PHY407 Formal Lab Report 3

Natalia Tabja

November 2024

1 Introduction

This lab explores a diverse set of computational physics techniques that leverage stochastic (i.e, randomly-determined) processes, optimization algorithms, and simulation methods to model complex physical problems. The report is divided into three main sections: Diffusion-Limited Aggregation (DLA), Simulated Annealing, and Monte Carlo Techniques for Integraion, each designed to investigate a different aspect of computational problem-solving in physics.

2 Simulating the DLA Process

First, we simulate the DLA process, which models the aggregation of particles undergoing random motion. DLA is a phenomenon seen in various physical systems, such as the formation of corals and snowflakes.

2.1 Part (a)

Initially, we simulate a random walk of a particle beginning at the center of a square grid (with sides of length $L = 101mm$).

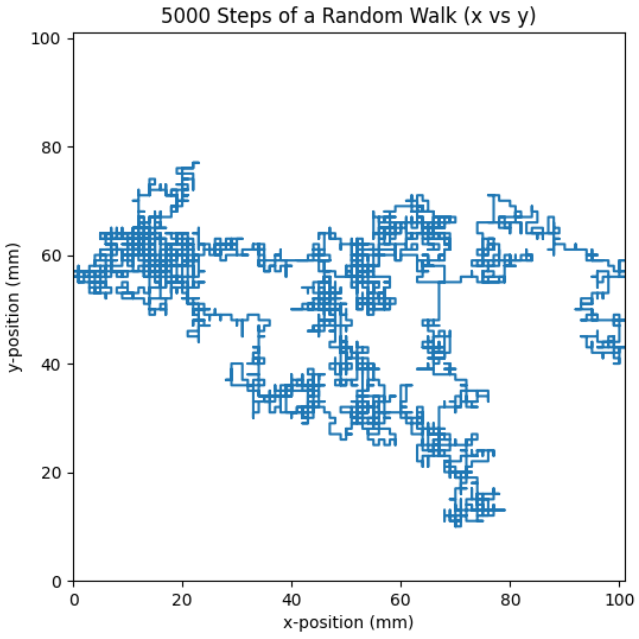

Figure 1: Trajectory of a particle undergoing a random walk on a 2D grid over 5000 steps. The particle starts at the center of the grid, and its path is shown in the figure.

For each iteration, the particle "chooses" a random direction (up, down, left, or right) and moves one step in that direction. The particle is restricted from moving outside the boundaries of the grid; If the particle reaches a wall, it can only move in the three directions that keep it within the grid. The simulation continues for $N = 5000$ time steps, and at each step, the particle's position is recorded so that the resulting trajectory can be visualized (Figure 1). Additionally, plots of x-position and y-position versus time are produced to illustrate the particle's evolution over time (Figures 2 & 3).

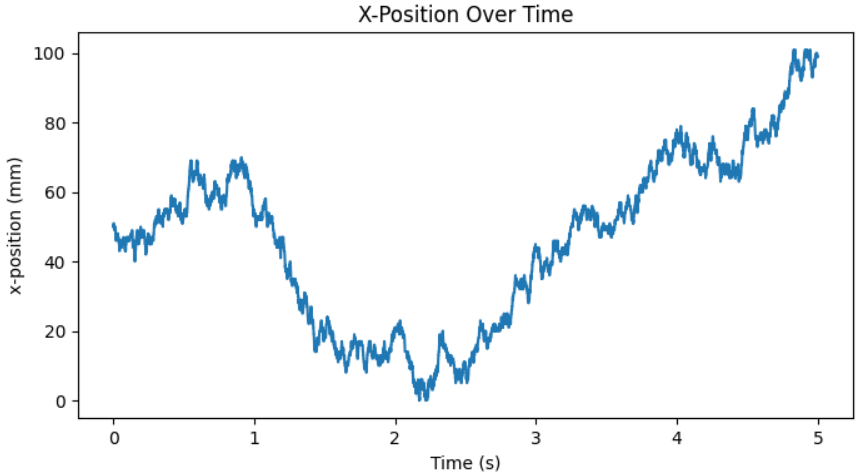

Figure 2: x-position of the particle as a function of time during the random walk. The x-coordinate fluctuates stochastically over 5000 time steps.

