PHY407 (Lab 10: Random numbers, Monte Carlo integration)

Utkarsh Mali[1] and Aslesha Pokhrel[1,2]

[1]Department of Physics, University of Toronto
[2]Department of Computer Science, University of Toronto

November 26, 2020

Contents

Work Allocation:
Utkarsh: Question 1
Aslesha: Question 2, Question 3

Question 1: Brownian motion, Diffusion Limited Aggregation

(a) **Here we are asked to formulate a Monte-Carlo Simulation of a random walk of a particle, commonly known as Brownian motion.** We used some of the functions defined in the given handout, set the number of steps $N = 5000$ to reduce run time. We choose not to animate this in order to save run time complexity.

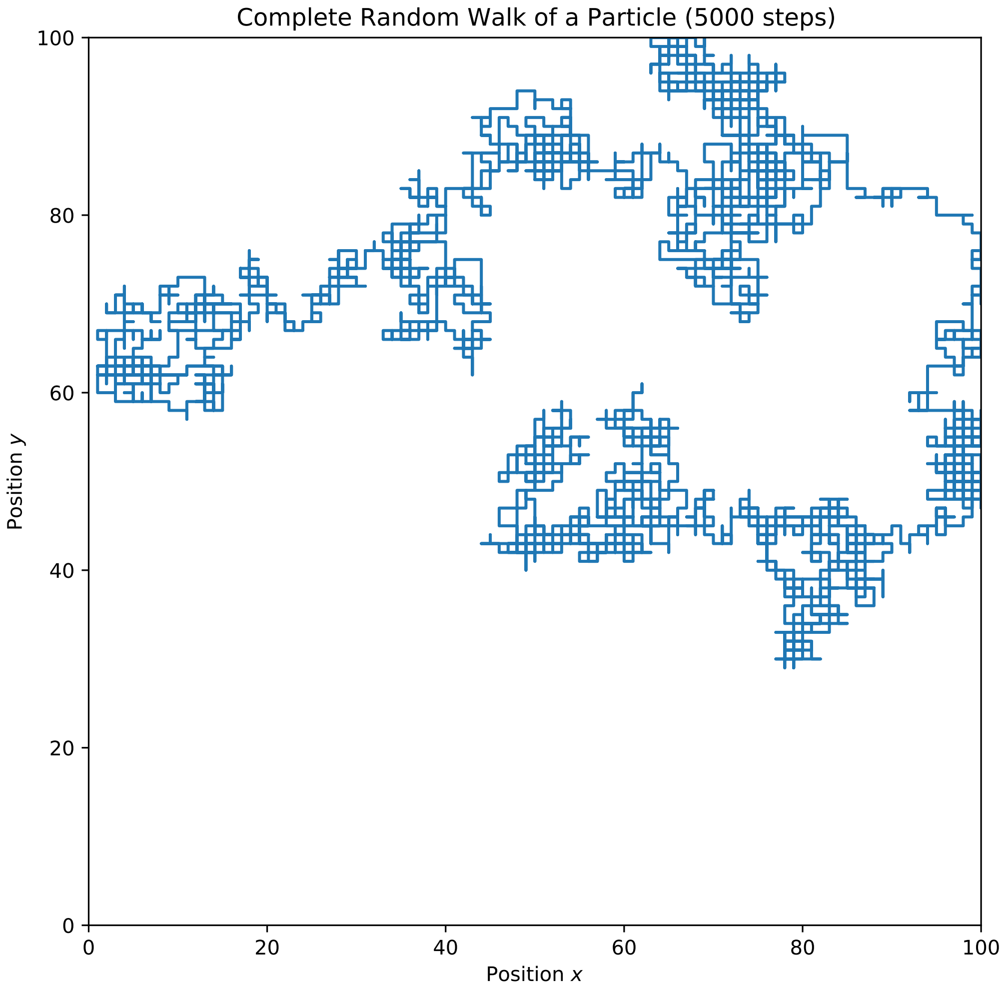

Figure 1: We noticed by observation that the particle follows a random-walk like Brownian motion with the boundary being set as the limitation, if the particle hits the boundary then it will be sent back into the allowed space inside the box.

