

Monte Carlo Simulations in the Teaching Process

Blanka Šedivá

Department of Mathematics, Faculty of Applied Sciences
University of West Bohemia, Plzeň, Czech Republic



CADGME 2018

Conference on Digital Tools in Mathematics Education
25-29 June 2018
University of Coimbra, Portugal

Outline

- 1 Motivation
- 2 The basic philosophy of Monte Carlo methods
- 3 The classic Monte Carlo applications
 - Calculation of area
 - Demonstration of unbiasedness and interval estimation
 - Buffon's needle
- 4 Modern Monte Carlo applications
 - Monte Carlo Solution of PDEs
- 5 Conclusion

Motivation

- Deterministic and stochastic problems are usually taught separately, without linking.
- Monte Carlo simulations (MCSs) are procedures for solving **nonprobabilistic-type problems** (problems whose outcome does not depend on chance) by **probabilistic-type methods** (methods whose outcome depends on chance).
- MCSs can be a valuable pedagogical tool for **several types of courses** - probability, statistics, econometrics but also basic and advanced courses in mathematical analysis.
- Today's, the using MCSs can help to understand the basic principles of some real problem but it also supports to train students to **practical programming in computers**.



A Brief History of Monte Carlo Methods

- The term „Monte Carlo“ was apparently first used by Ulam and von Neumann as a Los Alamos code word for the stochastic simulations they applied to building better atomic bombs.
- One of the first documented Monte Carlo experiments is Buffon's needle experiment (1733).
- In physics or economical-related problems, Monte Carlo methods are useful for simulating systems with many coupled degrees of freedom, such as fluids, disordered materials, strongly coupled solids, cellular structures or calculation of trade price of the derivatives.

The basic philosophy of Monte Carlo methods

- MCSs is a part of experimental mathematics.
- At the first step, the problem is described in the deterministic and the stochastic formulation and the connection between this two views are established.
- Next the repeated numerical experiments based on the random number are proceeded.
- By the law of large numbers, the expected value of some random variable can be approximated by taking the empirical mean (the sample mean) of independent samples of the variable.

Random samples

- For MCSs we need to be able to reproduce randomness by a computer algorithm.
- A pseudo-random number generator (RNG) is an algorithm for whose output the $U[0, 1]$ distribution is a suitable model.
- The number generated by the pseudo-random number generator should have the same relevant statistical properties as independent realisations of a $U[0, 1]$ random variable.
- The ability of a Monte Carlo method to work depends on the quality random numbers used.
- Random Number Generators in SW Excel: `=rand()`, GeoGebra: `RandomBetween[ 1, 10 ]`, MATLAB: `rand()`, `rng(s)`, PYTHON: `random()`, `seed()`, C++ `rand()`



Calculation of area

- To compute an MCSs of π using a simple experiment.
- Assume that we could produce „uniform rain“ on the square $[-1, 1] \times [-1, 1]$ such that the probability of a raindrop falling into a region $\Omega \subset [-1, 1]^2$ is proportional to the area of Ω , but independent of the position of Ω .

- The probability that a raindrop falls into the unit circle is

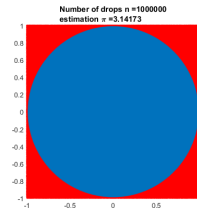
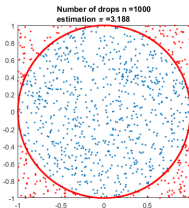
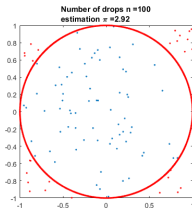
$$\mathbb{P}(\text{drop within circle}) = \frac{\text{area of the unit circle}}{\text{area of the square}} = \frac{\pi}{2 \cdot 2}.$$

- $\pi = 4 \cdot \mathbb{P}(\text{drop within circle})$
- The probability $\mathbb{P}$ is estimated using our raindrop experiment.

