Lab 10 Report

<u>PHY407 Week 9 Assignment</u>

Landon Wang, Yishi Liu

November 29, 2022

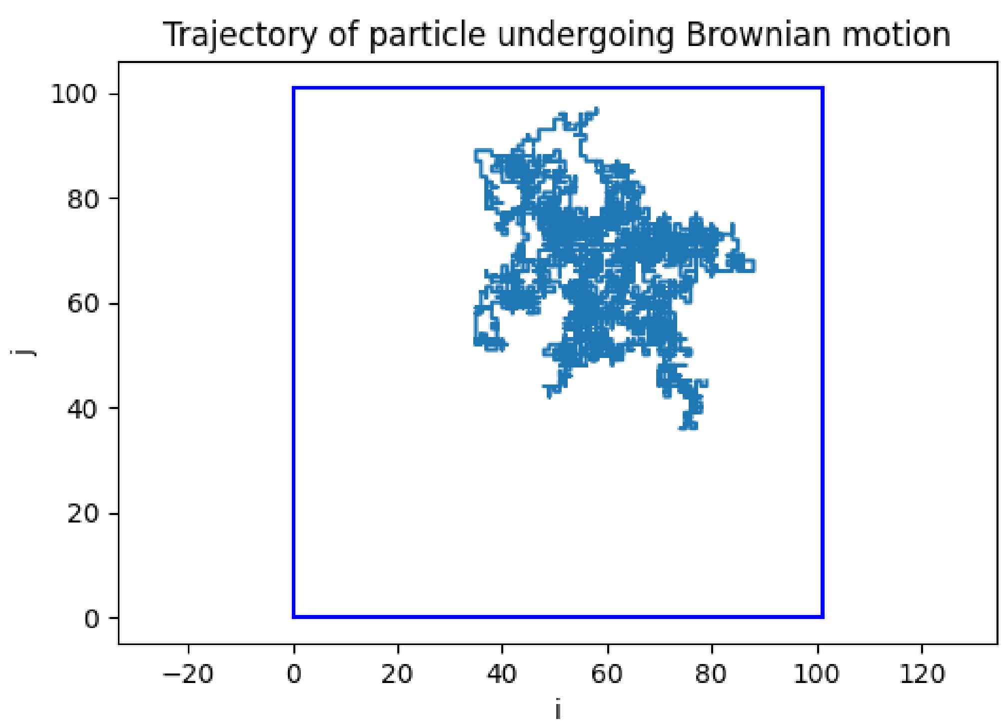
Contribution: Landon Wang writes the entirety of question 3 and completes this document, Yinshi Liu writes the entirety of question 1 and question 2.

Question 1

Q1a

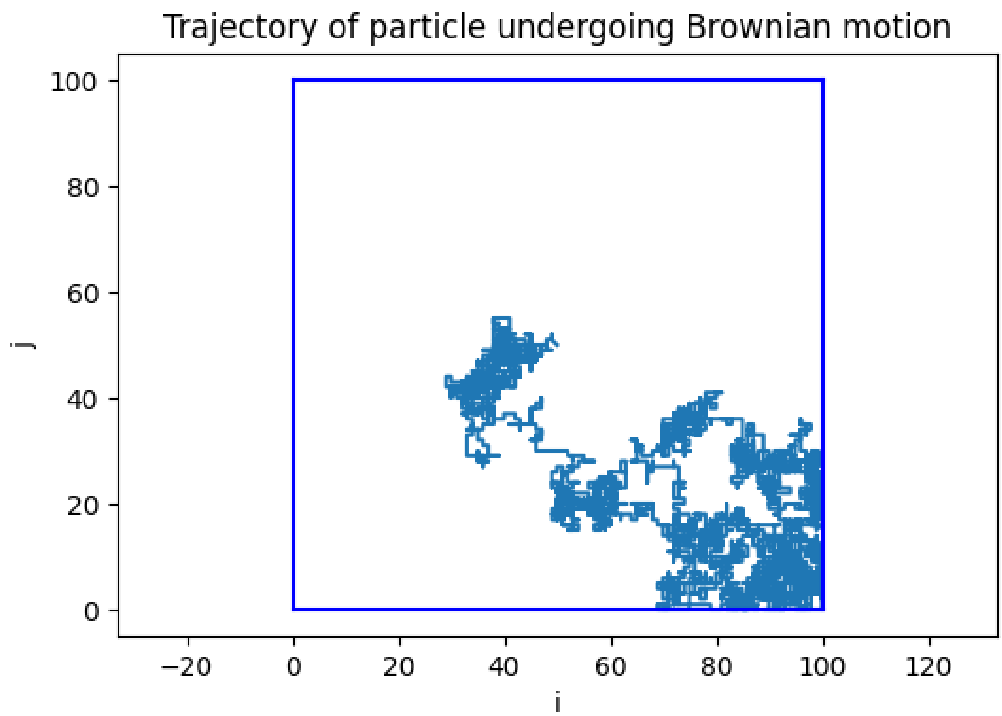
Note: The python code Brownian-start.py is modified to compute and print all figure in this part of problem.

