WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools

What links here
Related changes
Upload file
Special pages
Permanent link
Page information

Monte Carlo method

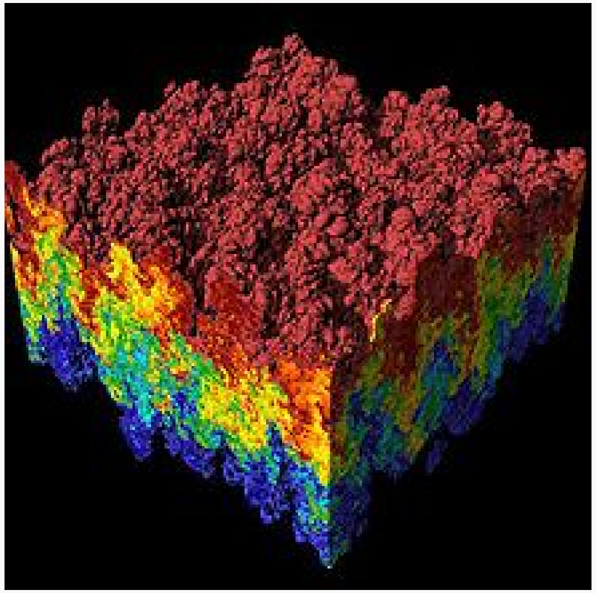
From Wikipedia, the free encyclopedia

Not to be confused with Monte Carlo algorithm.

Monte Carlo methods (or **Monte Carlo experiments**) are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. They are often used in physical and mathematical problems and are most useful when it is difficult or impossible to use other mathematical methods. Monte Carlo methods are mainly used in three distinct problem classes:[1] optimization, numerical integration, and generating draws from a probability distribution.

In physics-related problems, Monte Carlo methods are quite useful for simulating systems with many coupled degrees of freedom, such as fluids, disordered materials, strongly coupled solids, and cellular structures (see cellular Potts model, interacting particle systems, McKean-Vlasov

Computational physics



Numerical analysis · Simulation

Data analysis · Visualization

Potentials [show]

Fluid dynamics [show]

Monte Carlo methods [show]

Page information
Wikidata item
Cite this page

Print/export

Create a book
Download as PDF
Printable version

Languages

العربية
Azərbaycanca
Bosanski
Català
Čeština
Dansk
Deutsch
Eesti
Ελληνικά
Español
Euskara
فارسی
Français
한국어
Hrvatski
Bahasa Indonesia
Italiano
עברית
Қазақша
Latviešu
Lietuvių

Monte Carlo methods [show]

Particle [show]

Scientists [show]

V·T·E·

processes, kinetic models of gases). Other examples include modeling phenomena with significant uncertainty in inputs such as the calculation of risk in business and, in math, evaluation of multidimensional definite integrals with complicated boundary conditions. In application to space and oil exploration problems, Monte Carlo–based predictions of failure, cost overruns and schedule overruns are routinely better than human intuition or alternative "soft" methods.[2]

In principle, Monte Carlo methods can be used to solve any problem having a probabilistic interpretation. By the law of large numbers, integrals described by the expected value of some random variable can be approximated by taking the empirical mean (a.k.a. the sample mean) of independent samples of the variable. When the probability distribution of the variable is too complex, we often use a Markov Chain Monte Carlo (MCMC) sampler. The central idea is to design a judicious Markov chain model with a prescribed stationary probability distribution. By the ergodic theorem, the stationary probability distribution is approximated by the empirical measures of the random states of the MCMC sampler.

In other important problems we are interested in generating draws from a sequence of probability distributions satisfying a nonlinear evolution equation. These flows of probability distributions can always be interpreted as the distributions of the random states of a Markov process whose transition probabilities depends on the distributions of the current random states (see McKean-Vlasov processes, nonlinear filtering equation).[3][4] In other instances we are given a flow of probability distributions with an increasing level of sampling complexity (path spaces models with an increasing time horizon, Boltzmann-Bibbs measures associated with decreasing temperature parameters, and many others). These models can also be seen as the evolution of the law of the random states of a nonlinear Markov chain.[4][5] A natural way to simulate these sophisticated

Magyar
Nederlands
日本語
Norsk bokmål
Norsk nynorsk
Occitan
Oʻzbekcha/ўзбекча
Polski
Português
Русский
Simple English
Slovenčina
Srpskohrvatski / српскохрватски
Basa Sunda
Suomi
Svenska
Türkçe
Українська
اردو
Tiếng Việt
中文
Edit links

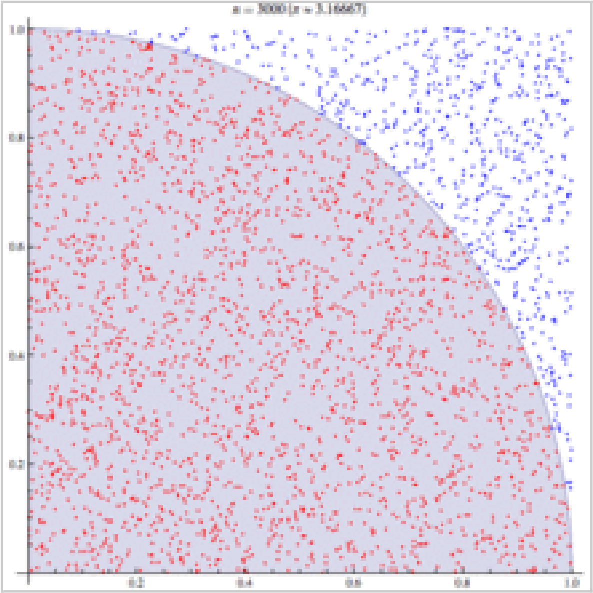
nonlinear Markov processes is to sample a large number of copies of the process, replacing in the evolution equation the unknown distributions of the random states by the sampled empirical measures. In contrast with traditional Monte Carlo and Markov chain Monte Carlo methodologies these mean field particle techniques rely on sequential interacting samples. The terminology mean field reflects the fact that each of the *samples (a.k.a. particles, individuals, walkers, agents, creatures, or phenotypes)* interacts with the empirical measures of the process. When the size of the system tends to infinity, these random empirical measures converge to the deterministic distribution of the random states of the nonlinear Markov chain, so that the statistical interaction between particles vanishes.

Contents [hide]

Introduction [edit]

Monte Carlo methods vary, but tend to follow a particular pattern:

1. Define a domain of possible inputs.
2. Generate inputs randomly from a probability distribution over the domain.
3. Perform a deterministic computation on the inputs.
4. Aggregate the results.



Monte Carlo method applied to approximating the value of π. After placing 30000 random points, the estimate for π is within 0.07% of the actual value. This happens with an approximate probability of 20%.

For example, consider a circle inscribed in a unit square. Given that the circle and the square have a ratio of areas that is π/4, the value of π can be approximated using a Monte Carlo method:[6]

1. Draw a square on the ground, then inscribe a circle within it.
2. Uniformly scatter some objects of uniform size (grains of rice or sand) over the square.

