

Integrative Approaches to Simulating Random Variables, Hamiltonian Dynamics, and Markov Chains

Author:

Alfonso Mateos Vicente

Tutor:

Noé Bassel



École des Ponts
ParisTech

Ingénierie Mathématique et Informatique
École des Ponts ParisTech
France
September 01, 2023

Contents

1 Simulation of Random Variables	4
1.1 Inverse CDF method	4
1.2 Rejection Sampling method	6
1.2.1 Selection of the enveloping distribution	6
1.2.2 Results of the Rejection Sampling method	7
1.3 Empirical validation of important statistical laws	8
1.3.1 Empirical validation of the Law of Large Numbers	9
1.3.2 Empirical Validation of the Central Limit Theorem	10
1.4 Variance reduction techniques	12
1.4.1 Preliminaries	13
1.4.2 Control variates	13
1.4.3 Importance sampling	14
1.4.4 Stratified sampling	15
1.4.5 Antithetic variates	17
1.4.6 Comparative Review	18
2 Hamiltonian dynamics	21
2.1 Introduction to Hamiltonian dynamics	21
2.2 Symplectic schemes	23
2.2.1 Analytical solution	23
2.2.2 Euler method	25
2.2.3 Failure of standard methods	26
2.2.4 Constructing Symplectic Schemes	27
2.2.5 Symplectic Euler method	28
2.2.5.1 Linear Stability Analysis of the Symplectic Euler Scheme	30
2.2.6 Störmer-Verlet method	31
2.2.6.1 Linear Stability Analysis of the Verlet Scheme	33
2.3 Solar System Simulation	34
2.3.1 Modeling the Solar System: Gravitational Dynamics	34
2.3.2 Störmer-Verlet Scheme for Gravitational Dynamics	35
2.3.3 Implementation of the Solar System Simulation	37
2.3.3.1 Experiment 1: Expanding to the N-Body Problem	39
2.3.3.2 Experiment 2: Using Symplectic Euler Method	40
3 Markov Chains and their Applications	42
3.1 Introduction to Markov Chains	43
3.1.1 Stationary Distribution	43
3.1.2 Ergodic Theorem	44
3.1.3 Irreducibility and Uniqueness	44

3.2	Markov Chain Monte Carlo Methods	44
3.2.1	Metropolis-Hastings Algorithm	44
3.2.1.1	Example: Sampling from a Gamma Distribution	45
3.2.2	Gibbs Sampling	46
3.2.2.1	Example: Sampling from a Bivariate Normal Distribution	46
3.2.3	Müller-Brown Potential	48
3.2.3.1	Boltzmann Distribution	49
3.2.3.2	Metropolis-Hastings Algorithm for the Müller-Brown Potential . .	50
3.2.3.3	Gibbs Sampling Algorithm for the Müller-Brown Potential	50
3.2.3.4	Comparison between Metropolis-Hastings and Gibbs Sampling	51
3.3	Molecular System Simulation	53
3.3.1	Theoretical Framework and Lennard-Jones Potential	53
3.3.2	Implementation of the model	54
3.3.2.1	Implementation 2D using the Metropolis-Hastings algorithm	55
3.3.2.2	Implementation 3D using the Gibbs Sampling algorithm	56

Introduction

This comprehensive paper delves into the intricacies of simulation techniques, spanning across the domains of random variable simulation, Hamiltonian dynamics, and Markov chain sampling, with an emphasis on their diverse applications in computational statistics and mathematical modeling.

In the first section, the focus is on the simulation of random variables. We begin with a detailed exploration of the Inverse Cumulative Distribution Function (CDF) method and the Rejection Sampling method, providing an extensive analysis of various enveloping distributions and their results. This part also includes an empirical validation of the Law of Large Numbers and the Central Limit Theorem, establishing the reliability and efficiency of these methods. We further investigate several variance reduction techniques, including control variates, importance sampling, stratified sampling, and antithetic variates, offering a comparative review of their effectiveness in enhancing simulation accuracy and efficiency.

The second section transitions to Hamiltonian dynamics, starting with its foundational principles. We discuss symplectic schemes, particularly focusing on their analytical solutions and critically evaluating the Euler method and the limitations of standard methods. Advanced symplectic schemes, such as the Symplectic Euler and Stormer-Verlet methods, are presented, including linear stability analysis. This section culminates with practical applications in simulating the Solar System's gravitational dynamics and expanding to the N-Body Problem, highlighting the use of the Symplectic Euler Method.

The final section is dedicated to Markov chain sampling, offering a deep dive into the Metropolis-Hastings algorithm and Gibbs Sampling. Practical applications are provided through sampling from Gamma and Bivariate Normal distributions and the simulation of the Müller-Brown Potential, with comparative analyses of their effectiveness. The section concludes with an innovative exploration of simulating molecular systems, detailing the implementation of a 2D molecular system using the Metropolis-Hastings algorithm and extending this simulation to a 3D molecular system using the Gibbs Sampling algorithm. The use of the Lennard-Jones potential in these simulations demonstrates the versatility of Markov chain sampling methods in modeling complex physical interactions in both two and three dimensions.

Overall, this paper not only provides a thorough overview of advanced simulation techniques but also showcases their practical applications, significantly contributing to the fields of computational statistics and mathematical modeling.

Chapter 1

Simulation of Random Variables

This chapter explores the simulation of random variables, a cornerstone in statistical and dynamical system studies. In Section 1.1 we introduce the Inverse CDF method providing a foundational overview of its mechanics and application in simulating random variables.

Progressing to Section 1.2, the manuscript transitions to the Rejection Sampling method. This section is further divided to enhance clarity and depth: Section 1.2.1 delves into the selection of the appropriate enveloping distribution, and Section 1.2.2 presents the outcomes and insights from applying this method.

The narrative continues in Section 1.3.1, where the empirical validation of the Law of Large Numbers is discussed, and in Section 1.3.2, focusing on the Central Limit Theorem. These sections underscore the practical validation and implications of these pivotal statistical laws in simulation scenarios.

Concluding this section, a detailed examination of various variance reduction techniques is presented in Section 1.4. Starting with fundamental concepts in Section 1.4.1, it then explores control variates (Section 1.4.2), importance sampling (Section 1.4.3), stratified sampling (Section 1.4.4), and antithetic variates (Section 1.4.5), culminating in a comparative review in Section 1.4.6. This part of the manuscript synthesizes and compares these techniques, highlighting their role in enhancing the accuracy and efficiency of simulation processes.

1.1 Inverse CDF method

Inverse CDF method is a basic method to generate pseudo-random numbers from any probability distribution given its cumulative distribution function. First of all, let's introduce the following statement extracted from [2]:

Theorem 1.1.1 (Inverse Transform Sampling). For any random variable $X \in \mathbb{R}$, the random variable $F_X^{-1}(U)$ has the same distribution as X , where F_X^{-1} is the generalized inverse of the cumulative distribution function F_X of X and U is uniform on $[0, 1]$.

For continuous random variables, the inverse probability integral transform is indeed the inverse of the probability integral transform, which states that for a continuous random variable X with cumulative distribution function F_X , the random variable $U = F_X(X)$ is uniform on $[0, 1]$.

Knowing this, the idea is to generate a uniform random variable U and then apply the inverse of

the cumulative distribution function F^{-1} to obtain a random variable with the desired distribution. The Algorithm 1 is as follows:

Algorithm 1: Inverse CDF method

1. Generate a set of random numbers $U \sim \mathcal{U}(0, 1)$;
 2. Find the inverse of the cumulative distribution function F^{-1} ;
 3. Apply the inverse to the set of random numbers

$$X = F^{-1}(U) ;$$
-

Let's see this with some examples. Using the exponential distribution, we know its probability density function is $f(x) = \lambda e^{-\lambda x}$. Also, we already know its cumulative distribution function which is $F(x) = 1 - e^{-\lambda x}$. Then, the inverse of the cumulative distribution function is:

$$F^{-1}(x) = -\frac{1}{\lambda} \ln(1 - x) \quad (1.1)$$

So, we can generate a set of random numbers $U \sim \mathcal{U}(0, 1)$ and apply the inverse to obtain a set of random numbers $X = F^{-1}(U) \sim \mathcal{E}(\lambda)$. The Figure 1.1 shows the histogram of the generated random numbers and the probability density function of the exponential distribution.

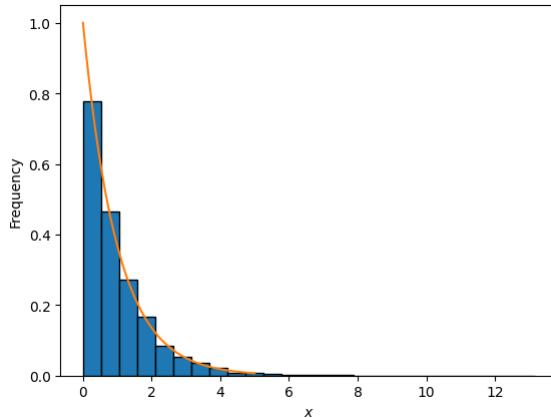


Figure 1.1: Histogram of Equation (1.1) with $\lambda = 1$ applied to \mathcal{U} .

With this picture, we can see that the histogram of the generated random numbers aligns remarkably well with the target distribution, however, we need to make more trials to be sure. We already know the theoretical mean and variance of the exponential distribution, so we can compare them with the mean of the samples while the number of trials increases.

We define the error as:

$$\text{error} = \frac{1}{n} \sum_{i=1}^n |\hat{x}_i - \mu|$$

where \hat{x}_i is the i -th random number generated and μ is the theoretical mean of the distribution. Recall that the uniform and exponential means are:

$$\mu_{\mathcal{U}} = \frac{a + b}{2}$$

$$\mu_{\mathcal{E}} = \frac{1}{\lambda}$$

The Figure 1.2 shows the error to the theoretical mean as the number of trials increases for the uniform and exponential distribution.

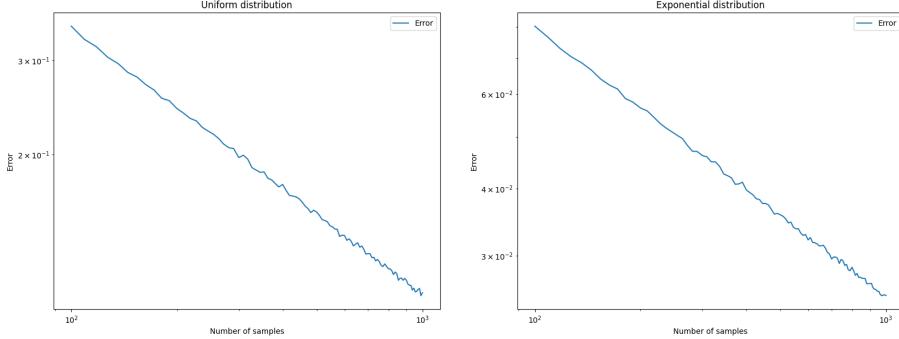


Figure 1.2: Error to the theoretical mean as the number of trials increases for the uniform distribution (left) and exponential distribution (right). Average of 10000 trials per point.

In drawing conclusions from the observed alignment of the histograms of generated random numbers with the target distribution, there appears to be a noteworthy relation, as evidenced by Figure 1.2, since the error tends to zero as the number of trials increases.

1.2 Rejection Sampling method

The Rejection Sampling method is a method to generate pseudo-random numbers for any probability distribution given its probability density function. The idea is to generate a set of random numbers from a probability distribution that is easy to sample from and then reject the numbers that are not in the desired distribution. The Algorithm 2 is as follows:

Algorithm 2: Rejection Sampling method

1. Generate a set of random numbers $X \sim g(x)$;
 2. Generate a set of random numbers $U \sim U(0, 1)$;
 3. If $U \leq \frac{f(X)}{Mg(X)}$ then accept X , otherwise reject X ;
-

Being $g(x)$ the probability density function with which we already know how to generate random numbers, denoted as the "enveloping" distribution; $f(x)$ the target probability density function; and M a factor we can choose manually and can be optimized. The formal explanation of why this algorithm works can be found in [12].

Let's see this with one example. We want to generate a set of random numbers from the following probability density function:

$$f(x) = 0.3e^{-0.2x^2} + 0.7e^{-0.2(x-5)^2} \quad (1.2)$$

1.2.1 Selection of the enveloping distribution

In this subsection we will explain the process of selecting an optimised "enveloping" distribution. In fact, we could use any distribution that envelops the desired one, but this will probably be very

inefficient because there will be many samples that will not be under the objective function, so our goal is to obtain a distribution that envelops the desired one but with the smallest possible space between them. In this report we have chosen the normal distribution as the "enveloping" distribution because it is easy to sample by using the *Quantile function* as the inverse CDF and applying the previous sampling method, and it is a good candidate for enveloping the desired distribution. Knowing this, we have three parameters to optimise: the mean, μ ; the variance, σ ; and the scale parameter, M . Note that for this problem, we have to define the bounds as a, b .

First of all, let's start by defining the scale parameter M . In this moment, what we are looking for is a parameter that envelops the desired distribution, therefore it is a good idea to define M as follows:

$$M = \max_{x \in \mathbb{R}} \frac{f(x)}{g(x)} \quad (1.3)$$

In this way, we ensure that whatever the function is, we will envelope it. Knowing this parameter, we only have to choose μ and σ in the order of minimizing M . So we have a problem of optimization with two variables. The problem is the following:

$$\begin{aligned} & \min_{\mu, \sigma} M(\mu, \sigma) \\ \text{s.t. } & M(\mu, \sigma) = \max_{x \in \mathbb{R}} \frac{f(x)}{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}} \\ & \sigma \geq 0, a \leq \mu \leq b \end{aligned} \quad (1.4)$$

We can solve this problem with multiple methods. In our case, we have chosen the Nelder-Mead's method (also called downhill simplex method) [11].

1.2.2 Results of the Rejection Sampling method

Once we have selected the "enveloping" distribution, we can apply the Rejection Sampling method. In our case, the enveloping function is the $N(\mu \approx 3.644, \sigma \approx 3.041)$ and the scalar parameter $M \approx 1.545$. The Figure 1.3 shows the histogram of the generated random numbers and the probability density function of the desired and the easy distribution.

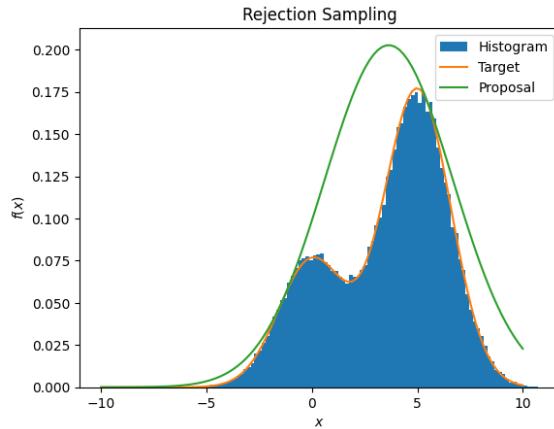


Figure 1.3: Histogram of the generated random numbers, Equation (1.2) and PDF of $\mathcal{N}(3.644, 3.041)$ scaled by $M \approx 1.545$.

First of all, note that the distribution chosen with the Equation 1.4 fits perfectly with the target distribution. Moreover, we can see that the histogram of the generated random numbers aligns remarkably well with the target distribution. In addition to the graph display itself, we can now also compare the mean of the samples with the theoretical mean as the number of trials increases. The Figure 1.4 shows the error to the theoretical mean. As disclaimer, we will use the following approximation for the mean: $\frac{1}{n} \sum_{i=1}^n x_i$ and the following approximation for the variance: $\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$.

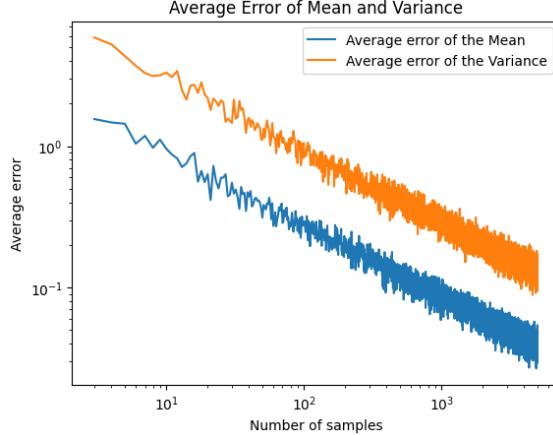


Figure 1.4: Error to the theoretical mean as the number of trials increases

In drawing conclusions we can see that the error tends to zero as the number of trials increases, so we can conclude that the Rejection Sampling method is a good method to generate pseudo-random numbers from any probability distribution given its probability density function.

1.3 Empirical validation of important statistical laws

In this section, we will empirically validate the Law of Large Numbers and the Central Limit Theorem. These laws are pivotal in probability theory and have profound implications in statistical

analysis and modeling. The empirical validation of these laws will serve to corroborate their theoretical underpinnings and substantiate their practical relevance in simulation scenarios.

1.3.1 Empirical validation of the Law of Large Numbers

This section explores the empirical validation of the Law of Large Numbers, a pivotal theorem in probability theory. The Law of Large Numbers (LLN) is a fundamental theorem in probability theory, asserting that the average of a sequence of independent and identically distributed random variables with finite expected value converges to this expected value as the number of terms in the sequence increases. Formally, the theorem can be expressed as follows, extracted from [10]:

Theorem 1.3.1 (Law of Large Numbers). Let $(X_j)_{j \geq 1}$ be a sequence of independent and identically distributed random variables with $\mathbb{E}(X_1^2) < +\infty$ then

$$\mathbb{E}(\bar{X}_n - \mathbb{E}(X_1))^2 = \frac{\text{Var}(X_1)}{n} \xrightarrow{n \rightarrow \infty} 0 \quad (1.5)$$

Given our established ability to generate random numbers reflecting specific probability distributions, we can select a distribution, generate a set of numbers corresponding to it, and then contrast the calculated averages and means of the target distribution while escalating the number of trials. Now we can define the following error:

$$\text{error} = \frac{1}{n} \sum_{i=1}^n (\hat{x}_i - \mu)^2 \quad (1.6)$$

where \hat{x}_i is the i -th random number generated and μ is the theoretical mean of the distribution. Note that in the theorem was stated that the esperance of the difference, behaves as the variance divided by the number of trials, so we can plot the variance as the number of trials increases and calculate the slope of the regression line. If the theory is correct, the slope should be around -1 since the variance of the normal distribution is $\text{Var}(\bar{X}) = \frac{\sigma^2}{n}$, therefore, the variance is inversely proportional to the number of variables. The Figure 1.5 shows the variance as the number of variables increases and the regression line.

