

Short Research Articles

# Theoretical and Experimental Analysis of the Efficiency of Deterministic and Stochastic (Monte Carlo) Numerical Integration Methods

## Abstract

This paper presents a comparative study between classical numerical integration methods (trapezoidal, midpoint, Simpson) and the stochastic Monte Carlo method. After a rigorous demonstration of the convergence orders, we experimentally validate the results using Python simulations. The experiments show that while deterministic methods outperform Monte Carlo in low dimensions for regular functions, the Monte Carlo method becomes more efficient in higher dimensions or when regularity breaks down.

*Keywords: Numerical Integration, Deterministic Quadrature, Monte Carlo Method, Convergence Analysis, Computational Efficiency, Curse of Dimensionality, Error Estimation, Simpson's Rule, Trapezoidal Rule, Midpoint Method, High-dimensional Integration, Variance Reduction, Quasi-Monte Carlo Sampling*

2010 Mathematics Subject Classification: 53C25; 83C05; 57N16

## 1 Introduction

Numerical integration is an essential component of modern numerical analysis. Classical methods rely on deterministic quadratures such as Newton–Cotes formulas, whereas the Monte Carlo method is based on probabilistic principles derived from the law of large numbers.

Our objective is to mathematically study the errors and convergence rates of these approaches, then to verify their validity through numerical simulation, highlighting the impact of dimensionality on precision.

## 2 Literature Review

Numerical integration constitutes a fundamental pillar of modern numerical analysis. Since the pioneering works of Newton and Cotes in the 18th century, classical quadrature formulas—such as the trapezoidal, Simpson, or Gauss–Legendre methods—have been central to methodological developments aimed at approximating integrals of continuous and sufficiently regular functions [7, 4].

---

These methods are based on deterministic schemes whose convergence strongly depends on the regularity of the function and on the discretization step  $h$ .

However, as problems have become more complex—particularly in computational physics, Bayesian statistics, and artificial intelligence—the integration of high-dimensional functions ( $d > 3$ ) has become a major challenge. This phenomenon, known as the *curse of dimensionality*, was highlighted as early as the 1960s by Bellman [5]. Indeed, the number of evaluations required to achieve a given accuracy grows exponentially with the dimension, making deterministic methods rapidly impractical. It is in this context that the **Monte Carlo method** emerged, introduced during World War II by Metropolis and Ulam [6]. Based on the law of large numbers and the central limit theorem, this stochastic approach estimates an integral by averaging randomly sampled values uniformly distributed over the integration domain. Unlike quadrature methods, the accuracy of Monte Carlo does not depend on the dimension of the space but rather on the number of samples  $n$ , with an asymptotic convergence of  $O(n^{-1/2})$ . This property gives the method a decisive advantage in high-dimensional spaces. Numerous works have since deepened its variants: **Quasi-Monte Carlo** methods, introduced by Halton (1960) and Sobol (1967), use low-discrepancy sequences to improve convergence toward  $O(n^{-1}(\log n)^d)$  [8, 9]. Other approaches, such as variance reduction or importance sampling, aim to reduce the multiplicative constant without changing the theoretical order [10, 11].

Despite this wealth of developments, the literature shows that few systematic comparative studies have been conducted between deterministic and stochastic approaches in a unified framework—one that evaluates not only precision but also **computational cost** and **robustness** to function regularity. In most numerical analysis textbooks, the comparison remains qualitative and limited to low-dimensional or regular-function cases [12, 13].

**Problem Statement and Relevance of This Work.** Our study precisely addresses this gap. We propose a joint theoretical and experimental analysis that aims to:

- mathematically demonstrate the convergence orders of classical methods (trapezoidal, midpoint, Simpson);
- empirically compare their performance with that of the Monte Carlo method for functions of increasing complexity;
- measure and visualize the efficiency of each method as a function of **computational time** and **problem dimensionality**.

This work therefore has a dual significance. On one hand, it clarifies the asymptotic behavior of classical methods under the effect of dimension; on the other, it highlights the universal character of Monte Carlo, able to maintain stable accuracy regardless of the dimension or regularity of the function. This analysis provides valuable insights for fields where numerical integration is ubiquitous: scientific computing, machine learning, computational economics, and the simulation of complex systems.

