

An Overview of Monte Carlo Algorithm

Shipra

Research Scholar (Mathematics), CMJ University, Shillong, Meghalaya

Abstract—Monte Carlo Method is used to represent a vast number of problem solving technologies which use probability statistics and random numbers. The methods of Monte Carlo are ways of using computers to solve crucial issues in a most improbable way. A Monte Carlo algorithm is also a computer algorithm and it is used to simulate the other system's behavior. It uses statistics and randomness to get an outcome. Thus this study discusses about the Monte Carlo methods, their major components, algorithms, benefits and drawbacks.

Index Terms— Monte Carlo Methods, Monte Carlo Algorithms, Components of Monte Carlo, Benefits and Drawbacks of Monte Carlo.

I. INTRODUCTION TO MONTE CARLO METHODS

According to Assaraf and Caffarel (1999) a Statistical Method which has been an occurrence for several years is the Monte Carlo Method. Using this method the speed of computation to arrive at a highly accurate solution can be slow since the accuracy is related to the square root of simulated particles. Because a numerous particle must be simulated it is not possible for a human being to resolve the issue by the Monte Carlo Method. Only after special computational technologies and computer science were discovered this method is used vastly in several fields and mostly in critical applications. To perceive this method the content of the Monte Carlo method must be known.

Contrary to that Janke (1998) has mentioned that Monte Carlo methods offer accurate solutions to different mathematical issues by performing experiments of statistical sampling. They can be referred loosely as statistical simulation methods where the statistical simulation is referred in normal terms to be any method that acquires random number sequences to execute the simulation. Thus Monte Carlo methods are a gathering of various methods that all execute the similar process basically. This process includes executing numerous simulations using probability and random numbers to get an accuracy of the response to the problem. The Monte Carlo Methods defining feature is its random number usage in its simulation. In facts these methods acquire their collective name from the fact that Monte Carlo the Monaco Capital has numerous casino roulette wheels and casinos are a better instance of a random number generator.

Similarly Stokes-Rees et al (2008) have described that since the early 1940s the Monte Carlo Simulation technology has occurred formally where it has research applications into nuclear fusion. However with the development in computer techniques and power has the technique become used more vastly. This is because now computers are able to execute millions of simulations much more quickly and effectively than before. This is an essential factor because it refers that the technique can offer an approximate response quickly and to a greater accuracy level because more simulations executed creates approximation more accurate. Conversely Sokal (1992) has described that the Monte Carlo methods offer only an approximation of the response. Thus the approximation error analysis is a main factor to take into account when estimating responses from these methods. The attempt to reduce these mistakes is the reason there are numerous Monte Carlo methods. The different methods can have various accuracy levels for their responses although this can rely on particular conditions of the question and some method's accuracy level differs relying on the issue.

II. HISTORY OF MONTE CARLO METHOD

Landau and Binder (2000) have described that in the community of materials the Monte Carlo method is an adaptation of a method used mainly to study the phase equilibrium statistical physics. During the Manhattan Project of II World War the term Monte Carlo was established by Metropolis (affected by the interest of Ulam in poker) because of the resemblance of statistical simulation to games of options and because Monte Carlo the Monaco capital was a gambling center. Now Monte Carlo defines to any method that acquires random number

sequences to execute statistical simulations. According to Radhakrishnan and Zacharia (1995) the major need to use the method of Monte Carlo for physical system simulation is that it must be applied to explain the system in probability density function terms also referred to as partition function (Z). Once the Z or a probability density function for a system is known then the simulation starts by random sampling from a probability density function and subsequently determining the desired sample properties by organizing some type of a trial. There must be a norm possible based on some cheap physical theory and/or mathematical theory to determine the result of such trial. Numerous trials are organized and all of these trials results are registered. In the Monte Carlo Method the last step is that the overall system behavior is acquired by computing the average of the trial results conducted.