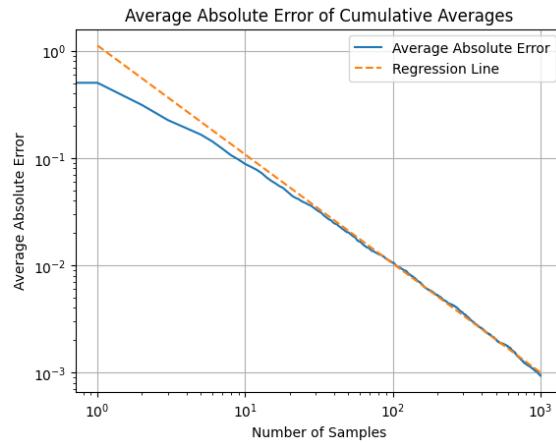


Figure 1.5: Logarithmic depiction of the absolute error relative to the mean. Average of 1000 trials per point.

The visual representation in Figure 1.5 substantiates that the error indeed diminishes as the number of trials augments, validating the assertion of the Law of Large Numbers that the experimental mean approaches the theoretical mean with an increasing number of observations. This is further corroborated by the linear regression, which yields a slope of -1.000 , in alignment with the theoretical prediction.

1.3.2 Empirical Validation of the Central Limit Theorem

The Central Limit Theorem (CLT) posits a pivotal foundational theorem in probability theory, signifying that, irrespective of the shape of the original distribution, the distribution of sample means will approximate a normal distribution as the sample size increases. To empirically validate this theorem, we can simulate a series of random numbers from any given probability distribution and systematically calculate their mean. By repeating this process, we can construct a histogram of the means and scrutinize whether it converges to a normal distribution, aligning with the theorem's prediction. Formally, the theorem can be expressed as follows: The Central Limit Theorem (CLT) posits a pivotal foundational theorem in probability theory, signifying that, irrespective of the shape of the original distribution, the distribution of sample means will approximate a normal distribution as the sample size burgeons. To empirically validate this theorem, we can simulate a series of random numbers from any given probability distribution and systematically calculate their mean. By perpetuating this process, we can construct a histogram of the means and scrutinize whether it converges to a normal distribution, aligning with the theorem's prediction. Formally, the theorem can be expressed as follows, extracted from [10]:

Theorem 1.3.2 (Central Limit Theorem). Let $(X_j)_{j \geq 1}$ be a sequence of independent and identically distributed random variables with $\mathbb{E}(X_1^2) < +\infty$ and $\sqrt{\text{Var}(X_1)} > 0$. We note $\bar{X}_n = \frac{1}{n} \sum_{j=1}^n X_j$ and $\sigma^2 = \text{Var}(X_1)$. Then for $n \rightarrow \infty$,

$$\sqrt{n} \frac{\bar{X}_n - \mathbb{E}(X_1)}{\sigma} \xrightarrow[n \rightarrow \infty]{} Y \sim \mathcal{N}(0, 1) \quad (1.7)$$

For illustrative purposes, consider the uniform distribution on $[0, 1]$, which yields a histogram as depicted in Figure 1.6.

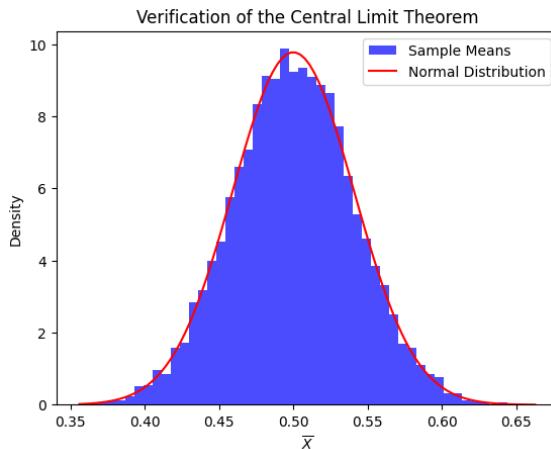


Figure 1.6: Histogram illustrating the convergence of the means of the uniform distribution with $n_{samples} = 10000$

In congruence with the methodology espoused in previous sections, we have juxtaposed the empirical mean of the samples with the theoretical mean, progressively augmenting the number of trials. Figure 1.7 delineates the deviation from the theoretical mean, and Figure 1.8 represents the deviation from the theoretical variance, both as functions of the number of trials. Therefore the formula of the deviation is:

$$error_{mean} = \frac{1}{n} \sum_{i=1}^n |\hat{x}_i - \mu|$$

where \hat{x}_i is the i -th random number generated and μ is the theoretical mean of the distribution.

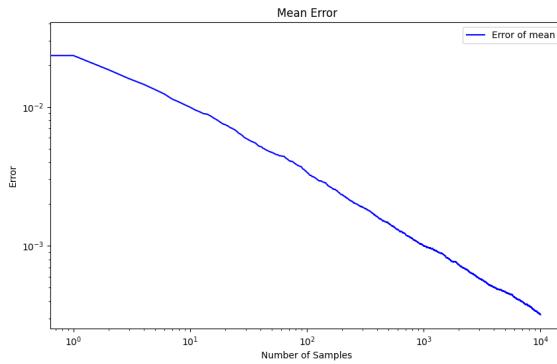


Figure 1.7: Error from the theoretical mean as a function of the number of trials. Average of 10000 trials per point.

And for the case of the variance:

$$error_{variance} = \frac{1}{n} \sum_{i=1}^n |\hat{x}_i - \sigma^2|$$

where \hat{x}_i is the i -th random number generated and σ^2 is the theoretical variance of the distribution.

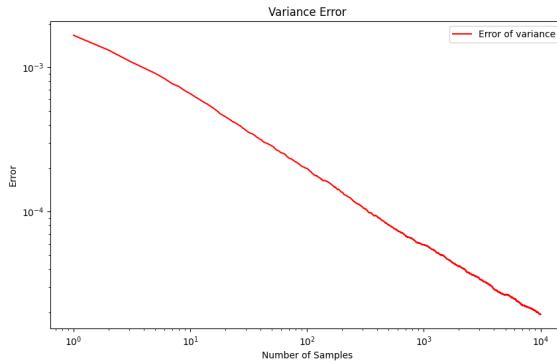


Figure 1.8: Error from the theoretical variance as a function of the number of trials. Average of 10000 trials per point.

A discernible insight gleaned from Figure 1.7 and Figure 1.8 is the palpable decrement in deviations from the theoretical values as the trials proliferate.

Finally, another way to validate the CLT is to plot the variance as the number of variables increases and calculate the slope of the regression line. If the theory is correct, the slope should be around -1 since the variance of the normal distribution is $\text{Var}(\bar{X}) = \frac{\sigma^2}{n}$, therefore the variance is inversely proportional to the number of variables. The Figure 1.9 shows the variance as the number of variables increases and the regression line.

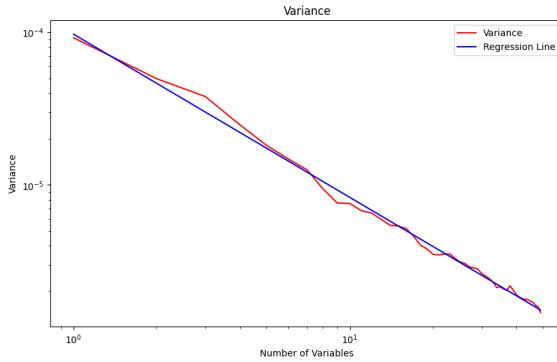


Figure 1.9: Deviation from the theoretical variance as a function of the number of trials. Average of 1000 trials per point.

The slope of the regression line is -1 in alignment with the theoretical prediction. This corroborates the validity of the Central Limit Theorem, as the variance of the means of the samples does indeed diminish as the number of trials increases.

1.4 Variance reduction techniques

In the theory of Monte Carlo methods, variance reduction techniques are a pivotal tool to increase the precision of the estimates of the expected value of a random variable. In this section, we will focus on three techniques: Control variates, Importance sampling and Antithetic variates. Also, we will introduce an example problem in which we will apply these techniques to compare them.

First of all, let introduce the variance reduction formally. Let X be a random variable drawn from a distribution of mean $\mathbb{E}(X) = \mu$ and finite variance $\text{Var}(X) = \sigma^2$. The estimator of the expected value of X , i.e. the empirical mean of the sample, is defined as follows:

$$\hat{\mathbb{E}}(X) = \frac{1}{n} \sum_{i=1}^n x_i \quad (1.8)$$

So we want to reduce the variance of our estimation. Since the variance is defined as follows:

$$\text{Var}(\hat{\mathbb{E}}(X)) = \frac{\text{Var}(X)}{n} \quad (1.9)$$

Consequently, we are presented with two avenues for optimization: increasing the value of n , or diminishing the variance of X . Assuming that n is predetermined and unalterable, our focus would then shift to minimizing the variance of X .

Since we are going to introduce and compare the three techniques, first of all we need to introduce the problem we are going to solve. The problem is that we want to estimate the following integral:

$$I = \int_0^1 x^2 dx \quad (1.10)$$

It should be noted that the selection of the integral for this demonstration was intentional; a readily solvable integral was chosen for its ease of analytical computation, allowing for a straightforward comparison with theoretical values. Nonetheless, the methods illustrated herein are equally applicable and potent for evaluating integrals that pose substantial challenges to analytical computation.

1.4.1 Preliminaries

First of all we have to compute which is the estimation we already can have without applying any variance reduction technique. We can compute the integral analytically. We know that the mean of the variable in a probability space is defined as follows:

$$\mathbb{E}(g(X)) = \int_a^b g(x)f(x) dx \quad (1.11)$$

Since $f(x) = \frac{1}{b-a}$ for the uniform distribution on $[a, b]$, we can apply Equation (1.11) to Equation (1.10) and we obtain:

$$I = \int_0^1 x^2 dx = \mathbb{E}(g(X)) = \int_0^1 x^2 \frac{1}{1-0} dx = \frac{1}{3} \quad (1.12)$$

So, we only have to generate a set of random numbers $X \sim \mathcal{U}(0, 1)$, apply $g(x) = x^2$ and get the mean of the sample which is the estimate of the integral.

1.4.2 Control variates

Control variates is a variance reduction technique with the following idea: Let μ the parameter we want to estimate, and assume we have a statistic Y such that $\mathbb{E}(Y) = \tau$. Then, we can estimate μ by estimating $\mathbb{E}(Y)$ as $\hat{\mathbb{E}}(Y)$ and correcting the bias with the following formula:

$$\hat{\mu} = \mu + c(\tau - \hat{\mathbb{E}}(Y)) \quad (1.13)$$

Being c a constant which minimize the variance of the estimation. It is computed as follows:

$$c = -\frac{\text{Cov}(\mu, \tau)}{\text{Var}(\tau)} \quad (1.14)$$

Proof. Using the Equation (1.13) we can compute the variance of the estimation as follows:

$$\begin{aligned} \text{Var}(\hat{\mu}) &= \\ &\mathbb{E}[(\mu + c(\tau - \hat{\mathbb{E}}(Y)))^2] - \mathbb{E}[\mu + c(\tau - \hat{\mathbb{E}}(Y))]^2 = \\ &\mathbb{E}[\mu^2 + c^2(\tau - \hat{\mathbb{E}}(Y))^2 + 2c\sigma(\tau - \hat{\mathbb{E}}(Y))] - \mathbb{E}[\mu]^2 - c^2\mathbb{E}[\tau - \hat{\mathbb{E}}(Y)]^2 - 2c\mathbb{E}[\sigma(\tau - \hat{\mathbb{E}}(Y))] = \\ &\text{Var}(\mu) + c^2\text{Var}(\tau) + 2c\text{Cov}(\mu, \tau) \end{aligned}$$

Therefore, since we want to minimize the variance of the estimation, we can differentiate with respect to c and equate to zero:

$$\frac{\partial \text{Var}(\hat{\mu})}{\partial c} = 2c\text{Var}(\tau) + 2\text{Cov}(\mu, \tau) = 0$$

And we obtain the following expression for c :

$$c = -\frac{\text{Cov}(\mu, \tau)}{\text{Var}(\tau)}$$

Since we want to minimize the variance, we can differentiate again with respect to c and we obtain:

$$\frac{\partial^2 \text{Var}(\hat{\mu})}{\partial c^2} = 2\text{Var}(\tau) > 0$$

Therefore, we can conclude that the variance is minimized when $c = -\frac{\text{Cov}(\mu, \tau)}{\text{Var}(\tau)}$. \square

In our case, we can use the knowledge of the mean of the uniform distribution to correct the bias, so in each iteration we can compute the mean of the sample of random numbers of $\mathcal{U}(0, 1)$, and knowing that the mean of the uniform distribution is 0.5 we can apply Equation (1.13) to obtain the estimate of the integral.

1.4.3 Importance sampling

Importance sampling is another variance reduction technique. Instead of using the knowledge of another estimator to reduce the bias of our sample, as done in the control variates method, in this case the idea is using the knowledge of the actual function we want to integrate, so instead of using a uniform distribution, we can use another distribution that is more similar to the function we want to integrate, in order to try more samples in the areas where the function is more important.

In our example, we know that the function is a parabola, so instead of using a sample which follows an uniform distribution, maybe we can use a distribution that fits better with the shape of the function. In this case, we have chosen $Beta(2.9, 1)$. In order to illustrate this, in the Figure 1.10 we can see the function we want to integrate, the PDF of $Beta(2.9, 1)$ and the PDF of $\mathcal{U}(0, 1)$.

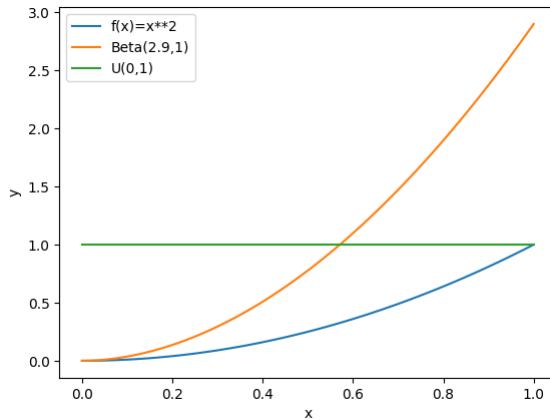


Figure 1.10: Function we want to integrate, PDF of $Beta(2.9, 1)$ and PDF of $\mathcal{U}(0, 1)$

As we can see, the PDF of $Beta(2.9, 1)$ fits better with the shape of the function we want to integrate, so we can expect that the error will decrease faster than the direct method and the control variates method.

So the idea is to generate a sample of random numbers $X \sim Beta(2.9, 1)$, and take $Y = \frac{f(X)}{g(X)}$ as the estimator of the integral, being $f(x) = x^2$ and $g(x)$ the PDF of $Beta(2.9, 1)$. The algorithm is as follows:

Algorithm 3: Importance Sampling method

1. Generate a set of random numbers $X \sim Beta(2.9, 1)$;
 2. Compute $Y = \frac{f(X)}{g(X)}$;
 3. Compute the mean of Y as the estimator of the integral ;
-

Now let's show why this method works.

Proof. The standard Monte Carlo estimator for $\mathbb{E}[g(X)]$ is:

$$\hat{\mathbb{E}}[g(X)] = \frac{1}{n} \sum_{i=1}^n g(X_i)$$

where X_i are i.i.d. samples from the distribution of X .

In Importance Sampling, we draw samples Y_i from a different distribution and compute the estimator:

$$\hat{\mathbb{E}}_{IS}[g(X)] = \frac{1}{n} \sum_{i=1}^n \frac{g(Y_i)f_X(Y_i)}{f_Y(Y_i)}$$

where f_X and f_Y are the probability density functions of X and Y , respectively.

The variance of the Importance Sampling estimator is:

$$\text{Var}(\hat{\mathbb{E}}_{IS}[g(X)]) = \frac{1}{n} \left(\mathbb{E} \left[\left(\frac{g(Y)f_X(Y)}{f_Y(Y)} \right)^2 \right] - \mathbb{E}^2[g(X)] \right)$$

The variance reduction comes from the fact that if $f_Y(y)$ is chosen such that it is closer to $|g(y)|$, the term $\frac{f_X(Y)}{f_Y(Y)}$ will stabilize the values of $g(Y)$, leading to a reduction in variance.

Therefore, if the sampling distribution Y is chosen appropriately, $\text{Var}(\hat{\mathbb{E}}_{IS}[g(X)])$ will be less than $\text{Var}(\hat{\mathbb{E}}[g(X)])$, achieving variance reduction. \square

1.4.4 Stratified sampling

The idea is to divide the interval of the stratified sampling is pretty simple. We divide the interval in n subintervals, and we generate a sample of random numbers for each subinterval. Doing this, we are trying to reduce the variance of the mean of the distribution we are considering, so the sample is more representative of the distribution. The alrogithm is as follows:

Algorithm 4: Stratified Sampling method

1. Divide the interval in n subintervals ;
 2. Generate a set of random numbers $X \sim \mathcal{U}(a_i, b_i)$ for each subinterval ;
 3. Compute the mean of X as the estimator of the integral ;
-

Now let's show why this method works.

Proof. Consider a population divided into k non-overlapping strata, and let a random variable X be defined over this population. We are interested in estimating the mean $\mathbb{E}[X]$.

For each stratum i , let X_i represent the values of X within that stratum. The mean and variance of X_i are $\mathbb{E}[X_i]$ and $\text{Var}(X_i)$, respectively.

In stratified sampling, we sample n_i observations from each stratum i . The estimator for the mean of X is a weighted average of the sample means of each stratum:

$$\hat{\mathbb{E}}[X] = \sum_{i=1}^k w_i \bar{X}_i$$

where $w_i = \frac{N_i}{N}$ is the weight of the i -th stratum (based on its proportion of the total population), N_i is the size of stratum i , N is the total population size, and \bar{X}_i is the sample mean for stratum i .