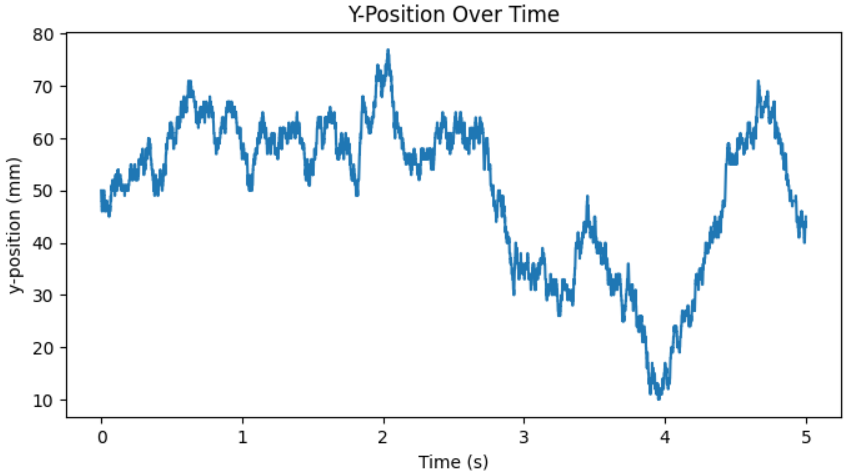

Figure 3: y-position of the particle as a function of time during the random walk. The y-coordinate fluctuates stochastically over 5000 time steps.

The random walk models the diffusion process and serves as a foundation for understanding the more complex behavior seen in DLA systems, where particles adhere to a growing structure.

2.2 Part (b)

In this part of the lab, we simulate the DLA process. The simulation begins with a particle performing a random walk on a two-dimensional grid, starting at the center. As additional particles are introduced, they

undergo random motion until they either attach to a pre-existing anchored particle or reach the boundaries of the grid.

The simulation stops once a particle anchors at the grid's center. By visualizing the aggregation pattern, we can observe how fractal-like structures emerge as a result of this stochastic process. This part of the lab introduces fundamental concepts in random walks, boundary conditions, and particle-based simulations.

