

Estimation of Monte Carlo Area Algorithm Error

Osama Y. Abu Tammam*

Department of Mathematical Modeling, Palestine Technical University-Kadoorie, Tulkarm, Palestine

Abstract

The paper introduces Monte Carlo Area Algorithm using Mat lab simulation to generate uniform random distribution points and evaluate the area accordingly and analyze the error. The method to calculate the error had done by comparing the real area value and estimated area values the repetition of the algorithm introduce superior efficiency and reduce the difference between real area value and estimated which considered as error. The relation between Error and number of random points is obviously an exponential relation which affected by number of repetition and random points. The true challenge in such studies is to limit the deviation in the exponential relation between error and number of random points.

Keywords: Mat lab • Random • Exponential relation • Monte Carlo

Introduction

Monte Carlo methods refers to a class of methods to solve mathematical problems using random samples. A straightforward example is the computation of the expectation value of a random variable; instead of computing the expectation value according to the definition (which may involve solving complex integrals) we observe the behavior of the random variable, i.e., we collect samples, and estimate its expectation value based on these samples. However, Monte Carlo methods may also be used for solving deterministic problems. This might seem odd at a first glance, but the idea is simply to find a random variable the statistic properties of which is depending on the solution to the deterministic problem. An example of this would be an opinion poll. Assume that there is going to be an election. If we ignore that people may change their preferences over time and consider just one specific point of time then the share of voters who are planning to vote for a certain candidate is deterministic. However, in order to compute the true value of this share, we would have to ask all voters which candidate they favour. An alternative would be to ask a limited number of randomly chosen voters and use these samples to estimate the share of votes the candidate will obtain [1].

Materials and Methods

Monte Carlo (MC) technique is a numerical method that makes use of random numbers to solve mathematical problems for which an analytical solution is not known. The first article, "The Monte Carlo Method" by Metropolis and Ulam, has appeared for the first time in 1949.

Speaking very broadly, Monte Carlo methods are tools for solving problems using random numbers. Although this might sound somewhat specific and not very promising, Monte Carlo methods are fundamental tools in many areas of modern science (ranging all the way from theoretical physics to political science) [2].

One of the earliest applications of random numbers was numerical computation of integrals, which is actually non-random (deterministic problem). Computing integrals with the aid of random numbers is known as

Monte Carlo integration and is one of the most powerful and widely used mathematical technique through science and engineering.

Monte Carlo (MC) methods are often used when a closed-form solution for a property being studied cannot be developed. Mathematically, we are attempting to evaluate $\int_{\Omega} p(\cdot) F(\cdot) d\Omega$ where $p(\cdot)$ is the combined probability function of all the input parameters, $F(\cdot)$ is the function describing the property being studied and Ω is the domain of interest [3,4].

Disposal a number of i.i.d. samples, x_1, x_N , drawn from $p(x)$. Then, the following approximation

$$E[f(x)] \cong \frac{1}{N} \sum_{i=1}^N f(x_i) - E_{f,N} \quad (1)$$

is justified by (a) the law of large numbers and (b) the central limit theorem [5-8]. Let us denote as $E[f(x)] = \mu_f$ and the respective variance as

$$\text{var}[f(x)] := E(f(x) - E[f(x)])^2 = \sigma^2 \quad (2)$$

Then, the previously referred two theorems guarantee that the limit in Eq. (2) refers to the notion of almost sure convergence, that is

$$\text{prob} \left\{ \lim_{N \rightarrow \infty} \left| \mu_f - \frac{1}{N} \sum_{i=1}^N f(x_i) \right| = 0 \right\} = 1 \quad (3)$$

The covariance and the correlation of two random variables:

$$\text{Cov}[x, y] = E[(x - E[x])(y - E[y])] = E[xy] - E[x]E[y]$$

$$\text{corr}[x, y] = \frac{\text{cov}[x, y]}{2\sqrt{\text{var}[x]\text{var}[y]}} \quad (4)$$

Mathematical properties for Monte Carlo Algorithm was presented by authors [9,10]

1. If the variance of $f(x)$ is finite, the MC estimate is consistent
2. The MC estimate is asymptotically unbiased
3. The MC estimate is asymptotically normally distributed
4. The standard deviation of the MC estimate is given by $\sigma = \frac{2\sqrt{\text{var}(f(x))}}{2\sqrt{n}}$

Computing areas by throwing random points

Think of some geometric region G in the plane and a surrounding bounding box B with geometry $[x_L, x_H] \times [y_L, y_H]$. One way of computing the

Address for Correspondence: Osama Y. Abu Tammam, Department of Mathematical Modeling, Palestine Technical University-Kadoorie, Tulkarm, Palestine; Tel: 970 594319982; E-mail: osamaabutammam@hotmail.com

Copyright: © 2022 Tammam OYA. This is an open-access article distributed under the terms of the creative commons attribution license which permits unrestricted use, distribution and reproduction in any medium, provided the original author and source are credited.

