Govind Pillai
gopillai@ucsc.edu
4/18/2021

CSE 13s Spring 2021
Assignment 2: A Small Numerical Library
Lab Writeup

PURPOSE:

The purpose of this lab is to learn how to create a mathematical library that has the implemented functions arcsin, arccos, arctan, and log, by only using Taylor series and Newton's method. The differences between the functions implemented in the program and the functions in math.h were observed and recorded. Because I used a Taylor Series that only terminates once there is a term with a value less than $1x10^{-10}$ in my arcsin, arccos, and arctan functions, I expected there to be no differences between my outputs and that of math.h. Because my log function only terminates once its guess is equal to the correct value with a margin of error of $1x10^{-10}$, I expected there to be no differences between my outputs and that of math.h

METHODS:

arcSin(x):

This function contains a for loop which mimics the taylor series of arcsin. In the loop, the next term of the taylor series is generated based on the previous term. For arcSin(x):

next term = current term * x^2 * $(k^2/(k+1)(k+2))$

Where k = 1 for the first non-trivial term in the series (2nd term in the series) and is incremented by 2 for each term.

A taylor series continues to grow with the addition of these terms until the value of a term is less than $1x10^{-10}$.

arcCos(x):

This function uses arcSin in order to calculate the value of arcCos(x). For arcCos(x):

arcCos(x) = (pi/2) - (arcsin(x))

arcTan(x):

This function uses arcSin in order to calculate the value of arcTan(x). For arcTan(x):

arcTan(x) = arcSin(x/sqrt(x^2+1))

Log(x):

This function uses Newton's method in order to obtain an accurate guess of the value of Log(x). It starts off with the guess y=1. If e^{guess} - x is more than $1x10^{-10}$ the function refines its guess using Newton's method. Newton's method states:

$y_{k+1}= y_k + (f(y_k)/f'(y_k))$

Using the function $f(x) = e^y-x$, we plug this into Newton's method. For Log(x):

new guess = current guess + $((x-e^y)/e^y)$

Results:

ArcSin(x), ArcCos(x), ArcTan(x):

Both these functions provided similar differences when running them against the <math.h> file. When I first implemented these functions I observed a high margin of error closer to the edges of the interval. This is because while the taylor series for arcsin and arccos will ultimately converge, they converge very slowly. Therefore, I used the trig identity:

arcsin(x) = arccos(sqrt(1-x^2)) for 0.75 <= x <= 1 (eq. 1)

You can multiply this value by -1 for the interval -1 <= x <= -0.75. For arccos(x) I used the trig identity:

arccos(x) = arcsin(sqrt(1-x^2)) for 0.75 <= x <= 1 (eq. 2)

To get the values for the interval -1 <= x <= -0.75, you can reflect the identity across the x-axis by multiplying it by -1 and then shifting it up by pi units. After adding these identities to my functions, the margins of error decreased significantly. The current differences are graphed below. As you may notice, I still see a relatively big error for x-values 0.7 and -0.7. I think this is because arcsin(0.7) and arccos(0.7) have very similar values:

arcsin(sqrt(2)/2) = arccos(sqrt(2)/2)