3. Count the number of objects inside the circle and the total number of objects.
4. The ratio of the two counts is an estimate of the ratio of the two areas, which is π/4. Multiply the result by 4 to estimate π.

In this procedure the domain of inputs is the square that circumscribes our circle. We generate random inputs by scattering grains over the square then perform a computation on each input (test whether it falls within the circle). Finally, we aggregate the results to obtain our final result, the approximation of π.

There are two important points to consider here: Firstly, if the grains are not uniformly distributed, then our approximation will be poor. Secondly, there should be a large number of inputs. The approximation is generally poor if only a few grains are randomly dropped into the whole square. On average, the approximation improves as more grains are dropped.

Uses of Monte Carlo methods require large amounts of random numbers, and it was their use that spurred the development of pseudorandom number generators, which were far quicker to use than the tables of random numbers that had been previously used for statistical sampling.

History [edit]

Before the Monte Carlo method was developed, simulations tested a previously understood deterministic problem and statistical sampling was used to estimate uncertainties in the simulations. Monte Carlo simulations invert this approach, solving deterministic problems using a probabilistic analog (see Simulated annealing).

An early variant of the Monte Carlo method can be seen in the Buffon's needle experiment, in which π can be estimated by dropping needles on a floor made of parallel and equidistant strips. In the 1930s, Enrico Fermi first experimented with the Monte Carlo method while studying neutron diffusion, but did not publish anything on it.[7]

The modern version of the Markov Chain Monte Carlo method was invented in the late 1940s by Stanislaw Ulam, while he was working on nuclear weapons projects at the Los Alamos National Laboratory. Immediately after Ulam's breakthrough, John von Neumann understood its importance and programmed the ENIAC computer to carry out Monte Carlo calculations. In 1946, physicists at Los Alamos Scientific Laboratory were investigating radiation shielding and the distance that neutrons would likely travel through various materials. Despite having most of the necessary data, such as the average distance a neutron would travel in a substance before it collided with an atomic nucleus, and how much energy the neutron was likely to give off following a collision, the Los Alamos physicists were unable to solve the problem using conventional, deterministic mathematical methods. Stanislaw Ulam had the idea of using random experiments. He recounts his inspiration as follows:

> The first thoughts and attempts I made to practice [the Monte Carlo Method] were suggested by a question which occurred to me in 1946 as I was convalescing from an illness and playing solitaires. The question was what are the chances that a Canfield solitaire laid out with 52 cards will come out successfully? After spending a lot of time trying to estimate them by pure combinatorial calculations, I wondered whether a more practical method than "abstract thinking" might not be to lay it out say one hundred times and simply observe and count the number of successful plays. This was already possible to envisage with the beginning of the new era of fast computers, and I immediately thought of problems of neutron diffusion and other questions of mathematical physics, and more generally how to change processes described by certain differential equations into an equivalent form interpretable as a succession of random operations. Later [in 1946], I described the idea to John von Neumann, and we began to plan actual calculations.
>
> –Stanislaw Ulam[8]

Being secret, the work of von Neumann and Ulam required a code name.[citation needed] A colleague of von Neumann and Ulam, Nicholas Metropolis, suggested using the name *Monte Carlo*, which refers to the Monte Carlo Casino in Monaco where Ulam's uncle would borrow money from relatives to gamble.[7] Using lists of "truly random" random numbers was extremely slow, but von Neumann developed a way to calculate pseudorandom numbers, using the middle-square method. Though this method has been criticized as crude, von Neumann was aware of this: he justified it as being faster than any other method at his disposal, and also noted that when it went awry it did so obviously, unlike methods that could be subtly incorrect.

Monte Carlo methods were central to the simulations required for the Manhattan Project, though severely limited by the computational tools at the time. In the 1950s they were used at Los Alamos for early work relating to the development of the hydrogen bomb, and became popularized in the fields of physics, physical chemistry, and operations research. The Rand Corporation and the U.S. Air Force were two of the major organizations responsible for funding and disseminating information on Monte Carlo methods during this time, and they began to find a wide application in many different fields.

The theory of more sophisticated mean field type particle Monte Carlo methods had certainly started by the mid-1960s, with the work of Henry P. McKean Jr. on Markov interpretations of a class of nonlinear parabolic partial differential equations arising in fluid mechanics.[9][10] We also quote an earlier pioneering article by Theodore E. Harris and Herman Kahn, published in 1951, using mean field genetic-type Monte Carlo methods for estimating particle transmission energies.[11] Mean field genetic type Monte Carlo methodologies are also used as heuristic natural search algorithms (a.k.a. Metaheuristic) in evolutionary computing. The origins of these mean field computational techniques can be traced to 1950 and 1954 with the work of Alan Turing on genetic type mutation-selection learning machines [12] and the articles by Nils Aall Barricelli at the Institute for Advanced Study in Princeton, New Jersey.[13][14]

Quantum Monte Carlo, and more specifically Diffusion Monte Carlo methods can also be interpreted as a mean field particle Monte Carlo approximation of Feynman-Kac path integrals.[15][16][17][18][19][20][21] The origins of Quantum Monte Carlo methods are often attributed to Enrico Fermi and Robert Richtmyer who developed in 1948 a mean field particle interpretation of neutron-chain reactions,[22] but the first heuristic-like and genetic type particle algorithm (a.k.a. Resampled or Reconfiguration Monte Carlo methods) for estimating ground state energies of quantum systems (in reduced matrix models) is due to Jack H. Hetherington in 1984[21] In molecular chemistry, the use of genetic heuristic-like particle methodologies (a.k.a. pruning and enrichment strategies) can be traced back to 1955 with the seminal work of Marshall. N. Rosenbluth and Arianna. W. Rosenbluth.[23]

The use of Sequential Monte Carlo in advanced Signal processing and Bayesian inference is more recent. It was in 1993, that Gordon et al., published in their seminal work[24] the first application of a Monte Carlo resampling algorithm in Bayesian statistical inference. The authors named their algorithm 'the bootstrap filter', and demonstrated that compared to other filtering methods, their bootstrap algorithm does not require any assumption about that state-space or the noise of the system. We also quote another pioneering article in this field of Genshiro Kitagawa on a related "Monte Carlo filter",[25] and the ones by Pierre Del Moral[26] and Himilcon Carvalho, Pierre Del Moral, André Monin and Gérard Salut[27] on particle filters published in the mid-1990s. Particle filters were also developed in signal processing in the early 1989-1992 by P. Del Moral, J.C. Noyer, G. Rigal, and G. Salut in the LAAS-CNRS in a series of restricted and classified research reports with STCAN (Service Technique des Constructions et Armes Navales), the IT company DIGILOG, and the LAAS-CNRS (the Laboratory for Analysis and Architecture of Systems) on RADAR/SONAR and GPS signal processing problems.[28][29][30][31][32][33] These Sequential Monte Carlo methodologies can be interpreted as an acceptance-rejection sampler equipped with an interacting recycling mechanism.