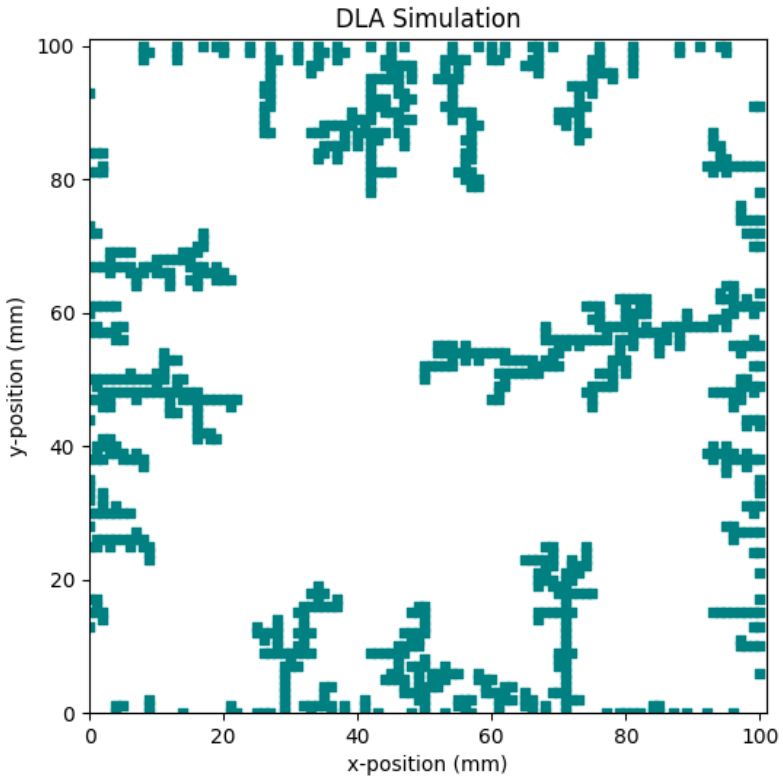

Figure 4: Fractal-like structure formed by Diffusion-Limited Aggregation (DLA). Particles start at the center and aggregate to create the characteristic branching patterns.

3 Efficiency of Simulated Annealing

The second section focuses on Simulated Annealing (SA), an optimization algorithm inspired by the physical process of annealing, where materials are heated and cooled slowly to reduce stress. The algorithm for SA is probabilistic and is used to find the global minimum of a function, where this minimum is akin to the ground state of the system. It involves exploring the solution space by repeatedly transitioning between states with a probability dependent on "temperature" (this is called the "acceptance probability"). As the temperature decreases, the acceptance probability lowers since the system has less energy. The temperature of the system is decreased over time to nearly zero, with a rate determined by the constant τ.

3.1 Part (a)

Initially, we apply SA to a simple two-dimensional function to identify its global minimum:

$$f(x, y) = x^2 - \cos(4\pi x) + (y - 1)^2.$$

The temperature is reduced exponentially according to a cooling schedule, with the following parameters:

- $T_0 = 10000$ (Initial Temperature)
- $T_f = 0.001$ (Final Temperature)
- $\tau = 100$ (Time Constant)

The global minimum of the function is located at $(x, y) = (0, 1)$. The simulation resulted in a final output of:

$$x = -0.00052, \quad y = 0.99573, \quad f(x, y) = -0.99996.$$

The trajectory of the system as it converges to this global minimum is visualized below.

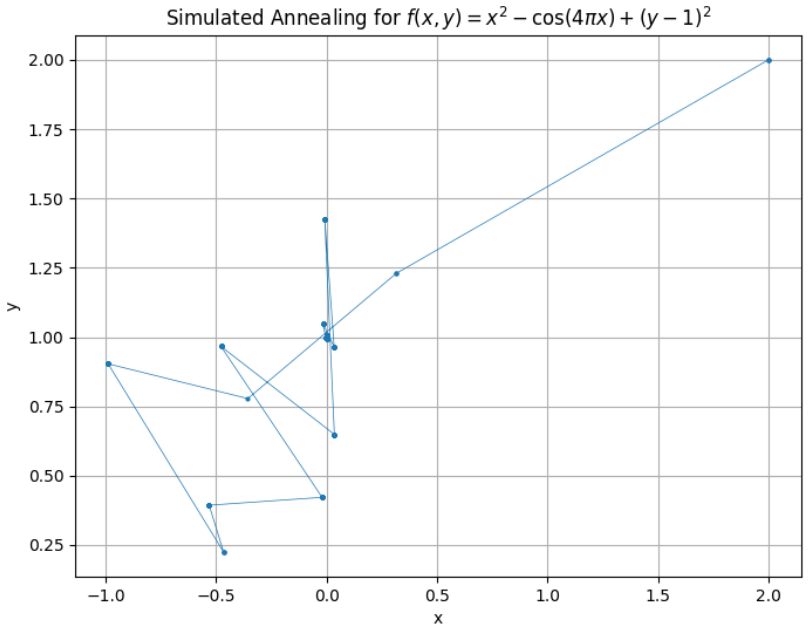

Figure 5: Simulated Annealing applied to $f(x, y) = x^2 - \cos(4\pi x) + (y - 1)^2$. The trajectory shows the system converging to the global minimum at $(x, y) \approx (0, 1)$.

