Fabrice Kurmann
fkurmann@ucsc.edu
16 April, 2021

CSE 13S Spring 2021
Assignment 2: A Small Numerical Library
Write-up

I used a Taylor series (specifically the Taylor series for arcsin, with modified inputs/adjusted output for the other trig functions) to compute arcsin, arccos, and arctan for this assignment and Newton's method to compute log.

Problem Analysis 1:

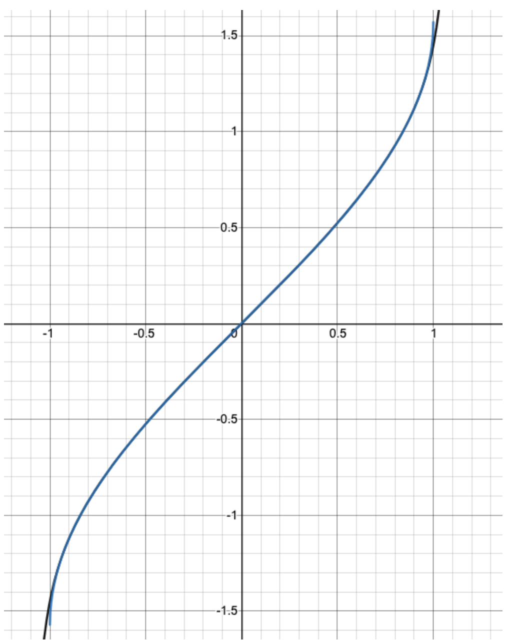
As I was expecting based on the Piazza discussion, my original implementation of arcsin would struggle to deal with the case where |x| was very close to 1. I decided to implement the function regardless, get an output, and deal with this case after. This allowed my to capture the results of this problem:

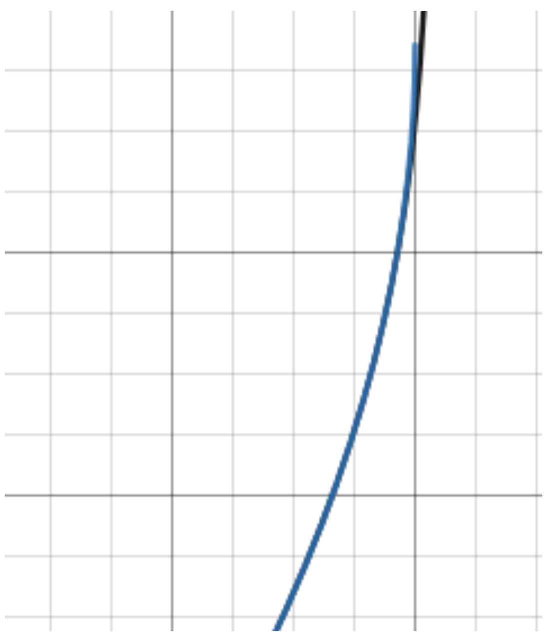
x	arcCos	Library	Difference
-	------	-------	----------
-1.0000	3.13388923	3.14159265	-0.0077034279
-0.9000	2.69056584	2.69056584	-0.0000000000
-0.8000	2.49809154	2.49809154	-0.0000000000
-0.7000	2.34619382	2.34619382	-0.0000000000
-0.6000	2.21429744	2.21429744	-0.0000000000
-0.5000	2.09439510	2.09439510	-0.0000000000
-0.4000	1.98231317	1.98231317	-0.0000000000
-0.3000	1.87548898	1.87548898	-0.0000000000
-0.2000	1.77215425	1.77215425	-0.0000000000
-0.1000	1.67096375	1.67096375	0.0000000000
-0.0000	1.57079633	1.57079633	0.0000000000
0.1000	1.47062891	1.47062891	-0.0000000000
0.2000	1.36943841	1.36943841	0.0000000000
0.3000	1.26610367	1.26610367	0.0000000000
0.4000	1.15927948	1.15927948	0.0000000000
0.5000	1.04719755	1.04719755	0.0000000000
0.6000	0.92729522	0.92729522	0.0000000000
0.7000	0.79539883	0.79539883	0.0000000000
0.8000	0.64350111	0.64350111	0.0000000000
0.9000	0.45102681	0.45102681	0.0000000000
1.0000	0.00770343	0.00000002	0.0077034068

Left is the output for arcCos. It and arcSin (arcCos is based off arcSin) did indeed have the predictable error near |1|.