From 1950 to 1996, all the publications on Sequential Monte Carlo methodologies including the pruning and resample Monte Carlo methods introduced in computational physics and molecular chemistry, present natural and heuristic-like algorithms applied to different situations without a single proof of their consistency, nor a discussion on the bias of the estimates and on genealogical and ancestral tree based algorithms. The mathematical foundations and the first rigorous analysis of these particle algorithms are due to Pierre Del Moral[26][34] in 1996. Branching type particle methodologies with varying population sizes were also developed in the end of the 1990s by Dan Crisan, Jessica Gaines and Terry Lyons,[35][36][37] and by Dan Crisan, Pierre Del Moral and Terry Lyons.[38] Further developments in this field were developed in 2000 by P. Del Moral, A. Guionnet and L. Miclo.[16][39][40]

Definitions [edit]

There is no consensus on how *Monte Carlo* should be defined. For example, Ripley[41] defines most probabilistic modeling as *stochastic simulation*, with *Monte Carlo* being reserved for Monte Carlo integration and Monte Carlo statistical tests. Sawilowsky[42] distinguishes between a simulation, a Monte Carlo method, and a Monte Carlo simulation: a simulation is a fictitious representation of reality, a Monte Carlo method is a technique that can be used to solve a mathematical or statistical problem, and a Monte Carlo simulation uses repeated sampling to determine the properties of some phenomenon (or behavior). Examples:

- Simulation: Drawing **one** pseudo-random uniform variable from the interval [0,1] can be used to simulate the tossing of a coin: If the value is less than or equal to 0.50 designate the outcome as heads, but if the value is greater than 0.50 designate the outcome as tails. This is a simulation, but not a Monte Carlo simulation.
- Monte Carlo method: Pouring out a box of coins on a table, and then computing the ratio of coins that land heads versus tails is a Monte Carlo method of determining the behavior of

repeated coin tosses, but it is not a simulation.

- Monte Carlo simulation: Drawing **a large number** of pseudo-random uniform variables from the interval [0,1], and assigning values less than or equal to 0.50 as heads and greater than 0.50 as tails, is a *Monte Carlo simulation* of the behavior of repeatedly tossing a coin.

Kalos and Whitlock[6] point out that such distinctions are not always easy to maintain. For example, the emission of radiation from atoms is a natural stochastic process. It can be simulated directly, or its average behavior can be described by stochastic equations that can themselves be solved using Monte Carlo methods. "Indeed, the same computer code can be viewed simultaneously as a 'natural simulation' or as a solution of the equations by natural sampling."

Monte Carlo and random numbers [edit]

Monte Carlo simulation methods do not always require truly random numbers to be useful — while for some applications, such as primality testing, unpredictability is vital.[43] Many of the most useful techniques use deterministic, pseudorandom sequences, making it easy to test and re-run simulations. The only quality usually necessary to make good simulations is for the pseudo-random sequence to appear "random enough" in a certain sense.

What this means depends on the application, but typically they should pass a series of statistical tests. Testing that the numbers are uniformly distributed or follow another desired distribution when a large enough number of elements of the sequence are considered is one of the simplest, and most common ones. Weak correlations between successive samples is also often desirable/necessary.

Sawilowsky lists the characteristics of a high quality Monte Carlo simulation:[42]

- the (pseudo-random) number generator has certain characteristics (*e.g.*, a long "period" before the sequence repeats)

- the (pseudo-random) number generator produces values that pass tests for randomness
- there are enough samples to ensure accurate results
- the proper sampling technique is used
- the algorithm used is valid for what is being modeled
- it simulates the phenomenon in question.

Pseudo-random number sampling algorithms are used to transform uniformly distributed pseudo-random numbers into numbers that are distributed according to a given probability distribution.

Low-discrepancy sequences are often used instead of random sampling from a space as they ensure even coverage and normally have a faster order of convergence than Monte Carlo simulations using random or pseudorandom sequences. Methods based on their use are called quasi-Monte Carlo methods.

Monte Carlo simulation versus "what if" scenarios [edit]

There are ways of using probabilities that are definitely not Monte Carlo simulations — for example, deterministic modeling using single-point estimates. Each uncertain variable within a model is assigned a “best guess” estimate. Scenarios (such as best, worst, or most likely case) for each input variable are chosen and the results recorded.[44]

By contrast, Monte Carlo simulations sample from a probability distribution for each variable to produce hundreds or thousands of possible outcomes. The results are analyzed to get probabilities of different outcomes occurring.[45] For example, a comparison of a spreadsheet cost construction model run using traditional “what if” scenarios, and then running the comparison again with Monte Carlo simulation and triangular probability distributions shows that the Monte Carlo analysis has a narrower range than the “what if” analysis.[*examples needed*] This is because the “what if” analysis gives equal weight to all scenarios (see quantifying uncertainty in corporate finance), while the Monte Carlo method hardly samples in the very low probability regions. The samples in such regions are called "rare events".

Applications [edit]

Monte Carlo methods are especially useful for simulating phenomena with significant uncertainty in inputs and systems with a large number of coupled degrees of freedom. Areas of application include:

Physical sciences [edit]

See also: Monte Carlo method in statistical physics

Monte Carlo methods are very important in computational physics, physical chemistry, and related applied fields, and have diverse applications from complicated quantum chromodynamics calculations to designing heat shields and aerodynamic forms as well as in modeling radiation transport for radiation dosimetry calculations.[46][47][48] In statistical physics Monte Carlo molecular modeling is an alternative to computational molecular dynamics, and Monte Carlo methods are used to compute statistical field theories of simple particle and polymer systems.[23][49] Quantum Monte Carlo methods solve the many-body problem for quantum systems.[3][4][15] In experimental particle physics, Monte Carlo methods are used for designing detectors, understanding their behavior and comparing experimental data to theory. In astrophysics, they are used in such diverse manners as to model both galaxy evolution[50] and microwave radiation transmission through a rough planetary surface.[51] Monte Carlo methods are also used in the ensemble models that form the basis of modern weather forecasting.

Engineering [edit]

Monte Carlo methods are widely used in engineering for sensitivity analysis and quantitative probabilistic analysis in process design. The need arises from the interactive, co-linear and non-linear behavior of typical process simulations. For example,

- In microelectronics engineering, Monte Carlo methods are applied to analyze correlated and uncorrelated variations in analog and digital integrated circuits.
- In geostatistics and geometallurgy, Monte Carlo methods underpin the design of mineral processing flowsheets and contribute to quantitative risk analysis.
- In wind energy yield analysis, the predicted energy output of a wind farm during its lifetime is calculated giving different levels of uncertainty (P90, P50, etc.)
- impacts of pollution are simulated[52] and diesel compared with petrol.[53]
- In Fluid Dynamics, in particular Rarefied Gas Dynamics, where the Boltzmann equation is solved for finite Knudsen number fluid flows using the Direct Simulation Monte Carlo [54] method in combination with highly efficient computational algorithms.[55]
- In autonomous robotics, Monte Carlo localization can determine the position of a robot. It is often applied to stochastic filters such as the Kalman filter or Particle filter that forms the heart of the SLAM (Simultaneous Localization and Mapping) algorithm.
- In telecommunications, when planning a wireless network, design must be proved to work for a wide variety of scenarios that depend mainly on the number of users, their locations and the services they want to use. Monte Carlo methods are typically used to generate these users and their states. The network performance is then evaluated and, if results are not satisfactory, the network design goes through an optimization process.
- In reliability engineering, one can use Monte Carlo simulation to generate mean time between failures and mean time to repair for components.
- In signal processing and Bayesian inference, particle filters and sequential Monte Carlo techniques are a class of mean field particle methods for sampling and computing the posterior distribution of a signal process given some noisy and partial observations using interacting

empirical measures.

