Derick Pan
dpan7

Asgn 2
A Small Numerical Library
DESIGN.PDF

Pre Lab Questions

1. How do I do Taylor expansion without using preexisting math libraries?
    a. When you see a sigma in math, think of a for loop. Use a for loop to imitate what taylor expansion is.
2. How does the code know when to stop doing taylor expansion?
    a. Stop doing taylor expansion when your answer's decimal points are smaller than epsilon.
3. Do I need to compute powers manually?
    a. Yes, You can not use the power function from math.h. Although, computing powers is simple; Use a for loop, starting from 1, and incrementing it by 1. Then multiply that to your base answer. Stop the loop when your incrementing reaches the power.
4. What's the formatting for the final printing
    a. The return function should do all the printing in the code. The arcsin, arccos, arctan, and log functions only return an array of it's own data.
5. Can I call pi from <math.h>?
    a. Yes, use M_PI from <math.h>
6. When, Where and Can I use anything given in <math.h>?
    a. When: Only in the print lines, and M_PI for the arcos() function.
    b. Where: Print lines, and arcos()
    c. Yes you can use <math.h>. BUT, YOU CAN NOT use it to help compute your functions.

The main function should be able to seperate the array of return values and print them with the respective X, Library, & Difference.

ex.

X	arcSin	Library	Difference
-1.0000	-1.57079633	-1.57079633	0.eeeeeeee
-0.9000	-1.11976951	-1.11976951	0.eeeee
-0.8000	-0.92729522	-0.92729522	0.eeeee

- Calculate values until the decimal points are smaller than epsilon.

- Σ = think of a for loop

- $0! = 1$

- CAN NOT use <math.h> in created functions
  L only use math.h in print statements

Derick Pan
dpan7

$\sin^{-1}$ arcSin() domain [-1, 1) step: 0.1 20 values to calculate

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

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

$\cos^{-1}$ arcCos() domain [-1, 1) step: 0.1 20 values to calculate

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

$\tan^{-1}$ arctan() domain [1, 10) step: 0.1 100 values to calculate

$$\arctan() = \arcsin\left(\frac{x}{\sqrt{x^2+1}}\right) = \arccos\left(\frac{1}{\sqrt{x^2+1}}\right), \ x > 0 \quad \text{or} \quad \sum_{k=0}^{\infty} \frac{2^{2k}(k!)^2}{(2k+1)!} \frac{x^{2k+1}}{(1+x^2)^{k+1}}$$

Log() domain [1, 10) step: 0.1

y is user input, ie: log(5), $y = 5$

$$x_{k+1} = x_k + \frac{y - e^{x_k}}{e^{x_k}}$$

x_k start with 1, end at

Arc Sin( )

~~There~~
1st draft of how the function could work/look

for( ) // increment x from -1 to 1, step 0.1

for( ) //Taylor Expansion   increment k++

$fk = 1$ $tk = 2 \cdot k$

$ftk = 1$

May be able to make another function

for( ) // calculate factorial for fk   increment i until $i == k$

$fk = fk \cdot i$

for( ) // calculate factorial for ftk   increment j until $j == tk$

$ftk = ftk \cdot j$

/* Now our Taylor expansion looks more like $\sum_{k=0}^{\infty} \frac{ftk}{2^{tk}(fk)^2} \cdot \frac{x^{tk+1}}{tk+1}$

double ans = 1

2x

Make this into a new function

~~for( )~~ //calculate powers */ ~~while( )~~ for( ) //start at exp, and -1 until 1

base = $\square$

ans = ans * base

do arithmetic then return it

The other functions aren't that different. They all still use Taylor Expansion. ~~Also all the~~ Using the two extra functions.

Extra Functions

data type ffactorial()

Parameters: base

declare ans=1 to return

for() /* increment i+until i==base */

ans = ans · i

return ans;

data type Power()

Parameters: base, ~~[illegible]~~ top

declare ans =1 to return

for() // exp -1 until==1

ans = ans * base

return ans;

Doing this stops me from declaring and writing so many for loops.

If I didn't write this, I would have to ~~[illegible]~~ declare and write the for loop, 6+times

exp() /* e^□ provided by class */

- Able to calculate exponents w/ base E

Creating the two functions above ~~[illegible]~~ creates LESS repetitive work.

Structure

main ( )

- able to call
one, or all arcSin, arcCos, arcTan, Log

- Print the array returned along with <math.h> values. Also print the x variable, and difference between our variables and <math.h>

arcSin
- compute taylor by math and calling
- return an array

arcCos
- call arcSin and receive an array
- Subtract $\frac{\pi}{2}$ − $\boxed{\text{each array item}}$
- return new array

arcTan
- Compute Taylor by math & calling
- return an array

Log
- compute Taylor by math & calling
- return an array

ff factorial ( )

power ( )

exp ( )

- In total, there should be 8 functions