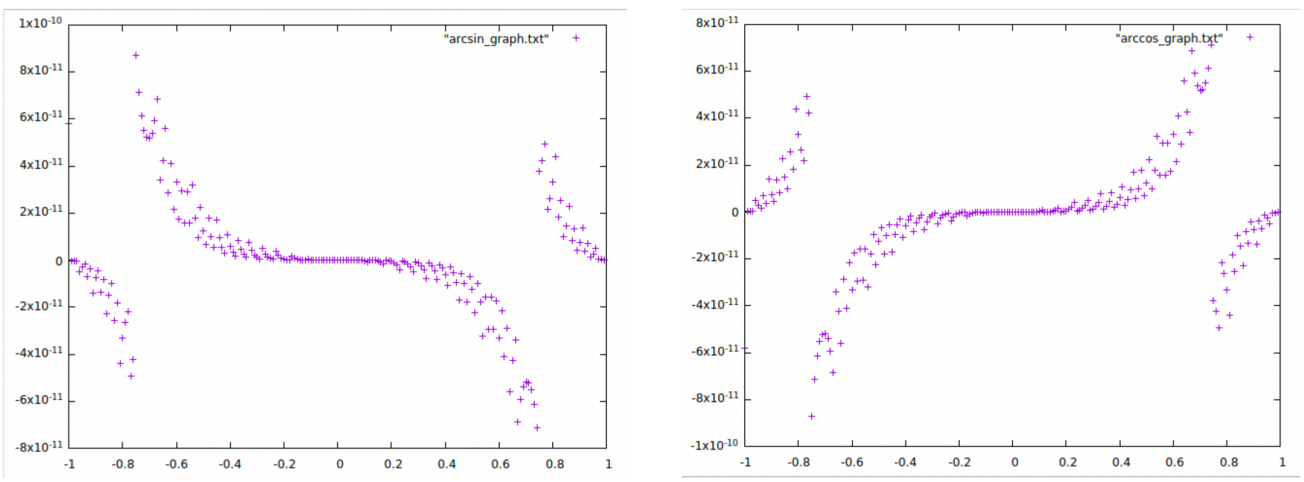
Assignment 2: A Small Numerical Library

Writeup

In this lab, we implemented four math functions and compared it to those in the <math.h> library. I used a Taylor series to compute the three inverse trig approximations and Newton's method for the log function. All four of my functions are accurate to the EPSILON value.

Below are my results and analysis for each function:

<u>Arcsine and Arccosine</u>

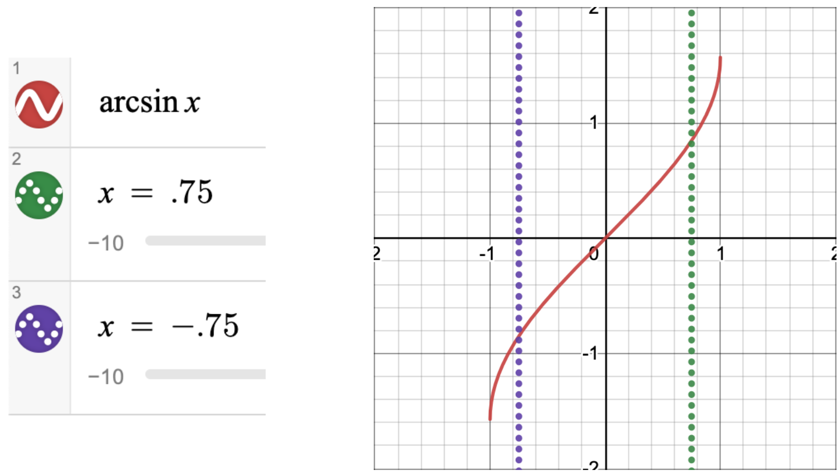


(Graph of arcsine/arccosine function differences using Gnuplot, my function minus library function)