Computational biology [edit]

Monte Carlo methods are used in various fields of computational biology, for example for Bayesian inference in phylogeny, or for studying biological systems such as genomes, proteins,[56] or membranes.[57] The systems can be studied in the coarse-grained or *ab initio* frameworks depending on the desired accuracy. Computer simulations allow us to monitor the local environment of a particular molecule to see if some chemical reaction is happening for instance. In cases where it is not feasible to conduct a physical experiment, thought experiments can be conducted (for instance: breaking bonds, introducing impurities at specific sites, changing the local/global structure, or introducing external fields).

Computer graphics [edit]

Path Tracing, occasionally referred to as Monte Carlo Ray Tracing, renders a 3D scene by randomly tracing samples of possible light paths. Repeated sampling of any given pixel will eventually cause the average of the samples to converge on the correct solution of the rendering equation, making it one of the most physically accurate 3D graphics rendering methods in existence.

Applied statistics [edit]

In applied statistics, Monte Carlo methods are generally used for two purposes:

1. To compare competing statistics for small samples under realistic data conditions. Although Type I error and power properties of statistics can be calculated for data drawn from classical theoretical distributions (*e.g.*, normal curve, Cauchy distribution) for asymptotic conditions (*i. e*, infinite sample size and infinitesimally small treatment effect), real data often

do not have such distributions.[58]

2. To provide implementations of hypothesis tests that are more efficient than exact tests such as permutation tests (which are often impossible to compute) while being more accurate than critical values for asymptotic distributions.

Monte Carlo methods are also a compromise between approximate randomization and permutation tests. An approximate randomization test is based on a specified subset of all permutations (which entails potentially enormous housekeeping of which permutations have been considered). The Monte Carlo approach is based on a specified number of randomly drawn permutations (exchanging a minor loss in precision if a permutation is drawn twice – or more frequently—for the efficiency of not having to track which permutations have already been selected).

Artificial intelligence for games [edit]

Main article: Monte Carlo tree search

Monte Carlo methods have been developed into a technique called Monte-Carlo tree search that is useful for searching for the best move in a game. Possible moves are organized in a search tree and a large number of random simulations are used to estimate the long-term potential of each move. A black box simulator represents the opponent's moves.[59]

The Monte Carlo Tree Search (MCTS) method has four steps:[60]

1. Starting at root node of the tree, select optimal child nodes until a leaf node is reached.
2. Expand the leaf node and choose one of its children.
3. Play a simulated game starting with that node.
4. Use the results of that simulated game to update the node and its ancestors.

The net effect, over the course of many simulated games, is that the value of a node representing a move will go up or down, hopefully corresponding to whether or not that node represents a good move.

Monte Carlo Tree Search has been used successfully to play games such as Go,[61] Tantrix,[62] Battleship,[63] Havannah,[64] and Arimaa.[65]

See also: Computer Go

Design and visuals [edit]

Monte Carlo methods are also efficient in solving coupled integral differential equations of radiation fields and energy transport, and thus these methods have been used in global illumination computations that produce photo-realistic images of virtual 3D models, with applications in video games, architecture, design, computer generated films, and cinematic special effects.[66]

Finance and business [edit]

See also: Monte Carlo methods in finance, Quasi-Monte Carlo methods in finance, Monte Carlo methods for option pricing, Stochastic modelling (insurance) and Stochastic asset model

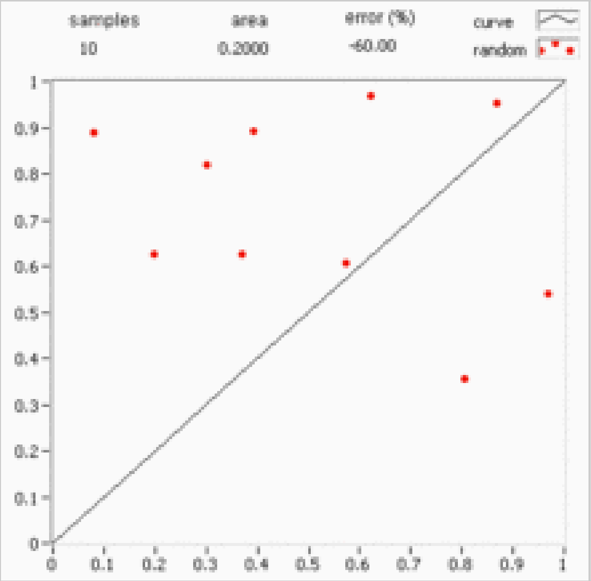
Monte Carlo methods in finance are often used to evaluate investments in projects at a business unit or corporate level, or to evaluate financial derivatives. They can be used to model project schedules, where simulations aggregate estimates for worst-case, best-case, and most likely durations for each task to determine outcomes for the overall project. Monte Carlo methods are also used in option pricing, default risk analysis.[67][68][69]

Use in mathematics [edit]

In general, Monte Carlo methods are used in mathematics to solve various problems by generating suitable random numbers (see also Random number generation) and observing that fraction of the numbers that obeys some property or properties. The method is useful for obtaining numerical solutions to problems too complicated to solve analytically. The most common application of the Monte Carlo method is Monte Carlo integration.

Integration [edit]

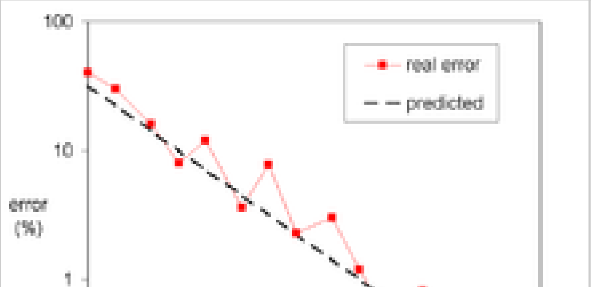
Main article: Monte Carlo integration



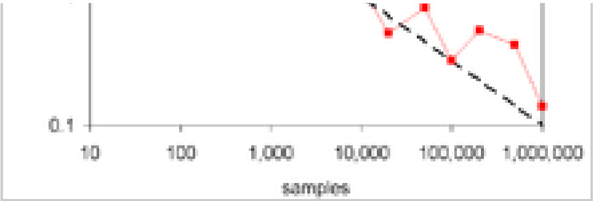
Monte-Carlo integration works by comparing random points with the value of the function

Deterministic numerical integration algorithms work well in a small number of dimensions, but encounter two problems when the functions have many variables. First, the number of function evaluations needed increases rapidly with the number of dimensions. For example, if 10 evaluations provide adequate accuracy in one dimension, then 10^{100} points are needed for 100 dimensions—far too many to be computed. This is called the curse of dimensionality. Second, the boundary of a multidimensional region may be very complicated, so it may not be feasible to reduce the problem to an iterated integral.[70] 100 dimensions is by no means unusual, since in many physical problems, a "dimension" is equivalent to a degree of freedom.