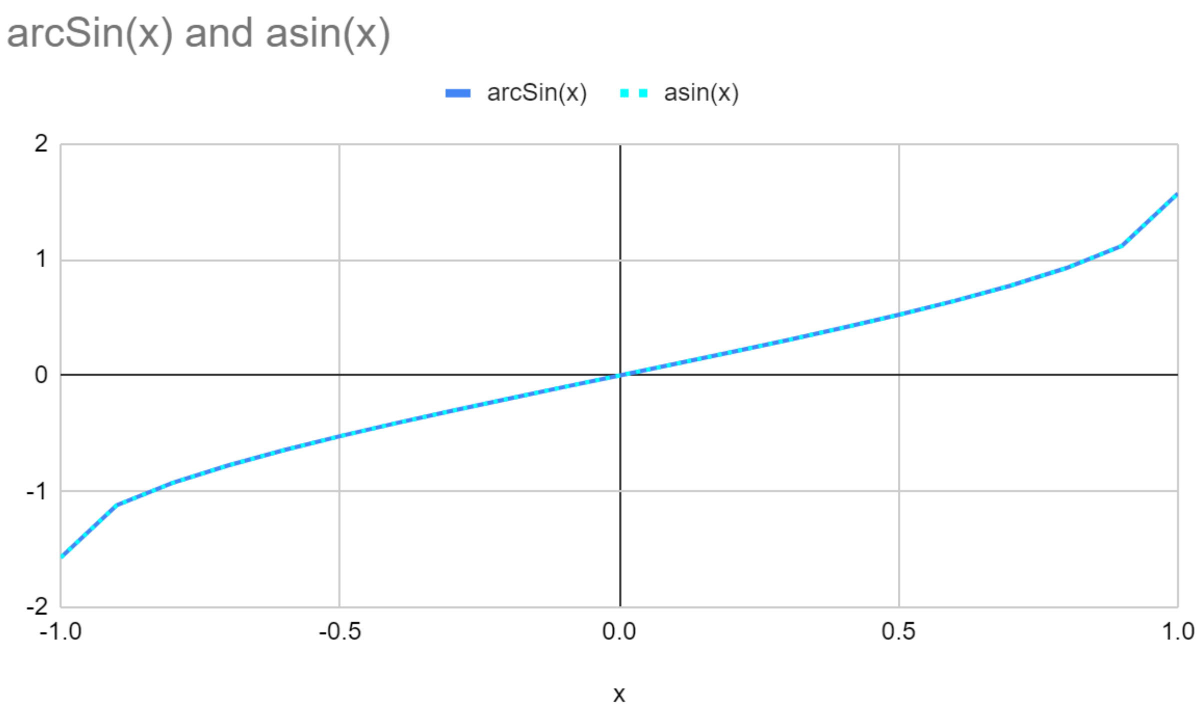
Sqrt(2)/2 equals 0.707 which is awfully close to 0.7. Therefore trying to use the identities shown in equation 1 and 2 will have little effect on the returned value. This is also why arctan experiences a high margin of error towards the start of its interval:

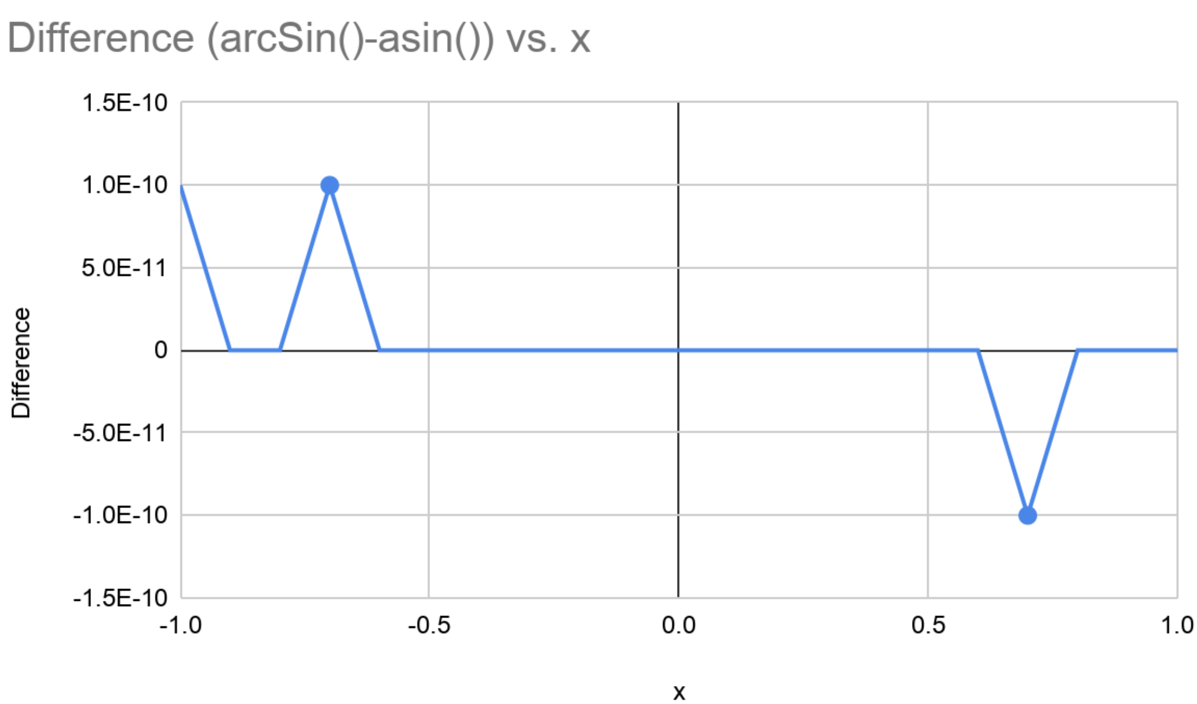
arctan(1) = arcsin(1/sqrt(1+x^2)) = arcsin(1/sqrt(2)) = arcsin(sqrt(2)/2)