(b) **This question asks us to make a copy of the Brownian Motion from Ex3 and repeatedly introduce new particles and let them hit the edge of the walls.** We ran the code as required. We have set up the anchored system in the for loop to first check whether the new position of the particle is at an anchored point, if it is then set the previous position of that particle to the list of all anchored points. Otherwise keep looping until that condition is satisfied.

```
PSEUDOCODE:
# Ex 10.13: Diffusion-limited aggregation
# Define Constants
# Set i, j
# Define starting position
# Use nextmove defenition from Prof Nicolas
# Define boundary checker
# Define anchor checker
# Run simulation through for loop
# Set number of particles
# Set list for anchored points

# For loop over all the particles
   # Print status update every few loops.
   # Empty xdata and ydata for each particle
   # Set each particle to the center at the start
   # Run Monte-Carlo Simulation for single particle
   # Loop over the number of steps we want each particle to take
      # Find new values of x and y.
      # Append to xdata and ydata
      # If item is at anchor then we can stop looping
          # Add location of particce to all anchors
      # Update current x and y and continue simualtion
      # If item is at boundary, we can stop looping
         # Add location of particle to all anchors
```

(c) **This part of the question asked us to keep running the code until we found that we had an anchored point at the origin.** We set the size to be larger and added the notion of an anchored system checker into our code. We kept running this code until there was an anchored point at the origin.