The variance of $\hat{\mathbb{E}}[X]$ is derived by considering the variance contribution of each stratum's estimator. Since the samples from different strata are independent, the variances add:

$$\text{Var}(\hat{\mathbb{E}}[X]) = \sum_{i=1}^k \text{Var}(w_i \bar{X}_i)$$

The variance of the sample mean \bar{X}_i for each stratum is $\frac{\text{Var}(X_i)}{n_i}$. Therefore, the variance of the weighted mean for each stratum is:

$$\text{Var}(w_i \bar{X}_i) = w_i^2 \frac{\text{Var}(X_i)}{n_i}$$

Combining these, we get the total variance:

$$\text{Var}_{\text{stratified}}(\hat{X}) = \sum_{i=1}^k w_i^2 \frac{\text{Var}(X_i)}{n_i}$$

Note that the weight w_i is the size of the stratum relative to the total population. Therefore, the variance of the stratified estimator is:

$$\text{Var}_{\text{stratified}}(\hat{X}) = \sum_{i=1}^k \left(\frac{N_i}{N} \right)^2 \frac{\text{Var}(X_i)}{n_i}$$

where N_i is the size of stratum i and N is the total population size.

The variance under simple random sampling (assuming a sample size of n) is:

$$\text{Var}_{\text{simple}}(\hat{X}) = \frac{\text{Var}(X)}{n}$$

Stratified sampling reduces the variance compared to simple random sampling if:

$$\sum_{i=1}^k \left(\frac{N_i}{N} \right)^2 \frac{\text{Var}(X_i)}{n_i} < \frac{\text{Var}(X)}{n}$$

This inequality holds true especially when the variance within each stratum $\text{Var}(X_i)$ is significantly lower than the overall variance $\text{Var}(X)$ due to the homogeneity within each stratum.

To further optimize, the number of samples n_i can be allocated to each stratum using proportional or optimal allocation strategies, often based on minimizing the variance of the estimator.

Therefore, by judiciously partitioning the domain into strata and allocating samples, stratified sampling significantly reduces the variance of the estimator compared to simple random sampling, particularly when variances within strata are substantially lower than the overall variance. \square

1.4.5 Antithetic variates

The Antithetic variates method consists on taking for each random sample, its antithetic, i.e. the symmetric with respect to the mean of the distribution. The idea is that the variance of the mean of the distribution is reduced, since the mean of the antithetic is the same as the mean of the distribution. In our case, we have generated a sample $\mathcal{U} \sim \mathcal{U}(0, 0.5)$ and its antithetic $\mathcal{U}' = \{1 - x : x \in \mathcal{U}\}$. We can apply Equation (1.8) to $f(X)$ to obtain the estimate of the integral. The algorithm is as follows:

Algorithm 5: Antithetic Variates method

1. Generate a set of random numbers $X \sim \mathcal{U}(0, 0.5)$;
 2. Compute the antithetic of X ;
 3. Compute the mean of $X \cup X'$ as the estimator of the integral ;
-

Now, let's show why this method works.

Proof. Let X be a random variable from which we are sampling to estimate $\mathbb{E}[X]$, and consider the use of Antithetic Variates to reduce the variance of this estimate.

For each sample X_i drawn from the distribution of X , an antithetic variate X'_i is constructed such that X_i and X'_i are negatively correlated. This is typically done by setting $X'_i = F^{-1}(1 - F(X_i))$ where F is the cumulative distribution function of X .

The estimator with Antithetic Variates is given by:

$$\hat{\mathbb{E}}_{AV}[X] = \frac{1}{2n} \sum_{i=1}^n (X_i + X'_i)$$

where n is the number of sample pairs.

To derive the variance of this estimator, consider:

$$\text{Var}(\hat{\mathbb{E}}_{AV}[X]) = \frac{1}{4n^2} \sum_{i=1}^n \text{Var}(X_i + X'_i)$$

Since X_i and X'_i are negatively correlated, we expand the variance term:

$$\text{Var}(X_i + X'_i) = \text{Var}(X_i) + \text{Var}(X'_i) + 2\text{Cov}(X_i, X'_i)$$

The covariance term $\text{Cov}(X_i, X'_i)$ is negative due to the negative correlation between X_i and X'_i . This reduces the variance of their sum.

Assuming $\text{Var}(X_i)$ and $\text{Var}(X'_i)$ are equal and represented by σ^2 , and letting the covariance be represented by $\text{Cov}(X_i, X'_i) = -\rho\sigma^2$ where $\rho > 0$, the variance becomes:

$$\text{Var}(X_i + X'_i) = 2\sigma^2 - 2\rho\sigma^2 = 2\sigma^2(1 - \rho)$$

Thus, the variance of the estimator is:

$$\text{Var}(\hat{\mathbb{E}}_{AV}[X]) = \frac{1}{4n^2} \sum_{i=1}^n 2\sigma^2(1 - \rho) = \frac{\sigma^2(1 - \rho)}{2n}$$

The variance under simple random sampling for n samples is $\text{Var}_{\text{simple}}(\hat{X}) = \frac{\sigma^2}{n}$.

Antithetic variates reduce variance compared to simple random sampling if:

$$\frac{\sigma^2(1 - \rho)}{2n} < \frac{\sigma^2}{n}$$

which simplifies to $1 - \rho < 2$, a condition satisfied since $\rho > 0$.

Therefore, the use of antithetic variates, through the introduction of negative covariance, effectively reduces the variance of the estimator compared to simple random sampling. This reduction is particularly significant when the negative correlation between the pairs of samples is strong. \square

1.4.6 Comparative Review

Having looked at the methods and confirmed the convergence of each approach described in this study, we now turn to a more focused comparison of these techniques. The goal is simple: find out which method works best for our specific problem. For this aims, we will be using the following formula to compute the error:

$$\text{error} = |\bar{\mathbb{E}}(X) - \frac{1}{3}|$$

See Figure 1.11 for a visual representation of the error in relation to the theoretical value as we increase the number of trials for each method. Note that for this comparision we have used 10 stratas for the stratified sampling method.

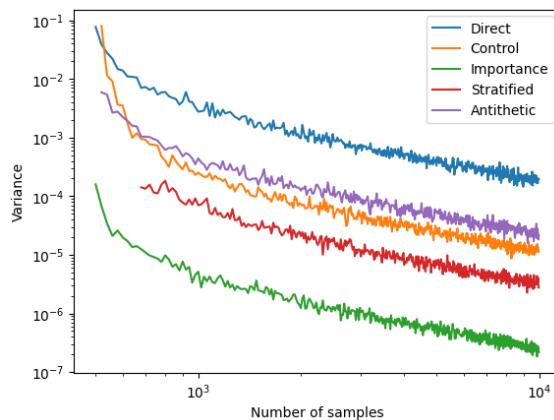


Figure 1.11: Error variation with increased trials for each method.

Looking at the data, it's clear that importance sampling takes the lead compared to the other methods. However, claiming it as the undisputed champion would be premature, especially when we have explored just one problem. To firm up this initial finding, we need to dig deeper and explore a variety of problems.

But before moving on to look at more problems, let's look at the comparative table of variances for each method:

Method	Variance
Direct sampling	$1.95839 \cdot 10^{-4}$
Control variates sampling	$1.12095 \cdot 10^{-5}$
Importance sampling	$2.17864 \cdot 10^{-7}$
Stratified sampling	$2.76024 \cdot 10^{-6}$
Antithetic sampling	$2.14899 \cdot 10^{-5}$

Recall that the formula used to compute the variance is:

$$\text{Var}(X) = \text{Var}(\hat{\mathbb{E}}(X)) n$$

With a sample size of $n = 1000$.

Upon examining the table, notable findings emerge. Initially, it's evident that direct sampling ranks as the least effective method, aligning with our theoretical insights. This approach primarily aims to reduce the variance in our estimates. In contrast, importance sampling emerges as the superior technique. However, it demands greater effort, as it involves identifying a more compatible distribution for the function we aim to integrate. Stratified sampling comes in as the next best option, showing potential for improved outcomes with an increased number of strata. Nevertheless, drawing definitive conclusions from a single case study is premature; further experimentation across various problems is necessary to determine the optimal method for each situation.

Next, we'll broaden our investigation to a more complex problem, applying the methods we have discussed to estimate the following integrals:

$$\text{Parabola} = \int_0^1 x^2 dx \quad (1.15)$$

$$\text{Gaussian} = \int_0^1 e^{-x^2} dx \quad (1.16)$$

$$\text{Sine} = \int_0^1 \sin(x) dx \quad (1.17)$$

$$\text{Polynomial} = \int_0^1 x^3 - 2x^2 + x dx \quad (1.18)$$

$$\text{Exponential} = \int_0^1 e^x dx \quad (1.19)$$

Using these examples, we can now draw comparisons between the methods in a broader context. Figure 1.12 illustrates the error of each method applied to each function.

This error is defined as:

$$\text{error} = \left| \text{method}(f, a, b, \text{iters}) - \int_a^b f dx \right|$$

where *method* is the method we are using, *f* is the function we are integrating, *a* and *b* are the limits of the integral and *iters* is the number of iterations we are using, in this case *iters* = 1000.

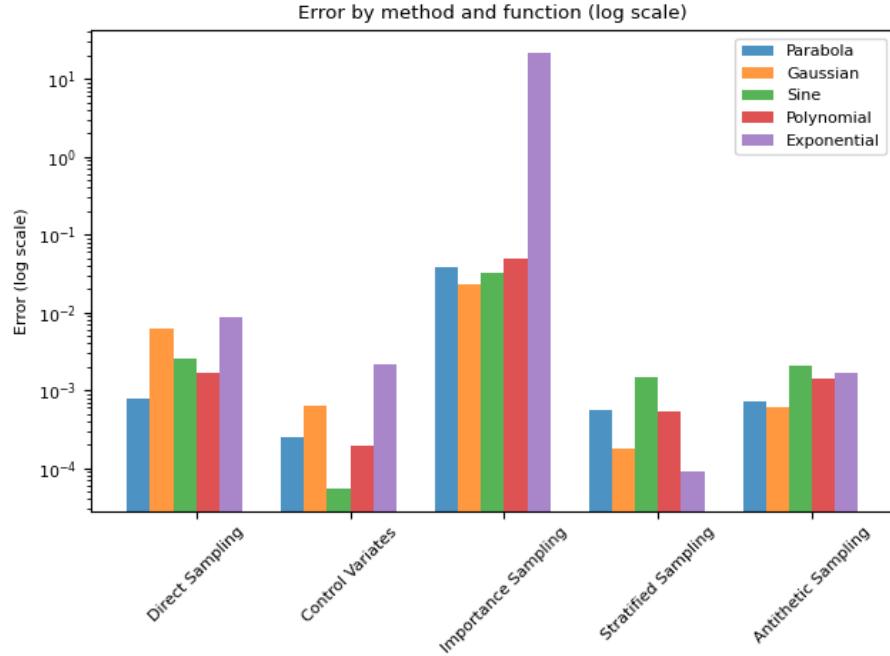


Figure 1.12: Method-wise error in each function.

Note that in the case of importance sampling, we have used the gaussian distribution as the sampling distribution, therefore the error could be higher than the other methods, since the gaussian distribution might not be the best sampling distribution for all the functions. To conclude, we can state that it depends on the problem which method is the best.

Chapter 2

Hamiltonian dynamics

This chapter delves into the realm of Hamiltonian dynamics, beginning with an introduction to hamiltonian dynamics in Section 2.1, followed by an in-depth exploration of symplectic schemes in Section 2.2. This section starts with a discussion on the analytical solution in Section 2.2.1, setting the groundwork for understanding symplectic integration methods.

In Section 2.2.2, the focus shifts to the Euler method, detailing its application and significance in Hamiltonian systems. Following this, Section 2.2.3 discusses the limitations and failures of standard methods in this context, highlighting the need for more sophisticated approaches.

Section 2.2.4 addresses the construction of symplectic schemes, a crucial step for accurate simulations in Hamiltonian dynamics. This is further expanded in Section 2.2.5, which specifically examines the Symplectic Euler method. A detailed Linear Stability Analysis of the Symplectic Euler scheme is then presented in Section 2.2.5.1.

The narrative continues with the Stormer-Verlet method in Section 2.2.6, another important technique in symplectic integration, followed by a Linear Stability Analysis of the Verlet scheme in Section 2.2.6.1. Section ?? concludes this part with a discussion on Backward Error Analysis, providing insights into the error characteristics of symplectic integrators.

Section 2.3 shifts the focus to the practical application of these concepts in simulating the solar system. Section 2.3.1 begins with modeling gravitational dynamics, followed by the implementation of the Störmer-Verlet scheme in Section 2.3.2. The implementation details of these models are further elaborated in Section 2.3.3, with two experiments: Experiment 1 in Section 2.3.3.1 expands to the N-Body problem, and Experiment 2 in Section 2.3.3.2 utilizes the Symplectic Euler Method, providing practical insights into their application in complex dynamical systems.

2.1 Introduction to Hamiltonian dynamics

Hamiltonian dynamics, also called "Hamiltonian mechanics", is a reformulation of the clasical mechanics which describes the temporal evolution of a physical system in terms of pairs of variables: the generalized coordinates q_i and their momenta p_i .

$$p(t) = \begin{bmatrix} p_1(t) \\ p_2(t) \\ \vdots \\ p_n(t) \end{bmatrix} \quad q(t) = \begin{bmatrix} q_1(t) \\ q_2(t) \\ \vdots \\ q_n(t) \end{bmatrix}$$

In this section we consider the time evolution of an isolated system described at a microscopic level, i.e. a system of particles. The state of the system is described by the position of the particles q_i and their momenta p_i . We denote D the dimension of the positions and momenta variables. Therefore $D = 3N$ when the system is composed of N particles in a 3-dimensional physical space. Also, we assume that for the system there is a function $H(q, p, t)$ which describes the energy of the system.

The Hamiltonian dynamics is defined by the following equations:

$$\begin{cases} \frac{dq(t)}{dt} = \nabla_p H(q(t), p(t)) \\ \frac{dp(t)}{dt} = -\nabla_q H(q(t), p(t)) \end{cases} \quad (2.1)$$

With initial condition $p(0) = p^0$, $q(0) = q^0$ that should be provided. Now, introducing the following matrix:

$$J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$$

And denoting $y = (q, p)$, we can rewrite the Hamiltonian dynamics equations as follows:

$$\frac{dy}{dt} = J \nabla H(y) = J \begin{bmatrix} \nabla_q H(y) \\ \nabla_p H(y) \end{bmatrix} \quad (2.2)$$

A Hamiltonian function is the sum of the kinetic energy and the potential energy of the system:

$$H(q, p) = K(p) + V(q)$$

A very common physical interpretation of the Hamiltonian function is the following:

$$H(q, p) = V(q) + \frac{1}{2} p^T M^{-1} p$$

Where M is the mass matrix of the system, note that we are supposing that the mass is stable over time. In this case, we can reformulate the Hamiltonian dynamics equations as follows:

$$\begin{cases} \frac{dq(t)}{dt} = M^{-1} p(t) \\ \frac{dp(t)}{dt} = -\nabla_q V(q(t)) \end{cases} \quad (2.3)$$

Therefore, if we consider this equations in terms of positions we get:

$$M \frac{d^2 q(t)}{dt^2} = -\nabla_q V(q(t))$$

Which is the Newton's second law of motion. So, we can see that the Hamiltonian dynamics is a generalization of the Newton's second law of motion.

One of the most important property of the Hamiltonian dynamics is the following:

Theorem 2.1.1 (Conservation of energy). Let $H(q, p)$ be the Hamiltonian function of a system. Then, the energy of the system is conserved over time, i.e. $H(q(t), p(t)) = H(q^0, p^0)$ for all t .

Proof. Deriving the Hamiltonian function with respect to time we get:

$$\begin{aligned}\frac{dH}{dt} &= \frac{\partial H}{\partial q} \frac{dq}{dt} + \frac{\partial H}{\partial p} \frac{dp}{dt} = \\ &= \nabla_q H \frac{dq}{dt} + \nabla_p H \frac{dp}{dt} = \\ &= \nabla_q H \nabla_p H - \nabla_p H \nabla_q H = 0\end{aligned}$$

Since the hamiltonian H is defined as the total energy of the system, we can conclude that the energy of the system is conserved over time. \square

2.2 Symplectic schemes

In this section we will introduce the symplectic schemes, which are a family of numerical methods to solve the Hamiltonian dynamics equations. The idea is to discretize the Hamiltonian dynamics equations (2.2) in order to obtain a numerical approximation of the solution. The symplectic schemes are a family of numerical methods which preserve the energy conservation property of the Hamiltonian dynamics equations, i.e. the energy of the system is conserved over time.

Definition 2.2.1. For an open set $U \subset \chi$, a mapping $g : U \rightarrow \mathbb{R}^{2D}$ of class C^1 is symplectic if $\nabla g(q, p)$ satisfies

$$(\nabla g)^T J \nabla g = J, \quad \forall (q, p) \in U$$

2.2.1 Analytical solution

Let us introduce a simple example problem to illustrate the symplectic schemes: the harmonic oscillator. The harmonic oscillator is a system composed of a particle of mass m attached to a spring with spring constant k . The position of the particle is denoted by $q(t)$. The potential energy of the system is defined as follows:

$$V(q) = \frac{1}{2} k q^2$$

Therefore, the Hamiltonian function of the system is defined as follows:

$$H(q, p) = \frac{1}{2m} p^2 + \frac{1}{2} k q^2 \tag{2.4}$$

Applying this Hamiltonian function to the Hamiltonian dynamics equations (2.2) we obtain the following equations:

$$\begin{cases} \frac{dq(t)}{dt} = \frac{1}{m} p(t) \\ \frac{dp(t)}{dt} = -k q(t) \end{cases}$$

Let's proceed step by step to solve the given system of differential equations for the harmonic oscillator. We are given the Hamiltonian:

$$H(q, p) = \frac{1}{2m}p^2 + \frac{1}{2}kq^2$$

And the Hamilton's equations:

$$\begin{cases} \frac{dq(t)}{dt} = \frac{1}{m}p(t) \\ \frac{dp(t)}{dt} = -kq(t) \end{cases}$$

We can combine the two first-order differential equations into a single second-order differential equation by substituting the expression for $\dot{x}(t)$ into the derivative $\dot{p}(t)$. Substituting $\dot{x}(t) = \frac{1}{m}p(t)$ into the derivative of the second equation gives us:

$$\frac{d^2q(t)}{dt^2} = -\frac{k}{m}q(t)$$

This is a second-order homogeneous linear differential equation.