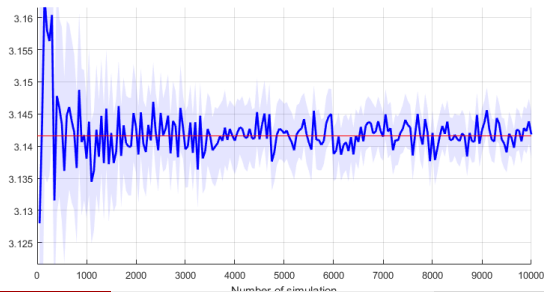
$$\hat{\mathbb{P}} = \frac{\text{number of drops inside the circle}}{\text{total number of drops}}.$$



Calculation of area, cont.



MCSs of π (with 95% confidence interval)



Demonstration of unbiasedness

- If random variables X_i are normally distributed iid with mean μ and standard deviation σ then $\bar{X}$ is the unbiased estimator of μ and s is bias estimator of σ .

- $(\hat{\mu} =) \dots \bar{X} = \frac{1}{n} \sum_{i=1}^n x_i, \quad (\hat{\sigma} =) \dots s = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{X})^2}$

- *In statistics, the bias (or bias function) of an estimator is the difference between this estimator's expected value and the true value of the parameter being estimated. An estimator or decision rule with zero bias is called unbiased.*

https://en.wikipedia.org/wiki/Bias_of_an_estimator

- $\mathbb{E}(\bar{X}) - \mu = 0 \quad \mathbb{E}(S) - \sigma \neq 0 \quad \mathbb{E}(S) = \frac{n}{n-1}\sigma$



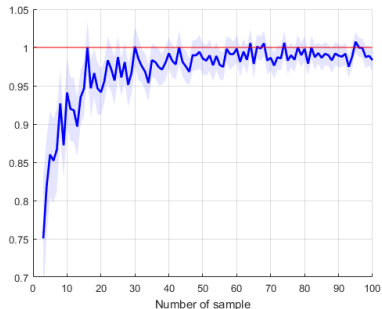
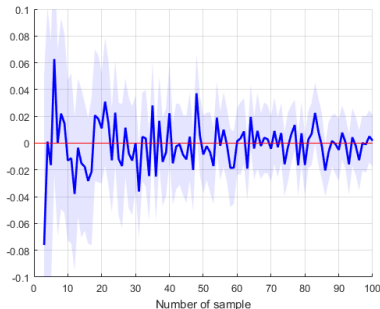
Demonstration of unbiasedness, cont.

- For fixed n we simulate $x_1, x_2, \dots, x_n$ as a realization of $X \sim N(\mu, \sigma)$
- We compute $\bar{x}$ and s as a estimation of parameters μ and σ .
- We repeat simulation process and at the end, we calculate mean of your estimated $\hat{\mu}$ and $\hat{\sigma}$.

simulation	sample	$\hat{\mu}$	$\hat{\sigma}$
1	$x_{11}, x_{12}, \dots, x_{1n}$	$\bar{x}_1$	s_1
2	$x_{21}, x_{22}, \dots, x_{2n}$	$\bar{x}_2$	s_1
...	...	...	...
k	$x_{k1}, x_{k2}, \dots, x_{kn}$	$\bar{x}_k$	s_1
mean		$\widehat{EX}$	$\widehat{ES}$

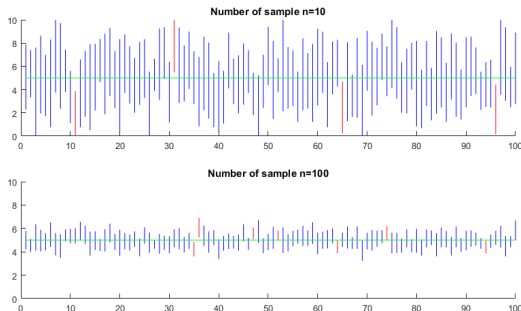
Demonstration of unbiasedness, cont.