Interestingly, despite also relying on arcSin, I noticed that my arcTan function was accurate across the entire range [1,10].

To get a better understanding of why this was the case, I graphed arcsin, and it's Taylor series to various term limits. Below are the results:





Left is the comparison of arcsin (blue) and my Taylor series with 20 terms. Note, 5 terms is already enough to make the graphs visually indistinguishable on [-0.7,0.7].

As is clear, towards |1|, the graphs diverge. Right is the Taylor series taken to 80 terms, and still the graphs diverge.

It was now clear, the Taylor series for arcSin could not give accurate results near 1.

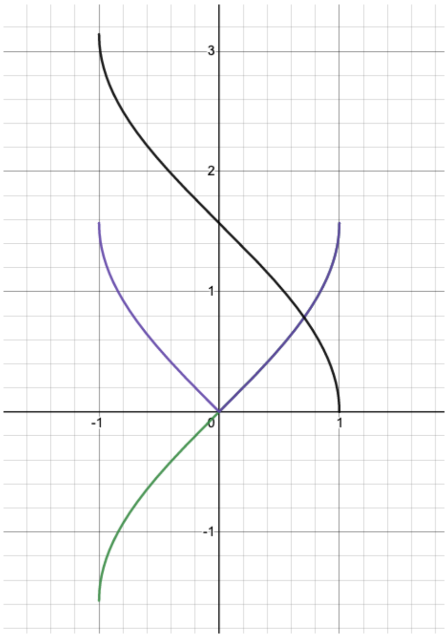
Incidentally, this realization also explains why arcTan, despite piggybacking off of the same flawed arcSin function worked fine: arcTan's input to arcsin was always: $\left(\frac{x}{\sqrt{x^2+1}}\right)$ On the entire range [1,10] this never approaches But doesn't come close enough to 1 to cause problems:

x	1	5	10
$\left(\frac{x}{\sqrt{x^2+1}}\right)$	0.7	0.98	0.995

I then focused on examining how the trigonometric identities provided on Piazza for exactly this error might be of use.

$$\sin^{-1}(x) = \cos^{-1}\left(\sqrt{1-x^2}\right) \quad 0 \le x \le 1$$

$$\cos^{-1}(x) = \sin^{-1}\left(\sqrt{1-x^2}\right) \quad 0 \le x \le 1$$



Left are the graphs of arccos (black), arccos($\sqrt{1-x^2}$) (purple), and arcsin (green).

Indeed using this identity, arccos could be made to mirror arcsin, I would simply have to account for the negative output given X = [-1,0) manually.

Furthermore, using $\sqrt{1-x^2}$ meant I would no longer have to plug values close to 1 into arcSin:

X	$\sqrt{1-x^2}$
1	0
0.99	0.0199
0.9	0.19

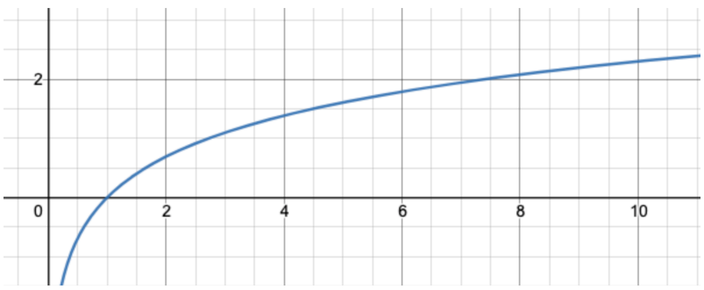
$$\operatorname{arcSin}(x) = \operatorname{arcCos}\left(\frac{x}{\sqrt{1-x^2}}\right)$$

$$=$$

$$\operatorname{arcSin}(x) = \frac{\pi}{2} - \operatorname{arcSin}\left(\frac{x}{\sqrt{1-x^2}}\right)$$

Having already concluded that arcSin was accurate near 0, these adjusted values would work perfectly. However, since my implementation of arcCos was based on arcSin, could I do the substitution. Yes, since arcCos = Pi/2 - arcSin, I simply implemented arcSin recursively for these edge cases.

