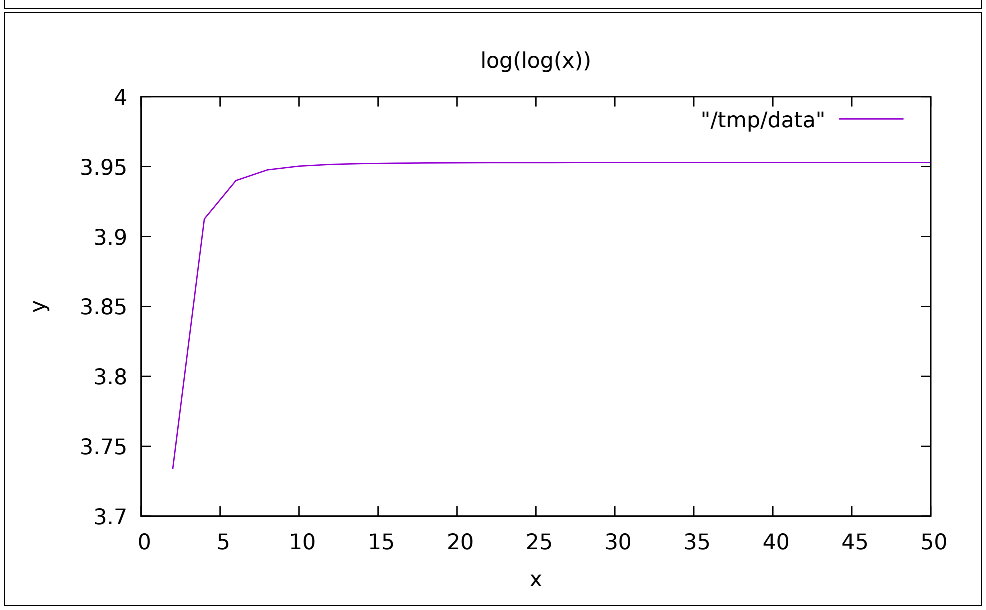


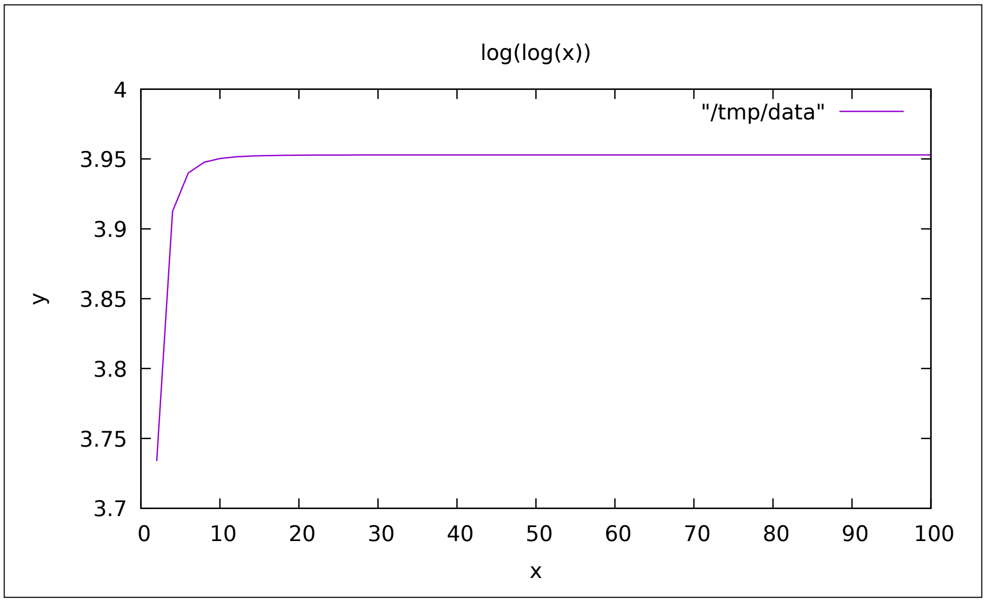
sin(x) / x
"/tmp/data"
y
x



sin(x) / x)
"/tmp/data"
y
x


log(log(x))
"/tmp/data"
4
3.95
3.9
3.85
3.8
3.75
3.7
y
2
3
4
5
6
7
8
9
10
x


log(log(x))
"/tmp/data"
4
3.95
3.9
3.85
3.8
3.75
3.7
y
0
5
10
15
20
25
30
35
40
45
50
x



From these 6 graphs (3 for each function), we can see that the result of the integration gets increasing closer to the value **2.984322451168924** and **3.952914142858876** respectively. From here, we can conclude that the value of an integration gets increasing more accurate the number of partitions increase.