Figure 1.1: Brownian Motion without Hitting Walls



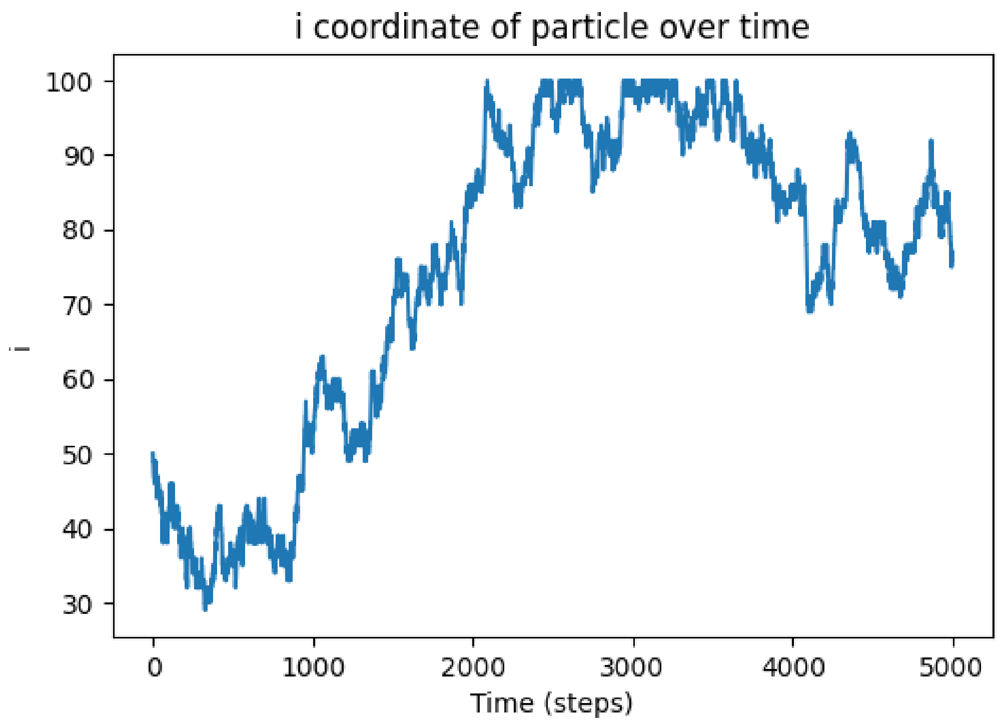
As shown from the plot, the trajectory of the particle is random.

