Name: Yashika Patil Roll No: 02115374

Solutions to Homework 2

Problem 1

The files for this problem are saved as Problem1_PartA, Problem1_PartB, Problem1_PartC, for part A, B, and C respectively.

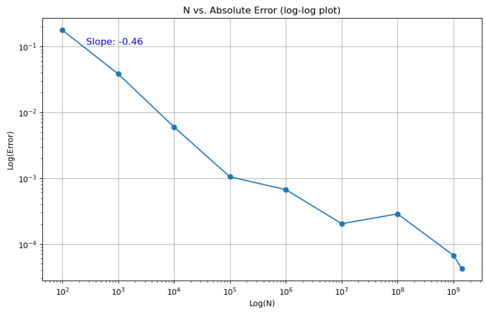
Part (a)

Use ./Problem1_PartA >> Problem1_PartA.dat to save the results from the following c program, i.e. Problem1_PartA.c, to the dat file.

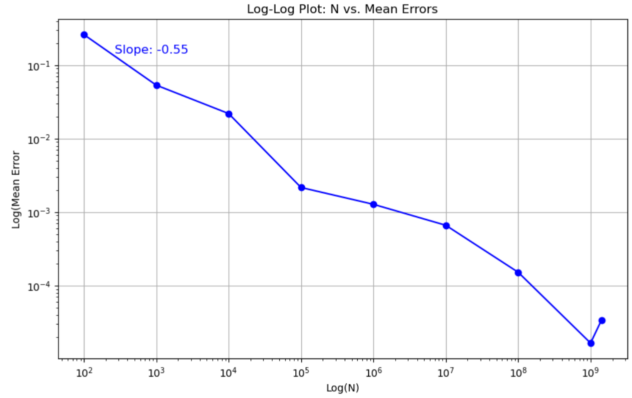
```c
// C program for computing pi using monte carlo integration
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

int main() {
    for (int k = 2; k <= 10; k++) {
        int N = 1;
        const float pi = 3.14159265358979323846;

        for (int i = 0; i < k; i++) {
            N *= 10;
        }
        srand((unsigned int)time(NULL));
        double count = 0;
        for (int i = 0; i < N; i++) {
            double x1 = ((double)rand() / RAND_MAX) * 2.0 - 1.0;
            double x2 = ((double)rand() / RAND_MAX) * 2.0 - 1.0;
            if (x1 * x1 + x2 * x2 <= 1.0) {
                count++;
            }
        }
        double result = 4.0 * (double)count / N;
        double abs_error = fabs(pi - result);
        printf("%d %i %f %f\n", k, N, result, abs_error);
    }
    return 0;
}
```

Part (b)

Using python code (Problem1_PartB.ipynb), plotted the following two graphs.



The above plot is created using Problem1_PartA.dat file, generated in Part A.



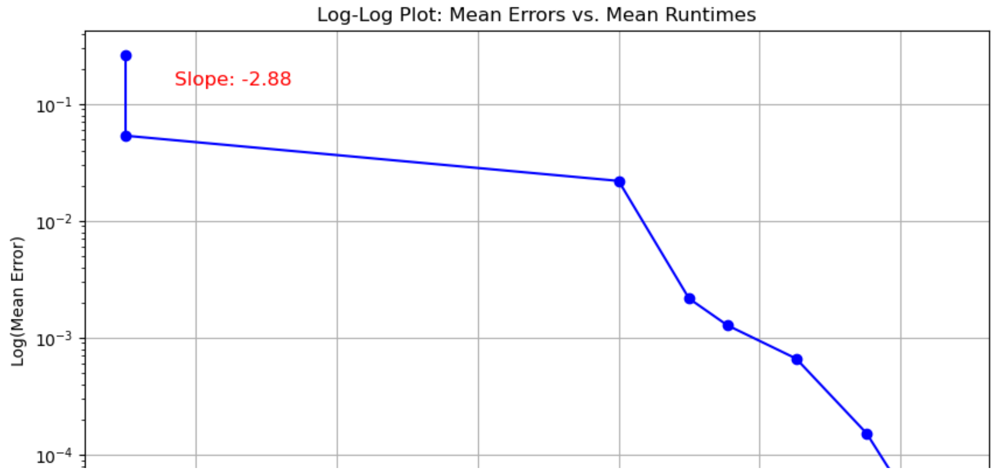
The above plot is created using 10 data files generated in Part C using .sh file. These files are saved in Problem1_data_files folder. The mean absolute error and mean runtime values are computed in ipynb file.

Part (c)

I modified the c program from part A and created an .sh file which passes 10^i, where i = 2...10, to Problem1_PartC.c The .sh file generates 10 data files (timing$i) which are saved in Problem1_data_files folder. The .sh file is as follows:

```
#!/bin/bash
gcc -std=c99 -o Problem1_PartC Problem1_PartC.c
j=1
while [[ j -le 10 ]]
do
    i=2
    while [[ i -le 10 ]]
    do
        /usr/bin/time -f " %e" ./Problem1_PartC $((10**i))\n &>> Problem1_data_files/timing$j.dat
        ((i = i + 1))
    done
    ((j = j + 1))
done
```

The following graph is plotted in Problem1_PartC.ipynb.



For estimating runtime for 10^-16 and 10^-70030 accuracy, here's the python code:

```
y = [-16, -70030]

for i in y:
    time = abs((i-intercept)/slope)
    print(f"Runtime for 10^{i} is 10^{time}")
```

Runtime for 10^-16 is 10^2.9857140594861247
Runtime for 10^-70030 is 10^24294.943337990815

Problem 2

The files are saved as Problem2.sh, Problem2_anvil.dat, Problem2_laptop.dat

I used unix script for N=10^7, 10^8, 10^9. The screenshot represents the mean runtime for 10 data files ran separately on the laptop and anvil.

```
#!/bin/bash
gcc -std=c99 -o Problem1_PartC Problem1_PartC.c
j=1
while [[ j -le 10 ]]
do
    for N in 10000000 100000000 1000000000
    do
        /usr/bin/time -f " %e" ./Problem1_PartC $N &>> Problem2_data_f
iles/timing_laptop$j.dat
        # To run on anvil, use the following command
        # /usr/bin/time -f " %e" ./Problem1_PartC $N &>> Problem2_data
_files/timing_anvil$j.dat
    done
    ((j = j + 1))
done
```

	N	Mean Error	Mean Runtime
0	1.000000e+07	0.000663	0.743
1	1.000000e+08	0.000152	6.151
2	1.000000e+09	0.000017	53.066

Mean Runtime on Laptop

	N	Mean Error	Mean Runtime
0	1.000000e+07	0.000457	0.21
1	1.000000e+08	0.000104	2.14
2	1.000000e+09	0.000045	21.17

Mean Runtime on Anvil

My laptop takes 59.96 seconds whereas Anvil takes 23.52 seconds on an average, which is about 2.55 times faster.