Monte Carlo methods provide a way out of this exponential increase in computation time. As long as the function in question is reasonably well-behaved, it can be estimated by randomly selecting points in 100-dimensional space, and taking some kind of average of the function values at these points. By the central limit theorem, this method



displays $1/\sqrt{N}$ convergence—i.e., quadrupling the number of sampled points halves the error, regardless of the number of dimensions.[70]



Errors reduce by a factor of $1/\sqrt{N}$

A refinement of this method, known as importance sampling in statistics, involves sampling the points randomly, but more frequently where the integrand is large. To do this precisely one would have to already know the integral, but one can approximate the integral by an integral of a similar function or use adaptive routines such as stratified sampling, recursive stratified sampling, adaptive umbrella sampling[71][72] or the VEGAS algorithm.

A similar approach, the quasi-Monte Carlo method, uses low-discrepancy sequences. These sequences "fill" the area better and sample the most important points more frequently, so quasi-Monte Carlo methods can often converge on the integral more quickly.

Another class of methods for sampling points in a volume is to simulate random walks over it (Markov chain Monte Carlo). Such methods include the Metropolis-Hastings algorithm, Gibbs sampling, Wang and Landau algorithm, and interacting type MCMC methodologies such as the Sequential Monte Carlo samplers.[73]

Simulation and optimization [edit]

Main article: Stochastic optimization

Another powerful and very popular application for random numbers in numerical simulation is in numerical optimization. The problem is to minimize (or maximize) functions of some vector that often has a large number of dimensions. Many problems can be phrased in this way: for example, a computer chess program could be seen as trying to find the set of, say, 10 moves that produces the best evaluation function at the end. In the traveling salesman problem the goal is to minimize distance traveled. There are also applications to engineering design, such as multidisciplinary design optimization. It has been applied with quasi-one-dimensional models to solve particle dynamics problems by efficiently exploring large configuration space.

The traveling salesman problem is what is called a conventional optimization problem. That is, all the facts (distances between each destination point) needed to determine the optimal path to follow are known with certainty and the goal is to run through the possible travel choices to come up with the one with the lowest total distance. However, let's assume that instead of wanting to minimize the total distance traveled to visit each desired destination, we wanted to minimize the total time needed to reach each destination. This goes beyond conventional optimization since travel time is inherently uncertain (traffic jams, time of day, etc.). As a result, to determine our optimal path we would want to use simulation - optimization to first understand the range of potential times it could take to go from one point to another (represented by a probability distribution in this case rather than a specific distance) and then optimize our travel decisions to identify the best path to follow taking that uncertainty into account.

Inverse problems [edit]

Probabilistic formulation of inverse problems leads to the definition of a probability distribution in the model space. This probability distribution combines prior information with new information obtained by measuring some observable parameters (data). As, in the general case, the theory linking data with model parameters is nonlinear, the posterior probability in the model space may not be easy to describe (it may be multimodal, some moments may not be defined, etc.).

When analyzing an inverse problem, obtaining a maximum likelihood model is usually not sufficient, as we normally also wish to have information on the resolution power of the data. In the general case we may have a large number of model parameters, and an inspection of the marginal probability densities of interest may be impractical, or even useless. But it is possible to pseudorandomly generate a large collection of models according to the posterior probability distribution and to analyze and display the models in such a way that information on the relative likelihoods of model properties is conveyed to the spectator. This can be accomplished by means of an efficient Monte Carlo method, even in cases where no explicit formula for the *a priori* distribution is available.

The best-known importance sampling method, the Metropolis algorithm, can be generalized, and this gives a method that allows analysis of (possibly highly nonlinear) inverse problems with complex *a priori* information and data with an arbitrary noise distribution.[74][75]

Petroleum reservoir management [edit]

Monte Carlo methods are very popular in hydrocarbon reservoir management in the context of nonlinear inverse problems. This includes generating computational models of oil and gas reservoirs for consistency with observed production data. For the goal of decision making and uncertainty assessment, Monte Carlo methods are used for generating multiple geological realizations.[76]

In popular culture [edit]

- *The Monte Carlo Method*, the 1998 album by the southern California indie rock band Nothing Painted Blue. (Scat. 1998).

See also [edit]

Statistics portal

- Auxiliary field Monte Carlo
- Biology Monte Carlo method
- Comparison of risk analysis Microsoft Excel add-ins
- Direct simulation Monte Carlo

- Dynamic Monte Carlo method
- Kinetic Monte Carlo
- Mean field particle methods
- Particle filter
- List of software for Monte Carlo molecular modeling
- Monte Carlo method for photon transport
- Monte Carlo methods for electron transport
- Morris method
- Genetic algorithms
- Quasi-Monte Carlo method
- Sobol sequence

Notes [edit]

1. ^ Kroese, D. P.; Brereton, T.; Taimre, T.; Botev, Z. I. (2014). "Why the Monte Carlo method is so important today". *WIREs Comput Stat* **6**: 386–392. doi:10.1002/wics.1314.
2. ^ Hubbard 2009
3. ^ *a* *b* Kolokoltsov, Vassili (2010). *Nonlinear Markov processes*. Cambridge Univ. Press. p. 375.
4. ^ *a* *b* *c* Del Moral, Pierre (2013). *Mean field simulation for Monte Carlo integration*. Chapman & Hall/CRC Press. p. 626. "Monographs on Statistics & Applied Probability"
5. ^ "Sequential Monte Carlo samplers - P. Del Moral - A. Doucet - A. Jasra - 2006 - Journal of the Royal Statistical Society: Series B (Statistical Methodology) - Wiley Online Library". *onlinelibrary.wiley.com*. Retrieved 2015-06-11.
6. ^ *a* *b* Kalos & Whitlock 2008
7. ^ *a* *b* Metropolis 1987
8. ^ Eckhardt 1987