3.2 Part (b)

We then extend this approach to a more complex function with multiple minima:

$$f(x, y) = \cos(x) + \cos(\sqrt{2}x) + \cos(\sqrt{3}x) + (y - 1)^2,$$

where $0 < x < 50$ and $-20 < y < 20$. The system rejects any (x, y) values outside this range. The correct global minimum of this function lies near $(x, y) = (16, 1)$, but other local minima exist, such as at $(x, y) \approx (2, 1)$ and $(x, y) \approx (42, 1)$. The simulation resulted in the following final output:

$$x = 2.15006, \quad y = 1.00622, \quad f(x, y) = -2.37741.$$

This demonstrates how SA avoids getting trapped in local minima and converges to the global minimum, showcasing its effectiveness in optimization problems.

The cooling schedule parameters used in this simulation are the same as in Part (a):

- $T_0 = 10000$ (Initial Temperature)
- $T_f = 0.001$ (Final Temperature)
- $\tau = 100$ (Time Constant)

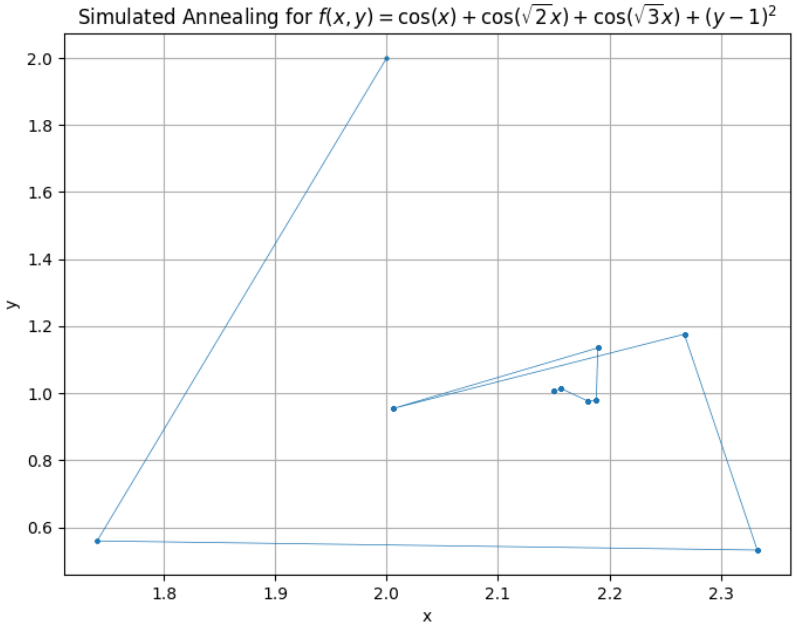

Figure 6: Simulated Annealing applied to $f(x, y) = \cos(x) + \cos(\sqrt{2}x) + \cos(\sqrt{3}x) + (y - 1)^2$. The system explores the solution space and converges to the global minimum near $(x, y) \approx (2.15, 1.01)$.

3.3 Error Analysis and Comparison

To evaluate the accuracy of the Simulated Annealing method, we compare the results for both functions to their known global minima. The table below summarizes the true values, computed results, and the absolute and relative errors for each case:

	$f(x,y) = x^2 - \cos(4\pi x) + (y-1)^2$	$f(x,y) = \cos(x) + \cos(\sqrt{2}x) + \cos(\sqrt{3}x) + (y-1)^2$
True Value	-1.00000	-2.31595
Result	-0.99996	-2.37741
Abs Err	0.00004	-0.06146
Rel Err	-0.00004	0.02654

Table 1: Comparison of Simulated Annealing results for the two functions, showing absolute and relative errors.