4 Interesting Observation and Learned Experience

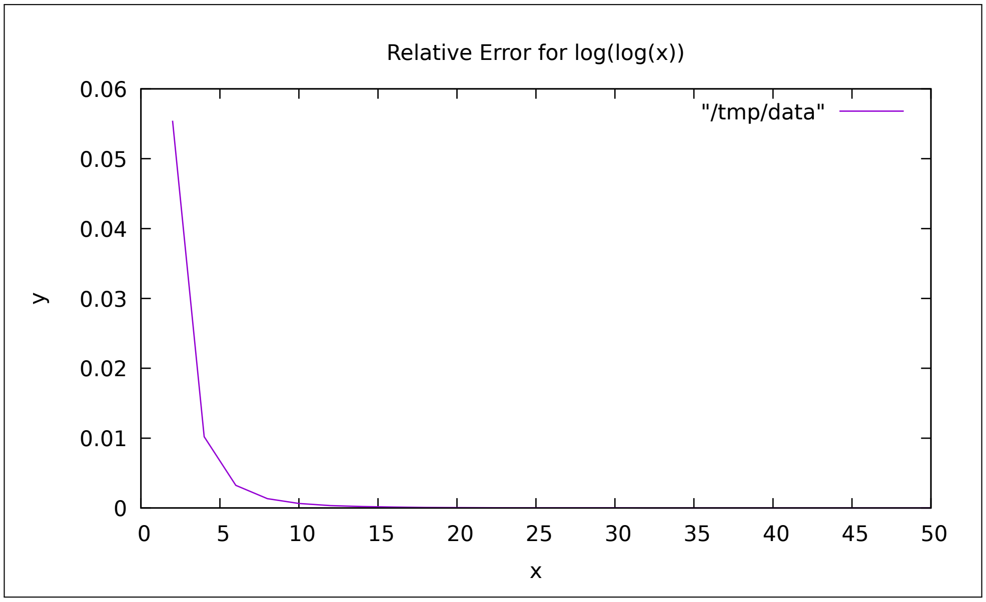
- As we have learned in section 3, the value of the integration gets closer and more accurate to the real value as we increase the number of partitions we used when doing the integration
- However, when we look closer at integrated value, we have to ask this important question "If the integration value gets more accurate as the number of partition increases, how accurate can they be?"
- To answer this question, we need to use a tool called the relative error in order to help see the accuracy of the estimated integration values to the actual value. The equation of relative error is:

$$\frac{|x' - x|}{|x|}$$

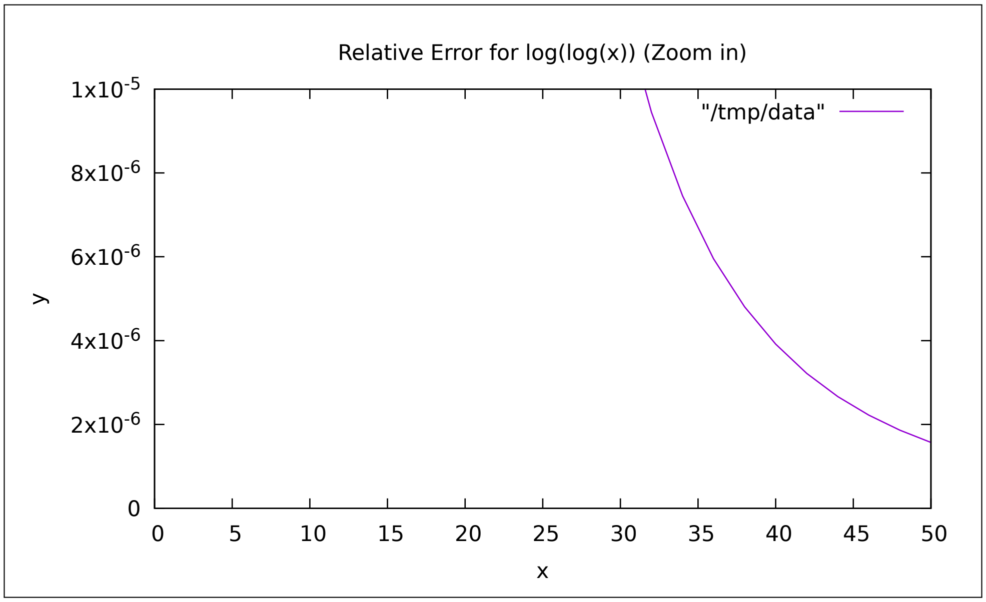
with **x'** being the estimated value and **x** being the real value. When the relative error equals to **0**, it means that the estimated value is exactly the same as the real value.
To see how accurate the results from the integration of $\frac{sin(x)}{x}$ and $log(log(x))$

