Proceedings of Science and Mathematics
Faculty of Science, Universiti Teknologi Malaysia
Vol. 9, 2022, page 128-138

Application of Monte Carlo Simulation in Integration and Poisson's Equation

Lee Wei Cong*, Yeak Su Hoe
Department of Mathematical Sciences, Faculty of Science
Universiti Teknologi Malaysia, 81310 Johor Bahru, Malaysia
*Corresponding author: cong5035@gmail.com

Abstract
Monte Carlo simulation is a multiple probability simulation, and it is mathematical technique that applied to appraisal viable outcomes of an unsure event. When dealing with integration with low dimensional, Gauss quadrature is really a good mathematical approach to solve the problem numerically. However, this approach facing the high complexity when deal with very high dimension of integration hence it is required another mathematical approach to overcome this problem. Monte Carlo was a good tool to solve the high dimensional integration problem and Poisson's equation problem since it can avert the complicated derivation of integral with mathematically. The objective of this study is to conduct a study on application of Monte Carlo simulation in solving integration problem and Poisson's equation in one-dimensional and two-dimensional. The purpose of this study also investigates the role of random number generator in Monte Carlo simulation and mainly focused on the one-dimensional integration, two-dimensional integration, one-dimensional Poisson's equation, and two-dimensional Poisson's equation. In conducting the Monte Carlo simulation, Python was used as a programming tool to observe the solution since it can be extended with compiled code and come out with a strong scientific program. The study introduces the alternative methods that can use to solve one-dimensional, two-dimensional integration as well as one-dimensional and two-dimensional Poisson's equation. In the study, Monte Carlo as computational algorithm was used to solve the difficult problem that very hard solve by the other approaches. There are some theorems was applied in Monte Carlo integration such as central limit theorem. Apart from that, all the Monte Carlo method use in principle constructed with Random number because it is more easily implementable. From the result, it is clearly that Monte Carlo simulation is a very good approach when the simulation take place with higher number of nodes and random number generation point in Poisson's equation. At the same time, Monte Carlo simulation solves the integration problem efficiency as the solution getting more approximately to exact solution as the number of tries increase. By applying Monte Carlo simulation, the study brings a more effectively and efficiency approach to solve the higher dimension integration problem and Poisson's equation. Monte Carlo can be said that it can model the complex system that with coupled degree of freedom.

Keywords: Monte Carlo simulation; one-dimensional and two-dimensional integration; one-dimensional and two-dimensional Poisson's equation; random walk.

1. Introduction

The study of Monte Carlo simulation has been deeply investigated due to it's important in various fields. Monte Carlo methods has been recognized by the world and now become one of the top ten algorithms in applied mathematics. Monte Carlo also called as multiple probability simulation and it is mathematical technique that applied to appraisal viable outcomes of an unsure event. Monte Carlo method being use in solving the Poisson's equation. Monte Carlo sampling was used in compute the Poisson probabilities and it can be apply to any mixed Poisson distribution [1]. Thus, it can say that Monte Carlo was a good tool to solve the Poisson's equation because it can avert the complicated derivation of integral with

mathematically. Monte Carlo multigrid method was used to increase the efficiency of stochastic smoothers and the stochastic models are as known as the solution for the nonlinear partial differential equation [2]. In conducting the Monte Carlo simulation, Python is a very famous programming language that widely use by people. Python is a strong programming language that can extend the compiled code efficiency [3]. By this, it will be more efficiency if using Python as a programming tool to conduct the Monte Carlo simulation since it can be extended with compiled code and come out with a strong scientific program.

Monte Carlo can be said that it can model the complex system that with coupled degree of freedom [4]. Monte Carlo also can model all type of probability distributions and this method is widely used. When solving the integration numerically with low dimension, the result of Gauss quadrature is much accuracy than the Monte Carlo [5]. However, the result from Monte Carlo simulation is more accurate if the dimensional of integration increase or degree of freedom of the integral increase. Other than that, finite difference method (FDM) was a very good approach to solve the systems of partial differential equations (PDEs) numerically. However, it is not suitable to solve the partial differential equations by using finite difference method (FDM) contain any sort of stochastic component. In the case, Monte Carlo Simulation is more recommended to vary the stochastic components for the partial differential equations. All this numerical method was a good tool to solve numerical problems but of course there exist weakness and problem when applying for each of the method above. To overcome it, Monte Carlo method is introduced.

