Assignment 2 Design Doc

Tanya Gyanmote; cruzid:tgyanmot

10/6/22

Details:

We are writing and implementing sin, cos, sin−1 (arcsin), cos−1 (arccos), tan−1 (arctan), and log functions. The Taylor series will be used for sin, cos, and arcsin. arccos and arctan can be calculated from arcsin while implementing a square root function. As well as creating a main() program and acting as a test harness.

Implements:

1. A function that approximates sin(x)
2. A function that approximates cos(x)
3. A function that approximates arcsin(x)
4. A function that approximates arccos(x)
5. A function that approximates arctan(x)
6. A function that approximates log(x)

Pseudocode:

For Absolute Value

1. Create an if statement to see if the value(x) is less than 0, then if it is multiple it with -1, else just return the value(x) with no changes

For Square Root

1. Using the function given in piazza “square_root”

For sin(x)

making taylor series of sin(x) using x/2n * x/2n+1 * |an-1| (from pdf)

1. will have some placeholder variables (n,final_val,temp1)
    a. setting n as n from equation
    b. setting final_val to keep track of the final value
    c. setting temp1 to x as a placeholder value for the while loop
2. Creating a while loop setting it to True
    a. Creating a temp value called temp and adding temp1 to it, as well as adding the iterative equation of sin(x)
        i. old value times with x squared divided by 2 times n times 2 times n plus 1
    b. check if n is even or odd, accordingly make it neg. or pos. from the sequence
    c. incrementing n with 1, to get the next value
    d. creating another temp variable to check the absolute value of the value
    e. if temp from d is less than EPSILON then break the function
3. Return the final value

For cos(x)

1. return pi divided by 2 minus x in the my_sin function
   (shortcut for doing cos)
2. Proof for this equation:

Using the sine subtraction formula

sin(a-b) = sin(a)cos(b) - cos(a)sin(b)

a = (pi/2)

b = x

$\sin(\pi/2-x) = \sin(\pi/2)\cos(x)-\cos(\pi/2)\sin(x)$

$(1x\cos(x)) - (0\times\sin(x))$

= cos(x) - 0

= cos(x)

proved sin(pi/2-x) = cos(x)

For arcsin(x)

1. Creating a val variable for the final variable, and a next variable for the next number
2. Creating a while loop
    a. Using the given equation from piazza,
    b. make the final value (val) equal to the next variable minus
        i. my_sin of next minus x divided by my_cos of next
    c. break the loop if the added difference is less than EPSILON
    d. changing the next value to add, next equals to val
3. Once the loop breaks, return val

For arccos(x)

1. return pi divided by 2 minus x of my_sin function (from pdf)

For arctan(x)

1. Using (x)
2. Using square root fcn for x*x plus one
3. returning arcsin divided by step 1 over 2
4. (equation given in asgn2 pdf)

For log(x)

1. Using the equation from asgn2 pdf
2. Creating two variables one for the temp value called first, and one for the final where I keep adding it
3. Create a while loop that runs till true
    a. To get my final value I'm adding the previous value to x- Exp(previous value) divided by Exp(previous value)
    b. creating an if statement to make sure that the absolute value of final-first is not less than EPSILON, and if it is then break if not keep running
    c. changing mu previous value to the final value so first = final
4. return final value

Main file

1. creating bool opt for options a,b,c, etc;
    a. setting them to false
2. Creating opt and setting it to 0, and creating a different variable for the difference between my function and the library function
3. creating switch opt

4. for every case

    a. set boolean: true

    b. set pointer: corresponding mathlib.c function

    C. for case a, make all the options true, and for the rest of the cases make the corresponding option true

5. Create a default case that prints an error

6. Create if statements for every case possible, -a, -s, and print the table accordingly

    a. Having a for loop to print the tables, according to the different domains and the steps

Files

1. mathlib.h: Supplied file, with the function prototypes for the math functions

2. mathlib.c: This file contains my math function implementations, as prototyped in mathlib.h.

3. mathlib-test.c: This file will contain the main() program and acts as a test harness for my math library. 4. Makefile: This file will allow the grader to type make to compile your program.

5. README.md: This file will describe how to build and run my program and list the command-line options it accepts and what they do.

6. DESIGN.pdf: Describe the purpose of your program and communicate the overall design of the program with enough detail

7. WRITEUP.pdf: Discussion of the results of my tests.