9. ^ McKean, Henry, P. (1967). "Propagation of chaos for a class of non-linear parabolic equations". *Lecture Series in Differential Equations, Catholic Univ.* **7**: 41–57.
10. ^ McKean, Henry, P. (1966). "A class of Markov processes associated with nonlinear parabolic equations" (PDF). *Proc. Nat. Acad. Sci. USA* **56** (6): 1907–1911. Bibcode:1966PNAS...56.1907M. doi:10.1073/pnas.56.6.1907. PMC 220210. PMID 16591437.
11. ^ Herman, Kahn; Harris, Theodore, E. (1951). "Estimation of particle transmission by random sampling" (PDF). *Natl. Bur. Stand. Appl. Math. Ser.* **12**: 27–30.
12. ^ Turing, Alan M. "Computing machinery and intelligence". *Mind* **LIX** (238): 433–460. doi:10.1093/mind/LIX.236.433.
13. ^ Barricelli, Nils Aall (1954). "Esempi numerici di processi di evoluzione". *Methodos*: 45–68.
14. ^ Barricelli, Nils Aall (1957). "Symbiogenetic evolution processes realized by artificial methods". *Methodos*: 143–182.
15. ^ *a* *b* Del Moral, Pierre (2004). *Feynman-Kac formulae. Genealogical and interacting particle approximations*. Springer. p. 575. "Series: Probability and Applications"
16. ^ *a* *b* Del Moral, Pierre; Miclo, Laurent (2000). *Branching and Interacting Particle Systems Approximations of Feynman-Kac Formulae with Applications to Non-Linear Filtering.* (PDF). *Lecture Notes in Mathematics* **1729**. pp. 1–145. doi:10.1007/bfb0103798.
17. ^ Del Moral, Pierre; Miclo, Laurent (2000). "A Moran particle system approximation of Feynman-Kac formulae.". *Stochastic Processes and their Applications* **86** (2): 193–216. doi:10.1016/S0304-4149(99)00094-0.
18. ^ Del Moral, Pierre (2003). "Particle approximations of Lyapunov exponents connected to Schrödinger operators and Feynman-Kac semigroups" (PDF). *ESAIM Probability & Statistics* **7**: 171–208. doi:10.1051/ps:2003001.
19. ^ Assaraf, Roland; Caffarel, Michel; Khelif, Anatole (2000). "Diffusion Monte Carlo Methods with a fixed number of walkers" (PDF). *Phys. Rev. E* **61**: 4566–4575. Bibcode:2000PhRvE..61.4566A. doi:10.1103/physreve.61.4566.

20. ^ Caffarel, Michel; Ceperley, David; Kalos, Malvin (1993). "Comment on Feynman-Kac Path-Integral Calculation of the Ground-State Energies of Atoms". *Phys. Rev. Lett.* **71**: 2159. Bibcode:1993PhRvL..71.2159C. doi:10.1103/physrevlett.71.2159.
21. ^ *a* *b* Hetherington, Jack, H. (1984). "Observations on the statistical iteration of matrices". *Phys. Rev. A.* **30** (2713): 2713–2719. Bibcode:1984PhRvA..30.2713H. doi:10.1103/PhysRevA.30.2713.
22. ^ Fermi, Enrique; Richtmyer, Robert, D. (1948). "Note on census-taking in Monte Carlo calculations" (PDF). *LAM* **805** (A). "Declassified report Los Alamos Archive"
23. ^ *a* *b* Rosenbluth, Marshall, N.; Rosenbluth, Arianna, W. (1955). "Monte-Carlo calculations of the average extension of macromolecular chains". *J. Chem. Phys* **23**: 356–359. Bibcode:1955JChPh..23..356R. doi:10.1063/1.1741967.
24. ^ Gordon, N.J.; Salmond, D.J.; Smith, A.F.M. (April 1993). "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". *Radar and Signal Processing, IEE Proceedings F* **140** (2): 107–113. doi:10.1049/ip-f-2.1993.0015. ISSN 0956-375X.
25. ^ Kitagawa, G. (1996). "Monte carlo filter and smoother for non-Gaussian nonlinear state space models". *Journal of Computational and Graphical Statistics* **5** (1): 1–25. doi:10.2307/1390750. JSTOR 1390750.
26. ^ *a* *b* Del Moral, Pierre (1996). "Non Linear Filtering: Interacting Particle Solution." (PDF). *Markov Processes and Related Fields* **2** (4): 555–580.
27. ^ Carvalho, Himilcon; Del Moral, Pierre; Monin, André; Salut, Gérard (July 1997). "Optimal Non-linear Filtering in GPS/INS Integration." (PDF). *IEEE-Trans. on Aerospace and electronic systems* **33** (3).
28. ^ P. Del Moral, G. Rigal, and G. Salut. Estimation and nonlinear optimal control : An unified framework for particle solutions
LAAS-CNRS, Toulouse, Research Report no. 91137, DRET-DIGILOG- LAAS/CNRS contract, April (1991).

29. ^ P. Del Moral, G. Rigal, and G. Salut. Nonlinear and non Gaussian particle filters applied to inertial platform repositioning.
LAAS-CNRS, Toulouse, Research Report no. 92207, STCAN/DIGILOG-LAAS/CNRS Convention STCAN no. A.91.77.013, (94p.) September (1991).
30. ^ P. Del Moral, G. Rigal, and G. Salut. Estimation and nonlinear optimal control : Particle resolution in filtering and estimation. Experimental results.
Convention DRET no. 89.34.553.00.470.75.01, Research report no.2 (54p.), January (1992).
31. ^ P. Del Moral, G. Rigal, and G. Salut. Estimation and nonlinear optimal control : Particle resolution in filtering and estimation. Theoretical results
Convention DRET no. 89.34.553.00.470.75.01, Research report no.3 (123p.), October (1992).
32. ^ P. Del Moral, J.-Ch. Noyer, G. Rigal, and G. Salut. Particle filters in radar signal processing : detection, estimation and air targets recognition.
LAAS-CNRS, Toulouse, Research report no. 92495, December (1992).
33. ^ P. Del Moral, G. Rigal, and G. Salut. Estimation and nonlinear optimal control : Particle resolution in filtering and estimation.
Studies on: Filtering, optimal control, and maximum likelihood estimation. Convention DRET no. 89.34.553.00.470.75.01. Research report no.4 (210p.), January (1993).
34. ^ Del Moral, Pierre (1998). "Measure Valued Processes and Interacting Particle Systems. Application to Non Linear Filtering Problems". *Annals of Applied Probability* (Publications du Laboratoire de Statistique et Probabilités, 96-15 (1996) ed.) **8** (2): 438–495. doi:10.1214/aoap/1028903535.
35. ^ Crisan, Dan; Gaines, Jessica; Lyons, Terry (1998). "Convergence of a branching particle method to the solution of the Zakai". *SIAM Journal on Applied Mathematics* **58** (5): 1568–1590. doi:10.1137/s0036139996307371.
36. ^ Crisan, Dan; Lyons, Terry (1997). "Nonlinear filtering and measure-valued processes". *Probability Theory and Related Fields* **109** (2): 217–244. doi:10.1007/s004400050131.