## 3 Theoretical Foundations

### 3.1 Composite Trapezoidal Method

**Theorem 3.1** (Error of the Composite Trapezoidal Rule). *Let  $f \in C^2([a, b])$ . Setting  $h = (b - a)/n$ ,*

$$T_n(f) = \frac{h}{2} \left( f(a) + 2 \sum_{i=1}^{n-1} f(a + ih) + f(b) \right),$$

*there exists  $\xi \in (a, b)$  such that:*

$$\int_a^b f(x) dx - T_n(f) = -\frac{(b-a)}{12} h^2 f''(\xi).$$

*Proof.* On each interval  $[x_i, x_{i+1}]$ , we linearly interpolate  $f$ . Let  $p(x)$  be the degree-1 polynomial coinciding with  $f$  at  $x_i$  and  $x_{i+1}$ . The Lagrange remainder gives:

$$f(x) - p(x) = \frac{f''(\eta_x)}{2}(x - x_i)(x - x_{i+1}),$$

where  $\eta_x \in (x_i, x_{i+1})$ . Integrating from  $x_i$  to  $x_{i+1}$ :

$$\int_{x_i}^{x_{i+1}} f(x) dx - \frac{h}{2}(f(x_i) + f(x_{i+1})) = \frac{f''(\xi_i)}{2} \int_{x_i}^{x_{i+1}} (x - x_i)(x - x_{i+1}) dx.$$

Computing gives:

$$\int_{x_i}^{x_{i+1}} (x - x_i)(x - x_{i+1}) dx = -\frac{h^3}{6}.$$

Thus,  $\int_{x_i}^{x_{i+1}} f(x) dx - T_i = -\frac{h^3}{12} f''(\xi_i)$ . Summing over  $i = 0, \dots, n - 1$  and applying the mean value theorem:

$$\int_a^b f(x) dx - T_n(f) = -\frac{(b-a)}{12} h^2 f''(\xi).$$

□

### 3.2 Midpoint Method

**Theorem 3.2** (Error of the Midpoint Rule). For  $f \in C^2([a, b])$ ,

$$\int_a^b f(x) dx - M_n(f) = \frac{(b-a)}{24} h^2 f''(\xi).$$

*Proof.* On  $[x_i, x_{i+1}]$ , let  $m_i = (x_i + x_{i+1})/2$ . We expand  $f(x)$  around  $m_i$ :

$$f(x) = f(m_i) + f'(m_i)(x - m_i) + \frac{f''(\eta_x)}{2}(x - m_i)^2.$$

Integration over  $[x_i, x_{i+1}]$  cancels the linear term:

$$\int_{x_i}^{x_{i+1}} f(x) dx - hf(m_i) = \frac{f''(\xi_i)}{2} \int_{x_i}^{x_{i+1}} (x - m_i)^2 dx = \frac{h^3}{24} f''(\xi_i).$$

Summing gives:

$$\int_a^b f(x) dx - M_n(f) = \frac{(b-a)}{24} h^2 f''(\xi).$$

□

### 3.3 Simpson's Method

**Theorem 3.3** (Error of the Composite Simpson Rule). If  $f \in C^4([a, b])$  and  $n$  is even,

$$\int_a^b f(x) dx - S_n(f) = -\frac{(b-a)}{180} h^4 f^{(4)}(\xi).$$

*Proof.* Over a block  $[x_0, x_2]$  of length  $2h$ , Simpson's rule corresponds to integrating the quadratic polynomial interpolating  $f(x_0), f(x_1), f(x_2)$ . Expanding  $f$  as a Taylor series of order 4 around the midpoint  $m = x_0 + h$  shows that terms up to order 3 are exact, and the first missing term is:

$$-\frac{h^5}{90} f^{(4)}(\xi_0).$$

Summing over all blocks yields:

$$E_n = -\frac{(b-a)}{180} h^4 f^{(4)}(\xi).$$

□

## 3.4 In-depth Analysis of the Monte Carlo Method

The Monte Carlo method represents one of the most powerful and versatile techniques in numerical integration, particularly when dealing with high-dimensional problems or integrands exhibiting irregular behavior. Unlike deterministic quadratures that rely on a fixed mesh of evaluation points, Monte Carlo methods use randomness to estimate integrals by statistical averaging.