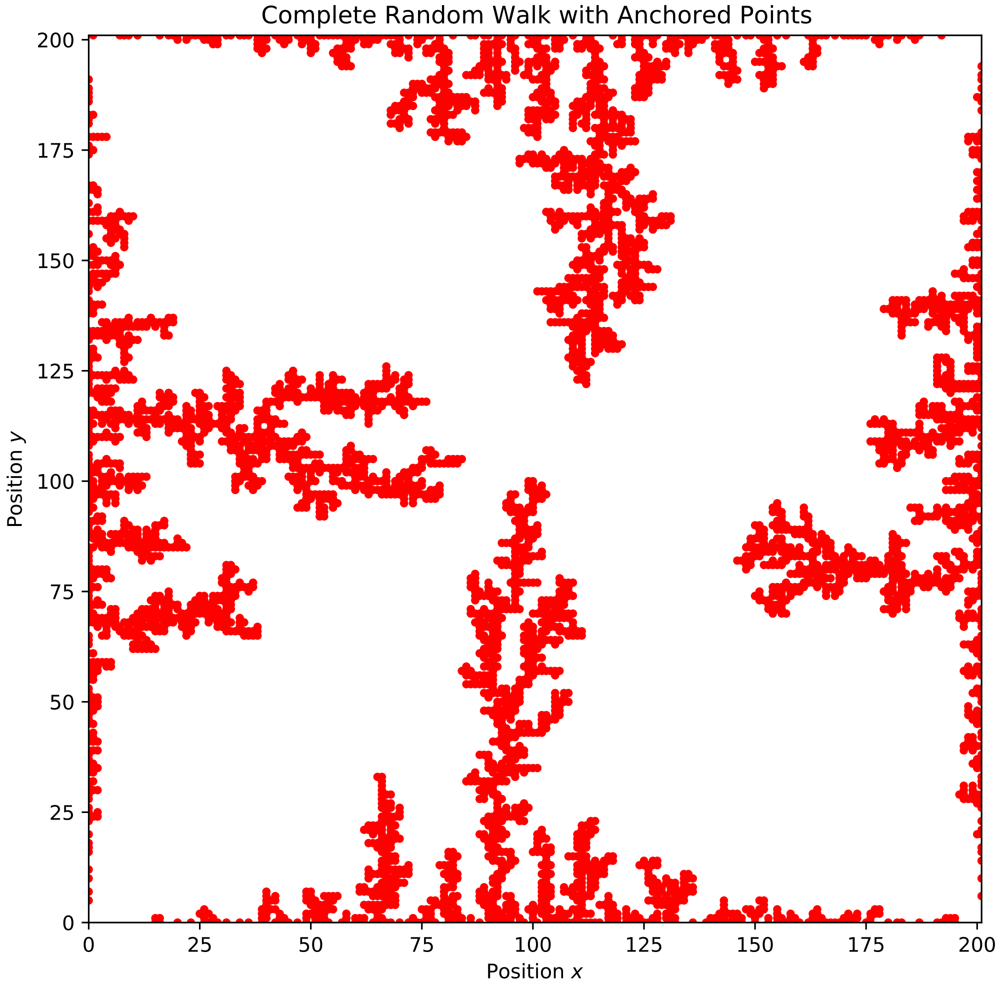

Figure 2: We noticed by observation that the particle follows a random-walk like Brownian motion with the boundary being set as the limitation, if the particle hits the boundary then it will be sent back into the allowed space inside the box. The same analogy can be applied to anchored points. Due to the number of particles and size of the frame being used we find that the image might be a little fuzzy.

Question 2: Volume of a 10-dimensional Hypersphere

In this section we estimate the volume of a 10-dimensional hypersphere using the mean value Monte Carlo method. Here, we used eq. 10.33 to compute the volume and the n-dimensional generalization of eq. 10.32 to compute the error as given in the textbook. The following equation was used to compute the volume of the hypersphere:

$$I \simeq \frac{V}{N} \sum_{i=1}^{N} f(\mathbf{r}_i)$$

where, V is the volume of the 10-dimensional hypercube given by L^{10} and $L = 2$ is the length of the hypercube. Similarly, N is the number of points sampled, we used a million points. Next, points $\mathbf{r}_i$ are picked uniformly at random from the volume V and f is a piecewise function which equals 1 if $\mathbf{r}_i \cdot \mathbf{r}_i \leq 1$ and 0 otherwise. Similarly, the error is given by:

$$\sigma = V \frac{\sqrt{\mathrm{var} f}}{\sqrt{N}}$$

where $\mathrm{var} f$ is the variance of f given by, $\mathrm{var} f = \langle f^2 \rangle - \langle f \rangle^2$.

The computed integral and the value of the error is given by the code to 2 decimal places as follows:

```
OUTPUT :

The estimated volume of a 10-dimensional hypersphere is 2.55 with the error of 5.10e-02.
```

Question 3: Importance sampling

(a) **Here, we estimate and compare the integral eq.4 in the handout whose integrand has a divergence using both the regular mean value method and the importance sampling methods.** The points are drawn from the probability distribution, $p(x) = \frac{1}{2\sqrt{x}}$. The integral is computed by using the setup given in the handout and the histogram of the resulting integral values using 10 bins from 0.80 to 0.88 is plotted to visualize the results.