Received: 20-Apr-2020, Manuscript No. JCSB-20-9464; **Editor assigned:** 23-Apr-2020, PreQC No. JCSB-20-9464; **Reviewed:** 07-May-2020, QC No. JCSB-20-9464; **Revised:** 24-Aug-2022, QI No. JCSB-20-9464; Manuscript No. JCSB-20-9464; **Published:** 31-Oct-2022, DOI: 10.37421/0974-7230.2022.15.417

area of G is to draw N random points inside B and count how many of them, M, that lie inside G. The area of G is then the fraction M/N (G's fraction of B's area) times the area of B $(x_L - x_H)(y_L - y_H)$. Phrased differently, this method is a kind of dart game where you record how many hits there are inside G if every throw hits uniformly within B.

Let us formulate this method for computing the integral $\int_a^b f(x)dx$

The important observation is that this integral is the area under the curve $y = f(x)$ and above the x axis, between $x = a$ and $x = b$. We introduce a rectangle B,

$$B = \{(x, y) \mid a \leq x \leq b, 0 \leq y \leq m\}, \quad (5)$$

where $m \leq \max_{x \in [a,b]} f(x)$. The algorithm for computing the area under the curve is to draw N random points inside B and count how many of them, M, that are above the x axis and below the $y = f(x)$ curve, see Figure 1. The area or integral is then estimated by

$$\frac{M}{N} m(b-a) \quad (6)$$

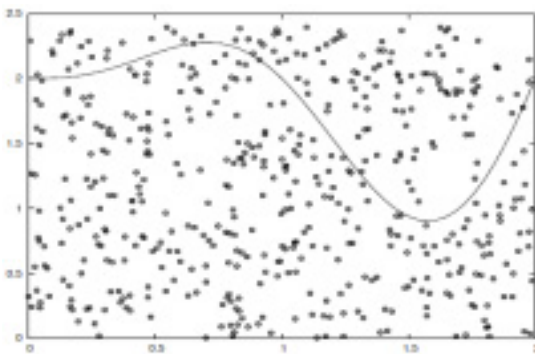


Figure 1. The area or integral of curve.

Results and Discussion

Two counters will be necessary: One to count the total points generated and a second to count those points that lie below the curve. You can then calculate an approximate value for the area under the curve by the following formula:

$$\frac{\text{area under the curve}}{\text{area of rectangular}} \approx \frac{\text{number of points counted below curve}}{\text{total number of random points}} \quad (7)$$

However, I will calculate the area under the curve many times and find the mean value of the approximated areas to reduce the error of the area to its maximum possible [7-9].

Monte Carlo area algorithm and flow chart

The following algorithm gives the sequence of calculations needed for a general computer simulation of this Monte Carlo technique for finding the area under a curve.

Define the domain of x values which is between a and b, and the domain of y values which is between 0 and M.

Generate random numbers for n times. The random numbers is i uniformly distributed and consider it as x values. The random numbers are bounded between a and b values.

Generate random numbers for n times. The random numbers is i uniformly distributed and consider it as y values. The random numbers are bounded between 0 and M value.

Calculate $f(x_i)$ for all of the generated x values.

Check the condition of existence y_i value under $f(x_i)$ curve to increase the counter by 1, otherwise leave the counter as is.

Calculate the area which equal to $\frac{M(b-a) \text{COUNTER}}{n}$

Irritate the steps (2) to (6) to calculate the area for z times.

Calculate the irritated area. It equal to mean value of irritated areas from step

$$(7). \text{ So the formula of output is } \frac{\sum_{i=1}^z \text{Area}(i)}{z}$$

The following flowchart clarify Monte Carlo for estimating the area under a curve (Figure 2).