Similarly Janke (1994) has mentioned that Monte Carlo methods are used in various manners for instance, as an integration technique of a function, as a component to estimate state properties, as a way to model stochastic processes such as T, E, V and P and as a model to simulate a system of interacting particles for example, ferromagnetic materials. However the Monte Carlo method in material science has been applied mainly to simulate the evolution of microstructure where the equilibrium exists on a local basis at best. Normally the system is from equilibrium and the kinetic study of the processes is attempted that lead to equilibrium time function for instance recrystallization or growth of grain. Although the model has assured to be helpful for numerous issues, it is essential to perceive that its capacity to simulate the physical behavior at the level of continuum is heuristic.

III. MAJOR COMPONENTS OF MONTE CARLO METHOD

Graves (2001) has mentioned that Monte Carlo method major components are the random number generator, probability density function, tallying, estimation of error and sampling. Each component is described in detail below:

The Monte Carlo Method continuously simulates the particle physical process migrating through the material. Every simulation is referred as a history. In every simulation the source particle has a probability density function to explain the position, direction distribution and energy. Every interaction of physics has a probability density function to explain the direction distribution and energy of discharging particles. In sampling these probability density functions a random number generator is used. The desired outcome is the number of histories average (Propp and Wilson, 1996).

Random Number Generator:

According to Per, Snook and Russo (2011) An essential component of a Monte Carlo method is the random number generator. All the random numbers are pseudo random numbers in Monte Carlo codes. A better algorithm of random number generates an unbiased and even random number with a big period. In a Monte Carlo code the random number generator is the foundation of all sampling. From distributions all samplings start with a random number generator.

Probability Density Function (PDF):

Trachtenberg (1990) has mentioned that another essential component of Monte Carlo Method is the probability density function. It is the statistical explanation of the physical system and explains the big particle samples statistical behavior. A big database is required to preserve the statistical information of a particle such as different interaction probability, particle's direction probability after an interaction and particle's energy probability after an interaction. Only if the probability density function is an accurate representation of the physical behavior of a particle can the outcome of a Monte Carlo simulation be precise.

Estimation of error and Tallying:

Gerald and Wheatley (2004) have described that the Monte Carlo Method builds a tally by averaging contributions in particular places over entire histories of simulation. Therefore a statistical mistake is produced as the histories variance. To develop the accuracy of tally the statistical mistake must gain a small mistake. The statistical mistake is proportional to the square root reciprocal to the history number according to the central limit theory:

$$R \propto \frac{1}{\sqrt{N}}$$

In the above equation N is the history number and R is the statistical mistake. Hence to acquire a greater precise solution a big number of histories are needed. This interprets into a greater expense of computational time. In several cases to acquire a precise outcome the researcher must wait for days or hours. Therefore the Monte Carlo method is regarded as a slowly converging method.

Janke (1998) has mentioned that the Monte Carlo method needs a repetitive domain representation and a discretized grid is not required. It can be applied to an arbitrary 3-D geometric design and numerous realistic systems can be simulated readily. These realistic systems can be complex geometries. Increased complexity of geometry modeling

will last longer the time of computation of present codes of Monte Carlo and will make the issue establishment more time consuming and crucial.

IV. MONTE CARLO ALGORITHM

According to Hayes(1993) a Monte Carlo Algorithm performs for a fixed number of steps and generates a response that is proper with probability $\geq 1/3$. A Las Vegas algorithm often generates the proper response its execution time is a random variable which has a bounded expectation. These expectations/probabilities are other random options done by algorithm independent of the input. Thus Monte Carlo algorithms independent repetitions reduce the probability of failure exponentially. The Monte Carlo algorithm computer is a randomized algorithm whose execution time is deterministic but whose results may be improper with a specific probability. Similarly Mascagni and Srinivasan (2000) have mentioned that the Las Vegas Algorithm is randomized but in a varied manner they acquire a time amount that differs randomly but often generates appropriate response. A Monte Carlo algorithm can be transformed into a Las Vegas algorithm whenever there occurs a process to check that the results generated by the algorithm is appropriate. If so then the resulting algorithm of Monte Carlo is merely to execute the algorithm of Monte Carlo until one of the runs generates a result that can be checked to verify. There are two kinds of Monte Carlo Algorithms:

One sided algorithm:

Luby (1996) has described that one sided algorithm makes errors only in one direction. For instance is the actual response is yes then $\Pr[\text{yes}] \geq \frac{1}{2} + \epsilon$ while if the actual response is no then $\Pr[\text{no}] = 1$. In other words the yes response is assured to be precise while a no response is doubtful. The researcher can develop his confidence in no response by executing the algorithm several times using independent random bits every time. So the researcher can get the result as yes if they view yes after t repetitions and otherwise results no. In this scheme the error probability is at most $(1 - \frac{1}{2} + \epsilon)^t$ so if the researcher need to make this less than any needed $\epsilon > 0$ it satisfies to take several trials t to be $O(\log 1/\epsilon)$ where the O in the constant relies on ϵ . The decision issues class is resolved by the polynomial time of Monte Carlo Algorithm with error on one sided is referred to as randomized polynomial time.

Two sided error:

In both the directions the algorithms of two sided errors make errors. Thus for example, if the actual response is yes then $\Pr[\text{yes}] \geq \frac{1}{2} + \epsilon$ and if the actual response is no then $\Pr[\text{no}] \geq \frac{1}{2} + \epsilon$. The researcher can develop their

confidence in this case by executing the algorithm several times and then making majority vote. The consequent estimation of standard reveals that several trials t needed to assure an error at most ϵ is again $O(\log 1/\epsilon)$ (Landau and Páez, 1997).

Example:

The probability that most of the vote algorithm represents an error is similar to the probability that the researcher can acquire at most $t/2$ appropriate results in t trials which is indicated by:

$$\begin{aligned} \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{t}{i} \left(\frac{1}{2} + \epsilon\right)^i \left(\frac{1}{2} - \epsilon\right)^{t-i} &\leq \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{t}{i} \left(\frac{1}{2} + \epsilon\right)^{t/2} \left(\frac{1}{2} - \epsilon\right)^{t/2} \\ &\leq \left(\frac{1}{4} - \epsilon^2\right)^{t/2} \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{t}{i} \\ &\leq \left(\frac{1}{4} - \epsilon^2\right)^{t/2} 2^t \\ &= (1 - 4\epsilon^2)^{t/2}. \end{aligned}$$

$$t = \frac{2 \log \delta^{-1}}{\log(1-4\epsilon^2)} = C \log \frac{1}{\delta},$$

Taking $\delta = \epsilon$ where C is a constant relying on ϵ makes this at $O(\log 1/\epsilon)$. The decision issues class is resolved by a Monte Carlo Algorithm's polynomial time with error on two sides is referred to as a bounded error polynomial time of probability.

Miodownik, Holm, et al. (2000) have mentioned that the result is often appropriate but the execution time may be unbounded in a Las Vegas algorithm. However the expected execution time is needed to be bounded. Similarly the researcher needs the bounded execution time but permits the algorithm to result either appropriate response or a special sign "?" so that the resulting probability "?" is $\frac{1}{2}$. The researcher can often transform a Las Vegas algorithm into a Monte Carlo algorithm by executing it for a fixed time amount and resulting an arbitrary response if it fails to stop. This executes because the researcher can bound the probability that the algorithm will execute a fixed limit past using inequality of Markov and the fact that the expectation is bounded. The converse is not true apparently because of the strict appropriate result need of Las Vegas algorithms. Thus there is no special term for transforming a Monte Carlo algorithm into a Las Vegas algorithm. The decision is issued class resolved by a Las Vegas algorithm polynomial time is referred to as zero error polynomial time probability.

V. ADVANTAGES OF MONTE CARLO SIMULATION

Monte Carlo simulation offers several benefits. Some of them are described below:

Probabilistic Outcomes:

Outcomes reveals not only what must exist but how probably every result is.

Analysis of sensitivity:

The deterministic analysis makes it crucial to view with just a small number of cases to view which variables influence the results the most. It is simple to view which inputs had the largest influence on bottom line outcomes (Laurenzi, Bartels and Diamond, 2002).

Graphical outcomes:

Manousiouthakis and Deem (1999) have described that because of the data produced by Monte Carlo simulation it is simple to make various results graphs and their opportunities of existence. This is essential for communicating observations to other stakeholders.

