asgn 2: DESIGN - A Little Slice of **π**

Eric Wang

May 3, 2023

Purpose:

In this series of programs, we are to write our own mathlib with different formulas to estimate the value of pi, square root of a number, and the value of e in various ways. In the test function, we are to write test using getopt to simulate different user inputs such as -a -e -r -b -m -v -w -n -s -h. When the users inputs one, or multiple of the said inputs, it will calculate the values of the input prompted by the user accordingly. -e stands for values of e, -r stands for euler's formula, -b stands for Bailey-Bornwin-Plouff formula, etc.

How to use the program:
Run the makefile, when you get the binary file, use ./mathlib-test with the command lines below.
Notice: if you don't input anything in the command line, it will output the help message instead.
Command line options:
-a : Runs all tests.
-e : Runs e approximation test.
-b : Runs Bailey-Borwein-Plouffe π approximation test.
-m : Runs Madhava π approximation test.
-r : Runs Euler sequence π approximation test.
-v : Runs Viete π approximation test.
-w : Runs Wallis π approximation test.
-n : Runs Newton-Raphson square root approximation tests, calling sqrt_newton() with various inputs for testing. This option does not require any parameters, and will only test within the range of [0, 10) (it will not test the value 10).
-s : Enable printing of statistics to see computed terms and factors for all tested functions.
-h : Display a help message detailing program usage. This can be anything you want, but must be specific and readable.

Pseudocode:

mathlib-test.c:

Take the number of arguments in a command line and the respective characters in the command line
Define a integer opt
While the opt is not equal to -1
        Switch cases for opt
                Case a
                        Execute all the commands

Case e
Get value from e
Case r
Get value from euler formula
Case b
Get value from bbp formula
Case m
Get value from madhava formula
Case v
Get value from viete formula
Case w
Get value from wallis formula
Case n
Get value from newton square root formula

There will be a count variable on top of every function. Depends on the function, the count variable will either be 1 or 0.

bbp.c
Declare 3 variables
While the terms in the bbp value is greater than epsilon
Adds the bbp term
Increment count
Increment k
16 to the power

e.c
Declare 3 variables, i, sum, and factorial
While 1 over factorial is less than epsilon
i to the power
Increment i
add 1 over factorial to sum

euler.c
Declare 2 variables, k and sum
While the euler term is less than epsilon
Add the euler term to sum
Increment k

madhava.c
declare 4 variables, k, i, store in old and new values
In the do while loop
Equate the old and new values
And add 2 to the k value

i to the power
replace the new value with the madhava term
while the absolute value of new value subtracted from the old value is less than epsilon
And then return the final value using the square root newton function.
newton.c
(this function was given us in the lab doc, I will write it again)

```
def sqrt_newton(x):
    next_y = 1.0
    y = 0.0
    while abs(next_y - y) > epsilon;
        y = next_y
        Next_y = 0.5 * (y + x / y)
    return next_y
```

viete.c
declare two variables, numerator and product
Set the numerator to the square root of 2
while the numerator is less than two minus epsilon
    Multiply the product to numerator over 2
    Set the numerator to the square root of 2 plus the old numerator
Return 2 over the product

wallis.c
Declare two variables, product and k
While the absolute value of 1 minus the wallis factor is greater than epsilon
    Multiply the product with the wallis factor
    Increment k

Return 2 multiplied by the product

Results:

Here are the results for my own binary that was output into a txt file:

e() = 2.718281828459046, M_E = 2.718281828459045, diff = -0.000000000000000
e() terms = 18
pi_bbp() = 3.141592653589791, M_PI = 3.141592653589793, diff = 0.000000000000002
pi_bbp() terms = 10
pi_madhava() = 3.141592653589800, M_PI = 3.141592653589793, diff = -0.000000000000007
pi_madhava() terms = 27
pi_euler() = 3.141592558095893, M_PI = 3.141592653589793, diff = 0.000000095493900
pi_euler() terms = 9999999
pi_viete() = 3.141592653589775, M_PI = 3.141592653589793, diff = 0.000000000000018
pi_viete() terms = 23

```
pi_wallis() = 3.141592495717032, M_PI = 3.141592653589793, diff = 0.000000157872762
pi_wallis() terms = 4974439
sqrt_newton(0.00) = 0.000000000000007, sqrt(0.00) = 0.000000000000000, diff = -0.000000000000007
sqrt_newton() terms = 47
sqrt_newton(0.10) = 0.316227766016838, sqrt(0.10) = 0.316227766016838, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(0.20) = 0.447213595499958, sqrt(0.20) = 0.447213595499958, diff = -0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(0.30) = 0.547722557505166, sqrt(0.30) = 0.547722557505166, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(0.40) = 0.632455532033676, sqrt(0.40) = 0.632455532033676, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(0.50) = 0.707106781186547, sqrt(0.50) = 0.707106781186548, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(0.60) = 0.774596669241483, sqrt(0.60) = 0.774596669241483, diff = 0.000000000000000
sqrt_newton() terms = 5
sqrt_newton(0.70) = 0.836660026534076, sqrt(0.70) = 0.836660026534076, diff = 0.000000000000000
sqrt_newton() terms = 5
sqrt_newton(0.80) = 0.894427190999916, sqrt(0.80) = 0.894427190999916, diff = 0.000000000000000
sqrt_newton() terms = 5
sqrt_newton(0.90) = 0.948683298050514, sqrt(0.90) = 0.948683298050514, diff = 0.000000000000000
sqrt_newton() terms = 5
sqrt_newton(1.00) = 1.000000000000000, sqrt(1.00) = 1.000000000000000, diff = -0.000000000000000
sqrt_newton() terms = 1
sqrt_newton(1.10) = 1.048808848170152, sqrt(1.10) = 1.048808848170151, diff = -0.000000000000000
sqrt_newton() terms = 5
sqrt_newton(1.20) = 1.095445115010332, sqrt(1.20) = 1.095445115010332, diff = 0.000000000000000
sqrt_newton() terms = 5
sqrt_newton(1.30) = 1.140175425099138, sqrt(1.30) = 1.140175425099138, diff = 0.000000000000000
sqrt_newton() terms = 5
sqrt_newton(1.40) = 1.183215956619923, sqrt(1.40) = 1.183215956619923, diff = -0.000000000000000
sqrt_newton() terms = 5
sqrt_newton(1.50) = 1.224744871391589, sqrt(1.50) = 1.224744871391589, diff = 0.000000000000000
sqrt_newton() terms = 5
sqrt_newton(1.60) = 1.264911064067352, sqrt(1.60) = 1.264911064067352, diff = -0.000000000000000
sqrt_newton() terms = 5
sqrt_newton(1.70) = 1.303840481040530, sqrt(1.70) = 1.303840481040530, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(1.80) = 1.341640786499874, sqrt(1.80) = 1.341640786499874, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(1.90) = 1.378404875209022, sqrt(1.90) = 1.378404875209022, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(2.00) = 1.414213562373095, sqrt(2.00) = 1.414213562373095, diff = -0.000000000000000
sqrt_newton() terms = 6
```

```
sqrt_newton(2.10) = 1.449137674618944, sqrt(2.10) = 1.449137674618944, diff = -0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(2.20) = 1.483239697419133, sqrt(2.20) = 1.483239697419133, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(2.30) = 1.516575088810310, sqrt(2.30) = 1.516575088810310, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(2.40) = 1.549193338482967, sqrt(2.40) = 1.549193338482967, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(2.50) = 1.581138830084190, sqrt(2.50) = 1.581138830084190, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(2.60) = 1.612451549659710, sqrt(2.60) = 1.612451549659710, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(2.70) = 1.643167672515499, sqrt(2.70) = 1.643167672515499, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(2.80) = 1.673320053068152, sqrt(2.80) = 1.673320053068151, diff = -0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(2.90) = 1.702938636592640, sqrt(2.90) = 1.702938636592640, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(3.00) = 1.732050807568878, sqrt(3.00) = 1.732050807568878, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(3.10) = 1.760681686165901, sqrt(3.10) = 1.760681686165901, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(3.20) = 1.788854381999832, sqrt(3.20) = 1.788854381999832, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(3.30) = 1.816590212458495, sqrt(3.30) = 1.816590212458495, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(3.40) = 1.843908891458578, sqrt(3.40) = 1.843908891458578, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(3.50) = 1.870828693386971, sqrt(3.50) = 1.870828693386971, diff = -0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(3.60) = 1.897366596101028, sqrt(3.60) = 1.897366596101028, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(3.70) = 1.923538406167135, sqrt(3.70) = 1.923538406167135, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(3.80) = 1.949358868961793, sqrt(3.80) = 1.949358868961793, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(3.90) = 1.974841765813150, sqrt(3.90) = 1.974841765813151, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(4.00) = 2.000000000000000, sqrt(4.00) = 2.000000000000000, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(4.10) = 2.024845673131659, sqrt(4.10) = 2.024845673131659, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(4.20) = 2.049390153191920, sqrt(4.20) = 2.049390153191920, diff = 0.000000000000000
sqrt_newton() terms = 6
```

sqrt_newton(4.30) = 2.073644135332772, sqrt(4.30) = 2.073644135332772, diff = 0.000000000000000
sqrt_newton() terms = 6
sqrt_newton(4.40) = 2.097617696340303, sqrt(4.40) = 2.097617696340303, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(4.50) = 2.121320343559643, sqrt(4.50) = 2.121320343559642, diff = -0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(4.60) = 2.144761058952722, sqrt(4.60) = 2.144761058952722, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(4.70) = 2.167948338867880, sqrt(4.70) = 2.167948338867880, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(4.80) = 2.190890230020664, sqrt(4.80) = 2.190890230020664, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(4.90) = 2.213594362117865, sqrt(4.90) = 2.213594362117865, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(5.00) = 2.236067977499789, sqrt(5.00) = 2.236067977499789, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(5.10) = 2.258317958127242, sqrt(5.10) = 2.258317958127242, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(5.20) = 2.280350850198276, sqrt(5.20) = 2.280350850198275, diff = -0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(5.30) = 2.302172886644267, sqrt(5.30) = 2.302172886644267, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(5.40) = 2.323790007724449, sqrt(5.40) = 2.323790007724449, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(5.50) = 2.345207879911714, sqrt(5.50) = 2.345207879911714, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(5.60) = 2.366431913239846, sqrt(5.60) = 2.366431913239845, diff = -0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(5.70) = 2.387467277262664, sqrt(5.70) = 2.387467277262664, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(5.80) = 2.408318915758458, sqrt(5.80) = 2.408318915758458, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(5.90) = 2.428991560298223, sqrt(5.90) = 2.428991560298223, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(6.00) = 2.449489742783177, sqrt(6.00) = 2.449489742783177, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(6.10) = 2.469817807045692, sqrt(6.10) = 2.469817807045693, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(6.20) = 2.489979919597745, sqrt(6.20) = 2.489979919597745, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(6.30) = 2.509980079602225, sqrt(6.30) = 2.509980079602225, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(6.40) = 2.529822128134702, sqrt(6.40) = 2.529822128134702, diff = 0.000000000000000
sqrt_newton() terms = 7