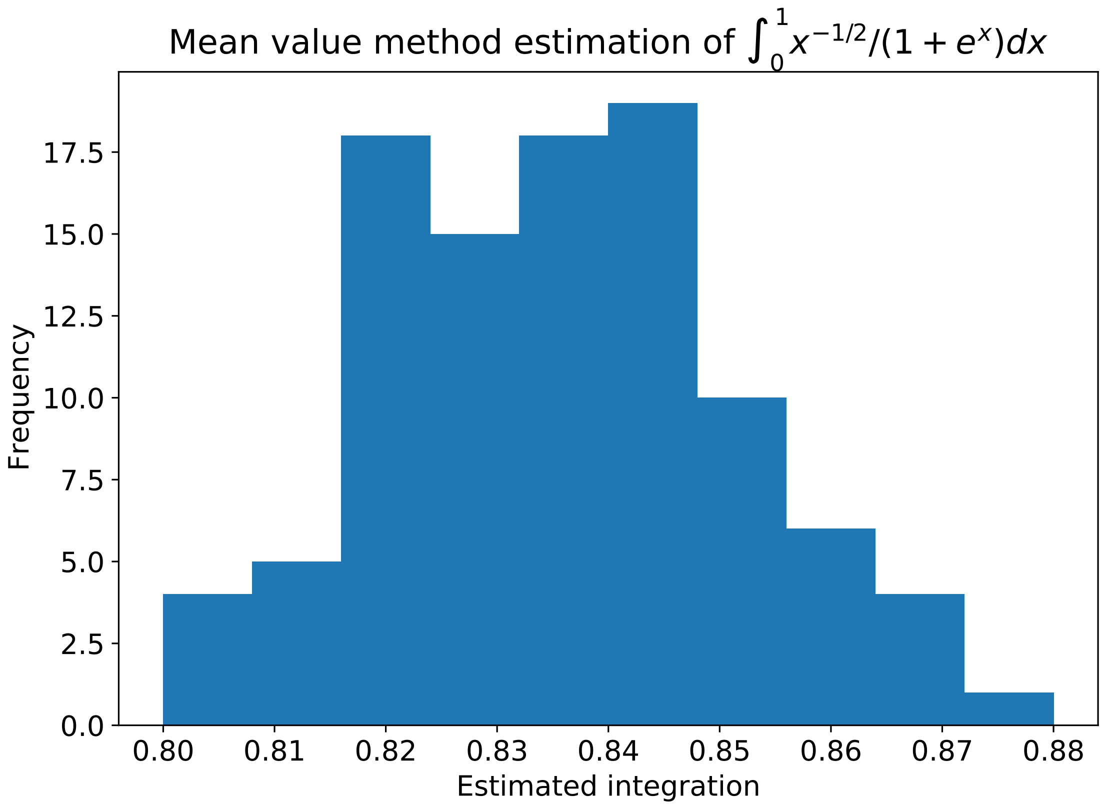

Figure 3: The histogram of the estimated integral $\int_0^1 \frac{x^{-1/2}}{1+e^x}dx$ using the mean value Monte Carlo method for 100 runs. The histogram is created using 10 bins from 0.80 to 0.88.

