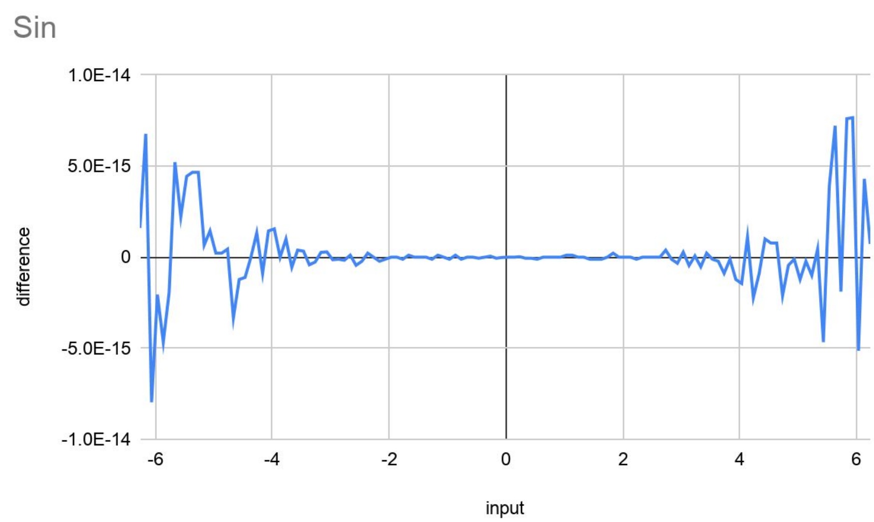
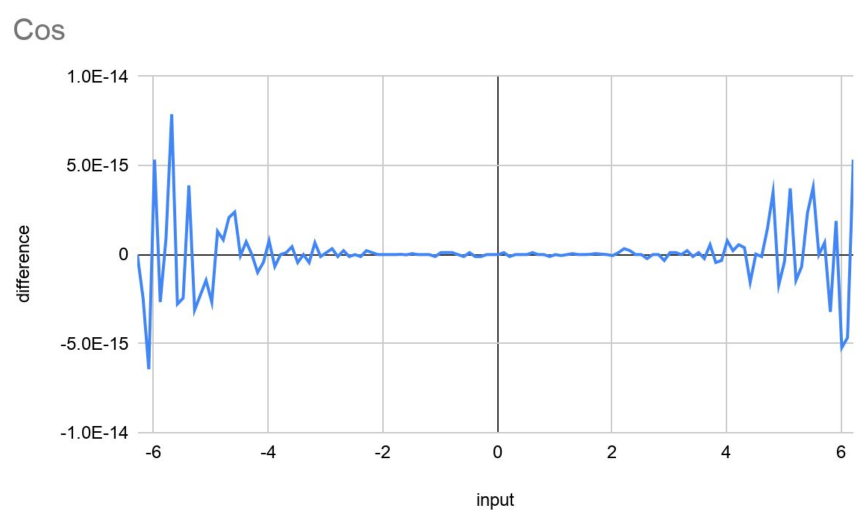
<u>WriteUp</u>

In this writeup we will be analyzing the differences in output of our program vs. the output from <math.h>.

