Fabrice Kurmann
fkurmann@ucsc.edu
12 April, 2021

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

Assignment: Create a numerical library (arcsin, arccos, arctan, log). Use taylor series approximation or the inverse method for the first three and Newton's method (for e^x inversed) for log.

Create a test harness in which this library will be compared to math.h

General Guidelines:

Test arcsin, arccos for [-1,1), step = 0.1
Test arctan and log for [1, 10), step = 0.1

Separate functions for each implementation

- arcSin()
- arcCos()
- arcTan()
- Log()

Halt computation at an epsilon value of 10^-10

Math, functions, and pseudocode:

Log using Newton's method:

Newton's method in general: $X_{K+1} = X_K - \frac{f(x_K)}{f'(x_K)}$

$$X_{K+1} = X_K + \frac{Y - e^{x_K}}{e^{x_K}}, \quad Y = \text{initial input, point to solve for}$$

Ex) input = 2

should approach 0.693147

$X_0 = 1$

$x_1 = \left(1 + \frac{2 - e^1}{e^1}\right) = 0.735$

$x_2 = \left(0.735 + \left(\frac{2 - e^{0.735}}{e^{0.735}}\right)\right) = 0.69477$

$x_3 = \left(0.69477 + \left(\frac{2 - e^{0.6948}}{e^{0.6948}}\right)\right) = 0.69312$

Convergence !

Taylor approximation pseudocode for Log:

log (double Y):
epsilon = 10^{-10} ______ accuracy threshold
______ step size of a particular iteration
└ (once smaller than ϵ, we have precision)
value = 1 ______ running value of function
└ initially 1 (on iteration 0)
for (int K = 1   step_size > epsilon , K += 1): ______ K = step counter

$$\text{step\_size} = \left(Y - e^{\text{value}}\right) / e^{\text{value}}$$ * — Exp(function needed)

$$\text{Value} = \text{value} + \text{step\_size}$$

return value

ArcSin (and arcCos) using Taylor series:

Taylor series for arcsin centered at 0:

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

$$= x + \left(\frac{1}{2}\right)\frac{x^3}{3} + \left(\frac{3}{8}\right)\frac{x^5}{5} + \left(\frac{1\cdot 3\cdot 5}{2\cdot 4\cdot 6}\right)\frac{x^7}{7}$$

Ex) $k=0$ $k=1$ $k=2$ $k=3$

input = 0.75

should approach 0.8481 convergence

$k=0$: 0.75
$k=1$: 0.82
$k=2$: 0.838
$k=3$: 0.849

For arc cos centered at 0:

$$\frac{\pi}{2} - \arcsin x = \text{value}$$

Ex) input = 0.4, should approach 1.1593

$k=0$: $\frac{\pi}{2} - 0.4 = 1.17$
$k=1$: 1.16
$k=2$: 1.1594

convergence

Taylor approximation **pseudocode** for arcSin and arcCos:

arcsin (double x):

epsilon = 10^{-10} ——— accuracy threshold

step_size = x ——— step size of a particular iteration
└ (once smaller than ϵ, we have precision)

value = x ——— running value of function
└ to be returned

0! = 1

for (int K = 1, step_size > epsilon, K += 1): ——— K = step counter

$$\text{step\_size} = \text{step\_size}\left(\frac{(2K-1)(2K-1)}{(2K)(2K+1)}x^2\right)$$

value = value + step_size

return value

arccos (double x):

epsilon = 10^{-10}

step_size = x

value = $\frac{\pi}{2} - x$ ——— arccos starting point differs from arcsin.

for (int K = 1, step_size > epsilon, K += 1):

$$\text{step\_size} = \text{step\_size}\left(\frac{(2K-1)(2K-1)}{(2K)(2K+1)}x^2\right)$$

value = value − step_size

return value

Note, values near one may require additional trig properties, to run code successfully, set for loop limit as reaching the desired epsilon value may not be possible for full range initially

---

* solving for step size in terms of previous step size - to avoid factorial

$$\sum_{K=0}^{\infty} \frac{(2K)!}{2^{2K}(K!)^2}\cdot\frac{x^{2k+1}}{2k+1}, \quad K(0) = x \qquad = x \cdot \left(\frac{(2k)(2k-1)}{4\cdot(k^2)}\right)\left(\frac{x^2}{\frac{2k+1}{2k-1}}\right) = \frac{(4K^2-2K)x^2}{4K^2\left(\frac{2k+1}{2k-1}\right)}$$

$k = 1$: $\frac{2!}{2^2(1!)^2}\cdot\frac{x^3}{3} = \frac{2x^3}{12} = \frac{x^3}{6}$

Simplifies to $\frac{(2K-1)(2K-1)}{(2K)(2K+1)}x^2$

$K = 2$: $\frac{4!}{2^4(2!)^2}\cdot\frac{x^5}{5} = \frac{24x^5}{320} = \frac{3x^5}{40}$

Credit for simplification:

$$4^K(K!)^2\cdot(2k+1) = \left[4^{(K-1)}\cdot(K-1)!^2\cdot(2k-1)\right]\cdot\left[4\cdot K^2\left(\frac{2k+1}{2k-1}\right)\right]$$

ArcTan using Taylor series:

— Note $\arctan(x) = \text{arccot}\left(\frac{1}{x}\right)$

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

$\text{ArcSin}(x)$ = approx ≈

$$x + \left(\frac{1}{2}\right)\frac{x^3}{3} + \left(\frac{3}{8}\right)\frac{x^5}{5} + \left(\frac{1\cdot3\cdot5}{2\cdot4\cdot6}\right)\frac{x^7}{7}$$

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

→ will need sqrt(x) fxn

Ex) input = 5.2, should approach 1.3808

$\frac{x}{\sqrt{x^2+1}} = 0.982$, $\arcsin(0.982) \to 1.3808$ — converges

Taylor approximation **pseudocode** for arcTan:

```
arctan(double x):
    plugin_value = (x / sqrt(x·x + 1))        ——— * Sqrt fxn needed
                                               ——— use tan to sin identity to solve
    value = arcsin(plugin_value)              ——— call arcsin of plugin_value
    return value
```

Function arguments for all math functions:

x = value for which we want to approximate (double)

Power fxn is a go:

$$e^{y \cdot \ln x} = x^y$$

No mod/truncate needed

try $x = 5$, $y = 3$

$$\underbrace{5^3 = 125 \quad e^{3\ln 5} = 125}_{\checkmark}$$