The main objective of the research is to study the roles of Monte Carlo and understand how it apply to solve the integration problem and partial differential equation. The study is focused on the one-dimensional integration, two-dimensional integration, one-dimensional Poisson's equation and two-dimensional Poisson's equation. In this study, Python Program is used to run the one-dimensional and two-dimensional integration. By applying Monte Carlo simulation, the study is hoped to bring a more effectively and efficiency approach to solve the higher dimension integration problem and Poisson's equation.

2. Literature Review

2.1. Monte Carlo Simulation of Integration Calculation

There are many of studies on Monte Carlo simulation being using in solving the integration problem. The uncertain data was being model by the Monte Carlo integration-based method to deal with the limitations for some unpredictability estimation empirically [6]. At most of the time, Monte Carlo simulation is used to solve the integration with a big scale of dimension and the large system size too. In solving the van der Waals density function vdW-DF, it needs to have a multi-dimension integration which Monte-Carlo technique is fulfilled this requirement [7]. Monte Carlo integration is said to be a very efficiency tool to handle with the various versions of van der Waals density function. In other words, Monte Carlo was likely to apply from low dimension problem till the complex structures. As the dimension of integration domain increase or infinite dimensionally, Monte-Carlo methods was recommended to solve the problem. Higher order Quasi-Monte Carlo integration was used to compute approximate Bayesian estimates for interest quantity [8]. Besides, multi-dimensional Monte Carlo integration also use on multiple Graphics Processing Units (GPUs). Multi-dimensional Monte Carlo integration was used on distributed multi–Graphics Processing Units devices [9].

2.2. Numerical Simulation of Poisson's Equation

There is various numerical simulation involve in solving Poisson equation. The different numerical simulation will undergoing different process and yield different value. One of the numerical simulations that can solve the Poisson equation is finite difference method (FDM). Besides, Poisson equation also can be solved by different type of finite difference method (FDM). Poisson equation on hierarchical Cartesian meshes was solved by using a cell centred FDM to reduce the truncation error coefficient which is in second order [10]. When there exists different boundary condition in solving the Poisson equation, different type of finite difference method will be use. Poisson equation with the Dirichlet

boundary condition was solved by Shortley-Weller method and finite difference method by Gibou et al. [11].

Other than finite difference method, Monte Carlo simulation also one of the numerical simulations that used to solve the Poisson equation. In solving the nonlinear Poisson equation, Monte Carlo simulation was used as technique for numerical solution such as fixed random walk, monotone iterative methods, and adaptive mesh [12]. For the two-dimensional (2D) Poisson equation and three-dimensional (3D) Poisson equation, quasi- Monte Carlo method was applied to avert the crisis of singularity and domain discretization [13]. When there exists different boundary condition such as Dirichlet boundary condition, Monte Carlo simulation was a very efficiency tool to solve the Poisson equation.

2.3. Application of Monte Carlo Simulation in Real Life Simulations

As one of the top mathematical algorithms in recent decade, Monte Carlo simulation is widely use around our real life. Monte Carlo simulation is a good methodology when simulate process that are consume time. Dynamic Monte Carlo (DMC) methodology was developed and utilized to achieve time dependent green's functions for transient analysis of the source-driven system [14]. At the same time, the ephemeral analysis of source driven nuclear systems was examined by using Dynamic Monte Carlo (DMC) methodology. In the existence of high-dimensional input uncertainties, multi-level Monte Carlo methods was used as a tool to perform uncertainty quantification for problems that depend on time [15].