To solve this second-order differential equation, we can use the characteristic equation method. Assume a solution of the form:

$$q(t) = e^{rt}$$

where r is a constant to be determined. Substituting this into the second-order equation gives:

$$r^2e^{rt} + \frac{k}{m}e^{rt} = 0$$

Since e^{rt} is never zero, we can divide through by it to get the characteristic equation:

$$r^2 + \frac{k}{m} = 0$$

Solving for r gives us:

$$r = \pm i\sqrt{\frac{k}{m}}$$

where i is the imaginary unit.

Given that the roots are complex, the general solution of the equation is:

$$\begin{cases} q(t) = A \cos(\omega t) + B \sin(\omega t) \\ p(t) = -m\omega A \sin(\omega t) + m\omega B \cos(\omega t) \end{cases}$$

where A and B are arbitrary constants determined by initial conditions and $\omega = \sqrt{\frac{k}{m}}$ is the angular frequency of the oscillator. In order to determine the constants A and B , we need initial conditions. Specifically, we need the initial position $x(0)$ and initial velocity $\dot{x}(0)$. Let's say, for example:

$$q(0) = q_0 \quad \text{and} \quad \dot{q}(0) = v_0$$

Substituting these into the general solution and its derivative gives:

$$q_0 = A \quad \text{and} \quad v_0 = B\omega$$

Thus, if initial conditions are provided, A and B can be determined to give the particular solution for the system. For example, if we take $x_0 = 1$ and $v_0 = 0$, we get that $A = 1$ and $B = 0$. Therefore, the solution of the harmonic oscillator is:

$$\begin{cases} q(t) = \cos(\omega t) = \cos(\sqrt{\frac{k}{m}}t) \\ p(t) = -m\omega \sin(\omega t) = -\sqrt{km} \sin(\sqrt{\frac{k}{m}}t) \end{cases}$$

The Figure 2.1 shows the phase space of the harmonic oscillator with $k = 1$ and $m = 1$.

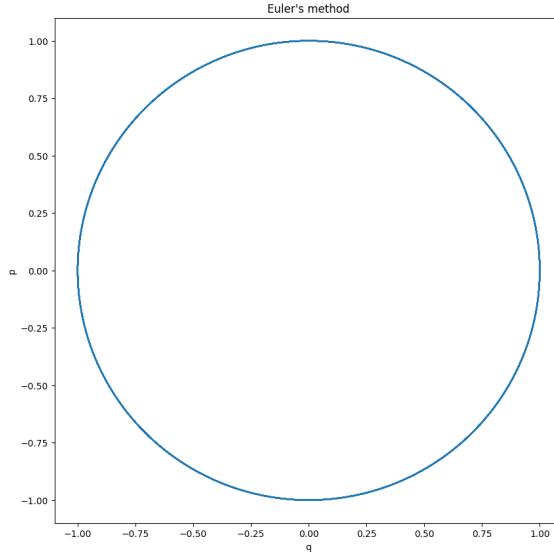


Figure 2.1: Phase space of the harmonic oscillator with $k = 1$ and $m = 1$ and initial conditions $q_0 = 1$ and $v_0 = 0$.

2.2.2 Euler method

The Euler method is a numerical method to solve ordinary differential equations. The idea is to discretize the differential equation in order to obtain a numerical approximation of the solution. The Euler method discretizes the differential equation as follows:

$$\begin{cases} p^{n+1} = p^n - \Delta t \frac{\partial H}{\partial q}(q^n, p^n) = p^n - \Delta t \nabla_q V(q^n) \\ q^{n+1} = q^n + \Delta t \frac{\partial H}{\partial p}(q^n, p^n) = q^n + \Delta t M^{-1} p^n \end{cases} \quad (2.5)$$

Being Δt the time step of the discretization and $p^n \approx p_{n\Delta t}$. Applying this sequence to the harmonic oscillator, which equations are described in (2.4), we obtain the following phase space:

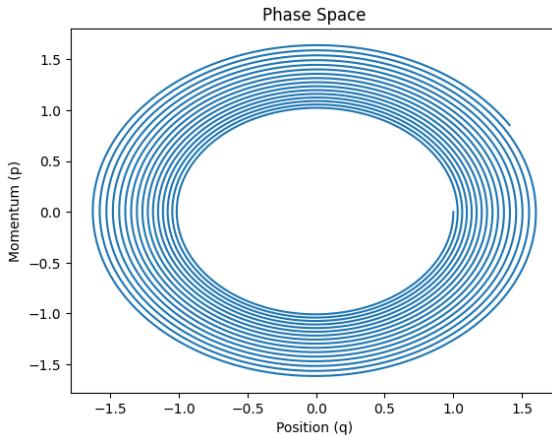


Figure 2.2: Phase space of the harmonic oscillator using the Euler method with $\Delta t = 0.01$.

As we can discern, this trajectory is not even near to the real trajectory of the harmonic oscillator in ideal conditions. The Figure 2.2 shows the phase space of the harmonic oscillator using the Euler method with $\Delta t = 0.1$. We can see that the phase space is not a closed curve, which is the correct behaviour of the harmonic oscillator. Also, the energy is not conserved over time, which is another property of the harmonic oscillator. The Figure 2.3 shows the energy of the harmonic oscillator using the Euler method with $\Delta t = 0.1$.

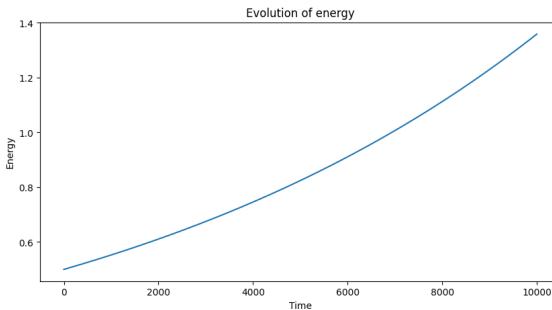


Figure 2.3: Energy of the harmonic oscillator using the Euler method with $\Delta t = 0.01$.

As we can see, the energy is not conserved over time, which is another property of the harmonic oscillator. The energy is increasing over time, which is not the correct behaviour of the harmonic oscillator. With this results we can conclude that the Euler method is not a good method to solve the Hamiltonian dynamics equations. This is a consequence of the fact that the Euler method does not preserve the symplectic structure of the Hamiltonian dynamics equations. Therefore, in the next section we will introduce a new method that preserves the symplectic structure of the Hamiltonian dynamics equations which is a slight modification of the Euler method.

2.2.3 Failure of standard methods

As we already know, the Hamiltonian dynamics is a standard ordinary differential equation (ODE), so it can be approximated by any standard integration scheme. However, as we saw using the Euler scheme, the energy increases over the time so it doesn't work. To provide a further view of

this, in this section, we want to introduce the mathematical approach to see why this methods are failing. Considering again the problem of the harmonic oscillator, whose Hamiltonian is provided in 2.4. And taking the euler scheme given in 2.5 we can rewrite it to:

$$\begin{pmatrix} p^{n+1} \\ q^{n+1} \end{pmatrix} = A^{n+1} \begin{pmatrix} p^0 \\ q^0 \end{pmatrix}$$

Being $A = \begin{pmatrix} 1 & -k\Delta t \\ \frac{1}{m}\Delta t & 1 \end{pmatrix}$. Now we diagonalize this matrix, so we obtain that its eigenvalues are:

$$\lambda_1 = 1 + i\Delta t \sqrt{\frac{k}{m}} = i + i\Delta t\omega, \quad \lambda_2 = 1 - i\Delta t \sqrt{\frac{k}{m}} = i - i\Delta t\omega$$

Since $\lambda_1\lambda_2 = 1 + (\Delta t\omega)^2 > 1$, we observe that the eigenvalues' magnitudes are greater than one. This indicates that the energy of the system, represented by the norm of the vector (p, q) , will grow exponentially over time. To see this, consider that the energy at the n -th step is proportional to the square of the norm of the vector (p^n, q^n) . Since each step of the Euler method multiplies the previous vector by A , and the eigenvalues of A have magnitudes greater than one, the norm of this vector, and hence the energy, will increase exponentially with each step. This exponential growth is a clear indication of the method's failure to conserve energy, a fundamental characteristic of the harmonic oscillator in a closed system. Therefore, the Euler method, along with other similar standard integration schemes, is unsuitable for simulating systems where energy conservation is crucial, necessitating the development of more sophisticated methods that can maintain the energy constant over time.

2.2.4 Constructing Symplectic Schemes

To develop an integrator, we split the Hamiltonian $H(q, p)$ into simpler components:

$$H(q, p) = H_1(q, p) + H_2(q, p)$$

With:

$$\begin{aligned} H_1(q, p) &= \frac{1}{2}p^T M^{-1}p \quad (\text{kinetic energy}) \\ H_2(q, p) &= V(q) \quad (\text{potential energy}) \end{aligned}$$

After the Hamiltonian is decomposed, the associated dynamics become:

For $H1$:

$$\begin{aligned} \dot{q} &= M^{-1}p \\ \dot{p} &= 0 \end{aligned}$$

For $H2$:

$$\begin{aligned} \dot{q} &= 0 \\ \dot{p} &= -\nabla V(q) \end{aligned}$$

Now, let's talk about flows. Think of a flow as a way to evolve a point in phase space over time based on our differential equations.

For the dynamics associated with H_1 , we can integrate with respect to time, t :

$$\int \dot{q} dt = \int M^{-1} p dt$$

This gives:

$$q(t) = q(0) + tM^{-1}p$$

Similarly, for momentum, $p(t) = p(0)$ because $\dot{p} = 0$.

Thus, the flow for H_1 , denoted as ϕ_1^t , evolves as:

$$\phi_1^t(q, p) = (q + tM^{-1}p, p)$$

For H_2 , given $\dot{q} = 0$, $q(t) = q(0)$. But for momentum, we get:

$$\int \dot{p} dt = \int -\nabla V(q) dt$$

$$p(t) = p(0) - t\nabla V(q)$$

Thus, the flow for H_2 , ϕ_2^t , is:

$$\phi_2^t(q, p) = (q, p - t\nabla V(q))$$

Combining these flows yields our symplectic schemes. The sequence of combining matters.

1. Kinetic (ϕ_1) followed by Potential (ϕ_2):

$$\begin{cases} q_{n+1} &= q_n + \Delta t M^{-1} p_n \\ p_{n+1} &= p_n - \Delta t \nabla V(q_{n+1}) \end{cases}$$

2. Potential (ϕ_2) followed by Kinetic (ϕ_1):

$$\begin{cases} p_{n+1} &= p_n - \Delta t \nabla V(q_n) \\ q_{n+1} &= q_n + \Delta t M^{-1} p_{n+1} \end{cases}$$

Theorem 2.2.1 (Symplecity of the Hamiltonian flow). Let $H(q, p)$ be a $C^2(U)$ function, where U is an open set of \mathbb{R}^{2D} . Then, for any fixed $t \in \mathbb{R}$ such that the flow ϕ^t is defined, the mapping ϕ^t is symplectic.

Proof. Proof in [14, Chapter 2.1.2]. □

2.2.5 Symplectic Euler method

The symplectic Euler method is a slight modification of the Euler method that preserves the symplectic structure of Hamiltonian systems. This new method discretizes the Hamiltonian dynamics equations (2.2) as follows:

$$\begin{cases} p^{n+1} = p^n - \Delta t \frac{\partial H}{\partial q}(q^n, p^n) = p^n - \Delta t \nabla_q V(q^n) \\ q^{n+1} = q^n + \Delta t \frac{\partial H}{\partial p}(q^{n+1}, p^{n+1}) = q^n + \Delta t M^{-1} p^{n+1} \end{cases} \quad (2.6)$$

Being Δt the time step of the discretization. Applying this sequence to the harmonic oscillator, which equations are described in (2.4), we obtain the following phase space:

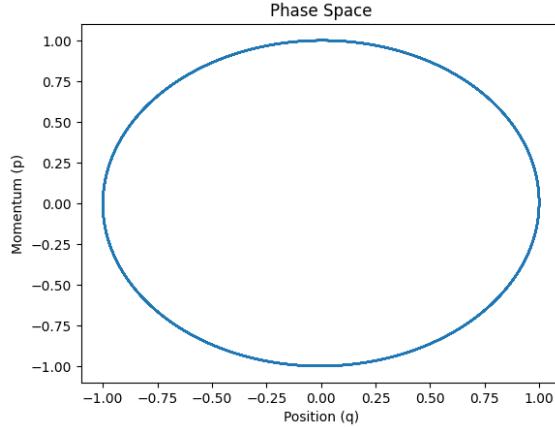


Figure 2.4: Phase space of the harmonic oscillator using the symplectic Euler method with $\Delta t = 0.01$.

The Figure 2.4 shows the phase space of the harmonic oscillator using the symplectic Euler method. We can see that the trajectory in the phase space is a closed curve, which is the correct behaviour of the harmonic oscillator.

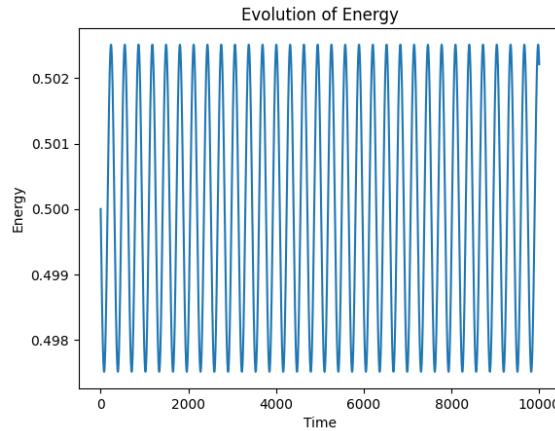


Figure 2.5: Energy of the harmonic oscillator using the symplectic Euler method with $\Delta t = 0.01$.

The Figure 2.5 shows the energy of the harmonic oscillator using the symplectic Euler method. We can see that the energy is approximately conserved over time, as we wanted for the Hamiltonian dynamics. Note that is not a straight line but the average of the energy is the same, so we can conclude that the energy is conserved over time.

Now, if we define the next map:

$$\Gamma_{\Delta t}^{Euler} = \max_{n=1\dots N} \{|H(p_{\Delta t}^n, q_{\Delta t}^n) - H(p_{\Delta t}^0, q_{\Delta t}^0)|\}$$

The Figure 2.6 shows the function $\Gamma_{\Delta t}^{Euler}$ over $\frac{1}{\Delta t}$.

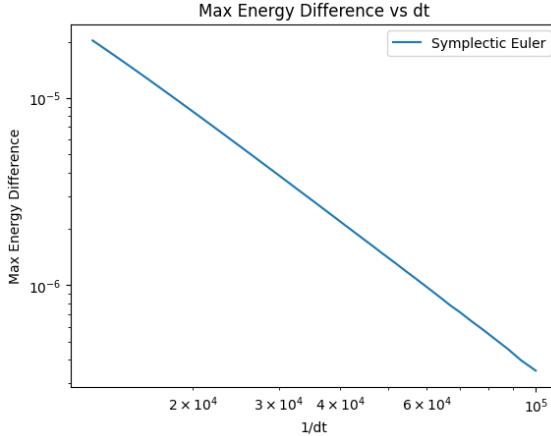


Figure 2.6: Function $\Gamma_{\Delta t}^{Euler}$ over $\frac{1}{\Delta t}$

To conclude the section, we can see in Figure 2.6 that the difference decreases as the time step decreases, which is the expected behaviour of any scheme that preserves the symplectic structure of the Hamiltonian dynamics equations. Therefore, we can conclude that the symplectic Euler method is a good method to solve the Hamiltonian dynamics equations.

2.2.5.1 Linear Stability Analysis of the Symplectic Euler Scheme

Using the equation 2.6 and replacing all the termusing the harmonic oscillator problem, one can rewrite the scheme as:

$$\begin{pmatrix} p^{n+1} \\ q^{n+1} \end{pmatrix} = A^{n+1} \begin{pmatrix} p^0 \\ q^0 \end{pmatrix}$$

With

$$A = \begin{pmatrix} 1 & -\Delta t k \\ \frac{\Delta t}{m} & 1 - \frac{\Delta t^2 k}{m} \end{pmatrix}$$

The eigenvalues are:

$$\begin{aligned} \lambda_1 &= \frac{-\Delta t^2 k - \Delta t \sqrt{k(\Delta t^2 k - 4m)}}{2m} + 1 \\ \lambda_2 &= \frac{-\Delta t^2 k + \Delta t \sqrt{k(\Delta t^2 k - 4m)}}{2m} + 1 \end{aligned}$$

The fact that $\lambda_1 \lambda_2 = 1$ indicates that the energy of the system is conserved over time.

2.2.6 Störmer-Verlet method

The Störmer-Verlet method, often simply referred to as the Verlet method, is a numerical technique used to integrate ordinary differential equations of the form $\dot{y} = f(y)$. It is particularly popular in molecular dynamics simulations and other problems modeled by Hamiltonian systems. The method can be derived directly from the Taylor series expansion of the solution. The central idea behind the method is to use information from both the current and previous time steps to predict the value at the next time step.

The Störmer-Verlet method applied to the Hamiltonian equations is as follows:

$$\begin{cases} p^{n+\frac{1}{2}} = p^n - \frac{\Delta t}{2} \nabla_q V(q^n) \\ q^{n+1} = q^n + \Delta t M^{-1} p^{t+1/2} \\ p^{n+1} = p^{n+\frac{1}{2}} - \frac{\Delta t}{2} \nabla_q V(q^{n+1}) \end{cases} \quad (2.7)$$

The numerical flow of this scheme is noted as ϕ_t^{Verlet} . In matter of fact, note that:

$$\Phi_{\Delta t}^{Verlet} = \phi_{\Delta t/2}^2 \circ \phi_{\Delta t}^1 \circ \phi_{\Delta t}^2$$

Therefore, it is easy to prove that, by construction, is a symplectic scheme. So given the problem of the harmonic oscillator using the Störmer-Verlet method, the update equations become:

$$\begin{cases} p^{n+\frac{1}{2}} = p^n - \frac{\Delta t}{2} k q^n \\ q^{n+1} = q^n + \Delta t M^{-1} p^{t+1/2} \\ p^{n+1} = p^{n+\frac{1}{2}} - \frac{\Delta t}{2} k q^{n+1} \end{cases} \quad (2.8)$$

The Figure 2.7 shows the phase space of the harmonic oscillator using the Störmer-Verlet method.