### 3.4.1 Principle of the Method

Let  $f : D \subset \mathbb{R}^d \rightarrow \mathbb{R}$  be an integrable function over a domain  $D$  of finite measure. The integral

$$I = \int_D f(x) dx$$

can be expressed as an expectation with respect to a random variable  $X$  uniformly distributed on  $D$ :

$$I = |D| \mathbb{E}[f(X)].$$

If  $X_1, \dots, X_n$  are i.i.d. samples from the uniform distribution on  $D$ , the Monte Carlo estimator of  $I$  is

$$\hat{I}_n = \frac{|D|}{n} \sum_{i=1}^n f(X_i).$$

According to the law of large numbers,  $\hat{I}_n \rightarrow I$  almost surely as  $n \rightarrow \infty$ . Furthermore, the central limit theorem provides a measure of statistical uncertainty:

$$\sqrt{n}(\hat{I}_n - I) \xrightarrow{\mathcal{L}} \mathcal{N}(0, |D|^2 \sigma^2),$$

where  $\sigma^2 = \text{Var}(f(X))$ .

### 3.4.2 Error and Convergence Properties

The Monte Carlo estimator is unbiased:

$$\mathbb{E}[\hat{I}_n] = I.$$

Its mean-square error is purely statistical:

$$\text{MSE}(\hat{I}_n) = \frac{|D|^2 \sigma^2}{n}.$$

Hence, the standard deviation of the error decreases as  $O(n^{-1/2})$ , independently of the dimension  $d$ . This independence from dimensionality makes Monte Carlo particularly attractive for integrals in large-dimensional spaces, where deterministic methods suffer from exponential growth in computational cost (*curse of dimensionality*).

### 3.4.3 Variance Reduction Techniques

Although the theoretical convergence rate of Monte Carlo cannot be improved beyond  $O(n^{-1/2})$  without altering the probabilistic framework, various techniques exist to reduce the variance of the estimator:

- **Importance Sampling:** changing the sampling distribution to concentrate samples where  $f$  has larger contributions.
- **Stratified Sampling:** partitioning the domain into subregions and sampling proportionally within each, reducing variability.

- **Antithetic Variates:** using negatively correlated samples  $(X_i, 1 - X_i)$  to cancel part of the random fluctuations.
- **Control Variates:** combining the target function  $f$  with auxiliary functions whose integrals are known analytically.

### 3.4.4 Quasi–Monte Carlo and Low-Discrepancy Sequences

A notable enhancement of the classical approach is the **Quasi–Monte Carlo** method, which replaces random sampling with deterministic low-discrepancy sequences (Halton, Sobol, Faure, etc.). These sequences fill the integration domain more uniformly than purely random samples. In this case, the error can be bounded by

$$|I - \hat{I}_n| \leq V(f)D_n^*,$$

where  $V(f)$  is the total variation of  $f$  and  $D_n^*$  the star-discrepancy of the sequence, which decreases approximately as  $O(n^{-1}(\log n)^d)$ . Thus, Quasi–Monte Carlo methods can surpass classical Monte Carlo when  $d$  is moderate and  $f$  has bounded variation.

### 3.5 Advantages and Limitations

- **Advantages:** Dimension-independent convergence rate, simple implementation, and applicability to complex or discontinuous integrands.
- **Limitations:** Slow convergence compared with deterministic methods in low dimension, and statistical uncertainty inherent to random sampling.

In summary, the Monte Carlo method provides a universal, probabilistic approach to numerical integration. Its simplicity, scalability, and robustness make it indispensable in modern computational mathematics, especially for problems arising in physics, finance, and machine learning.

### 3.6 Theoretical Comparison with Monte Carlo

**Proposition 3.1.** For  $f : [0, 1]^d \rightarrow \mathbb{R}$ ,

$$|I - Q_n(f)| = O(n^{-p/d}), \quad |I - I_n^{MC}| = O_{\mathbb{P}}(n^{-1/2}).$$

*Proof.* Regular quadratures use  $n \approx h^{-d}$  points, giving error  $O(h^p) = O(n^{-p/d})$ . For Monte Carlo, the variance of the estimator is  $\sigma^2/n$ , hence  $|I - I_n^{MC}| = O_{\mathbb{P}}(n^{-1/2})$ , independent of  $d$ .  $\square$

### 3.7 Central Limit Theorem for Monte Carlo

**Theorem 3.4.** Let  $X_i \sim \mathcal{U}([a, b])$  i.i.d. and  $Y_i = f(X_i)$ . Then:

$$\sqrt{n}(\hat{I}_n - I) \xrightarrow{\mathcal{L}} \mathcal{N}(0, (b - a)^2 \sigma^2),$$

where  $\sigma^2 = \text{Var}(f(U))$ .