37. ^ Crisan, Dan; Lyons, Terry (1999). "A particle approximation of the solution of the Kushner–Stratonovitch equation". *Probability Theory and Related Fields* **115** (4): 549–578. doi:10.1007/s004400050249.
38. ^ Crisan, Dan; Del Moral, Pierre; Lyons, Terry (1999). "Discrete filtering using branching and interacting particle systems" (PDF). *Markov Processes and Related Fields* **5** (3): 293–318.
39. ^ Del Moral, Pierre; Guionnet, Alice (1999). "On the stability of Measure Valued Processes with Applications to filtering". *C.R. Acad. Sci. Paris* **39** (1): 429–434.
40. ^ Del Moral, Pierre; Guionnet, Alice (2001). "On the stability of interacting processes with applications to filtering and genetic algorithms". *Annales de l'Institut Henri Poincaré*, **37** (2): 155–194. doi:10.1016/s0246-0203(00)01064-5.
41. ^ Ripley 1987
42. ^ *a* *b* Sawilowsky 2003
43. ^ Davenport 1992
44. ^ Vose 2000, p. 13
45. ^ Vose 2000, p. 16
46. ^ "GPU-based high-performance computing for radiation therapy". *Physics in Medicine and Biology* **59**: R151–R182. Bibcode:2014PMB....59R.151J. doi:10.1088/0031-9155/59/4/R151.
47. ^ "Advances in kilovoltage x-ray beam dosimetry". *Physics in Medicine and Biology* **59**: R183–R231. Bibcode:2014PMB....59R.183H. doi:10.1088/0031-9155/59/6/R183.
48. ^ "Fifty years of Monte Carlo simulations for medical physics". *Physics in Medicine and Biology* **51**: R287–R301. Bibcode:2006PMB....51R.287R. doi:10.1088/0031-9155/51/13/R17.
49. ^ Baeurle 2009
50. ^ MacGillivray & Dodd 1982
51. ^ Golden 1979
52. ^ Int Panis et al. 2001
53. ^ Int Panis et al. 2002
54. ^ G. A. Bird, Molecular Gas Dynamics, Clarendon, Oxford (1976)

55. ^ Dietrich, S.; Boyd, I. (1996). "A Scalar optimized parallel implementation of the DSMC technique". *Journal of Computational Physics* **126**: 328–42. Bibcode:1996JCoPh.126..328D. doi:10.1006/jcph.1996.0141.
56. ^ Ojeda & et al. 2009,
57. ^ Milik & Skolnick 1993
58. ^ Sawilowsky & Fahoome 2003
59. ^ http://sander.landofsand.com/publications/Monte-Carlo_Tree_Search_-_A_New_Framework_for_Game_AI.pdf
60. ^ Monte Carlo Tree Search - About
61. ^ Parallel Monte-Carlo Tree Search - Springer
62. ^ http://www.tantrix.com:4321/Tantrix/TRobot/MCTS%20Final%20Report.pdf
63. ^ http://www0.cs.ucl.ac.uk/staff/D.Silver/web/Publications_files/pomcp.pdf
64. ^ Improving Monte–Carlo Tree Search in Havannah - Springer
65. ^ http://www.arimaa.com/arimaa/papers/ThomasJakl/bc-thesis.pdf
66. ^ Szirmay-Kalos 2008
67. ^ Carmona, René; Del Moral, Pierre; Hu, Peng; Oudjane, Nadia (2012). Carmona, René A.; Moral, Pierre Del; Hu, Peng et al., eds. "An Introduction to Particle Methods with Financial Applications". *Numerical Methods in Finance*. Springer Proceedings in Mathematics (Springer Berlin Heidelberg) **12**: 3–49. doi:10.1007/978-3-642-25746-9_1. ISBN 978-3-642-25745-2.
68. ^ "Numerical Methods in Finance - Springer". *link.springer.com*. doi:10.1007/978-3-642-25746-9.
69. ^ Kroese, D. P.; Taimre, T.; Botev, Z. I. (2011). *Handbook of Monte Carlo Methods*. John Wiley & Sons.
70. ^ ***a*** ***b*** Press et al. 1996
71. ^ MEZEI, M (31 December 1986). "Adaptive umbrella sampling: Self-consistent determination of the non-Boltzmann bias". *Journal of Computational Physics* **68** (1): 237–248. Bibcode:1987JCoPh..68..237M. doi:10.1016/0021-9991(87)90054-4.

72. ^ Bartels, Christian; Karplus, Martin (31 December 1997). "Probability Distributions for Complex Systems: Adaptive Umbrella Sampling of the Potential Energy". *The Journal of Physical Chemistry B* **102** (5): 865–880. doi:10.1021/jp972280j.
73. ^ "Sequential Monte Carlo samplers - Del Moral - Doucet - Jasra- 2006 - Journal of the Royal Statistical Society: Series B (Statistical Methodology) - Wiley Online Library". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**: 411–436. doi:10.1111/j.1467-9868.2006.00553.x.
74. ^ Mosegaard & Tarantola 1995
75. ^ Tarantola 2005
76. ^ Shirangi, M. G., History matching production data and uncertainty assessment with an efficient TSVD parameterization algorithm, Journal of Petroleum Science and Engineering, http://www.sciencedirect.com/science/article/pii/S0920410513003227

References [edit]

- Anderson, Herbert L. (1986). "Metropolis, Monte Carlo and the MANIAC" (PDF). *Los Alamos Science* **14**: 96–108.
- Baeurle, Stephan A. (2009). "Multiscale modeling of polymer materials using field-theoretic methodologies: A survey about recent developments". *Journal of Mathematical Chemistry* **46** (2): 363–426. doi:10.1007/s10910-008-9467-3.
- Berg, Bernd A. (2004). *Markov Chain Monte Carlo Simulations and Their Statistical Analysis (With Web-Based Fortran Code)*. Hackensack, NJ: World Scientific. ISBN 981-238-935-0.
- Binder, Kurt (1995). *The Monte Carlo Method in Condensed Matter Physics*. New York: Springer. ISBN 0-387-54369-4.
- Caflisch, R. E. (1998). *Monte Carlo and quasi-Monte Carlo methods*. Acta Numerica **7**. Cambridge University Press. pp. 1–49.
- Davenport, J. H. "Primality testing revisited". *Proceeding ISSAC '92 Papers from the*

international symposium on Symbolic and algebraic computation: 123 129. doi:10.1145/143242.143290. ISBN 0-89791-489-9.

- Doucet, Arnaud; Freitas, Nando de; Gordon, Neil (2001). *Sequential Monte Carlo methods in practice*. New York: Springer. ISBN 0-387-95146-6.
- Eckhardt, Roger (1987). "Stan Ulam, John von Neumann, and the Monte Carlo method" (PDF). *Los Alamos Science, Special Issue* (15): 131–137.
- Fishman, G. S. (1995). *Monte Carlo: Concepts, Algorithms, and Applications*. New York: Springer. ISBN 0-387-94527-X.
- C. Forastero and L. Zamora and D. Guirado and A. Lallena (2010). "A Monte Carlo tool to simulate breast cancer screening programmes". *Phys. In Med. And Biol.* **55** (17): 5213–5229. Bibcode:2010PMB....55.5213F. doi:10.1088/0031-9155/55/17/021.
- Golden, Leslie M. (1979). "The Effect of Surface Roughness on the Transmission of Microwave Radiation Through a Planetary Surface". *Icarus* **38** (3): 451–455. Bibcode:1979Icar...38..451G. doi:10.1016/0019-1035(79)90199-4.
- Gould, Harvey; Tobochnik, Jan (1988). *An Introduction to Computer Simulation Methods, Part 2, Applications to Physical Systems*. Reading: Addison-Wesley. ISBN 0-201-16504-X.
- Grinstead, Charles; Snell, J. Laurie (1997). *Introduction to Probability*. American Mathematical Society. pp. 10–11.
- Hammersley, J. M.; Handscomb, D. C. (1975). *Monte Carlo Methods*. London: Methuen. ISBN 0-416-52340-4.
- Hartmann, A.K. (2009). *Practical Guide to Computer Simulations*. World Scientific. ISBN 978-981-283-415-7.
- Hubbard, Douglas (2007). *How to Measure Anything: Finding the Value of Intangibles in Business*. John Wiley & Sons. p. 46.
- Hubbard, Douglas (2009). *The Failure of Risk Management: Why It's Broken and How to Fix It*.