Log(x) and Final Thoughts:

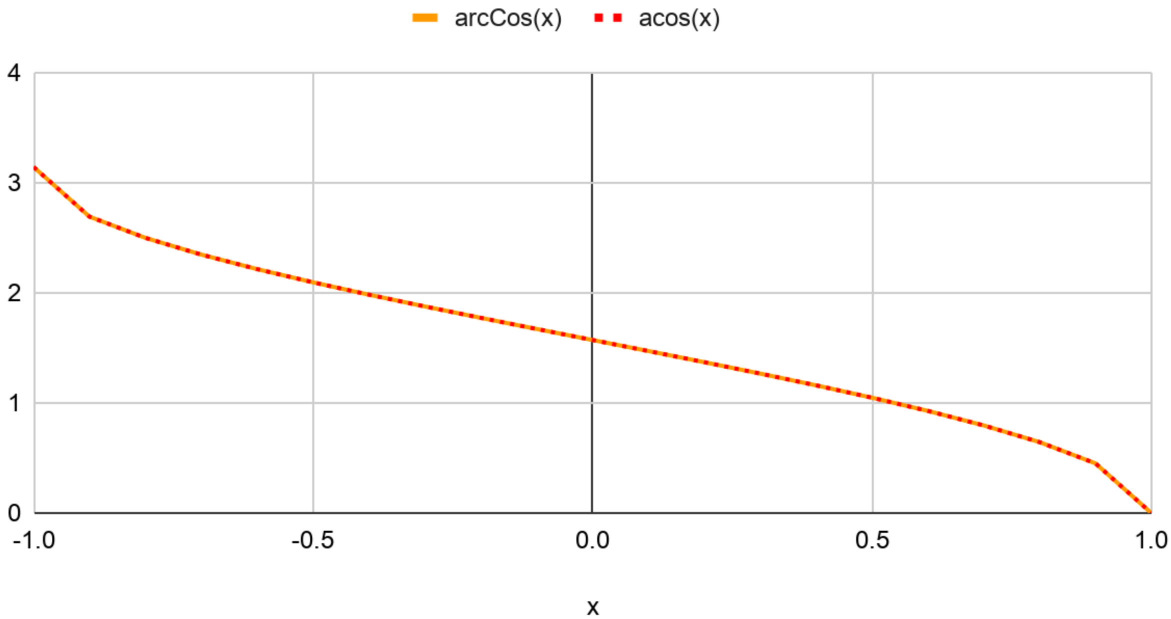
My Log(x) function had no difference up to 10 decimal places with the log(x) function in <math.h>. I suspect this is because Log(x) used Newton's method instead of a Taylor series. Therefore, it did not observe a slow convergence and was able to make a more accurate guess.