*Proof.* Let  $Z_i = f(X_i) - \mathbb{E}[f(X_i)]$ . The CLT gives:

$$\frac{1}{\sqrt{n}} \sum Z_i \xrightarrow{\mathcal{L}} \mathcal{N}(0, \sigma^2),$$

hence:

$$\sqrt{n}(\hat{I}_n - I) = (b - a) \frac{1}{\sqrt{n}} \sum Z_i \xrightarrow{\mathcal{L}} \mathcal{N}(0, (b - a)^2 \sigma^2).$$

$\square$

## 4 Results

### 4.1 Numerical Results in One Dimension

We consider five test functions on  $[0, 1]$ :

$$f_1(x) = \sin x, \quad f_2(x) = e^{-x^2}, \quad f_3(x) = \frac{1}{1+x^2},$$

$$f_4(x) = x(1-x) \sin(200x(1-x)), \quad f_5(x) = \frac{1}{\sqrt{x+10^{-12}}}.$$

Table 1 presents the absolute errors obtained for different values of  $N$ .

Table 1: Absolute integration errors for test functions (dimension 1).

Function	Method	N=100	N=1000	N=10000
$f_1(x) = \sin x$	Trapezoidal	$6.7 \times 10^{-3}$	$9.4 \times 10^{-4}$	$3.7 \times 10^{-4}$
	Simpson	$3.8 \times 10^{-5}$	$2.1 \times 10^{-7}$	$1.2 \times 10^{-9}$
	Monte Carlo	$1.1 \times 10^{-1}$	$1.9 \times 10^{-3}$	$3.4 \times 10^{-3}$
$f_2(x) = e^{-x^2}$	Trapezoidal	$9.8 \times 10^{-3}$	$3.0 \times 10^{-3}$	$8.9 \times 10^{-4}$
	Simpson	$5.0 \times 10^{-6}$	$3.8 \times 10^{-8}$	$2.5 \times 10^{-10}$
	Monte Carlo	$1.2 \times 10^{-1}$	$5.9 \times 10^{-3}$	$1.7 \times 10^{-3}$
$f_3(x) = 1/(1+x^2)$	Trapezoidal	$1.0 \times 10^{-2}$	$7.9 \times 10^{-3}$	$3.4 \times 10^{-3}$
	Simpson	$7.1 \times 10^{-6}$	$3.5 \times 10^{-8}$	$2.0 \times 10^{-10}$
	Monte Carlo	$9.5 \times 10^{-2}$	$1.0 \times 10^{-2}$	$7.0 \times 10^{-4}$
$f_4(x) = x(1-x) \sin(200x(1-x))$	Trapezoidal	$1.8 \times 10^{-2}$	$6.2 \times 10^{-6}$	$1.7 \times 10^{-8}$
	Simpson	$1.1 \times 10^{-3}$	$3.0 \times 10^{-7}$	$1.5 \times 10^{-9}$
	Monte Carlo	$2.8 \times 10^{-2}$	$7.2 \times 10^{-3}$	$4.1 \times 10^{-4}$
$f_5(x) = 1/\sqrt{x}$	Trapezoidal	$7.5 \times 10^3$	$1.3 \times 10^3$	$1.4 \times 10^2$
	Simpson	$6.8 \times 10^3$	$1.2 \times 10^3$	$1.3 \times 10^2$
	Monte Carlo	$1.2 \times 10^{-1}$	$1.0 \times 10^{-1}$	$2.7 \times 10^{-3}$

### Performance Summary by Function Type

- For regular functions ( $f_1, f_2, f_3$ ), deterministic methods quickly achieve near machine precision, confirming their theoretical orders  $O(h^2)$  and  $O(h^4)$ .
- For the oscillatory function  $f_4$ , accuracy remains good but depends on the step size: oscillations are better handled by Monte Carlo for small  $N$ .
- For the singular function  $f_5$ , deterministic methods fail, while Monte Carlo maintains a stable error ( $10^{-2}$ – $10^{-3}$ ).