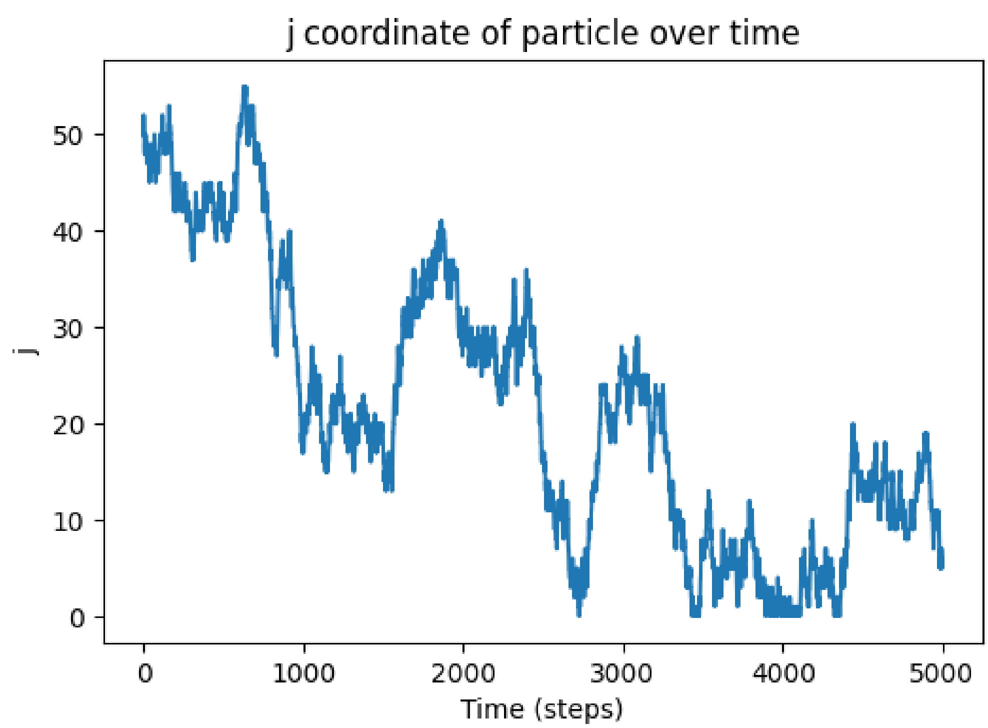
Figure 1.2: Brownian Particle Interacting with Walls



The interaction with walls caused the particle to move close to the walls. As Fig 1.3 and 1.4 has shown, the particle remained close at the boundary until the particle is able to move away due to the random force pointing away from the boundary.

Figure 1.3-1.4: Trajectory of i and j over Time.



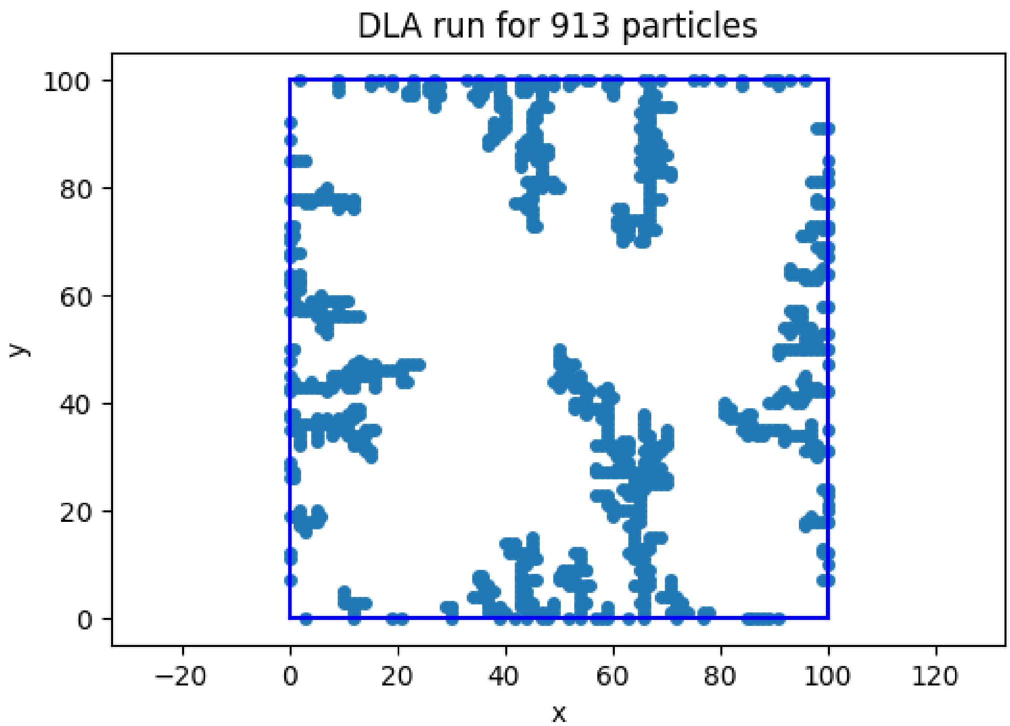


Q1b

Note: The python code DLA-start.py is modified to compute and print all figure in this part of problem.

Code Output

Figure 1.5 DLA Simulation for a 101×101 grid.