GRAPHS:

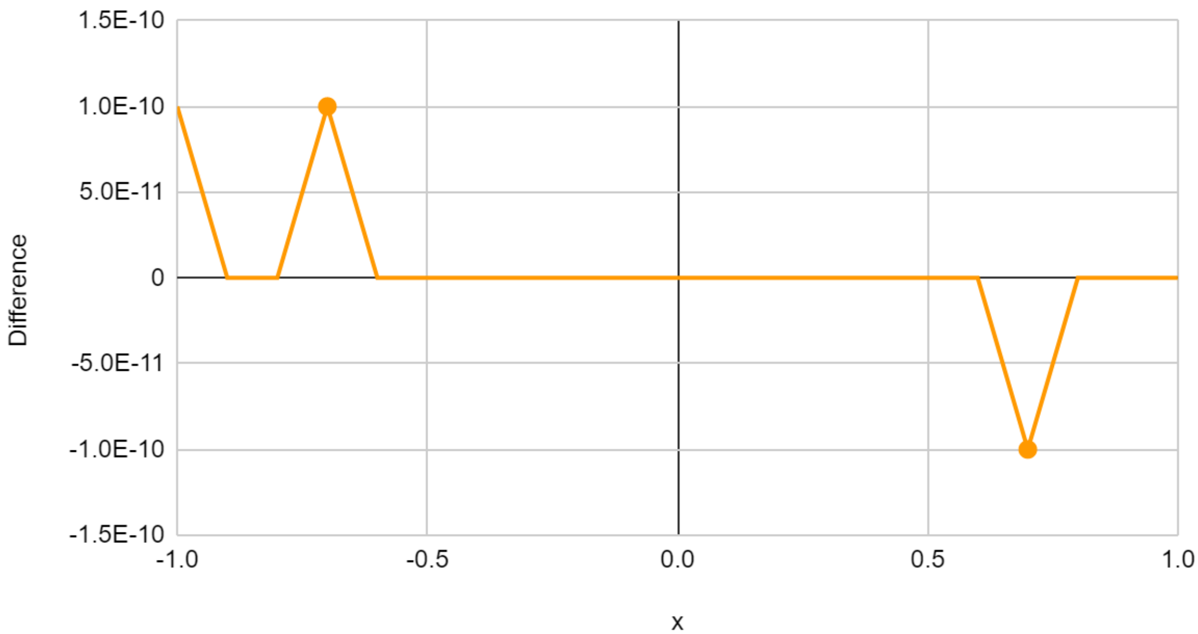

arcSin(x) and asin(x)
arcSin(x)
asin(x)
2
1
0
-1
-2
-1.0
-0.5
0.0
0.5
1.0
x

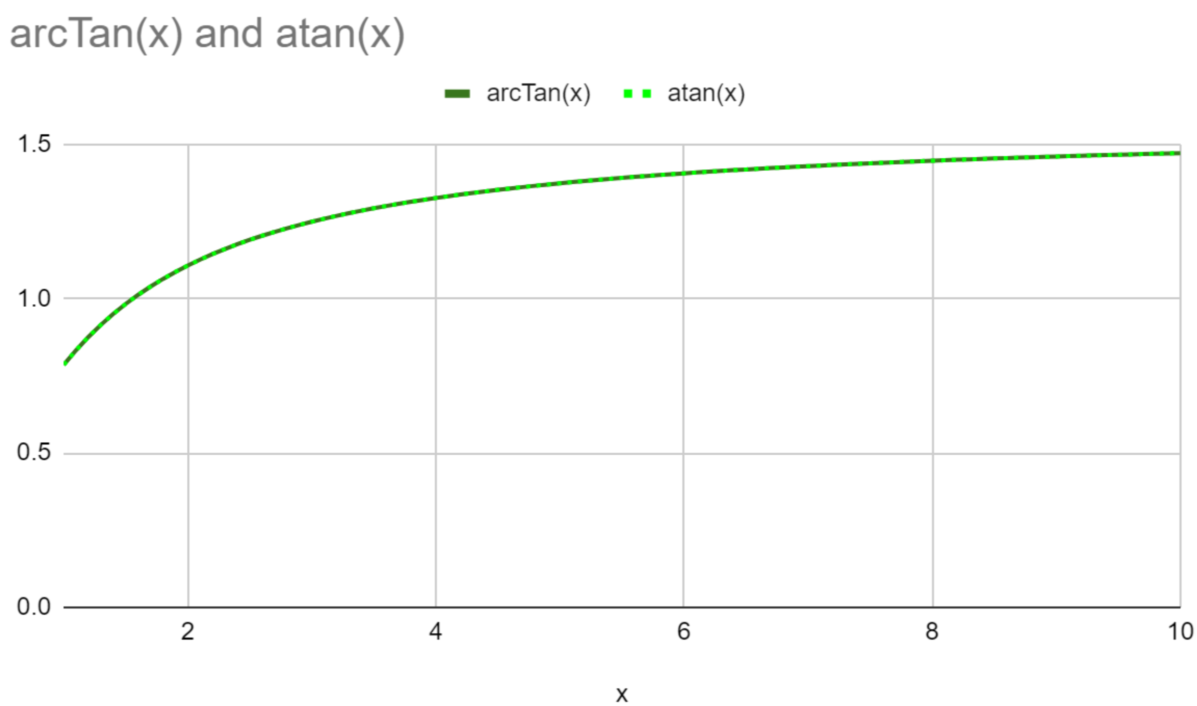

