High Performance Scientific Computing: HW 1

Krut Patel

Problem 1

Answer a: Mathematical formula

$$I_1 \approx \hat{I}_1(N) = \frac{V}{N} \sum_{i=1}^{N} \frac{1}{1 + x_i^2} \tag{1}$$

Answer b: Pseudo Code

```
// Include necessary libraries which will be relevant for using standard input/output functions,
// random number generation, and math operations

// Define a function to calculate f(x)
double function(double x) {
    return 1.0 / (1.0 + x * x);
}

// Interpret the value of N from the command line argument
// Seed the random number generator with the current time
// Initialize a variable to store the sum of terms

// Loop for Monte Carlo simulation
for (int i = 1; i <= N; i++) {
    // Generate a random value xi within the range [-1, 1]
    // Calculate f(xi) using the defined function
    double f_xi = function(xi);

    // Increment the sum by f(xi)
    sum_terms += f_xi;
}

// Define the value of V by integrating dx for the range set above as your limit

double I1_approx = (V / N) * sum_terms;   // Calculate the approximate I1
printf("%lf\n", I1_approx);   // Print out the result

return 0;
```

Answer c: The final code

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
```

```
// Define the value of pi
#define PI 3.14159265358979323846

// Define the function f(x)
double function(double x) {
    return 1.0 / (1.0 + x * x);
}

int main(int argc, char *argv[]) {   //to confirm that the correct value of N is
                                     //provided if not, print how to use the code
    if (argc != 2) {
        printf("Usage: %s N\n", argv[0]);
        return 1;
    }

    int N = atoi(argv[1]);  //interpret the value of N from the command line

    srand(time(NULL));  //random number generator with current time

    double sum_terms = 0.0;

    // Monte Carlo simulation
    for (int i = 1; i <= N; i++) {
        double xi = 2.0 * ((double)rand() / (double)RAND_MAX) - 1.0;    //generates
                //random values of xi from 0 to 1 and multiplies it by 2, then
                //substracting 1 so that we get values fromm -1 to 1

        double f_xi = function(xi);
        sum_terms += f_xi;
    }

    // Define the value of V
    double V = 2.0;

    //calculating the approximate I1 using Monte Carlo
    double I1_approx = (V / N) * sum_terms;

    //defining the true valuee of I_1 from the integral which is pi/2
    double I1_true = PI / 2.0;

    //calculates the error E(N)
    double error = fabs(I1_true - I1_approx);

    // Print N and E(N) as an array
    printf("%d %lf\n", N, error);
    return 0;
}
```

Answers d, e and f: Analysis

The graph of N vs $\langle E(N) \rangle$ approximately follows a line equation $\log\langle E(N) \rangle = 0.0138 - 0.0043N$. So a reasonable estimate for A and B can be -0.0043 and 0.0138 respectively.

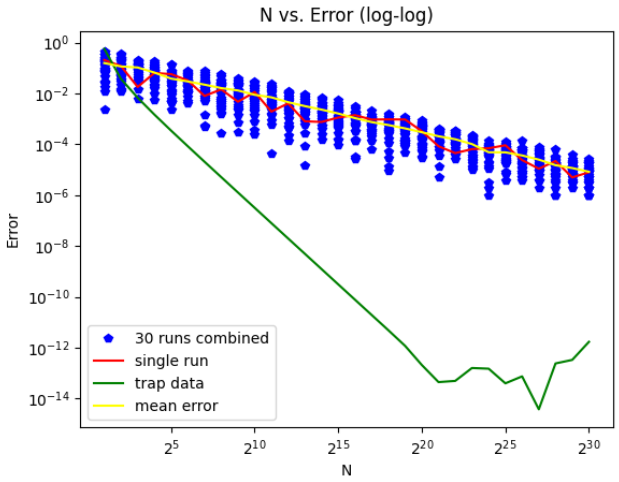

Figure 1: This is a combined plot for N vs $E(N)$ from the data generated from a single run and 30 reruns of a Monte Carlo simulation. It also includes $\langle E(N) \rangle$ and the trapezoidal simulation for the same equation for comparison.

Answers g: Run-time analysis

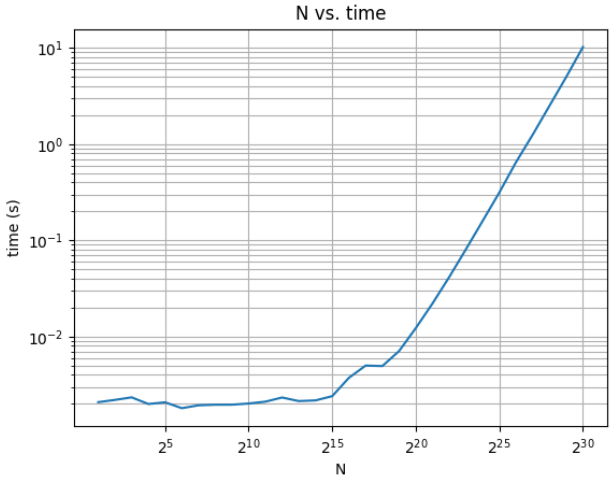

Figure 2: This plot shows how is time increases as we take higher and higher values of N on a log-log scale

As the Unix program 'time' was giving me just the total run-time of the whole program, I've written a bash script which calculates the start time and end time at the beginning and ending of the Monte Carlo program execution. At the end of the loop, I've calculated 'end time - start time' to find the elapsed time for each iteration.

After the value of $N = 2^{18}$, the graph of $Nvstime$ follows a straight line in log-log scale. From that we can derive a line equation to find out the time to be around 26 seconds for $N = 2^{32}$.

Problem 2

The value I found was 2.055 meters. I have ran the code for 30 different values of S from 2^1 to 2^{30} just as we did in the previous Monte Carlo integral. As the value of S increases, the integral converges to 2.055 meters.

Some parts of the bash scripts have been written with the help of ChatGPT.