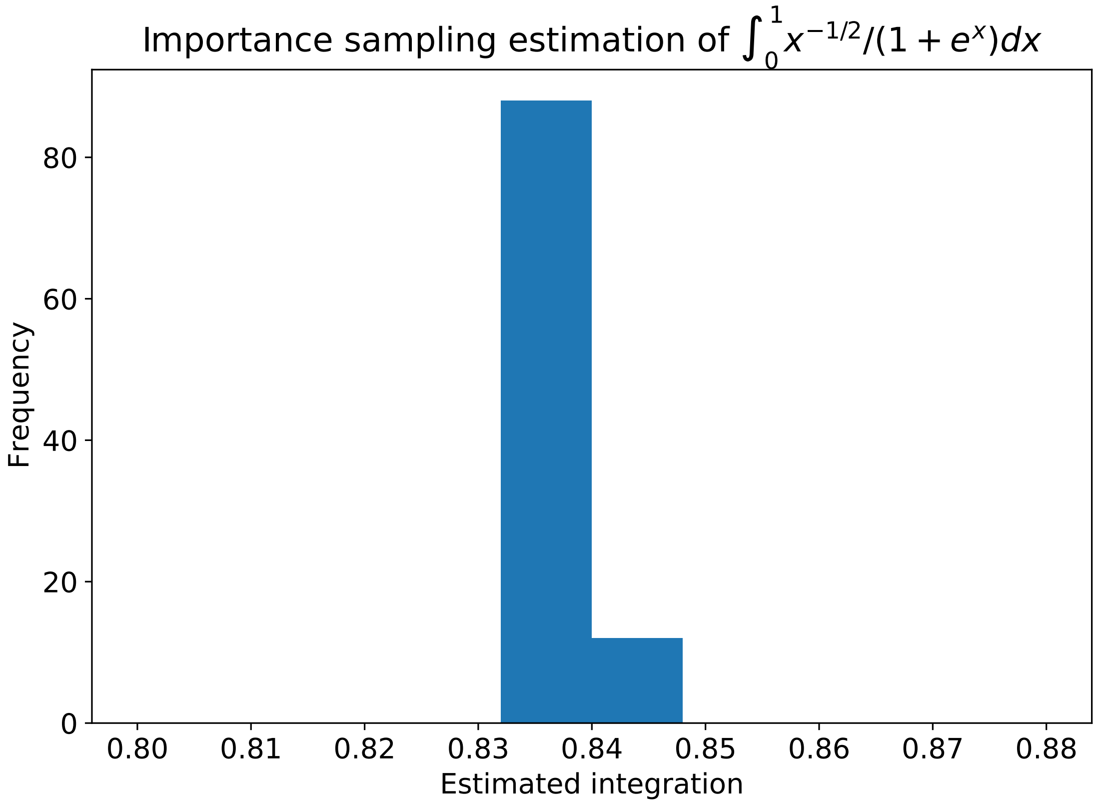

Figure 4: The histogram of the estimated integral $\int_0^1 \frac{x^{-1/2}}{1+e^x}dx$ using the importance sampling method for 100 runs. The histogram is created using 10 bins from 0.80 to 0.88.

Comparing fig. 3 to fig. 4 we notice that the integral estimation of the importance sampling method is more precise and accurate compared to the mean value method in the case of the given integral. All of the estimated integral values from the importance sampling method lie within 2 bins between 0.832 and 0.848 which is very close to the approximate real value of 0.839 (calculated using the wolfram alpha) whereas, the estimation of the mean value method varies widely from 0.8 to 0.88 with the majority and almost equal number of estimations between 0.816 and 0.848.

(b) **Next, we repeat the procedure in part (a) to estimate the integral eq. 6 given in the handout which is sharply peaked near $x = 5$.** Here, the points are sampled for a normal distribution with men of 5 and standard deviation of 1. The integral is computed by using the setup given in the handout and the histogram of the resulting integral values using 10 bins from 0.93 to 1.07 is plotted to visualize the results.

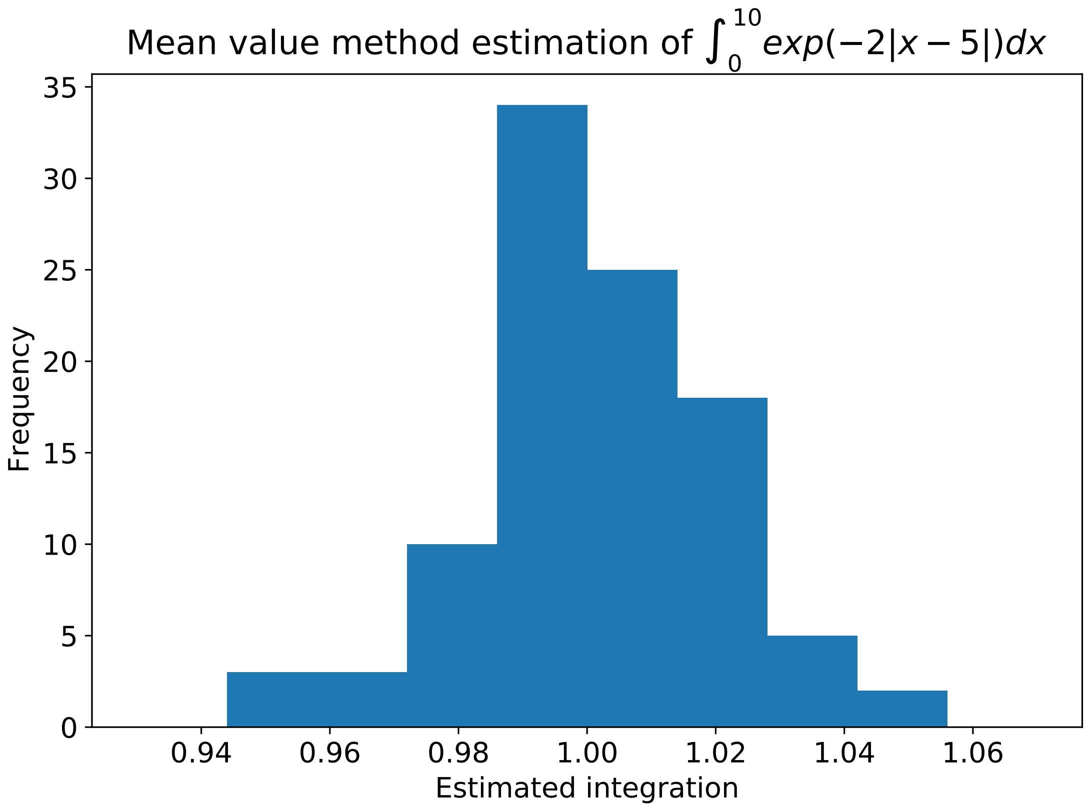

Figure 5: The histogram of the estimated integral $\int_0^{10} \exp(-2|x-5|)dx$ using the mean value Monte Carlo method for 100 runs. The histogram is created using 10 bins from 0.93 to 1.07.