Difference (arcSin()-asin()) vs. x
Difference
1.5E-10
1.0E-10
5.0E-11
0
-5.0E-11
-1.0E-10
-1.5E-10
-1.0
-0.5
0.0
0.5
1.0
x

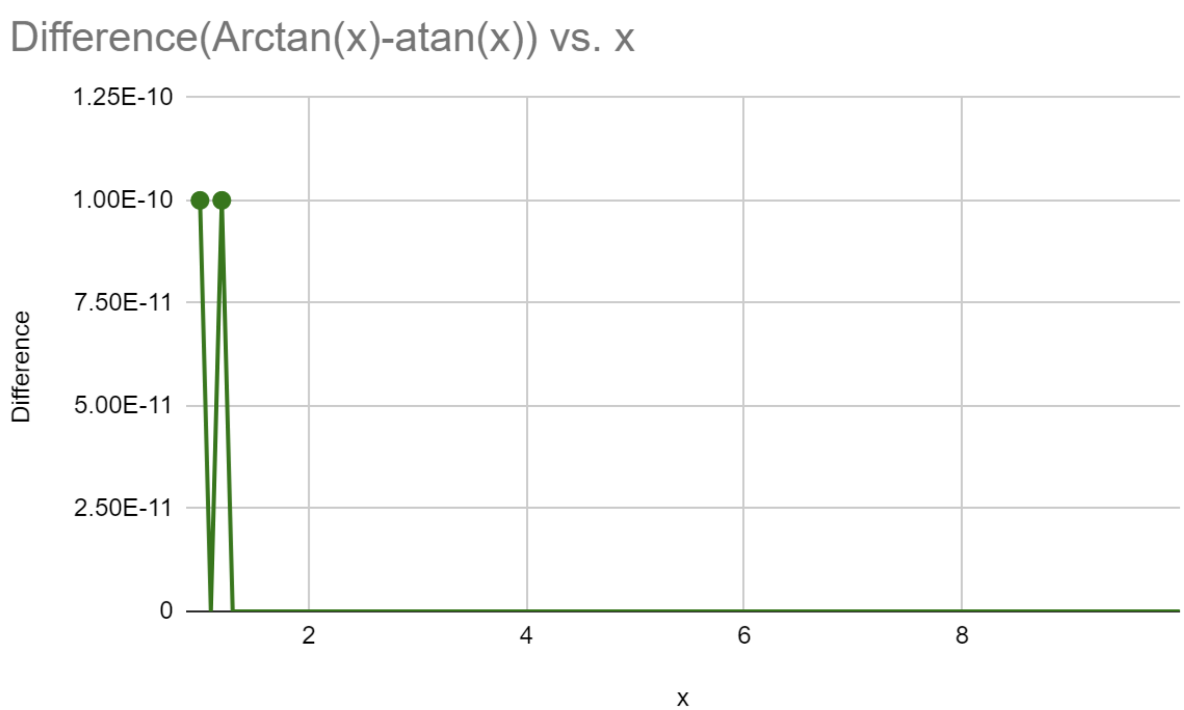
arcCos(x) and acos(x)

