Govind Pillai
gopillai@ucsc.edu
4/15/2021

CSE 13s Spring 2021
Assignment 2: A Small Numerical Library
Design Document

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 the four basic mathematical operations (addition, subtraction, multiplication, and division). A test bench is also implemented to see how accurate the functions in the small numerical library are compared to the functions given in <math.h>. In addition to this, makefiles and header files are also used.

Design Summary:

Because Arccos and Arctan values can be solved using the Arcsin function, much of the design was working on the Arcsin function. The way I created the function presented in pseudocode below, was finding a way to make the first non-trivial term of Arcsin's Taylor Series become the second non-trivial term. The first non-trivial term in the Taylor series is the second term of the series and it is:

$(½) * (x^3/3)$

The second non-trivial term in the Taylor series is the third term of the series and it is:

$(½)*(¾)*(x^5/5)$

What I first noticed was that the exponent of x was incremented by 2 for each term. So all I had to do was multiply the previous term by x^2. After that I noticed the (½) in both terms so I didn't have to change that. I needed to multiply (¾) to the 2nd term of the series. If k=1 for the second term of the series and I increment k by 2 for each term, then for the third term of the series k would be 3. Therefore, the (¾) could be represented as (k/k+1). Now that I figured out the (½), the x^5, and the (¾), I had to divide by 5. This was easy because k =3 so I just divided by (k+2). The last part is tricky. In the first non-trivial term, x^3 is divided by 3. That division is not present in the second non-trivial term. Therefore I needed to multiply by 3, or k, to cancel out that operation. In the end it came to:

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

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

After testing, this equation was found to have worked for all terms of the taylor series. This function was then used to design Arccos and Arctan. The log function did not use a taylor series but instead used Newton's formula. I store 1 in one variable which then stores a first guess of e in another variable. Then Newton's formula is used to refine the guess until it's error is within epsilon.

Taylor Series:

Arcsin:

$$\sum_{k=0}^{\infty} \frac{(2k)!}{2^{2k}(k!)} \frac{x^{2k+1}}{2k+1}$$

$x + (x^3/6) + (3x^5/40) + (5x^7/112) + \ldots$

Arccos:

$$\frac{\pi}{2} - \sum_{k=0}^{\infty} \frac{(2k)!}{2^{2k}(k!)} \frac{x^{2k+1}}{2k+1}$$

$(\pi/2) - (x + (x^3/6) + (3x^5/40) + (5x^7/112) + \ldots)$

Arctan:

$$\arcsin\left(\frac{x}{\sqrt{x^2+1}}\right)$$

OR

$$\arcsin\left(\frac{1}{\sqrt{x^2+1}}\right)$$

PSEUDOCODE:

mathlib.c:

```
arcsin(double x, double epsilon) {
        double current_sum = current sum of taylor approximation (initial value: x);
        double current_term = current term of taylor approximation(initial value: x);
        for (k = 1; increments by 2 until current_term is less than epsilon) {
                Current_term = current_term * x^2 * (k^2/((k+1)(k+2));
                current_sum += current_term;
        }
        return current_sum;
}
arccos(double x, double epsilon) {
        return (pi/2) - arcsin() function ;
}
arctan(double x, double epsilon) {
        Return the  arcsin of (x/sqrt(x^2+1)) using sqrt helper;
log(double x) {
        set k equal to 1.0;
        Set m equal to e^k using Exp helper;
        while(absolute_value(m-x) bigger than Epsilon) {
                Add k + (( m-x)/x) back to k
        }
        return k;
}
```