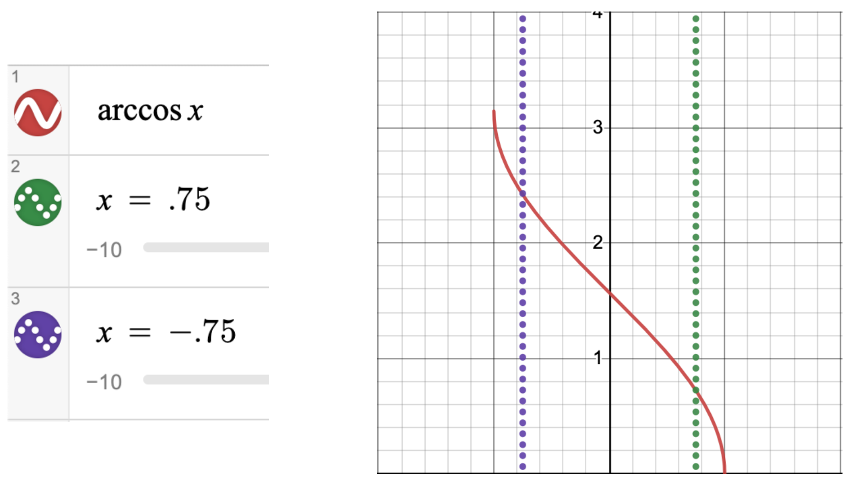
Originally, I had arcsine built only using a Taylor series, which yielded inaccurate results for when |x| approaches 1. So I decided to use an inverse trig identity to calculate the approximation around -1 and 1 to get better results. That would allow for a calculation using a x value far from -1 and 1 with the arccosine function which would be more accurate. In the gnuplot graph, at around |x| = .75 to 1 for arcsine are the areas with greatest difference (thus biggest amount error). It is also the place where we switched to using inverse trig identity to find the calculation, which is the possible cause for the break in the graph.

Similar patterns occur for the arccosine function. Originally, because I built arccosine using the identity: arccos(x) = pi/2 - arcsin(x), the arccosine function was just as inaccurate as the arcsine function at around |x| = 1.

A possible reason for the inaccuracy of the Taylor series at around |x| = 1 is because of the derivative. Taylor series uses derivatives, which is the slope, to calculate an estimation. If we look at the graph below, as x is approaching -1 and 1, the slope becomes very big (vertical) for both arcsine and arccosine, making the Taylor series estimate inaccurate.



(arcsine function, graphed on Desmos)

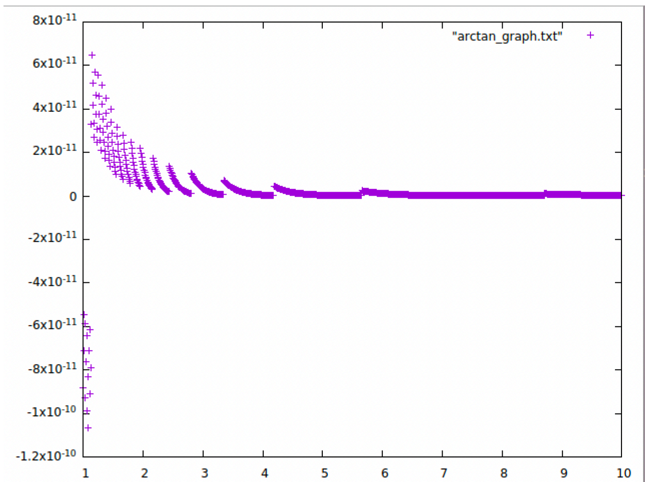


(arccosine function, graphed on Desmos)

Alternatively, when we use the inverse trig identity for arcsine at around |x| = 1, we will be taking the arccosine of an x value that is not close to -1 and 1. Same process for arccosine at around |x| = 1, we will take arcsine of an x value that is not close to -1, 1. When we look at the graphs of arcsine and arccosine, when x is not close to -1 and 1, the function does not have a very large (vertical) slope, so this will be a good estimation for arcsine/arccosine of when |x| is near 1.