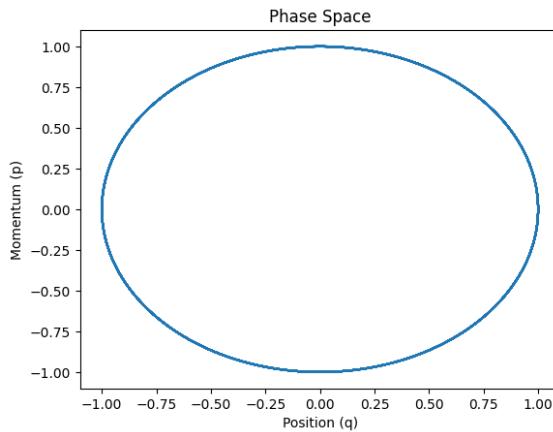


Figure 2.7: Phase space of the harmonic oscillator using the Störmer-Verlet method with $\Delta t = 0.01$.

The Figure 2.7 shows the phase space of the harmonic oscillator using the Störmer-Verlet method. We can see that the trajectory in the phase space is a closed curve, which is the correct behaviour of the harmonic oscillator.

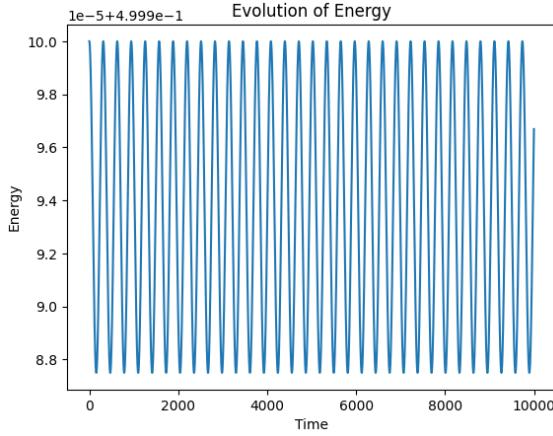


Figure 2.8: Energy of the harmonic oscillator using the Störmer-Verlet method with $\Delta t = 0.01$.

The Figure 2.8 shows the energy of the harmonic oscillator using the Störmer-Verlet method. We can see that the energy is approximately conserved over time, as we wanted for the Hamiltonian dynamics. Like the behaviour of the Euler Symplectic method, we can see that in this case, the energy continues to oscillate around the average value, however the scale of this oscillation is much smaller than in the Euler Symplectic method. To see this, we define the following function:

$$\Gamma_{\Delta t}^{Verlet} = \max_{n=1 \dots N} \{ |H(q_{\Delta t}^n, p_{\Delta t}^n) - H(q_{\Delta t}^0, p_{\Delta t}^0)| \}$$

So now we are prepared to overview the difference between both methods. The Figure 2.9 shows the function $\Gamma_{\Delta t}^{Euler}$ and $\Gamma_{\Delta t}^{Verlet}$ over $\frac{1}{\Delta t}$.

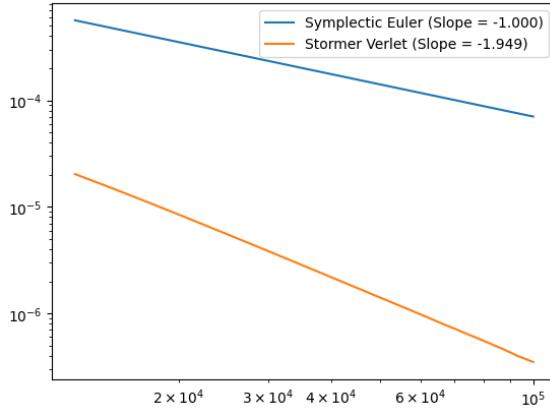


Figure 2.9: Function $\Gamma_{\Delta t}^{Euler}$ and $\Gamma_{\Delta t}^{Verlet}$ over $\frac{1}{\Delta t}$

The comparative analysis presented in Figure 2.9 clearly demonstrates the superiority of the Störmer-Verlet method over the Euler Symplectic method in solving Hamiltonian dynamics equations. This conclusion is evidenced by the slopes of their respective linear regressions: approximately -2 for the Störmer-Verlet method and -1 for the Euler Symplectic method. This difference is likely due to the Störmer-Verlet method being a second-order method, in contrast to the first-order nature of the Euler Symplectic method. Higher-order methods typically offer greater accuracy and stability in numerical simulations, as reflected in the more consistent energy conservation shown by

the Störmer-Verlet method. Therefore, the Störmer-Verlet method emerges as the more effective approach for accurately solving these equations.

2.2.6.1 Linear Stability Analysis of the Verlet Scheme

Using the equation 2.8 and replacing all the terms by its definition, and considering that $\omega = \sqrt{\frac{k}{m}}$, we can rewrite the scheme as:

$$\begin{pmatrix} p^{n+1} \\ q^{n+1} \end{pmatrix} = A \begin{pmatrix} p^n \\ q^n \end{pmatrix}$$

With

$$A = \begin{pmatrix} 1 - \frac{(\omega\Delta t)^2}{2} & \Delta t \\ -\omega^2\Delta t(1 - \frac{(\omega\Delta t)^2}{4}) & 1 - \frac{(\omega\Delta t)^2}{2} \end{pmatrix}$$

Noting $\xi = \frac{(\omega\Delta t)^2}{2}$, the eigenvalues are the solutions of λ in:

$$(1 - \xi - \lambda)^2 + \xi(2 - \xi) = 0$$

Therefore, the eigenvalues are:

$$\begin{cases} \lambda_1 = 1 - \xi + i\sqrt{\xi(2 - \xi)} & \lambda_2 = 1 - \xi - i\sqrt{\xi(2 - \xi)} \text{ if } \xi(2 - \xi) \geq 0 \\ \lambda_1 = 1 - \xi + \sqrt{\xi(\xi - 2)} & \lambda_2 = 1 - \xi - \sqrt{\xi(\xi - 2)} \text{ if } \xi(2 - \xi) \leq 0 \end{cases}$$

In linear stability analysis, a scheme is said to be stable if the eigenvalues of the amplification matrix lie inside the unit circle in the complex plane. This means that their absolute values should be less than or equal to 1 for all possible values of ξ . This ensures that errors do not grow unboundedly as we march forward in time.

Given the eigenvalues derived above:

$$\begin{aligned} \lambda_1 &= \begin{cases} 1 - \xi + i\sqrt{\xi(2 - \xi)} & \text{if } \xi(2 - \xi) \geq 0 \\ 1 - \xi + \sqrt{\xi(\xi - 2)} & \text{if } \xi(2 - \xi) \leq 0 \end{cases} \\ \lambda_2 &= \begin{cases} 1 - \xi - i\sqrt{\xi(2 - \xi)} & \text{if } \xi(2 - \xi) \geq 0 \\ 1 - \xi - \sqrt{\xi(\xi - 2)} & \text{if } \xi(2 - \xi) \leq 0 \end{cases} \end{aligned}$$

The eigenvalues are complex conjugates. Their magnitudes are:

$$|\lambda_1| = |\lambda_2| = \sqrt{(1 - \xi)^2 + \xi(2 - \xi)}$$

For stability, we require $|\lambda_1| \leq 1$. This translates to the condition:

$$(1 - \xi)^2 + \xi(2 - \xi) \leq 1$$

This inequality leads to a tautology so it is always stable. This is consistent with the observed property of the Störmer-Verlet scheme which conserves the energy over time. This, in combination

with its second-order accuracy, makes it preferable over the Euler symplectic method for problems modeled by Hamiltonian systems.

2.3 Solar System Simulation

In this section, we'll harness the power of Hamiltonian dynamics to craft a simulation of the solar system. We'll adopt the Stormer-Verlet scheme, renowned for its energy-conserving properties, as our primary numerical method.

First of all, let's do a brief introduction to the problem we want to solve. We can see the solar system as a set of N bodies, each of them with a mass m_i a position \mathbf{q}_i and a velocity \mathbf{v}_i . The position and velocity are vectors in \mathbb{R}^2 . The gravitational force between two bodies is given by:

2.3.1 Modeling the Solar System: Gravitational Dynamics

The motion of celestial bodies in the solar system, primarily planets, moons, and the Sun, can be described by Newton's law of universal gravitation. For N bodies, the gravitational force exerted on the i^{th} body due to the j^{th} body is given by:

$$\mathbf{F}_{ij} = -\frac{Gm_i m_j (\mathbf{q}_i - \mathbf{q}_j)}{|\mathbf{q}_j - \mathbf{q}_i|^3}$$

where G is the gravitational constant, approximately 6.67430×10^{-11} Nm $^2/\text{kg}^2$, \mathbf{q}_i and \mathbf{q}_j are the position vectors of the i^{th} and j^{th} bodies respectively and m_i and m_j are the masses of the i^{th} and j^{th} bodies respectively. Therefore, the total gravitational force acting on the i^{th} body due to all other bodies is:

$$\mathbf{F}_i = \sum_{j=1, j \neq i}^N \mathbf{F}_{ij}$$

To use Hamiltonian mechanics and the Störmer-Verlet scheme, we can represent the system in terms of its Hamiltonian, which is the sum of its kinetic and potential energies. For the i^{th} body:

1. Kinetic Energy, T_i :

$$T_i = \frac{1}{2} m_i \mathbf{v}_i \cdot \mathbf{v}_i$$

where \mathbf{v}_i is the velocity of the i^{th} body.

2. Potential Energy, U_{ij} , due to the interaction between the i^{th} and j^{th} bodies:

$$U_{ij} = -\frac{Gm_i m_j}{|\mathbf{q}_j - \mathbf{q}_i|}$$

The total Hamiltonian H for the system is the sum of the kinetic and potential energies for all body pairs:

$$H = \sum_{i=1}^N T_i + \frac{1}{2} \sum_{i=1}^N \sum_{j=1, j \neq i}^N U_{ij}$$

This Hamiltonian can be used to derive the equations of motion using Hamilton's equations, which can then be solved using the Störmer-Verlet scheme.

In the next section, we'll dive deeper into the implementation details, but this sets up the mathematical foundation for our solar system simulation based on gravitational interactions.

2.3.2 Störmer-Verlet Scheme for Gravitational Dynamics

Now that we have the Hamiltonian for the system, we can derive use it inside the Störmer-Verlet scheme to derive the equations of motion. The Störmer-Verlet scheme for the i^{th} body is:

$$\begin{cases} \mathbf{p}_i^{n+\frac{1}{2}} = \mathbf{p}_i^n - \frac{\Delta t}{2} \nabla_{\mathbf{q}_i} U(\mathbf{q}_i^n) \\ \mathbf{q}_i^{n+1} = \mathbf{q}_i^n + \Delta t M^{-1} \mathbf{p}_i^{n+1/2} \\ \mathbf{p}_i^{n+1} = \mathbf{p}_i^{n+\frac{1}{2}} - \frac{\Delta t}{2} \nabla_{\mathbf{q}_i} U(\mathbf{q}_i^{n+1}) \end{cases}$$

where $U(\mathbf{q}_i)$ is the potential energy of the i^{th} body due to all other bodies. The total potential energy of the system is the sum of the potential energies of all body pairs:

$$U = \frac{1}{2} \sum_{i=1}^N \sum_{j=1, j \neq i}^N U_{ij}$$

where U_{ij} is the potential energy of the i^{th} body due to the j^{th} body. Therefore, the gradient of the potential energy of the i^{th} body is:

$$\nabla_{\mathbf{q}_i} U(\mathbf{q}_i) = \nabla_{\mathbf{q}_i} U(\mathbf{q}_i) = \sum_{j=1, j \neq i}^N \nabla_{\mathbf{q}_i} U_{ij}(\mathbf{q}_i, \mathbf{q}_j)$$

Given that:

$$U_{ij} = -\frac{Gm_i m_j}{|\mathbf{q}_j - \mathbf{q}_i|}$$

We can compute the gradient of U_{ij} with respect to \mathbf{q}_i . Firstly, we can express the distance $|\mathbf{q}_j - \mathbf{q}_i|$ as r_{ij} :

$$r_{ij} = |\mathbf{q}_j - \mathbf{q}_i|$$

Thus, the potential energy becomes:

$$U_{ij} = -\frac{Gm_i m_j}{r_{ij}}$$

To find the gradient of U_{ij} with respect to \mathbf{q}_i , we need to differentiate with respect to \mathbf{q}_i :

$$\nabla_{\mathbf{q}_i} U_{ij} = \frac{\partial U_{ij}}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial \mathbf{q}_i}$$

First, differentiate U_{ij} with respect to r_{ij} :

$$\frac{\partial U_{ij}}{\partial r_{ij}} = Gm_i m_j \frac{1}{r_{ij}^2}$$

Next, differentiate r_{ij} with respect to \mathbf{q}_i :

$$r_{ij} = \sqrt{(\mathbf{q}_j - \mathbf{q}_i) \cdot (\mathbf{q}_j - \mathbf{q}_i)}$$

$$\frac{\partial r_{ij}}{\partial \mathbf{q}_i} = \frac{\mathbf{q}_j - \mathbf{q}_i}{r_{ij}}$$

Putting it all together:

$$\nabla_{\mathbf{q}_i} U_{ij} = Gm_i m_j \frac{1}{r_{ij}^2} \frac{\mathbf{q}_i - \mathbf{q}_j}{r_{ij}} = Gm_i m_j \frac{\mathbf{q}_i - \mathbf{q}_j}{r_{ij}^3}$$

Thus, the gradient of the potential energy U_{ij} with respect to \mathbf{q}_i is:

$$\nabla_{\mathbf{q}_i} U_{ij} = -Gm_i m_j \frac{\mathbf{q}_i - \mathbf{q}_j}{|\mathbf{q}_j - \mathbf{q}_i|^3}$$

Now, plugging this back into the equation for the gradient of the total potential energy with respect to \mathbf{q}_i :

$$\nabla_{\mathbf{q}_i} U(\mathbf{q}_i) = \sum_{j=1, j \neq i}^N -Gm_i m_j \frac{\mathbf{q}_i - \mathbf{q}_j}{|\mathbf{q}_j - \mathbf{q}_i|^3}$$

This is the gradient of the potential energy with respect to the position of the i^{th} body, taking into account all the interactions with other bodies in the system. Before executing the Störmer-Verlet scheme, let's show the potential energy of the system. The Figure 2.10 shows the potential energy of the system.

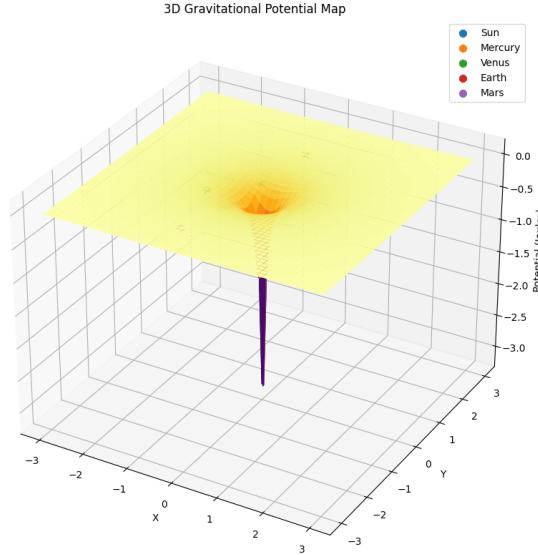


Figure 2.10: Potential energy of the part of the solar system.

As we could have expected, the Sun has, by far, the most negative potential energy, since it is the body with the highest mass. Also we can observe that the potential energy is negative, which is the expected behaviour of the gravitational potential energy.

2.3.3 Implementation of the Solar System Simulation

First of all, we need to define the parameters of the simulation. The Table 2.1 shows the parameters of the simulation.

Celestial Body	Mass (Relative to Sun)	Position (AU)	Mean velocity (AU/Year)
Sun	1.0000	0.000	0.00
Mercury	1.6505e-7	0.390	9.99
Venus	2.4335e-6	0.720	7.38
Earth	2.9860e-6	1.000	6.28
Mars	3.2085e-7	1.520	5.08
Jupiter	9.4950e-4	5.187	2.76

Table 2.1: Data of Celestial Bodies in the Solar System

Therefore, the units that we have used are:

- Mass: M_{\odot} (Solar Mass)
- Distance: AU (Astronomical Unit)
- Time: Earth year

This is because we want to do a simulation of the solar system inside a machine that uses discrete numbers, therefore if we use the SI units, we could get some troubles with the precision of the machine, while if we use the units that we have used, we get numbers that are not too big or too small, so we can get a good precision. Also, using this numbers we get that the gravitational constant is:

$$G = 4\pi^2 \frac{\text{AU}^3}{\text{year}^2 M_{\odot}}$$

Note that we have used the real masses of the planets and the Sun, and the real distance between them. Therefore, we can see that the simulation is very similar to the real solar system. All of these parameters can be found on the NASA website [1]. Now we are prepared to implement the simulation.

For illustrative purposes, we'll simulate the orbits of the Sun, Mercury, Venus, Earth and Mars. The Figure 2.11 shows the orbits of the planets of the solar system during 365 days and with a time step of 1 day.

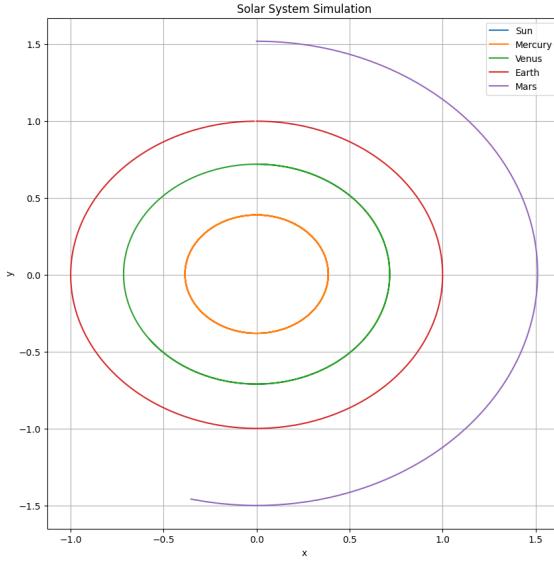


Figure 2.11: Simulation of the orbits of part of the planets of the solar system using the Störmer-Verlet scheme for $\Delta t = 1$ day and $T = 365$ days.