Besides, Monte Carlo also known as a very effective tool when run many experiments in a very short timeframe. There is various of studies that had been applied the Monte Carlo algorithms when solving the large-scale systems due its computational efficiency. The computation of magnetization temperature dependence in big-scale granular thin films was completed by assisting of Monte Carlo algorithm [16]. Other than that, Monte Carlo also widely used in analysis especially for the large complex network. A distributed linear algebra solver was developed based on Monte Carlo at where it applies the random walks over the system matrix [17].

2.4. Random Number Generator

Monte Carlo simulation is based mainly on the random number generator as debugging aid by the intelligence to undergo the sane sequence of random number repeatedly by start from the same random seed. The application of randomness produces various of method to generate the random data or random walk. Thus, choosing an appropriate random number generator is important so that it can be optimally function in Mote Carlo simulation to solve the integration problem and Poisson's equation.

Generally, uniform distribution of pseudorandom number can be generated by the linear congruential generator. Suppose that *m*, *a*, and *b* is integer and the recurrence relation is shown as below:

$$x_{n+1} = (ax_n + b) \bmod m \tag{1}$$

where a represents the multiplier, b represents the increment and m represents the modulus.

3. MONTE CARLO SIMULATION IN POISSON'S EQUATION AND THE INTEGRATION

3.1. Introduction

In the study, Monte Carlo as computational algorithm was used to solve the difficult problem that very hard solve by the other approaches. Python Program is used to run the Monte Carlo simulation to obtain the result. In the integration problem, Monte Carlo contributes a solution that can solve problem with exponential increase in computation time. There are some theorems was applied in Monte Carlo integration such as central limit theorem. By this theorem, $\frac{1}{\sqrt{n}}$ convergence was displayed. Apart from that, Monte Carlo always has a close relation with random number. All the Monte Carlo method is said to be in principle constructed with Random number because it is more easily implementable. For the Monte Carlo method, the integral was computed by considering a function f on the interval [a, b] and it

compute as $I = \int_a^b f(x)dx$ for certain bounded function f. Thus, $f \geq 0$ can be assume if it is general problem. Otherwise, f will be replaced by $f + c$ at where c is a constant. Other than that, $f \leq 1$ can be assume if not f will be replaced by cf at where the constant c is sufficiently small. Next, a can be assumed is 0 (a=0) and b can be assumed as 1 (b=1). Otherwise, variable y will be change to $y = \frac{(x-a)}{(b-a)}$.

3.2. One-dimensional Monte Carlo Integration

To deal with the Monte Carlo simulation in solving integration problem, the one-dimensional Monte Carlo equation was derived.

$$\int_a^b f(x)dx = (b-a)\frac{1}{N}\sum_{i=1}^{N} f(x_i) \tag{2}$$

where *N* is the simulation node.

3.2.1. Framework of One-dimensional Monte Carlo Integration

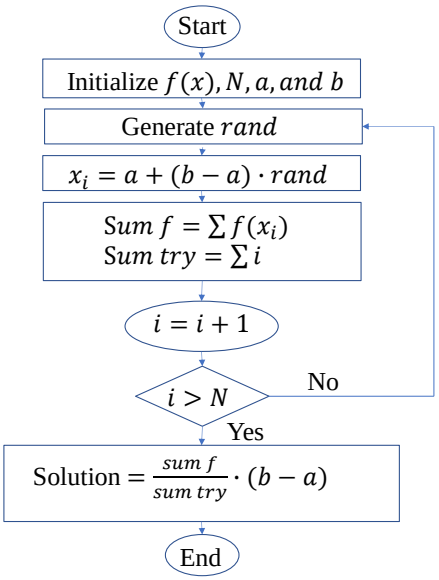

Figure 3.1 Framework of one-dimensional Monte Carlo integration

3.3. Two-dimensional Monte Carlo Integration

After deal with one-dimensional Monte Carlo integration, we investigate the role of Monte Carlo integration in solving higher dimensional integration problem. Therefore, the study proceeds with Monte Carlo simulation in solving two-dimensional integration problem. The two-dimensional Monte Carlo integration equation is derived as follows:

$$\int_c^d \int_a^b f(x,y)dxdy = (b-a)(d-c)\frac{1}{N_i N_j}\sum_{j=1}^{N_j}\sum_{i=1}^{N_i} f(x_i, y_j) \tag{3}$$

where i = 1,2,3, …N_i ; j=1,2,3, …N_j

3.3.1 Framework Two-dimensional Monte Carlo Integration

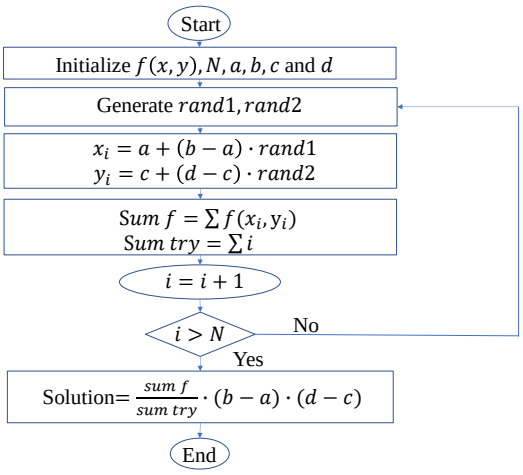

Figure 3.2 Framework of two-dimensional Monte Carlo integration

3.4. Monte Carlo in Solving One-dimensional Poisson's Equation

Beside integration problem, Monte Carlo method also play a role in solving partial differential equation. In this study, Monte Carlo method was used to solve the one-dimensional Poisson's equation and the result was observed to investigate the accuracy of this method. Therefore, the Monte Carlo simulation was introduced and applied in Poisson's equation as below:

$$\phi(x) = -f \cdot (h^2) + \frac{\phi(x-h)}{2} + \frac{\phi(x+h)}{2} \tag{4}$$

For applying the Monte Carlo simulation in solving Poisson's equation, random walk (RW) is the concept that we should understand and use it. Assume that domain Ω of φ is in a rectangular lattice where there is a distance *h* for each lattice point from the neighbours. Then, we start to move randomly from a point in the lattice where we denoted the point as *A*. The moving of the point is to another point that is closest neighbours of that point with the step with equal probability. This movement was repeated until reach the boundary point *a* of the domain. Then, that point is the point where the random walk *w* calculates. In this case, the term F is as follows:

$$F(w) = \sum_{\substack{i \in interior\ points\ in \\ random\ walk}} f(i)h^2 \tag{5}$$

The random walk was repeated for a finite number of times (*N*). After the walk concluding at each boundary point *a*, the probability of the walk that starting at *A* to the boundary point *a* was calculated and denoted as $P_A(a)$. The equation to calculate the $P_A(a)$ is as follows:

$$P_A(a) = \frac{Number\ of\ times\ random\ walk\ ends\ at\ a}{N} \tag{6}$$

Therefore, the quantity $u(A)$ can be compute by using the boundary condition *g* at each boundary point *a.* The equation is as follows:

$$u(A) = \sum_{a \in boundary\ points} g(a)\, P_A(a) - \sum_{for\ all\ w} F(w) \tag{7}$$

Lastly, *u* for all points that in the domain was calculated and this was the approximate solution to the differential equation.

$$\varphi(A) \approx u(A) \;\; A \in Lattice\ points\ in\ \Omega \tag{8}$$

3.4.1. Framework of Monte Carlo in Solving One-dimensional Poisson's Equation

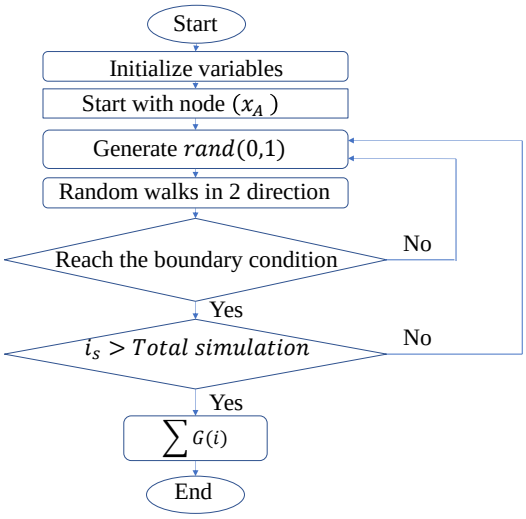

Figure 3.3 Framework of Monte Carlo in solving one-dimensional Poisson's equation

