DESIGN - Assignment 2

Brian Nguyen

October 7, 2021

1 Description

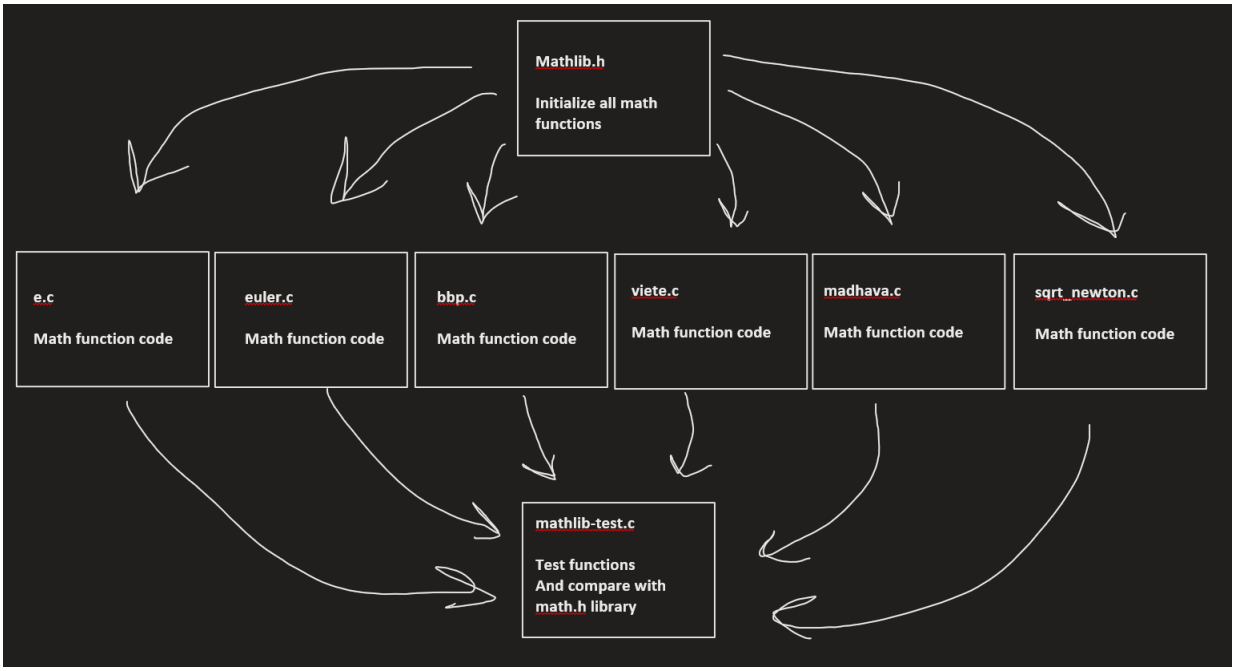
This program will contain a library of files with functions that all try to calculate the approximation of either π or e based on various traditional methods in an attempt to replicate <math.h>. It also contains a test harness that will be used to compare the accuracy between this math library and the built-in <math.h> library.

2 Files

1. bbp.c
    - This source file contains the code that calculates the approximation of π using Bailey-Borwein-Plouffe's formula and also returns the value of the number of terms computed.
2. e.c
    - This source file contains the code that calculates the approximation of e using Euler's formula and also returns the value of the number of terms computed.
3. euler.c
    - This source file contains the code that calculates the approximation of π using Euler's formula and also returns the value of the number of terms computed.
4. madhava.c
    - This source file contains the code that calculates the approximation of π using the Madhava series and also returns the value of the number of terms computed.
5. newton.c
    - This source file contains the code that calculates the approximation of $\sqrt{x}$ using Newton's method and also returns the value of the number of terms computed.

6. viete.c
    - This source file contains the code that calculates the approximation of π using Viete's formula and also returns the value of the number of terms computed.
7. mathlib-test.c
    - This source file contains the main() function that runs and tests the functions of the created math library.
8. mathlib.h
    - This header file contains the interface of the created math library.
9. Makefile
    - This make file contains the code that builds and compiles the math library program to be run. It also cleans all compiler generated files and formats the code to be submitted.
10. README.md
    - This markdown file describes the program, how to build it, how to run it, and also lists and explains all the command-line options that the math program accepts. It also documents any false positives given by scan-build.
11. DESIGN.pdf
    - This pdf is the manual that explains the program, files included, layout or structure, and pseudo-code of the math library.
12. WRITEUP.pdf
    - This pdf is a scientific writeup made up of graphs and explanations by the gnuplot tool which shows the differences in the values between the mathlib and <math.h>.

3 Structure



4 Pseudo-code

4.1 bbp.c

set static counter variable
define pi bbp
set top variable
set total variable
for loop until reaching epsilon using k
    do 16^{-k} using for loop
    add $\frac{1}{top} * \frac{(k(120k+151)+47)}{k(k(k(512k+1024)+712)+194)+15}$ to total
    increment counter
return total

define pi bbp terms
    return counter

4.2 e.c

set static counter variable to 1
define e

make total variable
set term variable
for loop until reaching epsilon using k
    do increment term by multiply with 1/k
    add term to total
    increment counter variable
return total

define e terms
    return counter

4.3 euler.c

set static counter variable
define euler
set total variable
    for loop until reaching epsilon using k
    do $\frac{1}{k^2}$
    add to total var
    increment counter variable
return sqrt of 6 * total

define euler terms
    return counter

4.4 madhava.c

set static counter variable
define madhava
set total variable
set power variable
for loop until reaching epsilon using k
    set power variable to 1
    for loop power operation
        set power to (-3)$^{-k}$
    do $\frac{1/power}{2k+1}$
    add to total var
    increment counter variable
return $\sqrt{12}$ * total

define madhava terms
    return counter

4.5 newton.c

Use assignment pdf provided pseudocode
    add a counter variable within while function

def newton factors
    return counter

4.6 viete.c

set static counter variable
define viete
set total variable to $\sqrt{2}/2$
set sqrt variable to $\sqrt{2}$
for loop until reaching epsilon using k
    do sqrt = sqrt(2 + sqrt)
    multiply sqrt(2)/2 with total
    increment counter variable
    return 2/total

define viete factors terms
    return counter

4.7 mathlib-test.c

use psuedocode from asgn2.pdf
provide own arguments instead of pi
use booleans to determine which argument is being called on
use if statements to print out values, etc. based on true booleans
difference for most = absolute(my program - constant)
difference for sqrt_newton = for loop from [0,10) comparing with math.h sqrt

5 Credits

1. I used the asgn2.pdf from Professor Long for formulas and explanations.
2. I watched the Lab Section recording from Eugene held on 10/05.

3. I watched the Sloan's Section recording held on 10/06.

4. I watched the Christian's Section recording held on 10/08.