From the table, we observe that the Simulated Annealing method performs more efficiently for the simpler function in Part (a), achieving a much smaller relative error. For the more complex function in Part (b), the presence of multiple local minima results in a higher relative error, highlighting the increased difficulty of optimization in this case. Nonetheless, the method successfully converges to a value near the global minimum, demonstrating its robustness in navigating challenging solution spaces.

4 Monte Carlo Techniques and Importance Sampling

The third section of the lab delves into Monte Carlo (MC) methods for numerical integration, which are useful for evaluating integrals that are analytically hard to solve or involve high-dimensional spaces. Two specific methods are explored: the Mean Value Monte Carlo (MVMC) method and Importance Sampling (IS).

The MVMC method estimates the integral by averaging the integrand's value over randomly sampled points within the domain. Although straightforward, this method struggles with integrands that vary sharply or contain singularities. In order to limit these issues, Importance Sampling takes points from a distribution that mimics the shape of the integrand, thus improving accuracy.

4.1 Part (a): Mean Value Monte Carlo Method

We begin by evaluating the integral:

$$\int_0^1 \frac{x^{-1/2}}{1+e^x}\,dx,$$

using the MVMC method. Note that the integrand has a singularity at $x = 0$ due to the $x^{-1/2}$ term. The MVMC method involves uniformly sampling points within the domain $[0, 1]$ and averaging the integrand's value at these points:

$$I \approx (b-a)\frac{1}{N}\sum_{i=1}^{N} f(x_i),$$

where $f(x)$ is the integrand, and $N = 10,000$ is the number of sample points. The calculation is repeated 1,000 times to account for variations in the results, and the mean of these results is taken as the final estimate.

Using this method, the computed value of the integral is:

$$\text{Mean Value MC Result: } 0.8382644309708749 \pm 0.01176.$$

However, while MVMC is simple and effective for most integrals, it is not optimal for those with singularities, as uniformly sampling points leads to poor coverage of the most significant regions (e.g., near $x = 0$).

4.2 Part (b): Importance Sampling Method

To address the aforementioned issues of uniform sampling, we apply IS, which rewrites the integral as:

$$I = \int_0^1 \frac{f(x)}{w(x)} w(x)\,dx.$$

Here, $w(x)$ is a weighting function chosen to closely match the shape of the integrand and to remove the singularity. For this integral, we set $w(x) = x^{-1/2}$, which cancels the singularity in the integrand. This choice leads to a simpler evaluation:

$$I \approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{w(x_i)},$$

where x_i are sampled from the probability density function $p(x) = \frac{w(x)}{\int_0^1 w(x)\,dx}$.

Using IS, the computed value of the integral is:

$$\text{Importance Sampling MC Result: } 0.8388699629092403 \pm 0.00141.$$

The IS result closely matches the MVMC result but with a significantly reduced error. By allocating more points near $x = 0$, where the integrand is most significant, IS demonstrates its effectiveness in handling singularities.

4.3 Comparison of Methods

To compare the accuracy of the two methods, we present the following table summarizing the results and errors:

	MVMC	**IS**
Result	0.8382644309708749	0.8388699629092403
Error	± 0.01176	± 0.00141

Table 2: Comparison of results and errors for Mean Value Monte Carlo and Importance Sampling

Analysis: From the table, we observe that IS significantly reduces the error compared to MVMC. This improvement demonstrates the advantage of sampling points from a distribution closely matching the integrand's shape, particularly for challenging cases like singular integrals.

4.4 Part (c): Histograms of Monte Carlo Results

To compare the results obtained from MVMC and IS, we plot histograms of the integral estimates generated over 1,000 iterations. Each histogram uses 100 bins to show the distribution of the estimates.

Mean Value Monte Carlo (MVMC): The MVMC histogram reveals that the estimates are widely spread around the computed mean, reflecting the method's limited precision when dealing with singular integrands. The broad spread indicates that MVMC requires a large number of samples to achieve acceptable accuracy for challenging integrands.