### 4.2 Multidimensional Results

for all  $f_i$ , let  $\psi(x)$  be defined by:

$$\psi_d(x) = (f_i(x))_{d \in [1,5]}$$

. We can then compute the integral for each dimension  $d$  and evaluate the error. Reference values obtained by high-precision Monte Carlo are:

$$\begin{aligned} d = 1 & : I \approx 0.636922, \\ d = 2 & : I \approx 0.557687, \\ d = 3 & : I \approx 0.535803, \\ d = 4 & : I \approx 0.164268, \\ d = 5 & : I \approx 0.100962. \end{aligned}$$

Table ?? summarizes the absolute errors according to dimension.

Table 2: Comparison between Monte Carlo and Trapezoidal methods by dimension and sample size (Part 1).

Dimension	N=100	N=1000	N=5000
1	MC: $2.9 \times 10^{-2}$ / TP: $6.7 \times 10^{-3}$	MC: $1.9 \times 10^{-3}$ / TP: $9.4 \times 10^{-4}$	MC: $3.6 \times 10^{-3}$ / TP: $4.3 \times 10^{-4}$
2	MC: $2.9 \times 10^{-3}$ / TP: $1.0 \times 10^{-2}$	MC: $5.9 \times 10^{-3}$ / TP: $3.0 \times 10^{-3}$	MC: $1.7 \times 10^{-3}$ / TP: $1.3 \times 10^{-3}$
3	MC: $6.3 \times 10^{-3}$ / TP: $1.8 \times 10^{-2}$	MC: $1.0 \times 10^{-2}$ / TP: $7.9 \times 10^{-3}$	MC: $3.5 \times 10^{-3}$ / TP: $4.4 \times 10^{-3}$

Table 3: Comparison between Monte Carlo and Trapezoidal methods by dimension and sample size (Part 2).

Dimension	N=100	N=1000	N=5000
4	MC: $5.5 \times 10^{-3}$ / TP: $1.5 \times 10^{-1}$	MC: $7.2 \times 10^{-3}$ / TP: $9.5 \times 10^{-2}$	MC: $3.4 \times 10^{-3}$ / TP: $7.4 \times 10^{-2}$
5	MC: $8.4 \times 10^{-3}$ / TP: $2.3 \times 10^{-2}$	MC: $1.1 \times 10^{-3}$ / TP: $1.5 \times 10^{-2}$	MC: $1.6 \times 10^{-3}$ / TP: $1.1 \times 10^{-2}$

**Trend Analysis:**

- For  $d = 1$ , classical methods remain more accurate.
- Starting from  $d = 3$ , quadrature degradation becomes noticeable.
- For  $d \geq 4$ , Monte Carlo becomes significantly more efficient.

The results confirm the theoretical orders established earlier: deterministic methods are ideal for smooth and low-dimensional integrals, but become impractical as  $d$  increases due to the exponential computational cost. The Monte Carlo method maintains linear cost and a constant convergence order  $O(n^{-1/2})$ , making it preferable for high-dimensional or irregular domains.

Promising perspectives include:

- the study of **Quasi-Monte Carlo** methods using Sobol or Halton sequences to accelerate convergence;
- variance reduction techniques (stratified sampling or importance sampling);
- and the adaptation of these methods to multidimensional integrals encountered in finance or machine learning.

## 5 Conclusion

The experiments confirm that:

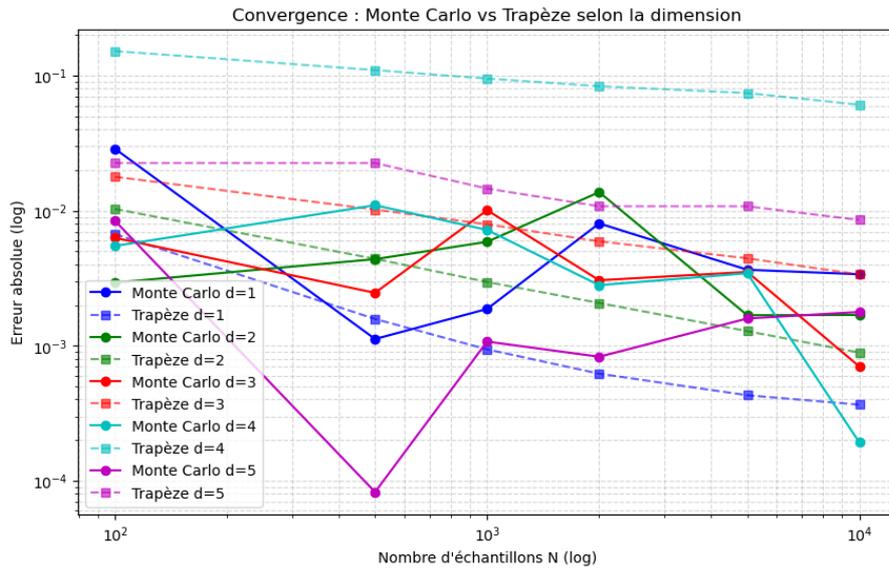


Figure 1: Convergence of absolute error by dimension ( $d = 1$  to  $5$ ).

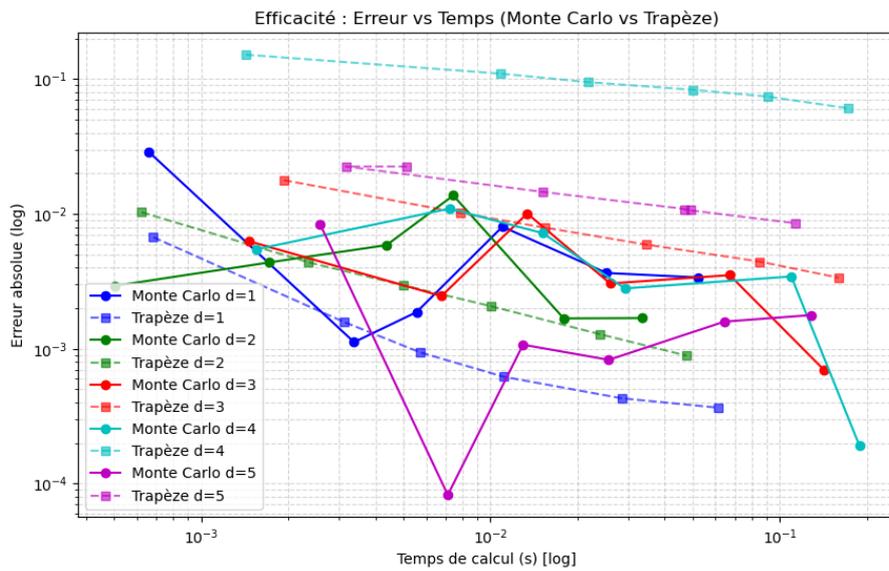


Figure 2: Time efficiency for dimensions  $d = 1$  to  $5$ .

1. Newton–Cotes methods (trapezoidal, Simpson) achieve higher order but are sensitive to dimension and singularities;
2. The Monte Carlo method, though of lower order, remains stable, robust, and scalable;
3. Time efficiency illustrates the *curse of dimensionality*: the cost of quadrature grows exponentially

---

with  $d$ , unlike Monte Carlo.

In conclusion, the Monte Carlo method stands out as a universal tool for high-dimensional numerical integration, while deterministic methods retain their relevance for low-dimensional and highly regular problems.

## References

- [1] Metropolis, N. & Ulam, S. (1949). *The Monte Carlo Method*. Journal of the American Statistical Association.
- [2] Dahlquist, G. & Björck, A. (2008). *Numerical Methods*. Dover Publications.
- [3] Müller, T. (2020). *Numerical Integration and Quadrature*. Springer.
- [4] Atkinson, K. (1989). *An Introduction to Numerical Analysis*. John Wiley & Sons.
- [5] Bellman, R. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton University Press.
- [6] Metropolis, N., & Ulam, S. (1949). *The Monte Carlo Method*. Journal of the American Statistical Association, 44(247), 335–341.
- [7] Dahlquist, G., & Björck, A. (2008). *Numerical Methods*. Dover Publications.
- [8] Niederreiter, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM.
- [9] Lemieux, C. (2009). *Monte Carlo and Quasi-Monte Carlo Sampling*. Springer.
- [10] Fishman, G. (1996). *Monte Carlo: Concepts, Algorithms, and Applications*. Springer.
- [11] Caflisch, R. (1998). *Monte Carlo and Quasi-Monte Carlo Methods*. Acta Numerica, 7, 1–49.
- [12] Quarteroni, A., Sacco, R., & Saleri, F. (2007). *Numerical Methods: Algorithms, Analysis and Applications*. Springer.
- [13] Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press.