We can see that the orbits are closed curves, which is the expected behaviour of the planets of the solar system. Also, we can see that the orbits are not perfect circles, which is also the expected behaviour of the planets of the solar system. Note that for illustrative purposes, since it is known that the Earth has an orbit of 365.25 days, we have simulated the orbits for 365 days. If we had plotted also other planets like Jupiter, the plot would have been very messy, so we have decided to plot only the inner planets.

Now let's see the energy of the system. The Figure 2.12 shows the energy of the system for the same simulation.

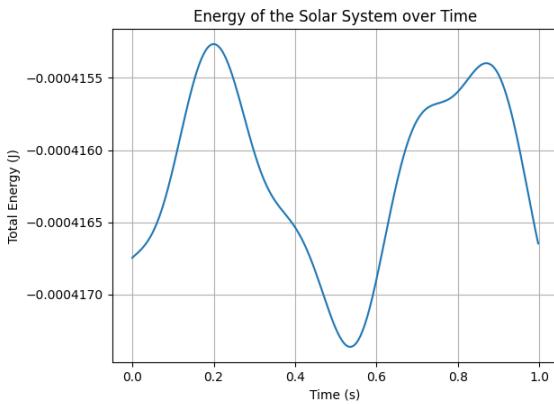


Figure 2.12: Energy of the simulation of part of the planets of the solar system using the Störmer-Verlet scheme for $\Delta t = 1$ day and $T = 365$ days.

We can see that the energy is approximately conserved over time, as we wanted for the Hamiltonian dynamics. Note that is not a straight line but the average of the energy is the same and we can see a slight of oscillations that may be repeated in the future if the simulation is done for more time. So, let's do the simulation for 20 years. The Figure 2.13 shows the energy of the system for

the same simulation.

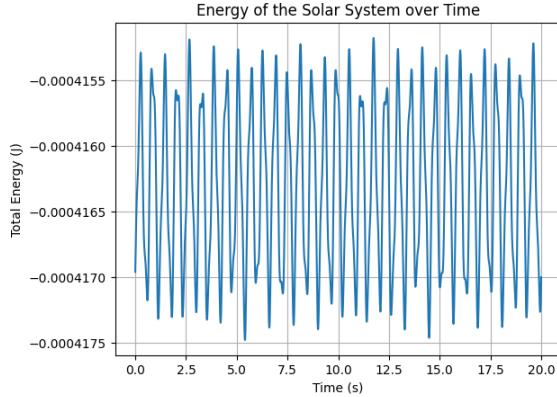


Figure 2.13: Energy of the simulation of part of the planets of the solar system using the Störmer-Verlet scheme for $\Delta t = 1$ day and $T = 20$ years.

As predicted, we can observe that the energy is approximately preserved.

Now that we have the system created and we have seen that the energy is preserved, we can do some experiments.

2.3.3.1 Experiment 1: Expanding to the N-Body Problem

In our initial simulation, as detailed in the referenced table, we included data for Jupiter but did not incorporate it into the actual simulation. This was primarily due to Jupiter's significant distance from the inner planets, which could potentially complicate the visual clarity of our model. However, given Jupiter's substantial mass – the largest among the planets – its inclusion is crucial for a more comprehensive understanding of the solar system's dynamics. This necessitates an expansion of our model to an N-body problem, allowing for the gravitational influence of not just the Sun but also the other planets to be accounted for.

To explore this, we have extended our simulation to include Jupiter, aiming to observe its impact on the orbital paths of the other planets. The updated Figure 2.14 illustrates the orbits of the solar system's planets over a period of 20 years with a daily time step, now incorporating Jupiter. This addition provides a more realistic and complex representation of the gravitational interplay within our solar system.

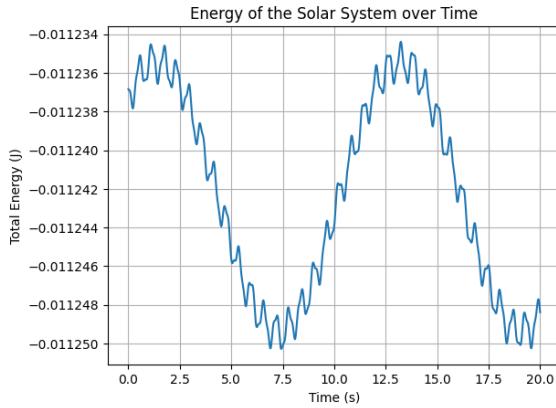


Figure 2.14: Simulation of the orbits of the planets of the solar system using the Störmer-Verlet scheme for $\Delta t = 1$ day and $T = 20$ years, now including Jupiter.

2.3.3.2 Experiment 2: Using Symplectic Euler Method

We already know that the Störmer-Verlet method performs better than the Euler Symplectic method, but let's see how the solar system behaves using the Euler Symplectic method. The Figure 2.15 shows the orbits of the planets of the solar system using the Euler Symplectic method for $\Delta t = 1$ day and $T = 20$ years.

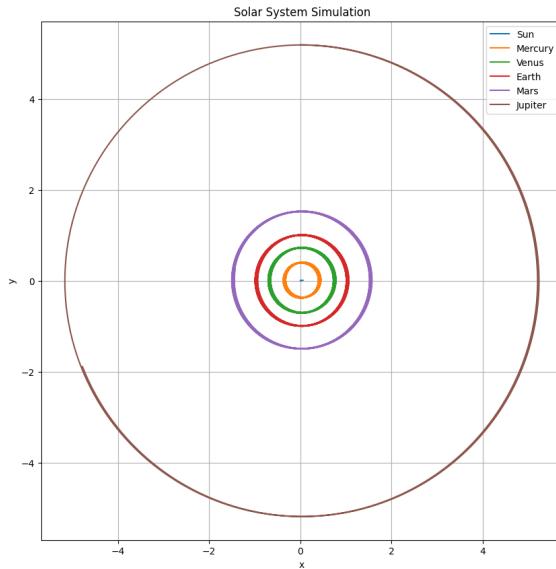


Figure 2.15: Simulation of the orbits of the planets of the solar system using the Euler Symplectic method for $\Delta t = 1$ day and $T = 20$ years.

Also we can plot the energy of the system. The Figure 2.16 shows the energy of the system for the same simulation.

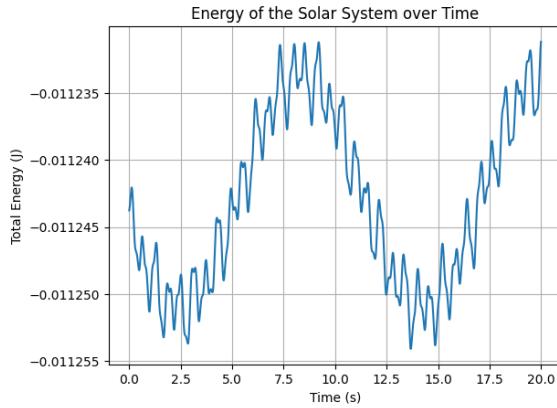


Figure 2.16: Energy of the simulation of the orbits of the planets of the solar system using the Euler Symplectic method for $\Delta t = 1$ day and $T = 20$ years.

Comparing it to the results obtained using the Störmer-Verlet method, again we can see that the first one is better in the following table:

Method	$\max H(x) - \min H(x)$	$\max H(X) - \mathbb{E}(H(X)) $
Symplectic Euler	2.43e-5	1.26e-5
Störmer-Verlet	1.59e-5	9.04e-6

Table 2.2: Data of Celestial Bodies in the Solar System

Therefore, in this problem we can conclude that the Störmer-Verlet method is better in terms of conservation of the Energy than the Euler Symplectic method.

Chapter 3

Markov Chains and their Applications

Chapter 3 embarks on an enthralling journey into the realm of Markov Chains and their myriad applications, a cornerstone in the fields of stochastic processes and computational statistics. We commence this exploration with an introduction to Markov Chains in Section 3.1, where we unravel their intricate structure and delve into their theoretical foundations. This segment includes a deep dive into the Stationary Distribution in Section 3.1.1, a brief state of the Ergodic Theorem in Section 3.1.2 and a comprehensive discussion on the concepts of Irreducibility and Uniqueness in Section 3.1.3. These discussions are pivotal in establishing a foundational understanding of Markov Chains' behavior and their profound implications in a variety of real-world scenarios.

As we progress to Section 3.2, the narrative shifts its spotlight to the Markov Chain Monte Carlo (MCMC) methods, a fascinating facet of Markov Chains. This section is meticulously structured to foster a holistic understanding of these methods. We begin with the Metropolis-Hastings Algorithm, a flagship technique in MCMC, in Section 3.2.1, complemented by a practical example of sampling from a Gamma Distribution in Section 3.2.1.1. Following this, we venture into the world of Gibbs Sampling in Section 3.2.2, another cornerstone MCMC technique, illustrated with an example involving a Bivariate Normal Distribution in Section 3.2.2.1. The discussion extends to the Müller-Brown Potential in Section 3.2.3 and to how the physical potentials can be sampled using MCMC methods thanks to the Boltzmann Distribution in Section 3.2.3.1, showcasing its integration into both the Metropolis-Hastings (Section 3.2.3.2) and Gibbs Sampling (Section 3.2.3.3) algorithms, culminating in a comparative analysis of these methods in Section 3.2.3.4.

The climax of this chapter unfolds in Section 3.3, where we immerse ourselves in the simulation of a 2D molecular system. Here, we present a theoretical exploration of the Lennard-Jones Potential in Section 3.3.1, followed by its practical application in simulating both 2-dimensional and 3-dimensional systems using the Metropolis-Hastings algorithm (Section 3.3.2.1) and the Gibbs Sampling algorithm (Section 3.3.2.2), respectively. This concluding segment not only showcases the practical utility of Markov Chains and MCMC methods but also underscores their remarkable versatility and efficacy in modeling and simulating intricate physical systems.

3.1 Introduction to Markov Chains

Let us start by introducing the concept of *Markov Chain*. A Markov Chain is a stochastic process that satisfies the Markov property. The Markov property states that the conditional probability distribution for the state at the next step depends only on the current state and not on the previous states. The Markov property can be expressed mathematically as:

$$\mathbb{P}(X_{n+1} = x | X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = \mathbb{P}(X_{n+1} = x | X_n = x_n)$$

for any states x, x_1, x_2, \dots, x_n .

A Markov Chain is usually characterised by the matrix of transitions, which is the matrix of the probabilities of transitioning between states of a Markov Chain. The Markov Chain is a square matrix $K(x, y)$ such that $K(x, y) \geq 0$ and $\sum_y K(x, y) = 1$, where the rows and columns are indexed by the states of the Markov Process. The entry in the i^{th} row and j^{th} column of the Markov Chain is the probability of transitioning from state i to state j .

3.1.1 Stationary Distribution

The final objective of the Chapter 3 is to introduce the Markov Chain Monte Carlo Methods (MCMC). As first step to introduce MCMC, we need to introduce the concept of stationary distribution.

Definition 3.1.1 (Stationary Distribution). Let $(X_n)_{n \geq 0}$ be a Markov Chain in E with transition matrix P . A probability measure π on E is called a *stationary distribution* for $(X_n)_{n \geq 0}$ if it satisfies

$$\pi P = \pi$$

The denomination of stationary distribution comes from the following result.

Proposition 3.1.1. Let π be a stationary distribution for $(X_n)_{n \geq 0}$. For any $n \geq 0$,

$$\mathbb{P}_\pi(X_n = y) = \pi(y) \text{ for all } y \in E$$

in other words, π is a stationary distribution if and only if the distribution of X_n is π for all $n \geq 0$.

The proof can be found in [6].

Theorem 3.1.1 (Fundamental Theorem of Markov Chains). Let χ be a finite set and $K(x, y)$ a Markov Chain indexed by χ . If there is $n_0 \in \mathbb{N}$ such that $K^n(x, y) \geq 0$ for all $n > n_0$, then K has a unique stationary distribution π such that

$$K^n(x, y) \xrightarrow{n \rightarrow \infty} \pi(y) \text{ for each } x, y \in \chi$$

This theorem has been extracted from [13].

3.1.2 Ergodic Theorem

After discussing the fundamental theorem and stationary distributions, it is crucial to introduce the Ergodic Theorem for Markov Chains. This theorem is a cornerstone in understanding the long-term behavior of Markov Chains.

Theorem 3.1.2 (Ergodic Theorem for Markov Chains). Let $(X_n)_{n \geq 0}$ be an irreducible and aperiodic Markov Chain on a finite state space E with a stationary distribution π . Then, for any initial distribution μ and for any function $f : E \rightarrow \mathbb{R}$, we have

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n f(X_i) = \sum_{x \in E} f(x)\pi(x) \quad \text{with probability 1.}$$

This theorem asserts that the time average of a function along the paths of the chain converges to a space average, weighted by the stationary distribution. This result is pivotal for the convergence analysis of Markov Chain Monte Carlo methods.

A detailed proof and further discussion can be found in [15].

3.1.3 Irreducibility and Uniqueness

One may ask if the stationary distribution is unique. The answer is no, it is not unique. However, there exists a condition that guarantees the uniqueness of the stationary distribution. This condition is called irreducibility.

Definition 3.1.2. A Markov Chain is said to be *irreducible* if for any two states x and y , there exists $n \geq 0$ such that $K^n(x, y) > 0$.

Proposition 3.1.2. Let K be a Markov Chain with a finite state space χ . If K is irreducible, then it has a unique stationary distribution π .

3.2 Markov Chain Monte Carlo Methods

In this section, we will introduce the Markov Chain Monte Carlo Methods (MCMC). The MCMC are a class of algorithms for sampling from a probability distribution. The idea of MCMC is to construct a Markov Chain that has the desired distribution as its stationary distribution. Therefore, the Markov Chain will converge to the desired distribution as the number of steps increases.

3.2.1 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is a MCMC algorithm for sampling from a probability distribution. The algorithm is defined as follows:

Algorithm 6: Metropolis-Hastings

1. Choose an initial state x_0
 2. For $n = 0, 1, 2, \dots$
 - a. Sample y from $q(x_n, \cdot)$
 - b. Compute the acceptance probability $\alpha(x_n, y)$
 - c. Sample u from $\mathcal{U}(0, 1)$
 - d. If $u < \alpha(x_n, y)$, set $x_{n+1} = y$
 - e. Else, set $x_{n+1} = x_n$
 3. Return $(x_n)_{n \geq 0}$
-

where $q(x, \cdot)$ is a probability distribution on χ and $\alpha(x, y)$ is the acceptance probability defined as:

$$\alpha(x, y) = \min \left(1, \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)} \right)$$

where π is the desired distribution. The proof of the correctness of the algorithm can be found in [3].

3.2.1.1 Example: Sampling from a Gamma Distribution

Let's see an example of the Metropolis-Hastings algorithm. We will use the Metropolis-Hastings algorithm to sample from a Gamma distribution. The Gamma distribution is a two-parameter family of continuous probability distributions. The probability density function of the Gamma distribution is:

$$\pi(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$$

where $\alpha, \beta > 0$ and $\Gamma(\alpha)$ is the Gamma function. The Gamma distribution is often used to model the time until an event occurs. For example, the time until a machine part fails, the time until a queue empties, the time until a customer arrives, etc. The Gamma distribution is also used to model the size of insurance claims and rainfall. Executing the algorithm with the following parameters: $\alpha = 2$ and $\beta = 1$ and $q(x, \cdot) = \mathcal{N}(x, \delta t)$ with $\delta t = 1$, we get the following results:

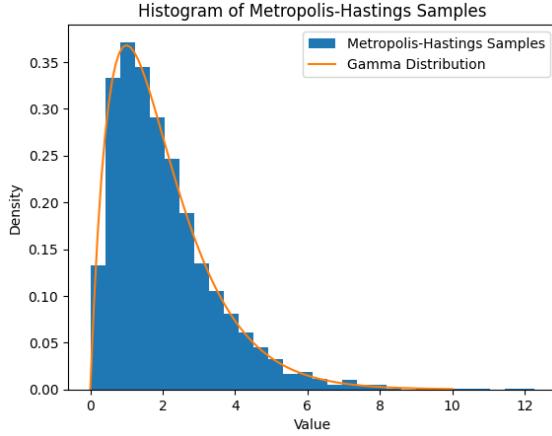


Figure 3.1: Sampling from a Gamma distribution using the Metropolis-Hastings algorithm with $\alpha = 2$ and $\beta = 1$ and $q(x, \cdot) = \mathcal{N}(x, \delta t)$ with $\delta t = 1$.

We can see that the samples are distributed according to the Gamma distribution.

3.2.2 Gibbs Sampling

Gibbs Sampling is a Markov Chain Monte Carlo (MCMC) algorithm used for generating samples from a multidimensional probability distribution π . The essence of this algorithm lies in its iterative approach, sampling from conditional distributions of each variable in sequence. This process is detailed as follows:

Algorithm 7: Gibbs Sampling Algorithm

1. Initialize with a starting state $x_0 = (x_0^{(1)}, x_0^{(2)}, \dots, x_0^{(d)})$
 2. For each iteration $n = 0, 1, 2, \dots$ do:
 - a. Sample $x_{n+1}^{(1)}$ from the conditional distribution $p(x_1|x_n^{(2)}, x_n^{(3)}, \dots, x_n^{(d)})$.
 - b. Sample $x_{n+1}^{(2)}$ from $p(x_2|x_1^{(1)}, x_n^{(3)}, \dots, x_n^{(d)})$.
 - c. Continue in this fashion for each subsequent dimension.
 - d. Finally, sample $x_{n+1}^{(d)}$ from $p(x_d|x_1^{(1)}, x_2^{(2)}, \dots, x_{n+1}^{(d-1)})$.
 3. The sequence $(x_n)_{n \geq 0}$ forms the Markov chain.
-

In this algorithm, $p(x_i|x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_d)$ represents the conditional probability distribution of variable x_i given all other variables in the state. This conditional dependence is the key to understanding the Gibbs Sampling process.