Analysis of scenario:

It is crucial to model various integrations of values for various inputs to view truly various scenario effects in deterministic models. Analysts can view exactly which inputs had which values integrated when particular results existed using the Monte Carlo simulation. For pursuing the further analysis this is not valuable.

Inputs Correlation:

It is possible to model interdependent rapport between variables of input in Monte Carlo simulation. It is essential for accuracy to indicate how, in reality, when some factors develops and others increases or decreases accordingly (Siepmann and Frenkel, 1992).

Simplicity:

According to Dodd, Boone and Theodorou (1993) the Monte Carlo Simulation major benefits contrast to other numerical processes that can resolve the similar issue is that it is simple conceptually. It does not need a particular knowledge of solution form or its analytic assets. On a computer Monte Carlo is similarly simple to implement.

Dimension Independence:

Betancourt (2005) has described that the amount of work to acquire a similar precision amount is dimension independent d of the underlying random variables. Thus the integration of Monte Carlo is the only process to

compute high dimensional integrals numerically. The techniques of traditional quadrature normally need an amount of work exponential in several dimensions d since they need grid sampling in d i.e. dimensional space. On the other side the integration of Monte Carlo is normally not rivalry with quadrature for less dimensional integration.

Parallelizable Easily:

Several processes of a computer can be involved in simulations of Monte Carlo simultaneously. Every simulation is independent of another simulation (Korniss, Novotny, and Rikvold, 1999).

Unrestricted option of functions:

The functions to combine with Monte Carlo can be arbitrary practically. No smoothness circumstances or boundedness circumstances are required for instance offering the integral is finite.

VI. DRAWBACKS OF MONTE CARLO SIMULATION

Though there are several benefits in Monte Carlo Simulation there are some drawbacks also. Some of the drawbacks of Monte Carlo Simulation are:

Slowness:

Caffisch and Numerica (1998) have described that the Monte Carlo integration major drawback is that it is slow. Numerous samples may be needed on the order of 1000s or even millions to acquire agreed precision in the response. Specifically since the probabilistic mistake bound reduces as the root of the reciprocal square of iteration numbers to gain more than one precision's decimal digit of the response needed $10^2 = 100$ times greater than the iteration numbers.

Crucial in calculating mistakes:

There are no difficult bounds on the mistake of the evaluated outcomes. The probabilistic mistake bound which is significantly based on the variance for skewed distributions may not be a better extent of the mistake (Geyer, 1991).

Black Box approach:

Earl and Deem (2005) has described that one can study the solution behavior with some kinds of analytical approximation if the starting parameters are modified. Normally this is difficult to do with Monte Carlo's black box approach.

Mistakes May Rely On Distribution:

Tessi et al (1996) has mentioned that the calculation of the expectation EX may be influenced severely if X the distribution is skewed heavily or has tails of heavier than normal. A non random numerical process may prevent these deficiencies or at least be as influenced severely.

VII. CONCLUSION:

In conclusion the Monte Carlo method is simple to use program to resolve various decision making probability issues. It permits the researchers to execute several trial simulations while accounting for doubtful variables. It can be inferred that as computers are becoming available easily, better networked and more numerous the computations of Monte Carlo algorithms will become more useful and common. .