3.5. Monte Carlo in Solving Two-dimensional Poisson's Equation

In this study, higher dimensional Poisson's equation was solved by using Monte Carlo method. The accuracy of Monte Carlo simulation in solving two-dimensional Poisson's equation was investigated and specified Poisson equations was used in this study as shown as below:

$$\phi(x,y) = -f \cdot \frac{(2h^2k^2)}{2(k^2+h^2)} + \frac{k^2\big(\phi(x-h,y)+\phi(x+h,y)\big)+h^2\big(\phi(x,y-k)+\phi(x,y+k)\big)}{2(k^2+h^2)} \tag{9}$$

In solving two-dimensional Poisson's equation, we start to move randomly from a point in the lattice where we denoted the point as (x_A, y_A). The moving of the point is to another point that is closest neighbours of that point with the step with equal probability where it is similarly as in one-dimensional Poisson' equation.This movement was repeated until reach the boundary point *a* of the domain. Then, that point is the point where the random walk *w* calculates. The random walk was repeated for a finite number of times (*N*). After the walk concluding at each boundary point *a*, the probability of the walk that starting at (x_A, y_A) to the boundary point *a* was calculated and denoted as $P_{x_A,y_A}(a)$. The equation to calculate the $P_{x_A,y_A}(a)$ is as follows:

$$P_{x_A,y_A}(a) = \frac{Number\ of\ times\ random\ walk\ ends\ at\ a}{N} \tag{10}$$

Therefore, the quantity $u(x_A, y_A)$ can be compute by using the boundary condition *g* at each boundary point *a.* The equation is as follows:

$$u(x_A,y_A) = \sum_{a \in boundary\ points} g(a)\, P_{x_A,y_A}(a) - \sum_{for\ all\ w} F(w) \tag{11}$$

Lastly, *u* for all points that in the domain was calculated and this was the approximate solution to the differential equation.

$$\varphi(x_A, y_A) \approx u(x_A, y_A) \;\; A \in Lattice\ points\ in\ \Omega \tag{12}$$

3.5.1. Framework of Monte Carlo in Solving One-dimensional Poisson's Equation

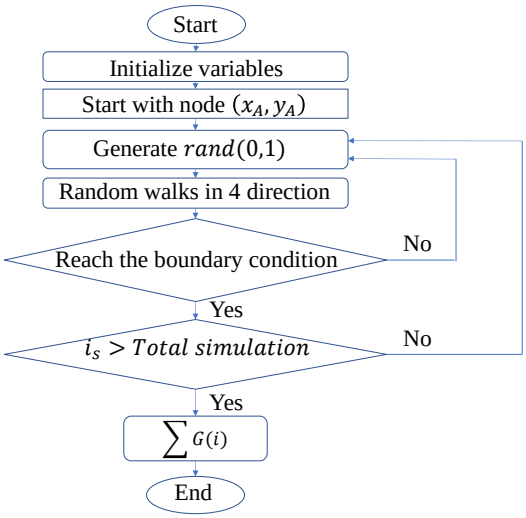

Figure 3.4 Framework of Monte Carlo in solving one-dimensional Poisson's equation

4. Results and discussion

4.1. Result of One-dimensional Monte Carlo Integration

For one-dimensional Monte Carlo integration, the result yield from the Python Programming was likely approximate to the exact solution. When the number of tries increase, the result from the Monte Carlo integration is more likely approximate to the exact solution. In solving one-dimensional Monte Carlo integration for following equation:

$$\int_0^{\pi} \frac{(\sin(\sqrt{x}) + 1)(e^{\sqrt{x}})}{\sqrt{x}}\, dx \tag{13}$$

To compare the results of Monte Carlo simulation in solving one-dimensional integration problem, different number of tries was set in the Python Programming for error observation.