As more particles are attached to the edge of the grid, the anchored cells allow more chance for particles to connect with the anchored cells. This has created a fractal like pattern for each edge of the grid that reaches towards the center cell. The simulation ends when one of the edge has extended to the central cell, allowing no more particles to be released. The mean amount of particle for it to reach the cell in a 101 by 101 grid is around 800-900 particles.

Question 2

Code Output

After running multiple trials, the volume of a 10-D unit hypersphere by Monte Carlo is as follows:

Best run

```
The volume of a 10 Dimensional Hypersphere is: 2.546685453314547
The relative error is: 0.0013640638438951213
```

Worst Run

```
The volume of a 10 Dimensional Hypersphere is: 2.4401895598104404
The relative error is: 0.04312447291515965
```

The volume simulated via Monte Carlo is quite close to the theoretical volume of $\frac{\pi^5}{120} r^{10}$. The worst trial is still able to calculate its volume within 4% of the theoretical value. Therefore, it is safe to conclude that Monte Carlo method is able to determine the volume with good accuracy.

Question 3

For this question, we are tasked to solve the integral I using Monte Carlo methods.

$$I = \int_0^1 \frac{x^{-\frac{1}{2}}}{e^x + 1} dx \tag{3.1}$$

PART A

Let function inside the integral be f

$$f(x) = \frac{x^{-\frac{1}{2}}}{e^x + 1} \tag{3.2}$$

Now, use the mean value method, we can evaluate the integral using the formula

$$I \approx \frac{b-a}{N} \sum_i^N f(x_i) \tag{3.3}$$

Where $x_i \sim U[a, b]$, constant $a = 0, b = 1$ are the upper and lower limit of the integral

We use $N = 10000$ samples per estimation of I, and 100 estimations was made.

Result will be shown in part C

PART B

The mean value method is not very efficient, to improve it, we use the Importance sampling method.

Notice the numerator part of the function f diverges, we can treat it separately by define the function w

$$w(x) = \frac{1}{\sqrt{x}} \tag{3.4}$$

Which is the numerator of the f.

For any function g, the weighted expectation follows that

$$\langle g \rangle_w = \frac{\int w(x)g(x)dx}{\int w(x)dx} \tag{3.5}$$

Which w can be any chosen function

Use the definition 3.5 and chose weighting function as equation 3.4, we have

$$\left\langle \frac{f(x)}{w(x)} \right\rangle_w = \frac{\int (w(x)f(x))/g(x))\, dx}{\int w(x)dx} = \frac{\int f(x)dx}{\int w(x)dx} = \frac{I}{\int w(x)dx}$$

$$\Rightarrow I = \left\langle \frac{f(x)}{w(x)} \right\rangle_w \int_0^1 w(x)dx \tag{3.6}$$

Since we have chosen a weighing function, the random sample must be drawn from the weighted probability density. We define a normalized probability function p

$$p(x) = \frac{w(x)}{\int w(x)dx} \tag{3.7}$$

With the probability density function, the weighted average for any function g follows that

$$\langle g \rangle_w = \int_a^b p(x)g(x)dx \approx \frac{1}{N}\sum_i^N g(x_i) \tag{3.8}$$

Where x_i follows the probability density function p

Combine equation 3.6 and 3.8, we arrived to the solution

$$I \approx \frac{1}{N}\sum_i^N \frac{f(x_i)}{w(x_i)}\int_0^1 w(x)dx \tag{3.9}$$

Evaluate the integral about w

$$\int_0^1 w(x)dx = \int_0^1 \frac{1}{\sqrt{x}}dx = 2 \tag{3.10}$$

Plug in f and the result of 3.10, we have

$$I \approx \frac{2}{N}\sum_i^N \frac{1}{e^x + 1} \tag{3.11}$$

The probability mass function 3.7 becomes

$$p(x) = \frac{w(x)}{\int w(x)dx} = \frac{1}{2\sqrt{x}} \tag{3.12}$$

Now, we need to find a way to generate random sample following 3.12, which no prebuilt python package can do.

We can use a transformation to take a uniformly distributed sample to the desired probability

Let $p(x) = u$ where u is the uniformly distributed sample and x is the desired non-uniform transfomed sample

$$u = \frac{1}{2\sqrt{x}}$$

$$\Longrightarrow x = \left(\frac{1}{2u}\right)^2 \tag{3.13}$$

Therefore, we can apply the transformation 3.13 to the uniformly generated random samples to obtain the sample set for integrating.