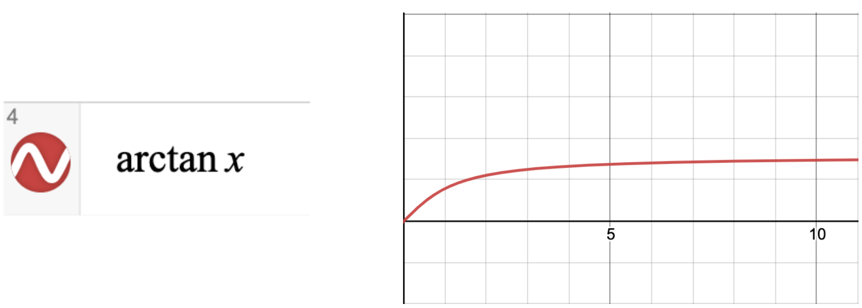
After implementing these changes, my arcsine and arccosine function have very small errors (less than EPSILON), mostly due to limited number of Taylor series terms (the more terms, the more accurate). We stop calculating terms for the Taylor series when the new term is smaller than our given EPSILON.

Arctangent:



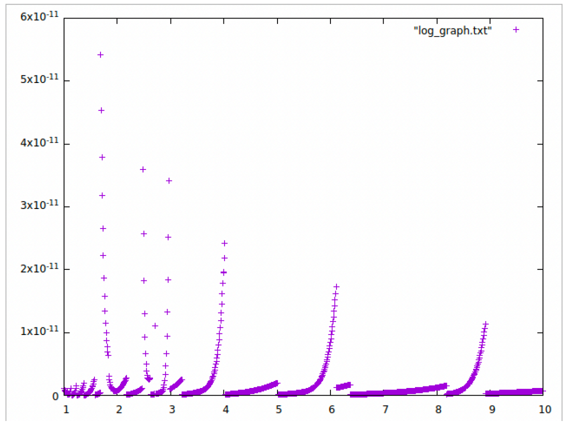
(Graph of arctangent function differences using Gnuplot, my function minus library function)

Arctangent was built also using inverse trig identity: arctan(x) = arcsin(x/sqrt(x^2+1)). So once the errors were minimized on my arcsine function, I didn't need adjust my arctangent function for accuracy. Arctangent didn't have as many issues are arcsine and arccosine because there were no points on a arctangent graph where the slope was undefined. If we look at the differences graph above, most points greater than ~2 were very close to 0. Looking at the below graph for arctangent, we can notice this is due to the very large slope from before x = 1. After the slope tapers to a horizontal asymptote, which is why at around x = 1, the approximation is less accurate, but after that, the inaccuracy is near 0.



(arctangent function, graphed on Desmos)

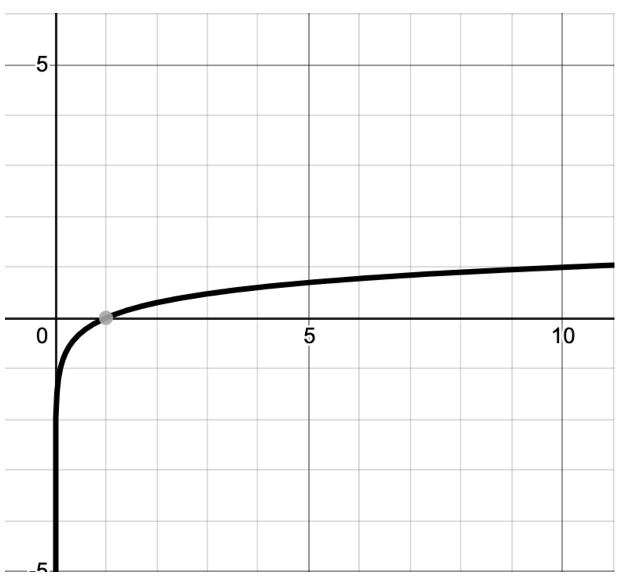
Logarithm



(Graph of log function differences using Gnuplot, my function minus library function)

Arctangent was built using Newton's method. There were no extra steps that I need to take to make my functions's errors less than EPSILON. As you can see from the differences plot, errors were greatest when x is close to 1. This is because the graph for log(x) (below) has a a very large slope before x = 1, which quickly tapers to a small positive slope as x passes 1 and continues to be a small positive slope, and so our log function is very accurate when x is farther from x = 1. The very large slope causes our estimations to be inaccurate.

A possible reason for spikes in this graph is that at those x values, the way the values are represented might not have enough precision, causing a jump in inaccuracy for some numbers.



(logarithm function, graphed on Desmos)