Table 4.1 Comparison of average absolute error for 10 simulations with different number of tries

Number of tries	Average absolute error
1000	0.129105
10000	0.060328

Through the comparison, it can clearly state that the average absolute error for 10000 tries is much smaller than the average absolute error for 1000 tries. Meanwhile, this indicate that the accuracy of Monte Carlo simulation in solving integration problems is higher when the number of tries increase.

4.2. Result of Two-dimensional Monte Carlo Integration

As mentioned as previously, Monte Carlo simulation is a very good tool to solve the higher order integration problem. When the integration problem is in higher order or dimension, Monte Carlo simulation can provide a greater solution with smaller absolute error.

In this session, the result will be display in visual or physical observation as it is easier for us to compare it with different number of tries. Before that, the exact solution for this two-dimensional Monte Carlo integration problem is as shown as below:

$$\int_0^{\pi}\int \frac{(\sin(\sqrt{xy})+1)(e^{\sqrt{xy}})}{\sqrt{xy}}\,dx\,dy = 16.978388 \tag{14}$$

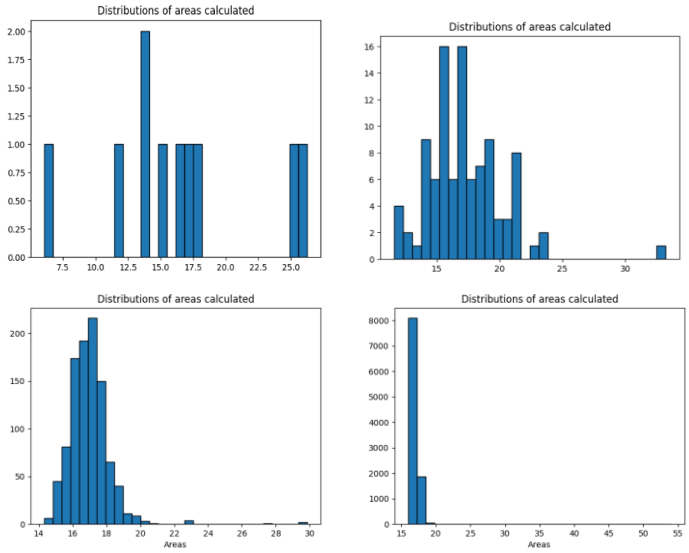

Figure 4.1 Value of two-dimensional Monte Carlo integration simulation with different tries

From the result, it is significantly show that the value generated by two-dimensional Monte Carlo integration simulation for 10000 tries is likely approximate to the exact solution. From Figure 4.4, it shows that all the value of Monte Carlo simulation lies between 15 to 20 which is in the range of value that approach to the exact solution. From the result, the domain of histogram for Figure 4.4 was narrow down as the range lies between the range of 15 to 20. Therefore, it can be described that the graph result from normal distribution that standard deviation was reduce. Hence, Monte Carlo simulation can be said as a very good tool to solve the higher order integration efficiently with higher number of tries.

4.3. Result of Monte Carlo in Solving One-dimensional Poisson's Equation

The comparison for the average of integration of error surface was computed to ensure the accuracy of the result in solving one-dimensional Poisson's equation for following equation:

$$\frac{\partial^2 u}{\partial x^2} = \mathrm{f}(x) = x^2 + 2x + 10 \tag{15}$$

The integration of error surface and average integration of error surface was computed for 10 simulations, and it was computed with different number of nodes and random number generation points.