can get, we will generate the graphs of relative error to see the result

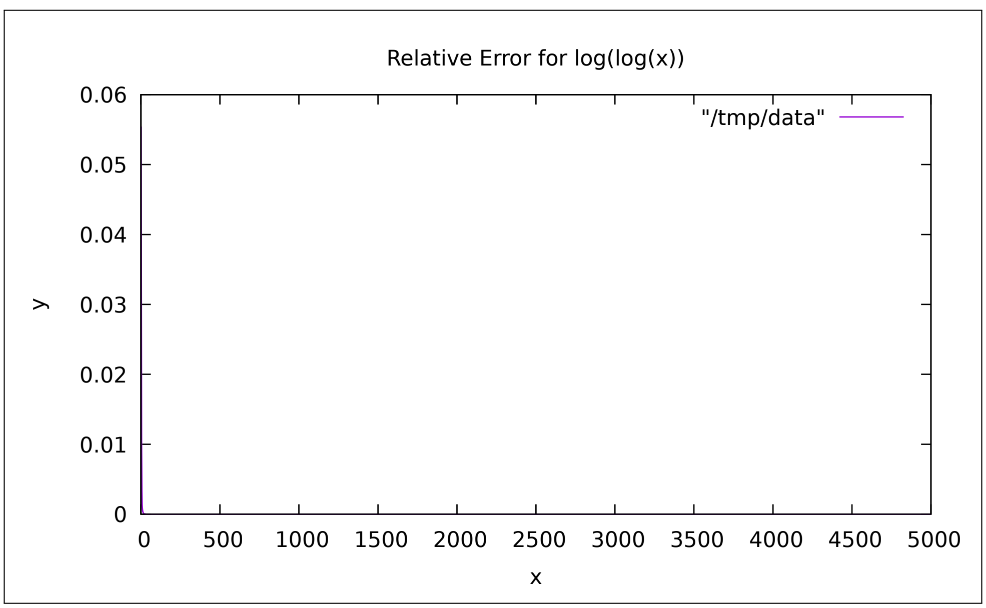
Let's start off with the relative error of $\int_2^{10} log(log(x))dx$ with the number of partitions being **50**

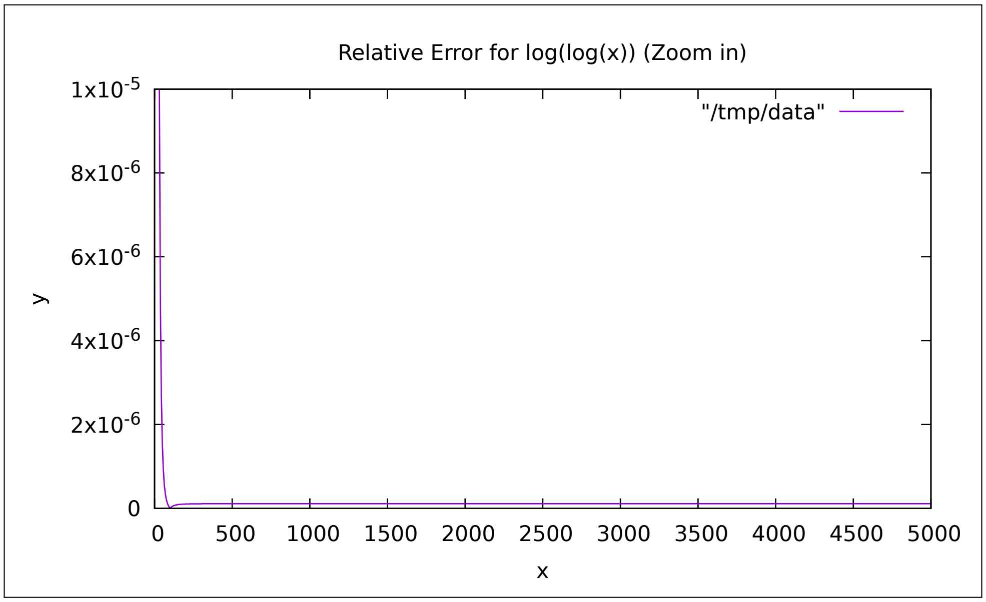


Looking at this graph you might think that at $\mathbf{x = 50}$ the y-value is **0**. However if we zoom in to the graph by setting the y range to an interval [**0:0.000005**], we can see that the y-value is not 0, therefore the approximation is not exactly the same as the real value



Now let's try to change the number of partitions to **5000**, which is going to produce an even more accurate value compared to the real value, we will also include the graph that has already been zoomed in like the previous graph since the graph without zooming can be very difficult to see