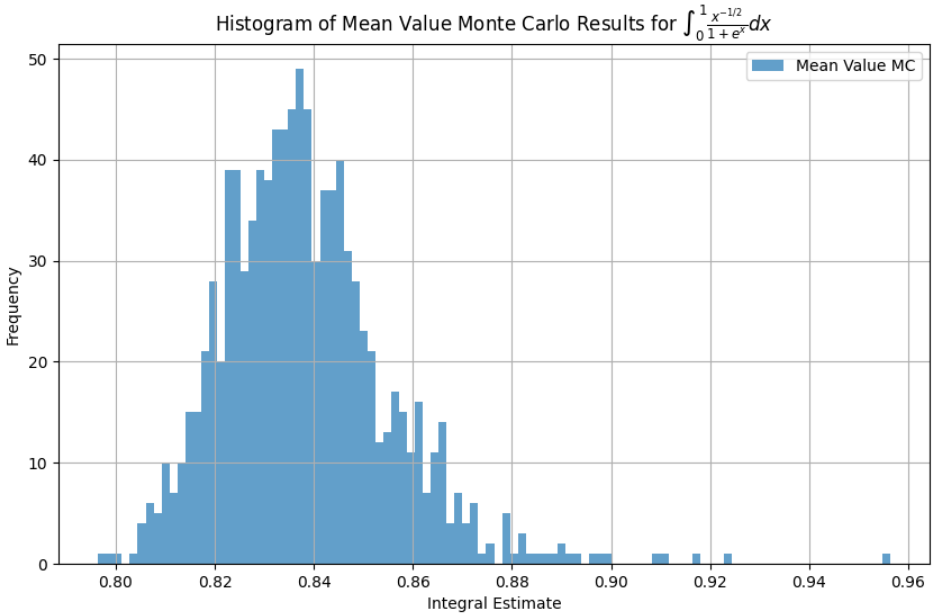

Figure 7: Histogram of integral estimates using the MVMC method. The wide spread reflects the method's difficulty in accurately estimating integrals with singularities.

Importance Sampling: The IS histogram demonstrates a much narrower distribution around the mean value (note that despite the wider appearance in the plot, this is due to the smaller range of the x-axis). By sampling more points near $x = 0$, IS ensures that the most significant contributions to the integral are well-represented. This leads to improved precision compared to MVMC, as reflected in the tighter clustering of the estimates.

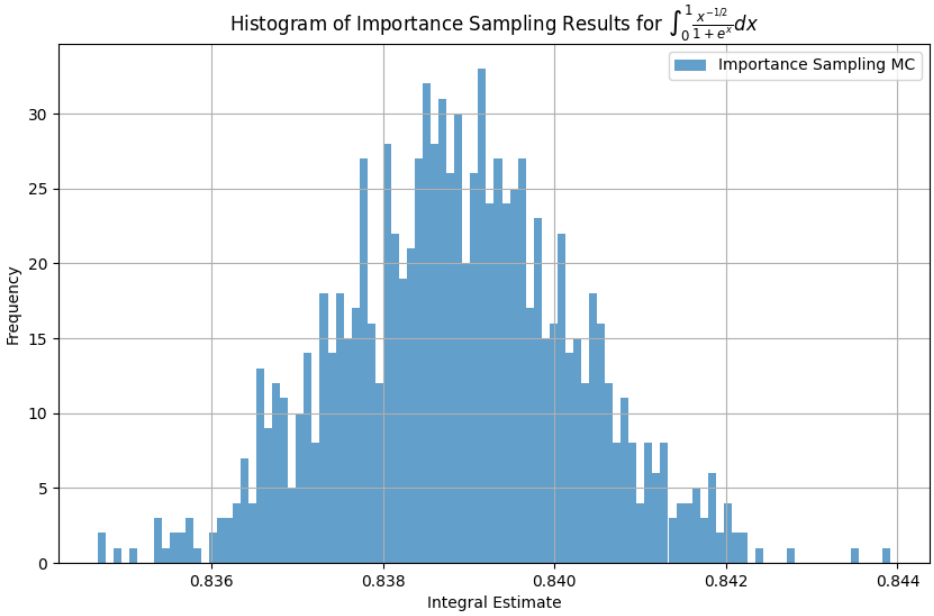

Figure 8: Histogram of integral estimates using the IS method. The narrow spread highlights its improved precision by focusing on regions where the integrand is most significant.