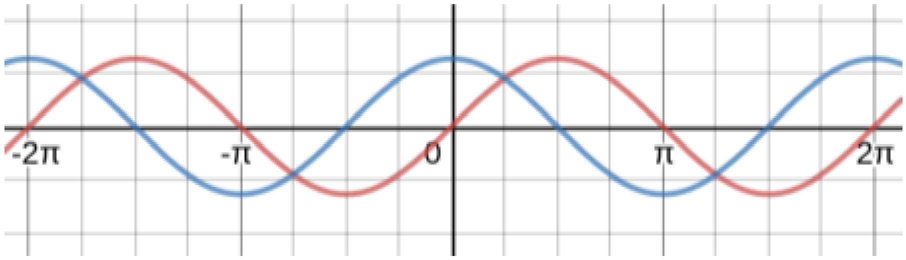
Sin and Cos

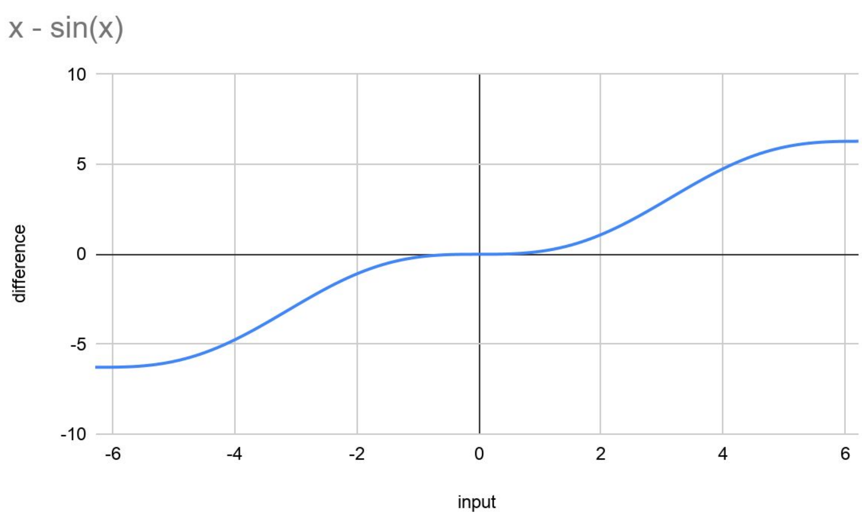




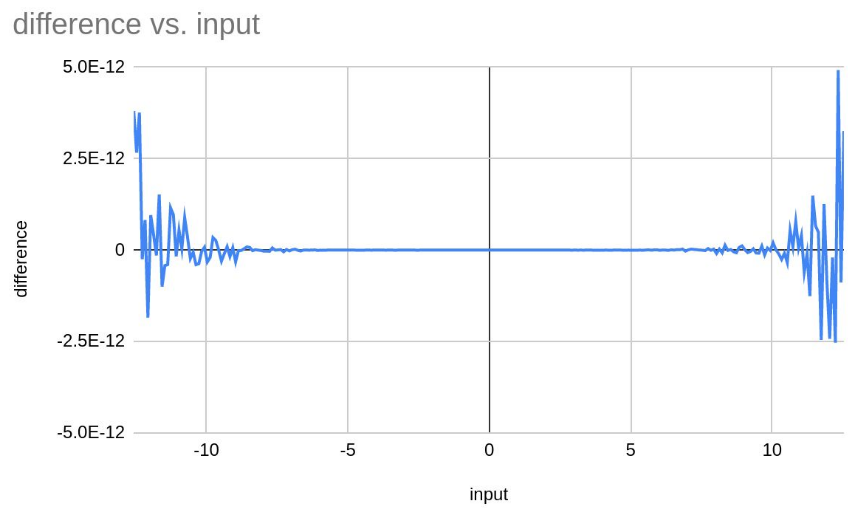
Here we have the graphs that show the differences between the outputs of our program vs. the outputs from <math.h>. The x-axis is the value of 'x' that we imputed to the functions, and the y-axis is the difference (my output - library output) between the outputs.

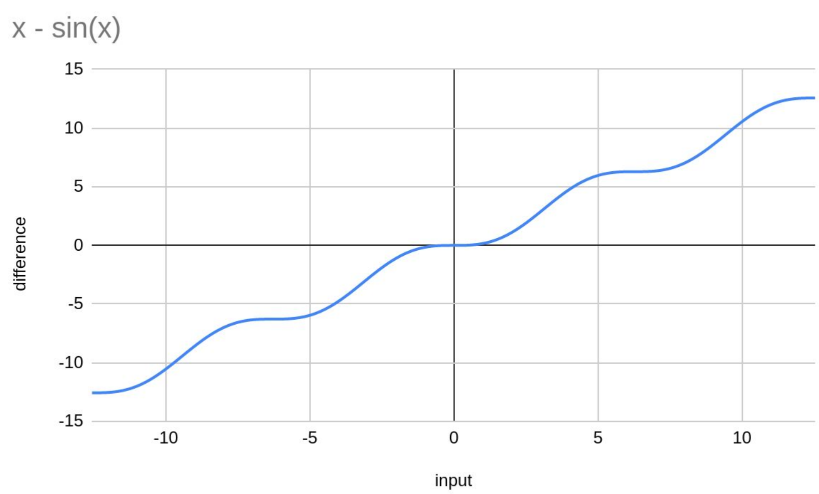
Both of these graphs look very similar in terms of the pattern they are showing. If you take a look at each of the graphs, it is very clear that there are large differences as we approach -2pi and 2pi. I think that the reason for the high discrepancies around these values is because the further away the real answer is from our initial guess, the less accurate the taylor series is.





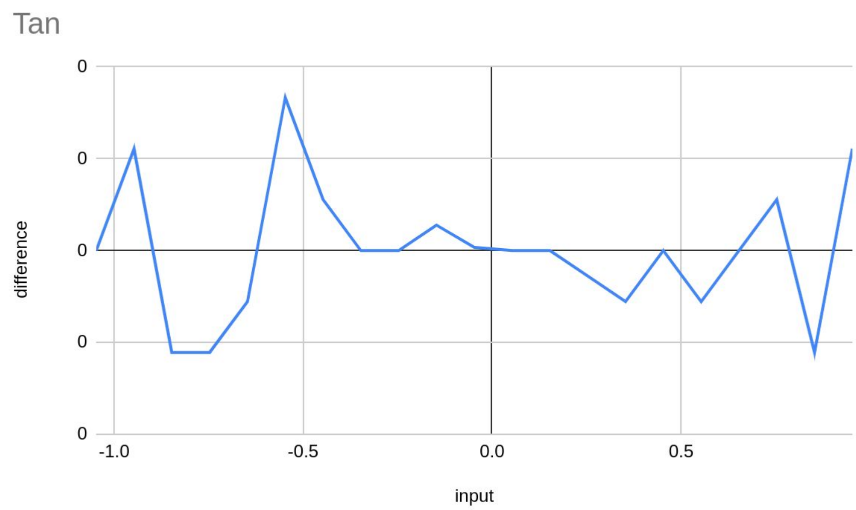
The top graph is a graph showing the cos(x) vs sin(x) graphs. Red is sin and Blue is cos. The bottom graph shows the difference between x and sin(x). If we look at the graphs, as our inputs approach 0, the difference between input and answer also approaches 0. Also, as our inputs stray from 0, the difference between input and answer start to get bigger. This would seem to be a logical explanation to why the library's answers and our answers start to deviate around these values. I was curious to see the graph of the functions on a range from [-4pi,4pi] without normalizing them (putting them between -2pi and 2pi) and the results seem to validate my suspicions.