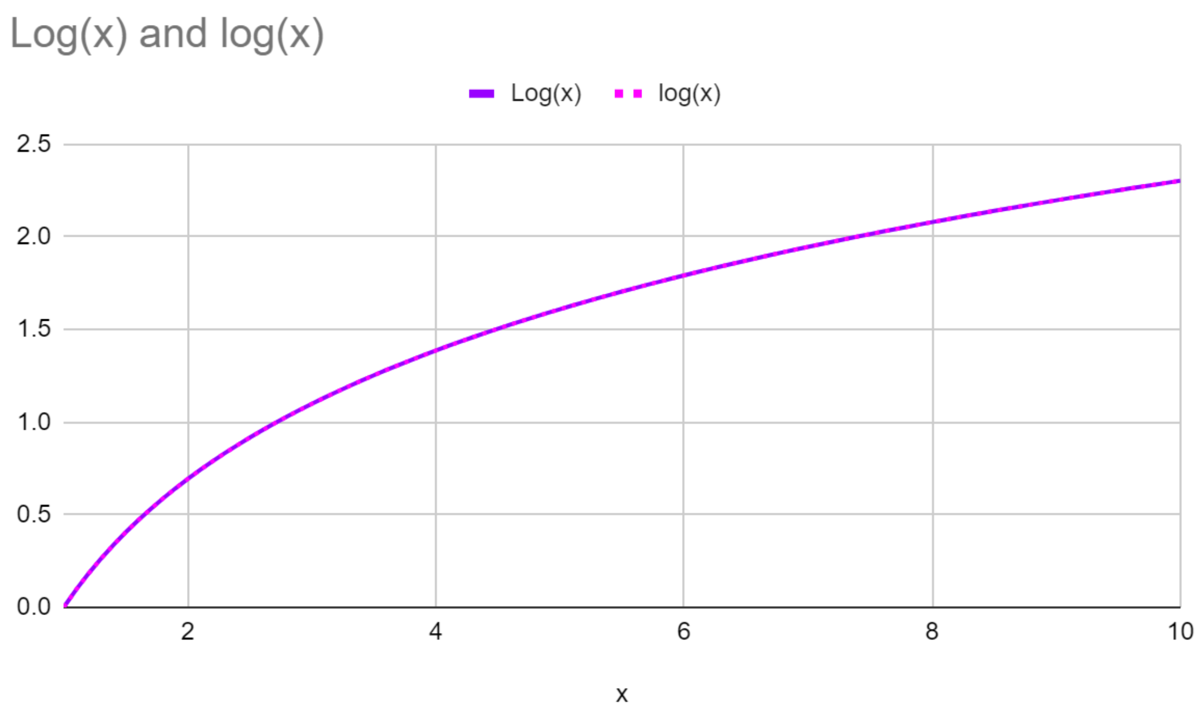


Difference (arcCos()-acos()) vs. x




arcTan(x) and atan(x)
arcTan(x)
atan(x)
1.5
1.0
0.5
0.0
2
4
6
8
10
x


Difference(Arctan(x)-atan(x)) vs. x
1.25E-10
1.00E-10
7.50E-11
5.00E-11
2.50E-11
0
Difference
2
4
6
8
x



Log(x) and log(x)
Log(x)
log(x)
2.5
2.0
1.5
1.0
0.5
0.0
2
4
6
8
10
x