PART C

One run of the result is shown in the figure 3.1 and 3.2. Notice that we did not set the seed for the random number generator, so each run might be different.

But something is very clearly shown through the two figures is the spread of the histogram is much larger with the mean value method, indicates a much larger variance to the data. In contrast to the mean value method, the importance sampling method gives a much sharper peak, with a significantly smaller variance.

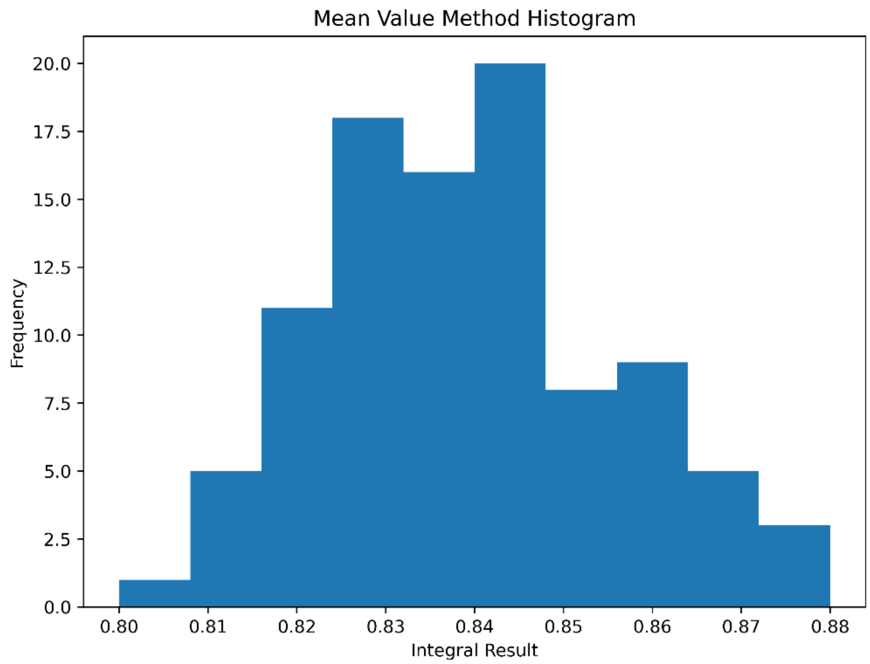


Figure 3.1: Histogram for Mean Value Method

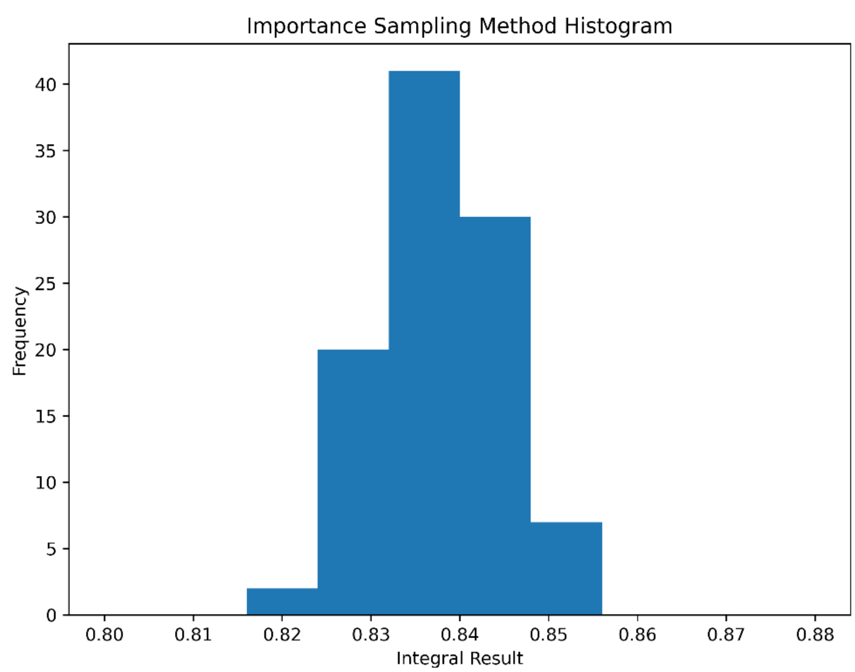


Figure 3.2: Histogram for Importance Sampling Method