This solved my problem with arcSin near |1|. But, to examine just why this problem happened for the arcSin Taylor series and not that for arcTan, or for that matter log, which used a slightly different method, I graphed some more functions:



Here is arctan (red). Unlike arcsin, it is continuous on the entire range of X values. Thus, a Taylor series to map values [1,10] as the assignment specified would not approach any boundaries of continuity - the areas where arcSin's Taylor struggled the most. — This begs the question, would it have been simpler to make an arcTan function and use that to find both arcCos and arcSin? Perhaps.

The same goes for log (blue). Being continuous and easy to map through an interative method, in this case Newton's Method, seems to be a trend.

Problem Analysis 2:
After having fixed the problem with arcSin close to 1, my functions were within a range or 1*10^-10 of the library functions. In fact, by setting my EPSILON value to 1*10^-11, the error was reduced to show 10 zeros, obviously at the cost of more iteration.

Below are some sample outputs. Log shows no error at all, arcSin, arcCos, and arcTan are the ones that still have tiny differences:

Most interesting is that the differences between arcSin and asin and arcCos and acos appear in the same spots. Perhaps unsurprising given these function differ only by a constant Pi/2, yet still notable.

x	arcSin	Library	Difference
-1.0000	-1.57079633	-1.57079633	0.0000000001
-0.9000	-1.11976952	-1.11976951	-0.0000000000
-0.8000	-0.92729522	-0.92729522	0.0000000001
-0.7000	-0.77539750	-0.77539750	0.0000000001
-0.6000	-0.64350111	-0.64350111	0.0000000000
-0.5000	-0.52359878	-0.52359878	0.0000000000
-0.4000	-0.41151685	-0.41151685	0.0000000000
-0.3000	-0.30469265	-0.30469265	0.0000000000
-0.2000	-0.20135792	-0.20135792	0.0000000000
-0.1000	-0.10016742	-0.10016742	0.0000000000
-0.0000	-0.00000000	-0.00000000	0.0000000000
0.1000	0.10016742	0.10016742	-0.0000000000
0.2000	0.20135792	0.20135792	-0.0000000000
0.3000	0.30469265	0.30469265	-0.0000000000
0.4000	0.41151685	0.41151685	-0.0000000000
0.5000	0.52359878	0.52359878	-0.0000000000
0.6000	0.64350111	0.64350111	-0.0000000000
0.7000	0.77539750	0.77539750	-0.0000000001
0.8000	0.92729522	0.92729522	-0.0000000001
0.9000	1.11976952	1.11976951	0.0000000000
1.0000	1.57079631	1.57079631	-0.0000000000

x	arcCos	Library	Difference
-1.0000	3.14159265	3.14159265	-0.0000000001
-0.9000	2.69056584	2.69056584	0.0000000000
-0.8000	2.49809154	2.49809154	-0.0000000001
-0.7000	2.34619382	2.34619382	-0.0000000001
-0.6000	2.21429744	2.21429744	-0.0000000000
-0.5000	2.09439510	2.09439510	-0.0000000000
-0.4000	1.98231317	1.98231317	-0.0000000000
-0.3000	1.87548898	1.87548898	-0.0000000000
-0.2000	1.77215425	1.77215425	-0.0000000000
-0.1000	1.67096375	1.67096375	-0.0000000000
-0.0000	1.57079633	1.57079633	0.0000000000
0.1000	1.47062891	1.47062891	0.0000000000
0.2000	1.36943841	1.36943841	0.0000000000
0.3000	1.26610367	1.26610367	0.0000000000
0.4000	1.15927948	1.15927948	0.0000000000
0.5000	1.04719755	1.04719755	0.0000000000
0.6000	0.92729522	0.92729522	0.0000000000
0.7000	0.79539883	0.79539883	0.0000000001
0.8000	0.64350111	0.64350111	0.0000000001
0.9000	0.45102681	0.45102681	-0.0000000000
1.0000	0.00000002	0.00000002	0.0000000000

x	arcTan	Library	Difference
1.0000	0.78539816	0.78539816	-0.0000000001
1.1000	0.83298127	0.83298127	-0.0000000001
1.2000	0.87605805	0.87605805	-0.0000000001
1.3000	0.91510070	0.91510070	-0.0000000001
1.4000	0.95054684	0.95054684	0.0000000000
1.5000	0.98279372	0.98279372	0.0000000000
1.6000	1.01219701	1.01219701	0.0000000000
1.7000	1.03907226	1.03907226	0.0000000000
1.8000	1.06369782	1.06369782	0.0000000000
1.9000	1.08631840	1.08631840	0.0000000000
2.0000	1.10714872	1.10714872	0.0000000000
2.1000	1.12637712	1.12637712	0.0000000000
2.2000	1.14416883	1.14416883	0.0000000000
2.3000	1.16066899	1.16066899	0.0000000000
2.4000	1.17600521	1.17600521	0.0000000000