sqrt_newton(6.50) = 2.549509756796391, sqrt(6.50) = 2.549509756796391, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(6.60) = 2.569046515733024, sqrt(6.60) = 2.569046515733024, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(6.70) = 2.588435821108956, sqrt(6.70) = 2.588435821108956, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(6.80) = 2.607680962081058, sqrt(6.80) = 2.607680962081058, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(6.90) = 2.626785107312738, sqrt(6.90) = 2.626785107312738, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(7.00) = 2.645751311064589, sqrt(7.00) = 2.645751311064589, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(7.10) = 2.664582518894844, sqrt(7.10) = 2.664582518894844, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(7.20) = 2.683281572999746, sqrt(7.20) = 2.683281572999746, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(7.30) = 2.701851217221257, sqrt(7.30) = 2.701851217221257, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(7.40) = 2.720294101747087, sqrt(7.40) = 2.720294101747087, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(7.50) = 2.738612787525828, sqrt(7.50) = 2.738612787525829, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(7.60) = 2.756809750418042, sqrt(7.60) = 2.756809750418042, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(7.70) = 2.774887385102319, sqrt(7.70) = 2.774887385102319, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(7.80) = 2.792848008753786, sqrt(7.80) = 2.792848008753786, diff = -0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(7.90) = 2.810693864511037, sqrt(7.90) = 2.810693864511037, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(8.00) = 2.828427124746188, sqrt(8.00) = 2.828427124746188, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(8.10) = 2.846049894151539, sqrt(8.10) = 2.846049894151539, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(8.20) = 2.863564212655268, sqrt(8.20) = 2.863564212655268, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(8.30) = 2.880972058177584, sqrt(8.30) = 2.880972058177584, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(8.40) = 2.898275349237886, sqrt(8.40) = 2.898275349237885, diff = -0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(8.50) = 2.915475947422648, sqrt(8.50) = 2.915475947422648, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(8.60) = 2.932575659723033, sqrt(8.60) = 2.932575659723033, diff = 0.000000000000000
sqrt_newton() terms = 7

sqrt_newton(8.70) = 2.949576240750523, sqrt(8.70) = 2.949576240750523, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(8.80) = 2.966479394838263, sqrt(8.80) = 2.966479394838263, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(8.90) = 2.983286778035257, sqrt(8.90) = 2.983286778035257, diff = -0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(9.00) = 2.999999999999997, sqrt(9.00) = 2.999999999999997, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(9.10) = 3.016620625799669, sqrt(9.10) = 3.016620625799669, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(9.20) = 3.033150177620618, sqrt(9.20) = 3.033150177620617, diff = -0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(9.30) = 3.049590136395378, sqrt(9.30) = 3.049590136395378, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(9.40) = 3.065941943351175, sqrt(9.40) = 3.065941943351175, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(9.50) = 3.082207001484485, sqrt(9.50) = 3.082207001484485, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(9.60) = 3.098386676965931, sqrt(9.60) = 3.098386676965931, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(9.70) = 3.114482300479485, sqrt(9.70) = 3.114482300479485, diff = 0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(9.80) = 3.130495168499703, sqrt(9.80) = 3.130495168499702, diff = -0.000000000000000
sqrt_newton() terms = 7
sqrt_newton(9.90) = 3.146426544510452, sqrt(9.90) = 3.146426544510452, diff = 0.000000000000000
sqrt_newton() terms = 7

By comparing to the given binary, the output for wallis, viete and euler are 1 digit off. However, error margin within one digit is allowed from the lab. Otherwise the output is the exact same as the given binary.

Error Handling:
When thinking of the errors, we have to consider that each test is run exactly once, even with duplicate options. I handled it with creating boolean values for every single tests. Within the switch case, I will turn the boolean value on and off depends on which test is triggered.

The second error we have to take into consideration is that the help message is printed both when “-h” is triggered and when invalid/no options are given. I did it by creating a notest boolean variable that checks if every other test variable is false. And then triggers the help message if either test h or the notest is true.

References:

1. Dr. Kerry Veenstra, Assignment 2 Document, 2023