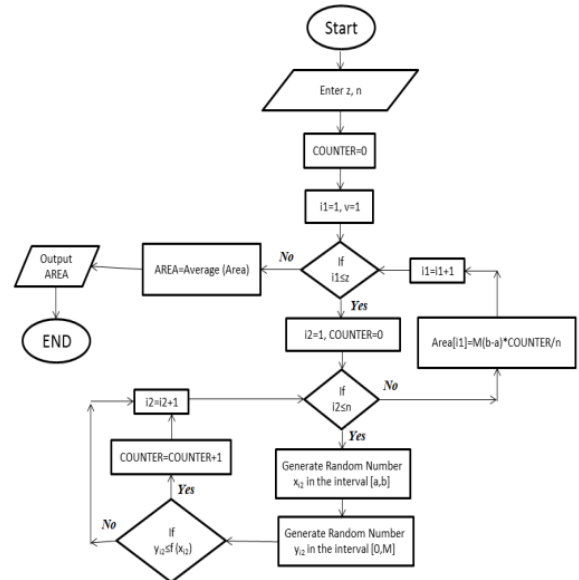


Figure 2. Monte Carlo area estimation flow chart.

Example solved by Mat lab

I considered the sine wave curve is the required curve to calculate the area under within surrounding bounding box B with geometry $[0,4] \times [0,1]$. It is clear that the area under the curve is equal to "4".

By estimation the area and compare it with the real value, so we can understand the mathematical relation between the real value and estimated using Monte Carlo Algorithm, so we can estimate the error [10].

Estimating the area can be done by checking random x_i and y_i for n times. But the estimated the area is done calculated by repeating the Monte Carlo algorithm for repetition z times and averaging it as shown in equation (8)

$$\text{Estimated Area} = \frac{\sum_{i=1}^z \text{Area}(i)}{z} \quad (8)$$

While area (i) can be calculated from equation (7)

Matlab Code:

```
clc
z=10;
for c1= 1:z
n=3000;
counter=0;
a=0;
b=4*pi;
M=1;
X=rand(n,1)*(b-a)+a;
Y=rand(n,1)*M;
for i= 1:n
```

```

F(i)=sin(X(i));
if (Y(i)<F(i))
counter=counter+1;
end
end
Area (c1)=(M*(b-a)*counter)/n;
end

```

We can calculate the error by using equation (9)

Error = | Real Area – Estimated Area | (9)

Where Estimated Area is calculated by equation (8)

Error analysis

The relation between error and number of random points shown in Figure 2 when $z=10$, Figure 3 when $z=500$ and Figure 4 when $z=1000$. These figures are showing the best fit curve too. Where x-axis represent the number of random points and y-axis represent the error which we get from equation (8).

Figures 5 are plotted through Mat lab and fitted through also. We get the next fitting curve equation (10).

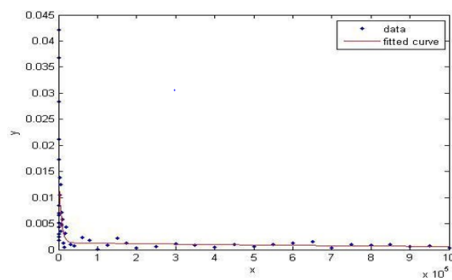


Figure 3. when $z=10$. Note : (●) data, (—) fitted curve.

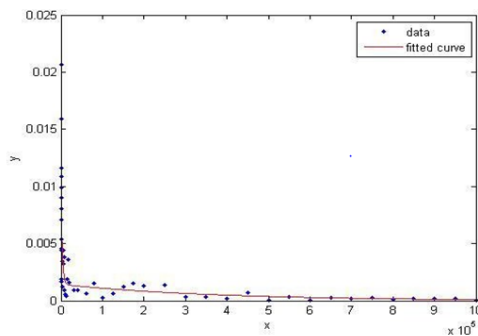


Figure 4. When $z=500$. Note : (●) data, (—) fitted curve.