4.5 Part (d): Importance Sampling for a Sharply Peaked Integrand

In this part, we apply Importance Sampling to evaluate the integral:

$$I = \int_0^{10} \exp(-2|x-5|)\, dx.$$

The integrand is sharply peaked near $x = 5$. Uniformly distributed points across the interval $[0, 10]$ would result in many samples barely contributing to the integral, as most of the function's value is concentrated near the peak. Thus, we use Importance Sampling with the weighting function:

$$w(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-5)^2}{2}\right).$$

Since $w(x)$ integrates to approximately 1 over the interval $[0, 10]$, it closely matches the shape of the integrand and ensures that more points are sampled near $x = 5$.

Implementation: The chosen weighting function $w(x)$ matches the probability density function (PDF) of a normal distribution with a mean $\mu = 5$ and a standard deviation $\sigma = 1$. Therefore, we directly sample x values from this normal distribution using the built-in function `numpy.random.normal`. This eliminates the need to compute a transformation from uniformly distributed random variables.

Results: Using 10,000 sample points per iteration and repeating the calculation 1000 times, the average integral estimate was:

$$\text{Importance Sampling Result (Part d): } 0.9999760237566945 \pm 0.00600.$$

Histogram: The histogram of the results demonstrates a tightly clustered distribution, reflecting the high precision achieved by sampling effectively from the region of greatest contribution.

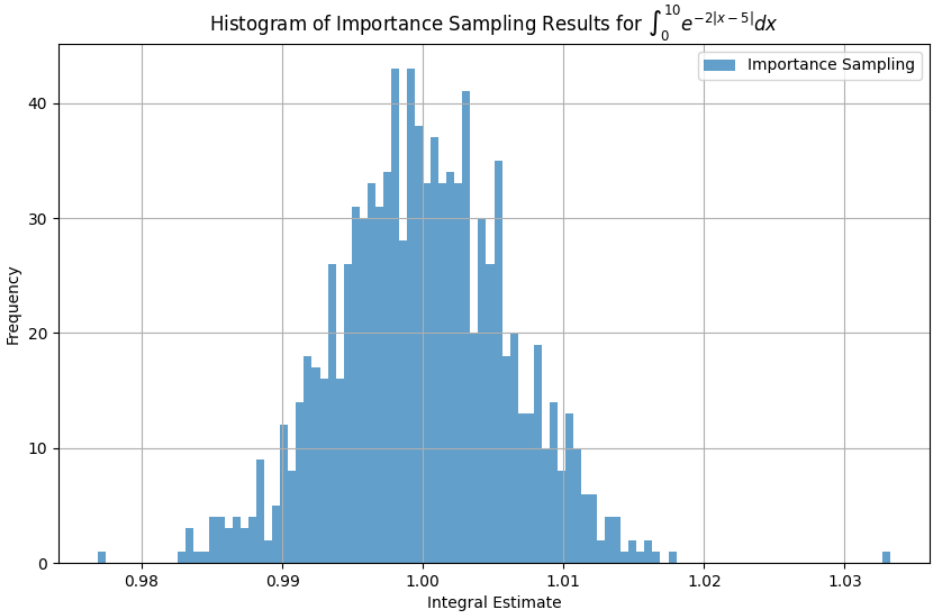

Figure 9: Histogram of IS results for $\exp(-2|x-5|)$. The narrow spread demonstrates the precision achieved by sampling from the normal distribution centered at $x = 5$.