Table 4.2 Comparison of average integration of error surface for *i*=10 with different *n* and 1000 random number generation points in 1D Poisson's equation

Random Number Generation Points	Number of Nodes	Average of Integration of Error Surface
1000	10	0.360972
1000	20	0.260005

Table 4.3 Comparison of average integration of error surface for *i*=10 with different *n* and 10000 random number generation points in 1D Poisson's equation

Random Number Generation Points	Number of Nodes	Average of Integration of Error Surface
10000	10	0.264593
10000	20	0.231988

Table 4.4 Comparison of average of integration of error surface for *i*=10 with *n*=10 and different random number generation points in 1D Poisson's equation

Number of Nodes	Random Number Generation Points	Average of Integration of Error Surface
10	1000	0.360972
10	10000	0.264593

Table 4.5 Comparison of average of integration of error surface for *i*=10 with n=20 and different random number generation points in 1D Poisson's equation

Number of Nodes	Random Number Generation Points	Average of Integration of Error Surface
20	1000	0.260005
20	10000	0.231988

From the comparison, the average of integration of error surface is smaller when the random number generation points are higher for fixed number of nodes, 20 nodes where the situation is similar with the fixed number of nodes at 10 nodes. Therefore, the result show that when number of nodes is same, the random number generation points will affect the average of integration of error surface. At the same time, the higher the random number generation points the smaller the average of integration of error surface.

4.4. Result of Monte Carlo in Solving Two-dimensional Poisson's Equation

For two-dimensional Poisson's equation, the following equation was used to compete the result for comparison:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y) = e^{-2x} + e^{-y} + 10 \tag{16}$$

The computational of result was varying with the number of nodes and random number generation points to perform a better view of comparison.

Table 4.6 Comparison of average integration of error surface for *i*=10 with different *n* and 1000 random number generation points in 2D Poisson's equation

Random Number Generation Points	Number of Nodes	Average of Integration of Error Surface
1000	10	0.026764
1000	20	0.025577

Table 4.7 Comparison of average integration of error surface for *i*=10 with different *n* and 10000 random number generation points in 2D Poisson's equation

Random Number Generation Points	Number of Nodes	Average of Integration of Error Surface
10000	10	0.026164
10000	20	0.025437

Table 4.8 Comparison of average of integration of error surface for *i*=10 with *n*=10 and different random number generation points in 2D Poisson's equation

Number of Nodes	Random Number Generation Points	Average of Integration of Error Surface
10	1000	0.026764
10	10000	0.026164

Table 4.9 Comparison of average of integration of error surface for *i*=10 with *n*=20 and different random number generation points in 2D Poisson's equation

Number of Nodes	Random Number Generation Points	Average of Integration of Error Surface
20	1000	0.025577
20	10000	0.025437

From all the result above, we noticed that when the number of tries increase, the result generated from the Monte Carlo simulation is more approximately to the exact solution for the integration problem. Similarly, the average of integration of surface error is smaller in solving Poisson's equation when the number of nodes is big and the random number generation points is more.

Conclusion

The study conducts successfully with the comparison of average absolute error and Monte Carlo integration value for Monte Carlo simulation in 1D and 2D integration problem. The study also investigates the role on Monte Carlo simulation in solving Poisson's equation in 1D and 2D by observing the average integration of error surface. From the study, Monte Carlo simulation provide a good solution for the integration problem especially when the dimensional of integration or degree of freedom is high. Monte Carlo also a mathematical approach that depends on random number generator. Therefore, the study also investigated the role of number of tries in solving integration problem and random number generation point in Poisson's equation. From the result, the absolute error is small when the number of tries is high in both 1D and 2D integration. Similarly, the integration of error surface in Poisson's equation is small when the random number generation point is high. The integration of error surface in both 1D and 2D Poisson's equation is small when the number of nodes increase. Thus, Monte Carlo simulation is a mathematical approach that very suitable to solve the problem that is complex and high degree of freedom by increase the number of tries and random number generation point. Monte Carlo can be said is a mathematical algorithm that suitable to solve the integration problem when the degree of integration for the problem is high and the complexity of problem is hard to solve by other mathematical approach. Monte Carlo simulation should highly recommend to widely use in solving the Poisson's equation as it solves the Poisson's equation efficiently and the result is approaching to the exact solution especially when the Poisson's equation is in high dimensional.

Acknowledgement
The researcher would like to express deeply appreciation to all people who have supported the research and given the support in mental or physical form.