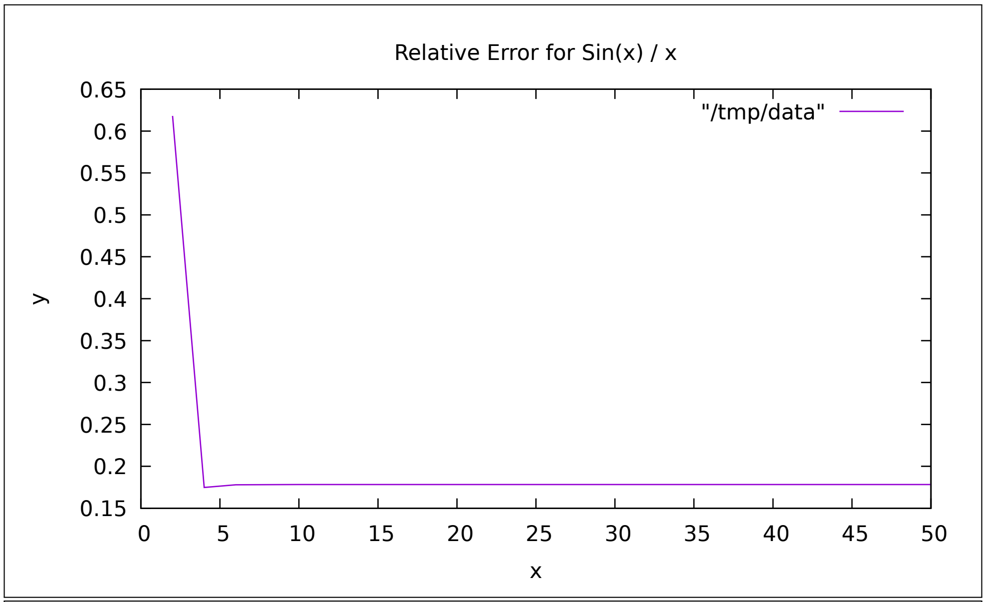


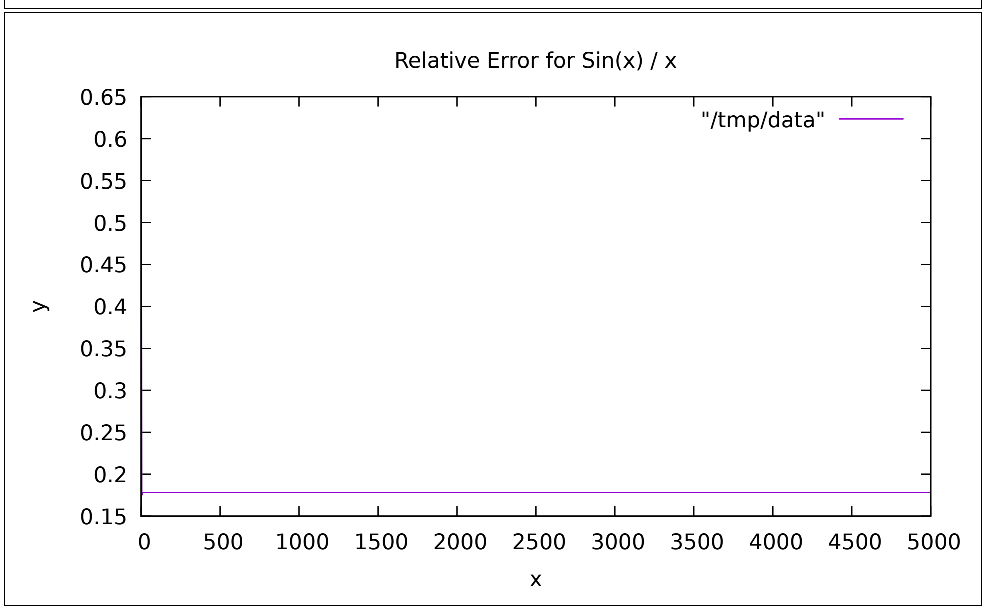


As you can see the value also didn't reach 0 completely, which means that even with this increasingly accurate result, the value can never be exactly equal to the real value.

This phenomenon doesn't happen only to $log(log(x))$, we can now look at the relative error graph of the equation $\int_{-4\pi}^{4\pi} \frac{sin(x)}{x} dx$ to see if the same phenomenon happens.

Trying with the number of partitions at 50 and 5000, we get the following graphs respectively





(note that the value in the second graph also starts from the approximately 0.63 and going down to approximately 0.17 like in the first graph but the scaling makes it very hard to see)

We can see the value also didn't touch 0, which means that the estimate value is not equal exactly to the real value. From these examples, we learned a lesson about floating point numbers in computers. Floating points in computers can never be exactly equal to the real number like in mathematics, floating point numbers in computers can only approximate as accurately as possible to the real number. Therefore don't ever use the equal operator in programming language to compare float numbers. The best course of action is to use an error tolerance value like **epsilon** $= 10^{-}14$ to see if the number is accurate enough to be acceptable.