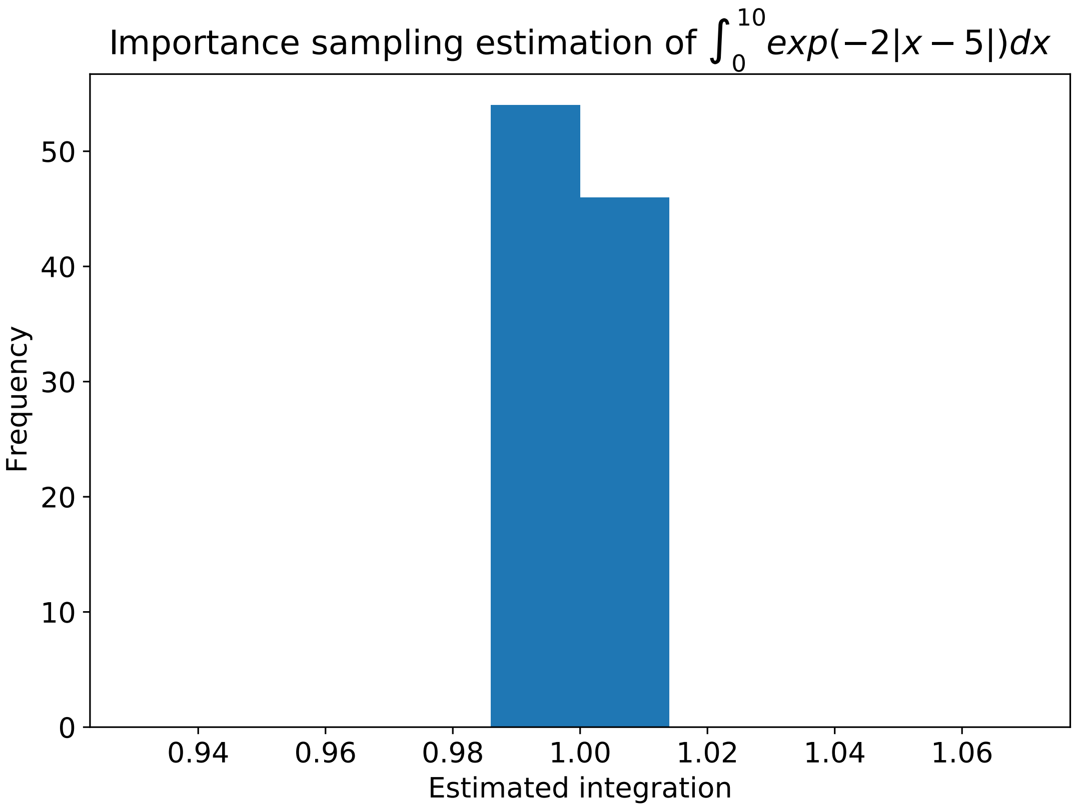

Figure 6: The histogram of the estimated integral $\int_0^{10} \exp(-2|x-5|)dx$ using the importance sampling method for 100 runs. The histogram is created using 10 bins from 0.93 to 1.07.

Again, comparing fig. 5 to fig. 6 we notice that the integral estimation of the importance sampling method is much more precise and accurate compared to the mean value method in the case of the given integral. All of the estimated integral values from the importance sampling method lie within 2 bins between 0.986 and 1.014 which is very close to the approximate real value of 0.999 (calculated using the wolfram alpha) whereas, the estimation of the mean value method varies widely from 0.944 to 1.056 with the majority of estimations between 0.972 and 1.028.