REFERENCES

1. Assaraf R and Caffarel M (1999), Zero-variance principle for Monte Carlo algorithms, *Physical Review Letters* , pp 4682–4685.
2. Janke W (1998), *Nonlocal Monte Carlo Algorithms for Statistical Physics Applications*, Mathematics and Computers in Simulations **47**, 329.
3. Stokes-Rees I, Baude F, Doan V D and Bossy M (2008), Managing parallel and distributed Monte Carlo simulations for computational finance in a grid environment, *Proceedings of the International Symposium on Grid Computing 2007*, Springer Verlag, Germany.
4. Sokal A D (1992), *Bosonic Algorithms*, in: *Quantum Fields on the Computer*, ed. M. Creutz, World Scientific , Singapore, p. 211.
5. Landau D P and Binder K (2000), *A Guide to Monte Carlo Simulations in Statistical Physics*, Cambridge University Press, Cambridge.
6. Radhakrishnan, B. and T. Zacharia (1995), Simulation of curvature-driven grain growth using a modified Monte Carlo algorithm. *Metallurgical and Materials Transactions* 26A: 167.
7. Janke W (1994), *Recent Developments in Monte Carlo Simulations of First-Order Phase Transitions*, in: *Computer Simulations in Condensed Matter Physics VII*, eds. D.P. Landau, K.K. Mon, and H.-B. Schuttler, Springer, Berlin, p. 29.
8. Graves , R . (2001) . Open and Closed: The Monte Carlo Model . *PM Network* . 15 (12) ,pp. 37 – 41 .
9. Propp J G and Wilson D B (1996), Exact sampling with coupled Markov chains and applications to statistical mechanics, *Random Structures and Algorithms*, pp 232–252.
10. Per M C, Snook I K and Russo S P (2011), Zero-variance zero-bias quantum Monte Carlo estimators for the electron density at a nucleus, *Journal of Chemical Physics* 135, 134112.
11. Trachtenberg, M. (1990). A general theory of software reliability modeling. *IEEE Transactions on Reliability*, 39(1), pp. 92-96.
12. Gerald, C. and Wheatley, P. O (2004),: *Applied Numerical Analysis*, 7th ed., Pearson Education, Inc. New York.
13. Janke W (1998), *Multicanonical Monte Carlo Simulations*, *Physica* **A254**, 164.
14. Hayes B (1993), "The Science of Computing: The Wheel of Fortune." *American Scientist*, Vol. 81, No. 2, pp. 114–118.
15. Mascagni M and Srinivasan A (2000), Algorithm 806: SPRNG: A Scalable Library for Pseudorandom Number Generation, *ACM Transactions on Mathematical Software*, 26: 436–461.
16. Luby M (1996), *Pseudorandomness and Cryptographic Applications*. Princeton University Press, Princeton, New Jersey.
17. Landau R H and Páez M J (1997), *Computational Physics: Problem Solving with Computers*, Wiley, New York, p 93-108.
18. Miodownik, M E, Holm, et al. (2000). Highly parallel computer simulations of particle pinning: Zener vindicated, *Scripta Materiala* 42: 1173-1177.
19. Laurenzi I J, Bartels S L and Diamond S L (2002), A general algorithm for exact simulation of multicomponent aggregation, *J. Comput. Phys.*, 177, 418–449, 2002.
20. Manousiouthakis, V. I. and Deem, M. W. (1999) Strict detailed balance is unnecessary in Monte Carlo simulation. *J. Chem. Phys.* 110, 2753–2756.
21. Siepmann, J. I. and Frenkel, D. (1992) Configurational-bias Monte Carlo: A new sampling scheme for flexible chains. *Mol. Phys.* 75, 59–70.
22. Dodd, L. R., Boone, T. D., and Theodorou, D. N. (1993), A concerted rotation algorithm for atomistic Monte

Carlo simulation of polymer melts and glasses. *Mol. Phys.* 78, 961–996.

23. Betancourt, M. R. (2005) Efficient Monte Carlo moves for polypeptide simulations. *J. Chem. Phys.* 123, 174905.

24. Korniss, G., Novotny, M. A. and Rikvold, P. A. (1999). Parallelization of a dynamic Monte Carlo algorithm: a partially rejection-free conservative approach. *Journal of Computational Physics* 153: 488-508.

25. Caflisch R E and Numerica A (1998), Monte Carlo and quasi-Monte Carlo methods, SAGE, London. pp. 1-49.

26. Geyer C J (1991), Markov chain Monte Carlo maximum likelihood. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, 156–163.

27. Earl, D. J. and Deem, M.W. (2005) Markov chains of infinite order and asymptotic satisfaction of balance: Application to the adaptive integration method, *J. Phys. Chem. B* 109, 6701–6704.

28. Tesi M C, Janse van Rensburg D J, Orlandini E and Whittington S G (1996), Monte Carlo study of the interacting self-avoiding walk model in three dimensions. *J. Stat. Phys.* 82, 155–181.