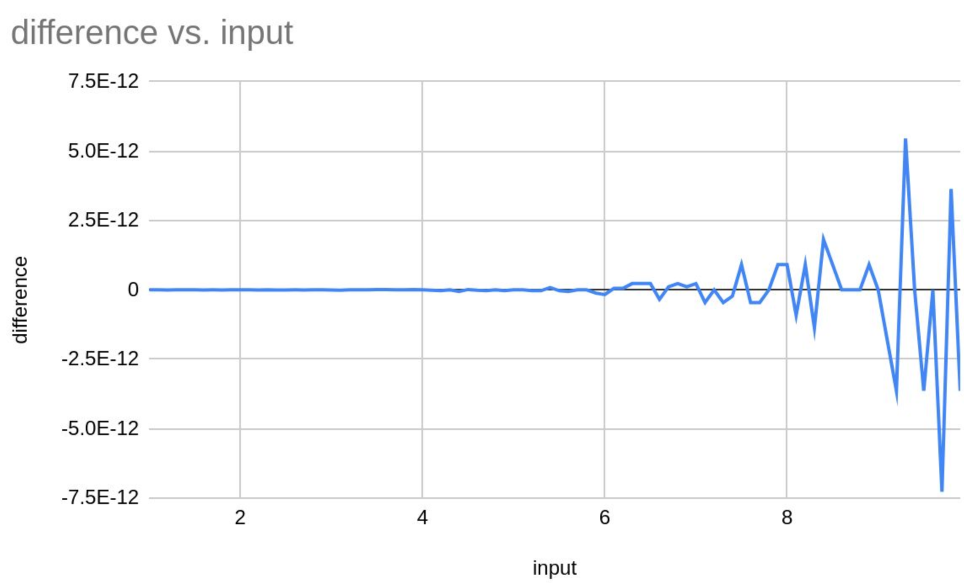
As the inputs(initial guesses) stray from the answer, the difference between the actual answer and our answer gets greater.

Tan



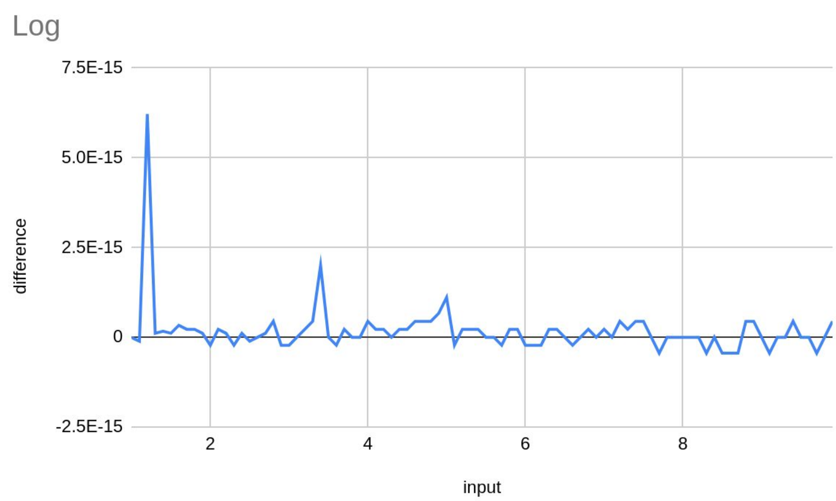
Tan(x) looks like a very random graph, but if you look at it carefully, it follows a similar pattern to the sin(x) and cos(x) graphs. It seems as though because this function is just a ratio of the previous two (Tan(x) = Sin(x)/Cos(x)), it's accuracy is dependent on the accuracy of sin(x) and cos(x) which lead it to follow a similar pattern to them.

Exp



The exp function seems to have the same symptomes of the sin and cos functions. This is probably because of how the taylor series works. Once again, when our answers move away from our initial guess, the likelihood of the answer being similar to the correct answer goes down.

Log



Finally we have the log graph. This function is the inverse to e^x. The graph seems to be very big at the beginning and gets smaller towards the end. I think that because this function is an inverse of the last function, the graph for it would be somewhat opposite to e^x.

After writing this write up, I realized that the issue may not have been held within the difference between the input and the initial guess, but just the size of the input. Because doubles only hold approximations of the true numbers they represent, as the inputs get bigger, the true value is less accurate than before.