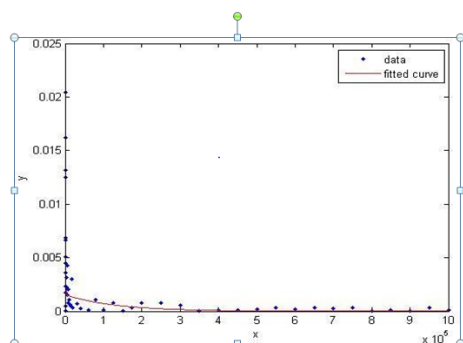


Figure 5. When $z=1000$. Note : (●) data, (—) fitted curve.

$$f(x) = ae^{bx} + ce^{dx} \quad (10)$$

As an example for the detailed coefficients we can see the below coefficients for repetition $z = 500$ with 95% confidence bounds.

$$a = 0.001432 \text{ } (-0.0007915, 0.003655)$$

$$b = -2.682 \times 10^{-6} \text{ } (-1.247 \times 10^{-5}, 7.109 \times 10^{-6})$$

$$c = 0.008694 \text{ } (0.005656, 0.01173)$$

$$d = -0.0003316 \text{ } (-0.0006954, 3.227 \times 10^{-5})$$

So we can write the equation of fitting curve for the relation between error and number of random points “n” when repetition is equal to 500 as shown in equation (11).

$$f(x) = 0.001432e^{-2.682 \times 10^{-6}x} + 0.008694e^{-0.0003316x} \quad (11)$$

Table 1 show the coefficients values with 95% confidence bounds for various z (number of repetition).

Table 1. Coefficients of exponential fitting curves for various number of repetition “z”.

z=10	a	Range
	b	Range
	c	Range
	d	Range
z=100	a	Range
	b	Range
	c	Range
	d	Range
z=500	a	Range
	b	Range
	c	Range
	d	Range
z=1000	a	Range
	b	Range
	c	Range
	d	Range

Table 2 show the error values for various repetition number and same number of random points ($n=100,000$).

Table 2. Error results incase random of points $n=100,000$ and number or repetition varies (10-1000).

n=100,000		
Error	z	Number of error points
0.00014	1,000	14.1
0.00029	900	29
0.00044	800	43.7
0.00023	700	23.2
2.6E-05	600	2.6
0.00024	500	24.1
3.6E-06	400	0.4
0.00021	300	21.2
0.00129	200	128.5
8.8E-05	100	8.8
0.0009	80	90
0.00129	60	129.2
0.00216	40	216.3
0.00314	20	313.7
0.00188	10	188.4

We notice the difference in error if we compare the small values of z such as 10, 20 or 80 with larger values such as 1000, 900 or 800. But the error sometimes varies with un-understood way due to probabilistic behaviour of the algorithm.

We can plot the previous table as shown in Figure 6 below

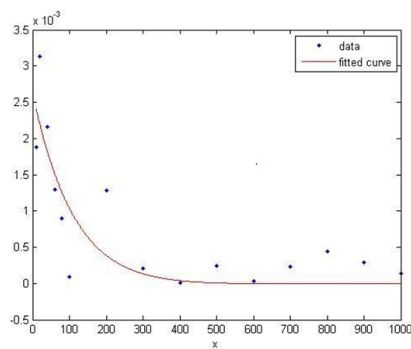


Figure 6. $n=100,000$ and number of repetition “z” varies and represented in x-axis. Note : (●) data, (—) fitted curve.

Error is represented in y-axis.

Figures 6 are plotted through Mat lab and fitted through also. We get the next fitting curve equation.

$$f(x) = 0.02212e^{-0.007468x} + -0.01948e^{-0.00724x} \quad (12)$$

In case we would to determine the error to be less than a value, difference between estimated area and actual area to be four digits and less, we can use equation (13) to determine the number of random points and repetitions.

$$a ebx + c edx \leq E_p \quad (13)$$

Where E_p is the preferred error.

Substituting equation (12) in the right hand side of equation (13) and considering $E_p = 0.001$ lead to evaluate x to be equal to 104.

It means that at random points $n=100,000$ and repetition $z=104$ the expected error of Monte Carlo Area Algorithm is equal to 0.001 and any increment in the number of repetition will decrease the error. So the change in the error value will appear in the fourth digit and above incase $z = 104$ and above.

By same way, to calculate when the change in the error start in the fifth digit and above, we consider $E_p=0.0001$ which lead to evaluate number of repetition to be equal to 326 at 100,000 random points [11].

Discussion

The authors described a modified sampling procedure for Monte-Carlo simulation of communication systems. The modified sampling technique leads to a considerable amount of savings in the sample size required to estimate low error rates in communication systems. But the authors in showed that classical MC fails to perform BER estimation in the region of high SNR.