Left, the arcSin and arcCos differences, note how they are exact negatives of each other - unsurprising given the function's construction.

Above, the arcTan difference, only for the first 4 terms is there an error, then the function stays aligned with that from the library.

What is interesting about these errors is that they all point to a lack of precision of the arcSin function around |0.75|, hence |0.7 and 0.8| are constantly getting an error.

Indeed the arcTan error stems from this exact same issue: The area where arcTan is struggling is 1 to 1.3, which is where that function would send to the arcSin function values of 0.7 to 0.8. Definitely not a coincidence.

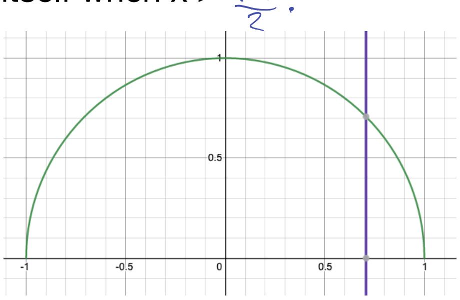
Why was I getting error at 0.8, but not at 0.9? The answer is quite simple. In my arcsin function, I used X > 0.8 as the threshold for when to use the arccosine identity that solved the issue of errors at |1|. Up to that threshold, I was getting tiny errors that stemmed from the diversion at |1|. This is why at 0.9, those errors disappear.

So, a solution could be to lower the threshold to 0.6, so that I could catch the error at 0.7 and 0.8.

However, this is not as simple as it seems. Lowering the threshold to use $(\sqrt{1-x^2})$ Could result in infinite recursion: Here's why:

My code runs arcSin($(\sqrt{1-x^2})$) recursively as long as X > Threshold. However, only after running the base case, which happens once $(\sqrt{1-x^2}) <$ Threshold, does the recursion stop.

When graphing $(\sqrt{1-x^2})$, the problem becomes clear. This function only yields a result smaller than itself when x > $\frac{\sqrt{2}}{2}$.



To the left is the graph for $(\sqrt{1-x^2})$, a semicircle. In in purple is x = $\frac{\sqrt{2}}{2}$

All inputs to the right of this purple line would cause infinite recursion, so I set my threshold right at the line. This had the effect of reducing error in all three of my trig functions, especially arcTan.

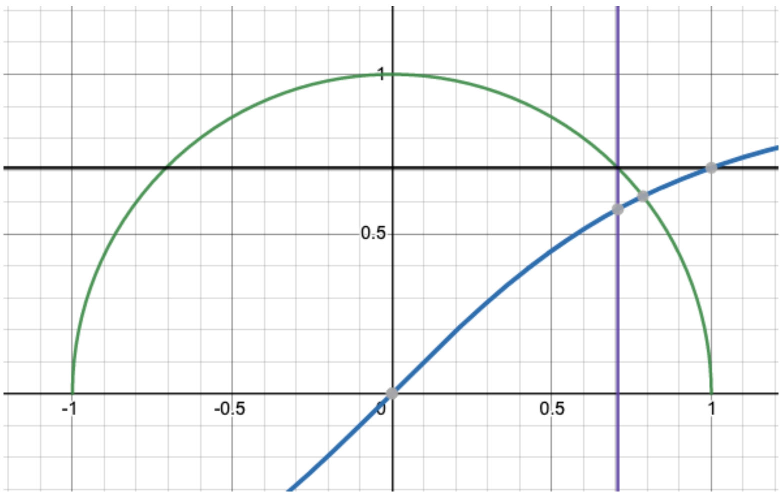
x	arcSin	Library	Difference
-1.0000	-1.57079633	-1.57079633	0.0000000001
-0.9000	-1.11976952	-1.11976951	-0.0000000000
-0.8000	-0.92729522	-0.92729522	-0.0000000000
-0.7000	-0.77539750	-0.77539750	0.0000000001
-0.6000	-0.64350111	-0.64350111	0.0000000000
-0.5000	-0.52359878	-0.52359878	0.0000000000
-0.4000	-0.41151685	-0.41151685	0.0000000000
-0.3000	-0.30469265	-0.30469265	0.0000000000
-0.2000	-0.20135792	-0.20135792	0.0000000000
-0.1000	-0.10016742	-0.10016742	0.0000000000
-0.0000	-0.00000000	-0.00000000	0.0000000000
0.1000	0.10016742	0.10016742	-0.0000000000
0.2000	0.20135792	0.20135792	-0.0000000000
0.3000	0.30469265	0.30469265	-0.0000000000
0.4000	0.41151685	0.41151685	-0.0000000000
0.5000	0.52359878	0.52359878	-0.0000000000
0.6000	0.64350111	0.64350111	-0.0000000000
0.7000	0.77539750	0.77539750	-0.0000000001
0.8000	0.92729522	0.92729522	0.0000000000
0.9000	1.11976952	1.11976951	0.0000000000
1.0000	1.57079631	1.57079631	-0.0000000000