3.2.2.1 Example: Sampling from a Bivariate Normal Distribution

Consider the example of sampling from a bivariate normal distribution using Gibbs Sampling. A bivariate normal distribution involves two random variables, X and Y , each normally distributed and potentially correlated. The distribution is defined by the parameters $\mu_x, \mu_y, \sigma_x, \sigma_y$, and the correlation coefficient ρ :

$$\begin{cases} X \sim \mathcal{N}(\mu_x, \sigma_x^2) \\ Y \sim \mathcal{N}(\mu_y, \sigma_y^2) \\ \text{Correlation coefficient: } \rho \end{cases}$$

The probability density function of the bivariate normal distribution is given by:

$$f(x, y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)} \left[\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} \right] \right)$$

Figure 3.2 illustrates the bivariate normal distribution.

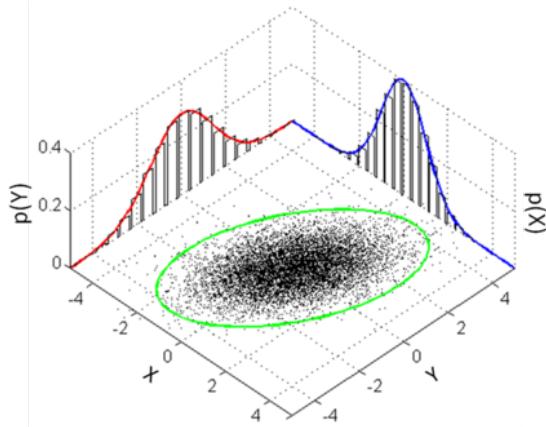


Figure 3.2: Bivariate normal distribution.

In the context of a bivariate normal distribution, the Gibbs Sampling algorithm involves iteratively sampling from the conditional distributions $P(x|y)$ and $P(y|x)$. These distributions can be derived from the joint distribution of X and Y .

Given a bivariate normal distribution with the parameters $\mu_x, \mu_y, \sigma_x, \sigma_y$, and correlation coefficient ρ , the conditional distributions are as follows:

1. $P(x|y)$: The distribution of X given $Y = y$ is a normal distribution with the following mean and variance:

$$\begin{aligned} \text{Mean: } \mu_{x|y} &= \mu_x + \rho \frac{\sigma_x}{\sigma_y} (y - \mu_y) \\ \text{Variance: } \sigma_{x|y}^2 &= (1 - \rho^2) \sigma_x^2 \end{aligned}$$

2. $P(y|x)$: The distribution of Y given $X = x$ is a normal distribution with the following mean and variance:

$$\begin{aligned} \text{Mean: } \mu_{y|x} &= \mu_y + \rho \frac{\sigma_y}{\sigma_x} (x - \mu_x) \\ \text{Variance: } \sigma_{y|x}^2 &= (1 - \rho^2) \sigma_y^2 \end{aligned}$$

The Gibbs Sampling algorithm then proceeds by iteratively sampling X from $P(x|y)$ and Y from $P(y|x)$, using the above conditional means and variances.

Executing this algorithm with parameters $\mu_x = 0$, $\mu_y = 0$, $\sigma_x = 1$, $\sigma_y = 1$, and $\rho = 0.5$, we simulate the sampling process. The resultant samples are expected to align with the bivariate normal distribution characteristics, as depicted in Figure 3.3.

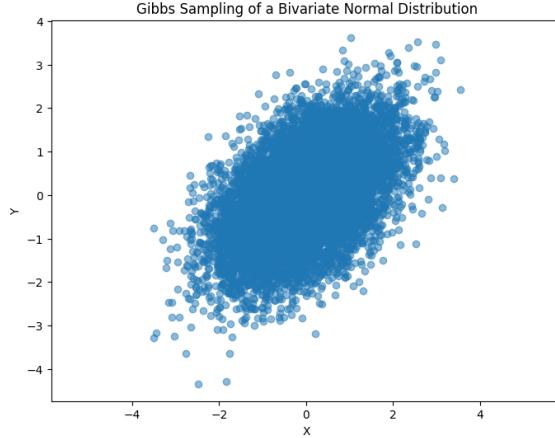


Figure 3.3: Sampling from a bivariate normal distribution using Gibbs Sampling with parameters $\mu_x = 0$, $\mu_y = 0$, $\sigma_x = 1$, $\sigma_y = 1$, and $\rho = 0.5$.

The figure demonstrates that the samples conform to the expected characteristics of a bivariate normal distribution, validating the effectiveness of the Gibbs Sampling algorithm in this context.

3.2.3 Müller-Brown Potential

In this section, we will introduce the Müller-Brown potential. The Müller-Brown potential is a function of two variables that is often used as a test function for optimization algorithms. The Müller-Brown potential is defined as:

$$U(x_1, x_2) = s \sum_{k=1}^4 A_k \exp(\alpha_k(x_1 - a_k)^2 + \beta_k(x_1 - a_k)(x_2 - b_k) + \gamma_k(x_2 - b_k)^2)$$

where A_k is the height of the k^{th} Gaussian, (a_k, b_k) is the location of the k^{th} Gaussian and $(\alpha_k, \beta_k, \gamma_k)$ is the shape of the k^{th} Gaussian. In our case, the parameters that we will use are:

$$\left\{ \begin{array}{l} (A_1, A_2, A_3, A_4) = (-200, -100, -170, 15) \\ (\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (-1, -1, -6.5, 0.7) \\ (\beta_1, \beta_2, \beta_3, \beta_4) = (0, 0, 11, 0.6) \\ (\gamma_1, \gamma_2, \gamma_3, \gamma_4) = (-10, -10, -6.5, 0.7) \\ (a_1, a_2, a_3, a_4) = (1, 0, -0.5, -1) \\ (b_1, b_2, b_3, b_4) = (0, 0.5, 1.5, 1) \\ s = 0.05 \end{array} \right.$$

The Figure 3.4 shows how the Müller-Brown potential looks like with these parameters.

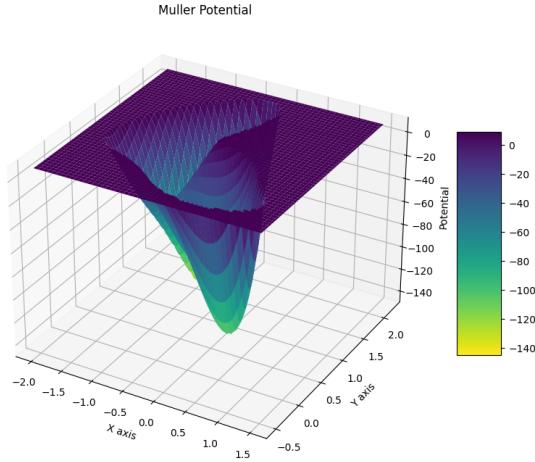


Figure 3.4: Müller-Brown potential.

The Müller-Brown potential is often used as a test function for optimization algorithms because it has many local minimum and one global minimum, therefore it is common that the optimization algorithms get stuck in a local minimum.

3.2.3.1 Boltzmann Distribution

Prior to delving into the application of the Metropolis-Hastings and Gibbs Sampling algorithms to the Müller-Brown potential, a natural query arises: given that the Müller-Brown potential is essentially a function describing energy, how does it relate to the concept of sampling a probability distribution? The answer lies in the realm of the Boltzmann distribution.

The Boltzmann Distribution plays a crucial role in connecting the concept of a potential, such as the Müller-Brown potential, to a probability distribution suitable for sampling methods. This distribution is pivotal in statistical mechanics and provides a fundamental bridge between the microstate of a system (described by the potential energy) and its macrostate (described by temperature and entropy).

The Boltzmann distribution is expressed as follows:

$$\pi(x, y) = \frac{e^{-\frac{U(x, y)}{kT}}}{Z}$$

In this equation, $\pi(x, y)$ denotes the probability of the system being in a state characterized by coordinates (x, y) . The function $U(x, y)$ represents the potential energy of that state, which in this context is given by the Müller-Brown potential. The temperature of the system is denoted by T , and k is the Boltzmann constant, which acts as a bridge between the macroscopic and microscopic worlds.

The denominator, Z , is the partition function, a crucial component in statistical mechanics. It serves as a normalizing factor, ensuring that the probabilities across all possible states sum up to 1. Mathematically, it's defined as an integral over all states:

$$Z = \int \int e^{-\frac{U(x, y)}{kT}} dx dy$$

The integration is performed over the entire configuration space, accounting for all possible states (x, y) .

The Boltzmann distribution's importance cannot be overstated in the context of sampling algorithms like Metropolis-Hastings and Gibbs Sampling. By converting the potential energy landscape into a probability distribution, these algorithms can effectively explore the state space, focusing on regions of high probability (low energy) while still allowing for exploration of less probable (higher energy) states. This approach is not only pivotal for the Müller-Brown potential but forms the foundation for the discussion of physical potentials in statistical mechanics and is extensively utilized in Chapter 3.

3.2.3.2 Metropolis-Hastings Algorithm for the Müller-Brown Potential

Now, we want to know how the Metropolis-Hastings algorithm behaves when we use the Müller-Brown potential as the desired distribution. The Figure 3.5 shows the results of the Metropolis-Hastings algorithm for the Müller-Brown potential.

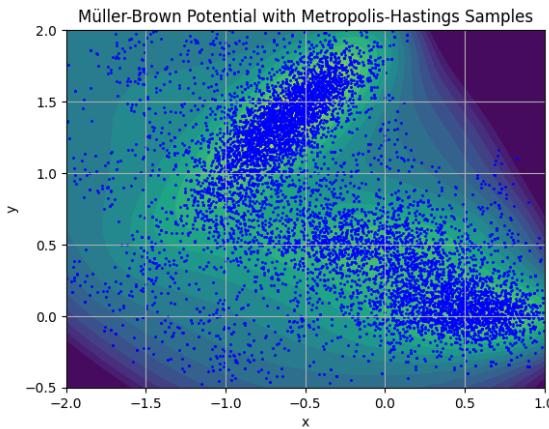


Figure 3.5: Sampling from the Müller-Brown potential using the Metropolis-Hastings algorithm with a temperature of 40 and 100000 samples.

We can see that the samples are distributed all around the Müller-Brown potential, which is the expected behaviour of the Metropolis-Hastings algorithm. But one could think whether this sampling is fitting the distribution or not. This will be checked in a future section after introducing the Gibbs sampling algorithm.

3.2.3.3 Gibbs Sampling Algorithm for the Müller-Brown Potential

In this section, our focus shifts to understanding the behavior of the Gibbs sampling algorithm when applied to the Müller-Brown potential. This potential represents our target distribution.

Previously, we successfully defined conditional distributions for simpler distributions like the bivariate normal. However, the Müller-Brown potential presents a more challenging scenario. Its distinct behavior from a bivariate normal distribution means that we lack a straightforward analytical expression for its conditional distributions. To overcome this hurdle, we turn to a one-dimensional (1D) sampling method for deriving these distributions. Specifically, we employ Rejection Sampling, as detailed in Section 1.2 on Rejection Sampling.

The procedure for sampling x given a fixed y_0 using Rejection Sampling is outlined below:

Algorithm 8: Sampling x given y_0 using Rejection Sampling

1. Determine K , the maximum negative value of the Müller-Brown potential, $-U(x, y_0)$.
 2. Continue sampling until an acceptable sample is found:
 - a. Randomly guess a value for x within the range $[0, b]$, where b represents the size of the considered box.
 - b. Draw a sample u from a uniform distribution between $[0, 1]$.
 - c. Accept the sample x if $u < \frac{e^{U(x, y_0)}}{e^K}$; otherwise, reject it and repeat.
-

With this approach, we can effectively execute the Gibbs Sampling algorithm, where Rejection Sampling facilitates obtaining $p(X|Y)$ and $p(Y|X)$. The results of applying the Gibbs Sampling algorithm to the Müller-Brown potential are illustrated in Figure 3.6.

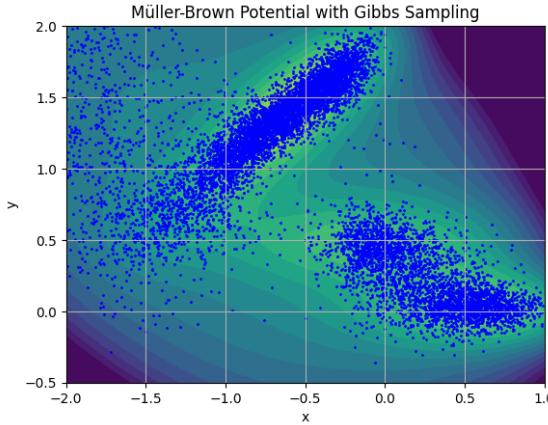


Figure 3.6: Sampling from the Müller-Brown potential using the Gibbs sampling algorithm with 10000 samples.

The distribution of the samples across the Müller-Brown potential landscape, as shown in the figure, aligns with our expectations from the Gibbs sampling method. The next section will further delve into comparing these samples with those obtained through the Metropolis-Hastings algorithm, ensuring they conform to the Müller-Brown potential's characteristics.

3.2.3.4 Comparison between Metropolis-Hastings and Gibbs Sampling

The objective of this section is twofold. Initially, we aim to verify whether the samples generated by the Metropolis-Hastings algorithm accurately reflect the Müller-Brown potential's distribution. Subsequently, we'll compare the Metropolis-Hastings method with the Gibbs Sampling approach.

To achieve these goals, we employ the Kullback-Leibler (KL) divergence, a statistical tool used to measure the difference between two probability distributions. The KL divergence quantifies how one probability distribution diverges from a reference distribution. Mathematically, it's defined as:

$$D_{KL}(P||Q) = \sum_{x \in \chi} P(x) \log \left(\frac{P(x)}{Q(x)} \right)$$

where χ denotes the state space of the distributions in question.

This metric allows us to quantitatively compare two probability distributions. A smaller KL divergence indicates that the distributions are similar, while a larger value suggests significant differences.

Figure 3.7 demonstrates the KL divergence between the samples obtained via the Metropolis-Hastings algorithm and the Müller-Brown potential, observing how this divergence evolves as the sample size increases.

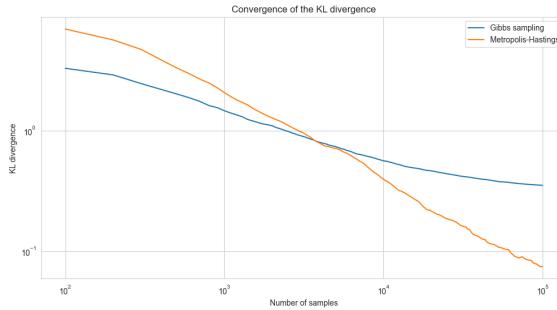


Figure 3.7: Kullback-Leibler divergence between the samples obtained with the Metropolis-Hastings algorithm, with temperature 40, and the Müller-Brown potential as the number of samples increases.

Two critical insights emerge from this figure. First, and most importantly, the decreasing trend of the KL divergence with larger sample sizes suggests that the Metropolis-Hastings algorithm's samples are indeed distributed in alignment with the Müller-Brown potential. Second, the convergence rate of the KL divergence for the Metropolis-Hastings algorithm appears faster than that for the Gibbs Sampling method. This might imply the superiority of the Metropolis-Hastings algorithm in some contexts. However, it's crucial to remember that Gibbs Sampling is based on the ability to obtain conditional distributions, and in this study, we utilized Rejection Sampling, which has potential for optimization. For instance, substituting the uniform distribution for a normal distribution in Rejection Sampling could yield different results. Therefore, we cannot definitively conclude that Metropolis-Hastings is superior to Gibbs Sampling as it heavily depends on the specific problem and the implementation methods used.

The final part of this analysis involves visualizing the heatmaps of both the theoretical distribution and those generated by the Metropolis-Hastings and Gibbs Sampling algorithms. These heatmaps provide a graphical representation of the distribution of samples. Figure 3.8 showcases these heatmaps, all computed using 10,000 samples.

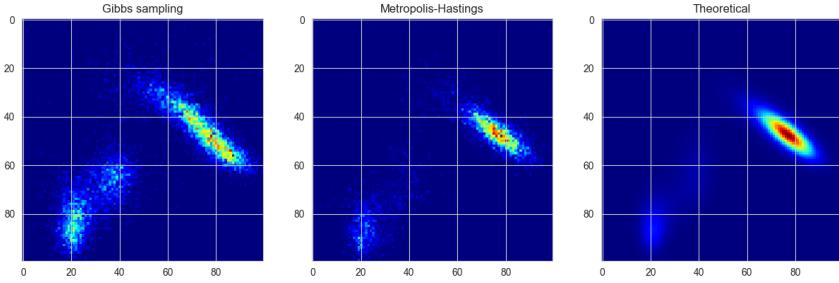


Figure 3.8: Heatmaps of the theoretical distribution, the Metropolis-Hastings algorithm, with tempeature 40, and the Gibbs sampling algorithm. The heatmaps are computed with 100000 samples.

These visual representations reinforce our findings that the samples from both the Metropolis-Hastings and Gibbs Sampling algorithms are indeed consistent with the Müller-Brown potential, as evidenced by the decreasing KL divergence with increasing sample numbers.

3.3 Molecular System Simulation

This final section aims to show how the MCMC algorithms can be used to simulate a molecular system. This subsection is divided in two parts. In the first part, we will introduce the potential we have used to simulate the ecosystem and we will delve a bit into the theory of the molecular dynamics. In the second part, we will do the implementation of the Gibbs sampling algorithm for the molecular system.

3.3.1 Theoretical Framework and Lennard-Jones Potential

At the heart of our molecular simulation lies the Lennard-Jones potential, a simplistic yet powerful model to represent interactions between a pair of atoms or molecules. It is mathematically expressed as:

$$U(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]$$

where $U(r)$ represents the potential energy between two particles separated by a distance r , ϵ signifies the depth of the potential well (indicating the strength of attraction), and σ is the distance at which the potential is zero. The r^{-12} term models the Pauli exclusion principle, responsible for the repulsive force at short distances due to overlapping electron orbitals. Conversely, the r^{-6} term accounts for the attractive van der Waals forces, dominant at longer ranges. This dualistic nature of the potential captures the essence of molecular interactions, balancing attraction and repulsion.