This open a discussion about the efficiency to analyze the results of a modified Monte-Carlo simulation technique for the evaluation of error rate in Digital Communication Systems for various number of random points and repetition. The author in discussed the major issues in classic econometric inference and defined Monte Carlo experiments as an experimental evidence on the actual characteristics of the finite sample distribution of estimators and tests for relevant parameter values can always be obtained, avoiding the complexities of analytical derivations. The author mentioned the limitations of Monte Carlo experiment and considered it non asymptotic regarding the size of the (econometric) sample n , but regarding the size of the sample of experiments performed in the Monte Carlo simulation, say R . He supposed that in case applying Monte Carlo Algorithm with large R , the asymptotic approximations involved in a Monte Carlo analysis can be made

very accurate and the remaining approximation errors can be controlled and checked very well [12-15].

Conclusion

The error in Monte Carlo Area algorithm has an exponential relation with number of random points” which is called n in our study. The error is decreasing every time we increase the number of repetition of Monte Carlo Algorithm “which is called z in our study” for the same number of random points. And we noticed that incase random points equal to 100,000 and repetition equal to 104, the variations in the error is equal to 1×10^{-5} and less. While it equals to and less in case repetition $z = 326$ and random points $n=100,000$ for estimating the area under sine wave within the within surrounding bounding box B with geometry $(0, 4) \times (0, 1)$. The probabilistic behavior of Monte Carlo Algorithm make the error of our estimation unexpected. By repetition of Monte Carlo Algorithm we decrease the deviation of the error with our exponential fit curve. In applications that need to calculate the area for some converging shapes in their shape and area and more accurately, the error analysis process followed in this research will be useful to know the number of random points and the repetition process required to obtain the required accuracy.

References

- Amelin M. “Monte Carlo Simulation in engineering.” (2013).
- Langtangen, Hans Petter, and Hans Petter Langtangen. “A Primer on Scientific Programming with Python”. *Springer* (2011).
- Metropolis, Nicholas, and Stanislaw Ulam “The Monte-Carlo Method”. *J Amer Stat Assoc* 44 (1949): 335–341
- Giordano, Frank R., William P. Fox, and Steven B. Horton. A. “A First Course In Mathematical Modelling”. *Cengage Learning* 2013.
- Tim Brereton “Methods of Monte Carlo Simulation” 2015-2016
- Oberle William. “Monte Carlo Simulations: Number of Iterations and Accuracy”. *Army Research Lab Aberdeen Proving Ground MD Weapons Materials Research Directorate* 2015.
- Theodoridis S. “Monte Carlo Methods”. *Machine Learn* (2015)707–744.
- Papoulis A, Pilla SU. “Probability, Random Variables and Stochastic Processes”(1965) 583.
- Terejanu, Gabriel A. “Tutorial on Monte Carlo Techniques.” *Department of Computer Science and Engineering. University at Buffalo* (2009).
- James, Frederick. “Monte Carlo Theory and Practice”. *Rep Prog Phys* 43(1980):1147–1188.
- Saoudi, Samir, Thomas Derham, Tarik Ait-Idir, and Patrice Coupe. “A Fast Soft Bit Error Rate Estimation Method.” *Eurasip J Wirel Commun Netw* (2010): 1-13.
- Shanmugam K, Balaban P. “A Modified Monte-Carlo Simulation Technique for the Evaluation of Error Rate in Digital Communication Systems.” *Institute Electrical Electronics Engineers Trans Commun* 28 (1980): 1916-1924.
- Kiviet, Jan F. “Monte Carlo Simulation for Econometricians.” *Found Trends Econom* 5 (2012): 1-181.
- Matilainen, Kaarina, Mäntysaari E A, Lidauer M H, Strandén I, and Thompson R. “Employing a Monte Carlo Algorithm In Newton-Type Methods For Restricted Maximum Likelihood Estimation of Genetic Parameters.” *Plos one* 8 (2013): e80821.
- Zhang, Ziling, Cheng Qiang, Qi Baobao, and Tao Zhiqiang. “A General Approach For The Machining Quality Evaluation of S-Shaped Specimen Based on POS-SQP Algorithm and Monte Carlo Method.” *J Manuf Syst* 60 (2021): 553-568.

How to cite this article: Abu Tammam, Osama Y. “Estimation of Monte Carlo Area Algorithm Error” *J Comput Sci Syst Biol*. 15 (2022): 417.