x	arcCos	Library	Difference
-1.0000	3.14159265	3.14159265	-0.0000000001
-0.9000	2.69056584	2.69056584	0.0000000000
-0.8000	2.49809154	2.49809154	0.0000000000
-0.7000	2.34619382	2.34619382	-0.0000000001
-0.6000	2.21429744	2.21429744	-0.0000000000
-0.5000	2.09439510	2.09439510	-0.0000000000
-0.4000	1.98231317	1.98231317	-0.0000000000
-0.3000	1.87548898	1.87548898	-0.0000000000
-0.2000	1.77215425	1.77215425	-0.0000000000
-0.1000	1.67096375	1.67096375	-0.0000000000
-0.0000	1.57079633	1.57079633	0.0000000000
0.1000	1.47062891	1.47062891	0.0000000000
0.2000	1.36943841	1.36943841	0.0000000000
0.3000	1.26610367	1.26610367	0.0000000000
0.4000	1.15927948	1.15927948	0.0000000000
0.5000	1.04719755	1.04719755	0.0000000000
0.6000	0.92729522	0.92729522	0.0000000000
0.7000	0.79539883	0.79539883	0.0000000001
0.8000	0.64350111	0.64350111	-0.0000000000
0.9000	0.45102681	0.45102681	-0.0000000000
1.0000	0.00000002	0.00000002	0.0000000000

x	arcTan	Library	Difference
1.0000	0.78539816	0.78539816	-0.0000000001
1.1000	0.83298127	0.83298127	0.0000000000
1.2000	0.87605805	0.87605805	0.0000000001
1.3000	0.91510070	0.91510070	0.0000000000
1.4000	0.95054684	0.95054684	0.0000000000
1.5000	0.98279372	0.98279372	0.0000000000
1.6000	1.01219701	1.01219701	0.0000000000
1.7000	1.03907226	1.03907226	0.0000000000
1.8000	1.06369782	1.06369782	0.0000000000
1.9000	1.08631840	1.08631840	0.0000000000
2.0000	1.10714872	1.10714872	0.0000000000
2.1000	1.12637712	1.12637712	0.0000000000
2.2000	1.14416883	1.14416883	0.0000000000
2.3000	1.16066899	1.16066899	0.0000000000
2.4000	1.17600521	1.17600521	0.0000000000
2.5000	1.19028995	1.19028995	0.0000000000
2.6000	1.20362249	1.20362249	0.0000000000

Circled are all the places where error used to exist but has been eradicated simply by moving my threshold for using the arcsin/ arccos identity as low as possible.

Most interesting to me is the way arcTan improved. Again, not surprising given that right around values of 1.3, where the arcTan function fed arcSin an input of ~0.75, the new approach payed off.



ArcTan curve (blue). Interesting to see how it hits y = $\frac{\sqrt{2}}{2}$, the critical value, right at 1. This is a level of interconnectedness I was not expecting to encounter beginning this assignment.