John Wiley & Sons.

- Kahneman, D.; Tversky, A. (1982). *Judgement under Uncertainty: Heuristics and Biases*. Cambridge University Press.
- Kalos, Malvin H.; Whitlock, Paula A. (2008). *Monte Carlo Methods*. Wiley-VCH. ISBN 978-3-527-40760-6.
- Kroese, D. P.; Taimre, T.; Botev, Z.I. (2011). *Handbook of Monte Carlo Methods*. New York: John Wiley & Sons. p. 772. ISBN 0-470-17793-4.
- MacGillivray, H. T.; Dodd, R. J. (1982). "Monte-Carlo simulations of galaxy systems" (PDF). *Astrophysics and Space Science* (Springer Netherlands) **86** (2).
- MacKeown, P. Kevin (1997). *Stochastic Simulation in Physics*. New York: Springer. ISBN 981-3083-26-3.
- Metropolis, N. (1987). "The beginning of the Monte Carlo method" (PDF). *Los Alamos Science* (1987 Special Issue dedicated to Stanislaw Ulam): 125–130.
- Metropolis, Nicholas; Rosenbluth, Arianna W.; Rosenbluth, Marshall N.; Teller, Augusta H.; Teller, Edward (1953). "Equation of State Calculations by Fast Computing Machines". *Journal of Chemical Physics* **21** (6): 1087. Bibcode:1953JChPh..21.1087M. doi:10.1063/1.1699114.
- Metropolis, N.; Ulam, S. (1949). "The Monte Carlo Method". *Journal of the American Statistical Association* (American Statistical Association) **44** (247): 335–341. doi:10.2307/2280232. JSTOR 2280232. PMID 18139350.
- M. Milik and J. Skolnick (Jan 1993). "Insertion of peptide chains into lipid membranes: an off-lattice Monte Carlo dynamics model". *Proteins* **15** (1): 10–25. doi:10.1002/prot.340150104. PMID 8451235.
- Mosegaard, Klaus; Tarantola, Albert (1995). "Monte Carlo sampling of solutions to inverse problems". *J. Geophys. Res.* **100** (B7): 12431–12447. Bibcode:1995JGR...10012431M.

doi:10.1029/94JB03097.

- P. Ojeda and M. Garcia and A. Londono and N.Y. Chen (Feb 2009). "Monte Carlo Simulations of Proteins in Cages: Influence of Confinement on the Stability of Intermediate States". *Biophys. Jour.* (Biophysical Society) **96** (3): 1076–1082. Bibcode:2009BpJ....96.1076O. doi:10.1529/biophysj.107.125369.
- Int Panis L; De Nocker L, De Vlieger I, Torfs R (2001). "Trends and uncertainty in air pollution impacts and external costs of Belgian passenger car traffic International". *Journal of Vehicle Design* **27** (1–4): 183–194. doi:10.1504/IJVD.2001.001963.
- Int Panis L, Rabl A, De Nocker L, Torfs R (2002). P. Sturm, ed. "Diesel or Petrol ? An environmental comparison hampered by uncertainty". *Mitteilungen Institut für Verbrennungskraftmaschinen und Thermodynamik* (Technische Universität Graz Austria). Heft 81 Vol 1: 48–54.
- Press, William H.; Teukolsky, Saul A.; Vetterling, William T.; Flannery, Brian P. (1996) [1986]. *Numerical Recipes in Fortran 77: The Art of Scientific Computing*. Fortran Numerical Recipes **1** (Second ed.). Cambridge University Press. ISBN 0-521-43064-X.
- Ripley, B. D. (1987). *Stochastic Simulation*. Wiley & Sons.
- Robert, C. P.; Casella, G. (2004). *Monte Carlo Statistical Methods* (2nd ed.). New York: Springer. ISBN 0-387-21239-6.
- Rubinstein, R. Y.; Kroese, D. P. (2007). *Simulation and the Monte Carlo Method* (2nd ed.). New York: John Wiley & Sons. ISBN 978-0-470-17793-8.
- Savvides, Savvakis C. (1994). "Risk Analysis in Investment Appraisal". *Project Appraisal Journal* **9** (1). doi:10.2139/ssrn.265905.
- Sawilowsky, Shlomo S.; Fahoome, Gail C. (2003). *Statistics via Monte Carlo Simulation with Fortran*. Rochester Hills, MI: JMASM. ISBN 0-9740236-0-4.
- Sawilowsky, Shlomo S. (2003). "You think you've got trivials?" (PDF). *Journal of Modern*

Applied Statistical Methods **2** (1): 218–225.

- Silver, David; Veness, Joel (2010). "Monte-Carlo Planning in Large POMDPs" (PDF). In Lafferty, J.; Williams, C. K. I.; Shawe-Taylor, J.; Zemel, R. S.; Culotta, A. *Advances in Neural Information Processing Systems 23*. Neural Information Processing Systems Foundation.
- Szirmay-Kalos, László (2008). *Monte Carlo Methods in Global Illumination - Photo-realistic Rendering with Randomization*. VDM Verlag Dr. Mueller e.K. ISBN 978-3-8364-7919-6.
- Tarantola, Albert (2005). *Inverse Problem Theory*. Philadelphia: Society for Industrial and Applied Mathematics. ISBN 0-89871-572-5.
- Vose, David (2008). *Risk Analysis, A Quantitative Guide* (Third ed.). John Wiley & Sons.

Wikimedia Commons has media related to ***Monte Carlo method***.

V · T · E ·

Statistics [hide]

Outline · Index ·

Descriptive statistics [show]

Data collection [show]

Statistical inference [show]

Correlation · Regression analysis · [show]

Categorical / Multivariate / Time-series / Survival analysis [show]

Applications [show]

Category · Portal · Commons · WikiProject ·

Authority control NDL: 00567842 ·

Categories: Monte Carlo methods | Numerical analysis | Statistical mechanics | Computational physics | Sampling techniques | Statistical approximations | Stochastic simulation | Probabilistic complexity theory | Risk analysis | Risk management

This page w as last modified on 20 September 2015, at 06:34.

Privacy policy About Wikipedia Disclaimers Contact Wikipedia Developers Mobile view