The parameters ϵ and σ are not merely mathematical constructs but hold profound physical meanings. ϵ relates to the energy scale of interactions, dictating how strongly particles attract or repel each other. σ , on the other hand, sets the spatial scale, determining the effective size of the particles. These parameters are often determined empirically or through quantum mechanical calculations for specific substances, making the Lennard-Jones potential a versatile tool for simulating a wide range of molecular systems.

3.3.2 Implementation of the model

When simulating the dynamics of molecular systems, accurately replicating the environmental context is crucial. To this end, we have conceptualized a 10x10 unit square space as the primary stage for the molecular interactions. However, to more closely mimic an infinite environment, where energy is not artificially confined or lost at the boundaries, we adopt a toroidal geometry. A torus, which is essentially a surface of revolution generated by revolving a circle in three-dimensional space around a coplanar axis, offers an elegant solution. This spatial configuration is achieved computationally through the application of modular arithmetic at the boundaries, ensuring a seamless and continuous space. This approach is crucial for maintaining the integrity of the system's energy dynamics and is visually represented in the Figure 3.9.

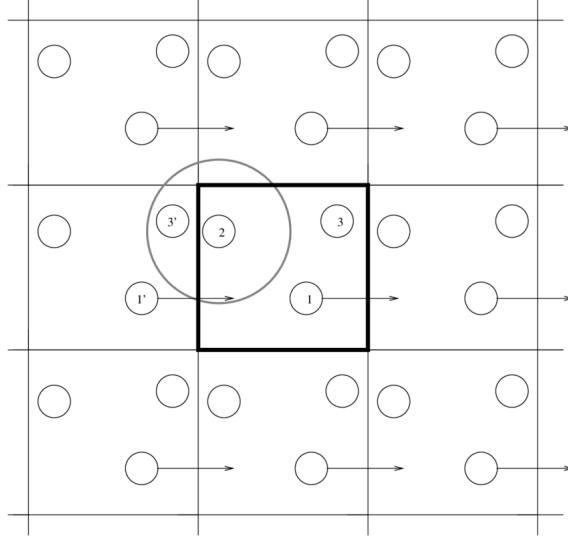


Figure 3.9: Illustration of boundary conditions demonstrating the toroidal topology, adapted from [9].

In our simulation, the initial placement of the particles within this toroidal space is generated using a uniform distribution, ensuring an unbiased starting point for the system. Given that these particles interact via the Lennard-Jones potential, we utilize both algorithms, the Metropolis-Hastings algorithm and the Gibbs sampling algorithm, to simulate their dynamics. These algorithms allow us to observe the evolution of the system under thermal fluctuations and intermolecular forces. For this whole experiment, we will be using the following parameters:

$$\begin{cases} \epsilon = 1 \\ \sigma = 1 \\ \Delta t = 0.01 \\ p^* = 0.9 \\ T^* = 1.5 \end{cases}$$

With ϵ and σ the parameters of the Lennard-Jones potential, Δt the time step, p^* the reduced pressure and T^* the reduced temperature and n the number of particles. Note that since $\epsilon = 1$ and $\sigma = 1$ then $p^* = p$ and $T^* = T$.

3.3.2.1 Implementation 2D using the Metropolis-Hastings algorithm

The Figure 3.10 depicts the spatial distribution of the particles within the torus at the outset and after 1000 iterations of the Metropolis-Hastings algorithm, showcasing the system's dynamic evolution.

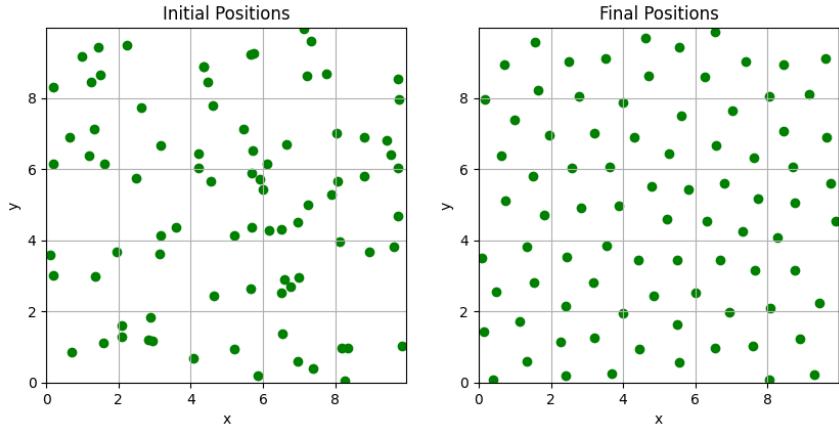


Figure 3.10: Spatial distribution of particles in the torus before and after 1000 iterations of the Metropolis-Hastings algorithm, demonstrating the system's evolution.

One thing that is extremely interesting of this simulation that can be observed in the Figure 3.10 is that, at the first case, we have distributed the particles randomly around the torus, but after 1000 iterations of the Gibbs Sampling algorithm, the particles have been distributed in a way in which we can see that there are a minimum distance between the particles. This is a very interesting result because it shows that the model is probably working as expected.

An integral aspect of our analysis is the examination of the system's energy profile over the course of the simulation. The Metropolis-Hastings algorithm, in theory, should conserve the total energy of the system, a hypothesis we put to the test. The Figure 3.11 captures the energy of the system at the beginning and after 1000 iterations, providing insights into the algorithm's efficacy in maintaining energy conservation.

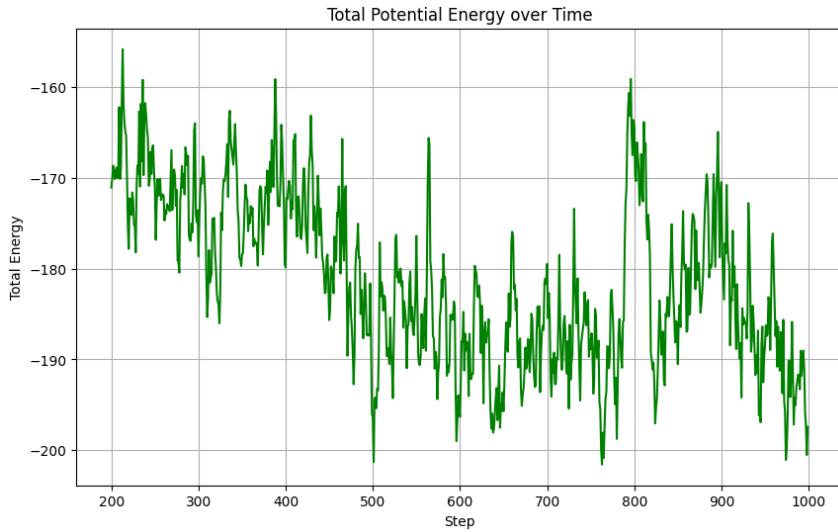


Figure 3.11: Energy profile of the system before and after 1000 iterations of the Metropolis-Hastings algorithm, illustrating the conservation of energy.

The stability of the system's energy, which remains largely unchanged, supports our expectations of the Metropolis-Hastings algorithm. This algorithm is known for maintaining the thermodynamic characteristics of molecular systems. The fact that the energy remains constant is a promising sign that the algorithm is functioning correctly. However, to confirm this more conclusively, we need to employ a more thorough method of verification. For this purpose, we plan to use the radial distribution function. Our current challenge is that most existing studies and references on the radial distribution function are based on three-dimensional systems. Therefore, to align our work with these studies, we will adapt our system to a three-dimensional model. Following this, we'll apply Gibbs Sampling to our three-dimensional system. This approach should allow us to derive a radial distribution function that is relevant and applicable to three-dimensional systems.

3.3.2.2 Implementation 3D using the Gibbs Sampling algorithm

As previously mentioned, while the outcomes from the Metropolis-Hastings algorithm implementation appear optimistic, we need to verify their accuracy. To do this, we will implement the Gibbs Sampling algorithm in a three-dimensional space. As we discussed in Section 3.2.3.3, this requires us to define the conditional distributions in three dimensions, similar to what we did for the Müller-Brown case. In situations like these, where an analytical expression for the conditional distributions is not available, a numerical approach is essential. We have chosen a specific algorithm for this purpose, detailed below.

The algorithm we're using can be summarized as follows:

Algorithm 9: Estimating Conditional Distributions in Three-Dimensional Gibbs Sampling

Input: Index of particle i , position matrix \mathbf{P} , variable range \mathcal{R} , variable dimension d

Output: Estimated conditional probability distribution \mathbf{Pr}

1. Let n be the number of elements in \mathcal{R} .
 2. For each $r \in \mathcal{R}$, construct a matrix \mathbf{P}_{varied} where:
 - Each row is a copy of \mathbf{P}_i , the i -th row of \mathbf{P} .
 - The d -th element of each row in \mathbf{P}_{varied} is set to a corresponding value from \mathcal{R} .
 3. Construct a comprehensive position matrix \mathbf{P}_{all} by replicating \mathbf{P} n times and replacing its i -th row with \mathbf{P}_{varied} .
 4. Calculate the potential energy for each configuration in \mathbf{P}_{all} , denoted as $U(\mathbf{P}_{all})$.
 5. Compute probabilities \mathbf{Pr} from U using the exponential function: $\mathbf{Pr} = e^{-U(\mathbf{P}_{all})}$.
 6. Normalize \mathbf{Pr} so that its sum equals one.
 7. Return the normalized probability distribution \mathbf{Pr} .
-

This outlines our approach for estimating the conditional distributions numerically in three dimensions, as required for our implementation. Note that by using this methodology what we are really doing is sampling a discrete version of the Gibbs Sampling. If we wanted to sample the continuous version of the Gibbs Sampling, we would need to use the Metropolis-Hastings algorithm to sample the conditional distributions. However, this would be computationally expensive and would require a lot of time to compute. Therefore, we will use the discrete version of the Gibbs Sampling algorithm.

Now that we have a method to compute the conditional distributions, we can proceed with the implementation of the Gibbs Sampling algorithm. The Figure 3.12 depicts the spatial distribution of the particles within the torus at the outset and after 100 iterations of the Gibbs Sampling algorithm, showcasing the system's dynamic evolution.

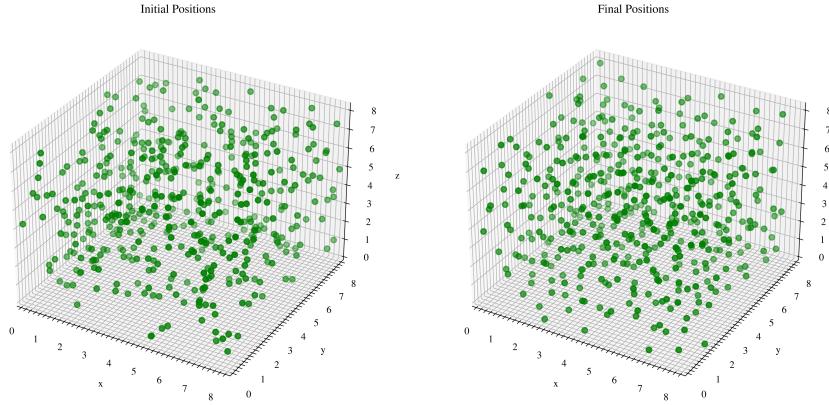


Figure 3.12: Spatial distribution of particles in the torus in 3 dimensions after 100 iterations of the Gibbs Sampling algorithm.

We have seen before the interesting result of the minimum distance between the particles, so now we want to see if indeed the theoretical aspects of the Lennard-Jones potential are being fulfilled. For this aim we can move on to the concept of the radial distribution function.

The radial distribution function (RDF), $g(r)$, is a key concept in understanding the spatial distribution of particles in a system, particularly in molecular simulations like the ones performed using the Gibbs Sampling algorithm. Mathematically, the RDF is defined as the ratio of the local density at a distance r from a reference particle to the average density of the system.

In a more formal sense, if ρ represents the average number density of particles in the system, and $\rho(r)$ represents the local number density at a distance r from a reference particle, then the RDF is given by:

$$g(r) = \frac{\rho(r)}{\rho}$$

For computational purposes, $g(r)$ can be estimated by considering a series of concentric shells of thickness δr around a reference particle. The probability of finding a particle in each shell is then computed and normalized by the volume of the shell and the average number density of particles. Mathematically, for a shell between radii r and $r + \delta r$, the RDF is estimated as:

$$g(r) \approx \frac{N(r)}{4\pi r^2 \delta r \rho}$$

where $N(r)$ is the average number of particles found in the shell.

To rewrite the caption and accompanying text for the provided figure, which illustrates the process of estimating the radial distribution function (RDF), you can use the following revised version:

For clarity, let us visualize the methodology behind calculating the radial distribution function. Figure 3.13 provides a schematic representation of this concept.

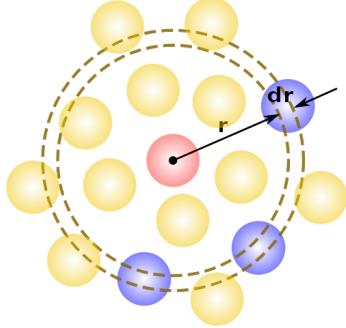


Figure 3.13: Schematic representation of radial distribution function estimation.

In this schematic, the central red particle serves as the reference point around which concentric shells (depicted by dashed circles) of infinitesimal thickness, δr , are considered. The RDF is estimated by counting the number of particles within the r and δr (illustrated as blue circles) and normalizing this count by the volume and the overall particle density. This process is replicated across all particles and averaged to yield the RDF, $g(r)$, which quantifies the probability of finding another particle at a distance r from a given reference particle.

This function is fundamental in characterizing the structure of materials, particularly in the context of the Lennard-Jones potential, as it provides insights into the arrangement of particles at various scales. Peaks in $g(r)$ indicate preferred inter-particle distances due to energetic favorability, while valleys suggest distances that particles tend to avoid.

In the context of our study using the Gibbs Sampling algorithm, the RDF helps validate the accuracy and realism of the simulated particle distributions. By comparing the RDF obtained from the simulation (as shown in Figure 3.15) with the theoretical or experimental RDF (as depicted in Figure 3.14), we can assess how well the simulation captures the physical properties of the system under study. The agreement between the simulated and theoretical RDFs, despite the finite number of particles and computational limitations, reinforces the effectiveness of the Gibbs Sampling algorithm in replicating the physical behavior of the system.

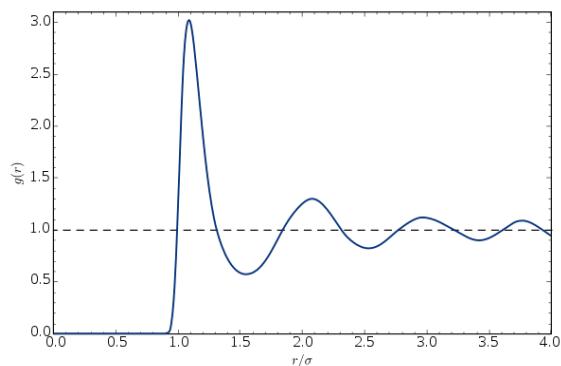


Figure 3.14: Radial distribution function for a liquid.

If we try to compute the radial distribution function for our system, we will get the Figure 3.15.

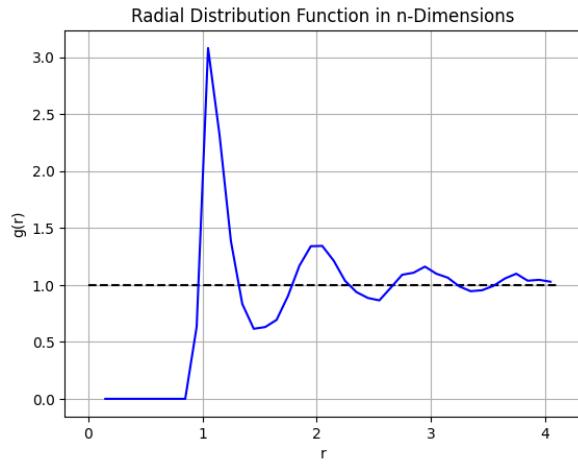


Figure 3.15: Radial distribution function for the system after 1000 iterations of the Gibbs Sampling algorithm.

We can see that the radial distribution function is almost of the same shape as the theoretical one, even though there is not as smooth as the other one since we have a finite number of particles. Therefore, we can conclude that it seems that the Gibbs Sampling algorithm is working as expected, concluding that we have created the aimed molecular system.

Bibliography

- [1] Planetary fact sheet, 2023. Accessed: 2023-11-02.
- [2] Wikipedia contributors. Inverse transform sampling, 2023. Accessed: 2023-11-24.
- [3] Wikipedia contributors. Metropolishastings algorithm - wikipedia. Wikipedia, The Free Encyclopedia, 2023. Accessed: 2023-11-29.
- [4] Wikipedia contributors. Multivariate normal distribution - wikipedia. Wikipedia, The Free Encyclopedia, 2023. Accessed: 2023-11-29.
- [5] Wikipedia contributors. Radial distribution function - wikipedia. Wikipedia, The Free Encyclopedia, 2024. Accessed: 2024-01-20.
- [6] Persi Diaconis. The markov chain monte carlo revolution. *Bulletin of the American Mathematical Society*, 46:179–205, 2008.
- [7] V. Frayssé. Hdr thesis, 2018.
- [8] Nicholas J. Higham. What is backward error?, 3 2020.
- [9] Thomas A. Hunt and Billy Dean Todd. On the arnold cat map and periodic boundary conditions for planar elongational flow. *Molecular Physics*, 101(23):3445–3454, 2003.
- [10] Benjamin Jourdain. *Probabilités et statistique*. Ellipses, 2 edition, 2016. Bibliogr. p. 185. Index.
- [11] K. I. M. McKinnon. Convergence of the nelder–mead simplex method to a nonstationary point. *SIAM Journal on Optimization*, 9(1):148–158, 1998.
- [12] Roger D. Peng. 6.3 rejection sampling, 2023. Accessed: 2023-11-25.
- [13] Julien Reygner. Méthodes numériques probabilistes, 2023. Course material for Master de Mathématiques et Applications, Spécialité Mathématiques de la Modélisation, Sorbonne Université et École des Ponts ParisTech, Academic Year 2023/2024.
- [14] Gabriel Stoltz. *An Introduction to Computational Statistical Physics*. Master de Mathématiques et Applications Spécialité Mathématiques de la Modélisation, 1 2023.
- [15] University of Cambridge. Ergodic theorem for markov chains. Online.