Rahul Vaidun
rvaidun@ucsc.edu
1 April 2021

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

This program utilizes Newton's method to calculate the value the inverse trig functions and log functions

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Newton's Method

Estimate roots of a function

$\sin^{-1}$ with Newton's method

$x = \sin^{-1}(a)$ $\quad$ $f(x) = \sin(x) - a$

$\sin(x) = a$ $\quad$ $x_{n+1} = x_n - \left(\frac{\sin(x_n) - a}{\cos(x_n)}\right)$

$\sin(x) - a = 0$

$\cos^{-1}$

$\cos^{-1}$ can be implemented by doing:

$\frac{\pi}{2} - \arcsin(x)$

$\tan^{-1}$ can be implemented by doing:

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

log

$x = \ln(a)$ $\quad$ $f(x) = \cancel{\sin}\ e^x - a$

$a = e^x$ $\quad$ $x_{n+1} = x_n - \left(\frac{(e^x - a)}{e^x}\right)$

$e^x - a = 0$

For arcsin when x gets closer to the edges (1 and -1) the answer rapidly loses accuracy. To combat this when the absolute value of x greater than 0.9 then solve it using trig identities and arccos. arcsin(x) = arccos(sqrt(1-x^2)), 0 <= x <= 1. If x is negative then return the negative of the answer

I. Pseudocode

```
double arcSin(double x) {
    double a;
    if x is an edge case(absolute value close to 1) {
        double a = sqrt(1-(x*x)
    } else {
        a = x;
    }
    double answer = a;
    while abs(answer) -a > epsilon {
        answer = answer - ((sin(answer) - a) / cos(answer));
    }
    if (Abs(x) > 0.9) {
        If x < 0 return -((PI / 2) - answer) else return (PI / 2) - answer
    } else {
        return answer;
    }

}
double arcCos(double x) {
    return (PI/2)-arcsin(x)
}

double arcTan(double x) {
    return arcsin(x/sqrt((x^2)+1))

}

double Log(double x) {
    double answer = a;
    while (abs(e^x - a) > epsilon) {
        answer = answer - ((e^x - a) / e^x);
    }
    return answer;
}
```