References

[1] Ong, S. H., Lee, W. J., & Low, Y. C. 2020. A general method of computing mixed Poisson probabilities by Monte Carlo sampling. *Mathematics and Computers in Simulation*, *170*, 98–106.
[2] Marshall, G. 1989. Monte Carlo methods for the solution of nonlinear partial differential equations. *Computer Physics Communications*, *56*(1), 51–61.
[3] Nilsen, J. K. 2007. MontePython: Implementing Quantum Monte Carlo using Python. *Computer Physics Communications*, *177*(10), 799–814.
[4] Smid, J., Verloo, D., Barker, G., & Havelaar, A. 2010b. Strengths and weaknesses of Monte Carlo simulation models and Bayesian belief networks in microbial risk assessment. *International Journal of Food Microbiology*, *139*, S57–S63.
[5] Bender, E. A., & Canfield, E. 1999. An Approximate Probabilistic Model for Structured Gaussian Elimination. *Journal of Algorithms*, *31*(2), 271–290.
[6] Sharma, K. K., & Seal, A. 2019. Modeling uncertain data using Monte Carlo integration method for clustering. *Expert Systems with Applications*, *137*, 100–116.
[7] Nabok, D., Puschnig, P., & Ambrosch-Draxl, C. 2011. noloco: An efficient implementation of van der Waals density functionals based on a Monte-Carlo integration technique. *Computer Physics Communications*, *182*(8), 1657–1662.
[8] Dick, J., Gantner, R. N., le Gia, Q. T., & Schwab, C. 2019. Higher order Quasi-Monte Carlo integration for Bayesian PDE Inversion. *Computers & Mathematics with Applications*, *77*(1), 144–172.
[9] Wu, H. Z., Zhang, J. J., Pang, L. G., & Wang, Q. 2020. ZMCintegral: A package for multi-dimensional Monte Carlo integration on multi-GPUs. *Computer Physics Communications*, *248*, 106962.
[10] Raeli, A., Bergmann, M., & Iollo, A. 2018. A finite-difference method for the variable coefficient Poisson equation on hierarchical Cartesian meshes. *Journal of Computational Physics*, *355*, 59–77.
[11] Yoon, G., & Min, C. 2015. Analyses on the finite difference method by Gibou et al. for Poisson equation. *Journal of Computational Physics*, *280*, 184–194.
[12] Li, Y., Lu, H. M., Tang, T. W., & Sze, S. 2003. A novel parallel adaptive Monte Carlo method for nonlinear Poisson equation in semiconductor devices. *Mathematics and Computers in Simulation*, *62*(3–6), 413–420.
[13] Chen, C., Golberg, M., & Hon, Y. 1998. Numerical justification of fundamental solutions and the quasi-Monte Carlo method for Poisson-type equations. *Engineering Analysis with Boundary Elements*, *22*(1), 61–69.
[14] R. Maleki, B., Goluoglu, S., & Tombakoglu, M. 2021. Time-dependent Green's function generation with Monte Carlo for transient analysis of source-driven systems. *Progress in Nuclear Energy*, *137*, 103744.
[15] ben Bader, S., Benedusi, P., Quaglino, A., Zulian, P., & Krause, R. 2021. Space-time multilevel Monte Carlo methods and their application to cardiac electrophysiology. *Journal of Computational Physics*, *433*, 110164.
[16] Lepadatu, S., Mckenzie, G., Mercer, T., MacKinnon, C. R., & Bissell, P. R. 2021. Computation of magnetization, exchange stiffness, anisotropy, and susceptibilities in large-scale systems using GPU-accelerated atomistic parallel Monte Carlo algorithms. *Journal of Magnetism and Magnetic Materials*, *540*, 168460.
[17] Magalhães, F., Monteiro, J., Acebrón, J. A., & Herrero, J. R. 2022. A distributed Monte Carlo based linear algebra solver applied to the analysis of large complex networks. *Future Generation Computer Systems*, *127*, 320–330.