MCSs demonstration of estimation μ and σ based on iid n samples of normal distributed variable $X \sim N(0, 1)$



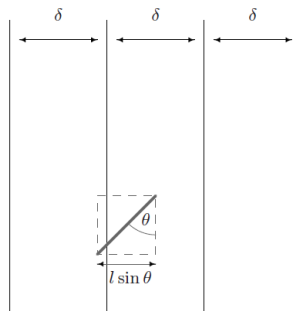
Demonstration of interval estimation

- Confidence intervals consist of a range of potential values of the unknown population parameter.
- A 95% confidence interval does not mean that 95% of the sample data lie within the interval.
- For Gaussian distribution $\mathbb{P} \left(\bar{x} - t^* \frac{s}{\sqrt{n}} < \mu < \bar{x} + t^* \frac{s}{\sqrt{n}} \right) = 1 - \alpha$



Buffon's needle

In 1733, the Comte de Buffon, George Louis Leclerc, asked the following question (Buffon, 1733): Consider a floor with equally spaced lines, a distance δ apart. What is the probability that a needle of length $l < \delta$ dropped on the floor will intersect one of the lines? Buffon answered the question himself in 1777 (Buffon, 1777).



(a) Illustration of the geometry behind Buffon's needle



(b) Results of the Buffon's needle experiment using 50 needles. Dark needles intersect the thin vertical lines

Solution of the problem „Buffon's needle“

- Assume the needle landed such that its angle is θ (see previous figure 1.5).
- Then the question whether the needle intersects a line is equivalent to the question whether a box of width $l \sin \theta$ intersects a line.
- The probability of this happened is $\mathbb{P}(\text{intersect}) = \frac{l \sin \theta}{\delta}$
- Assuming that the angle θ is uniform on $[0, \pi)$ we obtain
$$\mathbb{P}(\text{intersect}) = \int_0^\pi \mathbb{P}(\text{intersect}|\theta) \frac{1}{\pi} d\theta = \int_0^\pi \frac{l \sin \theta}{\delta} \frac{1}{\pi} d\theta = \frac{2l}{\pi \delta}$$
- When dropping n needles the expected number of needles crossing a line is thus $X = \frac{n2l}{\pi \delta}$.

Modern Monte Carlo applications

- Methods for the solving partial differential and integral equations based on random walks.
- Credit Risk and the Valuation of Corporate Securities. Valuation of Credit Insurance Portfolios
- Computing ValueAtRisk (VaR) and Principal Components Analysis (PCA) by MCSs.
- Stochastic DEs and PDEs. For example calculation of pricing options (European, American and Exotic options).
- Application in Linear Algebra, that MCSs can be used to approximate sums of huge number of terms such as high-dimensional inner products.



Monte Carlo Solution of PDEs

- Approximate solution of the Dirichlet problem.

- Problem : to find $u(x, y)$ that satisfies

$$\text{PDE: } u_{xx} + u_{yy} = 0 \quad x \in \Omega$$

$$\text{BC: } u(x, y) = g(x, y) \quad x \in \partial\Omega$$

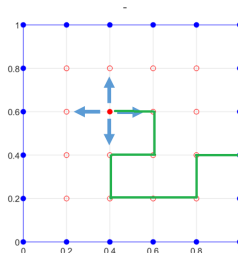
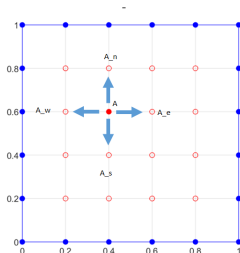
- $\Omega = \{[x, y] \subset \mathbb{R}^2 : 0 < x < 1, 0 < y < 1\}$

- $g(x, y) = \begin{cases} 1 & \text{on the top of the square} \\ 0 & \text{on the sides and bottom of the square} \end{cases}$

To illustrate the Monte Carlo method in this problem, we introduce a game called *tour du wino*. To play it, we need a board on which grid lines are drawn on the next slide.



Random walk process



- The process starts from an arbitrary point (fill red point).
- At each stage of the game, the wino staggers off randomly to one of the four neighbouring points.
- The process wandering from point to point until eventually hitting a boundary point (blue point).

Simulation process

- For every interior point we repeat many random walk.
- Keep track of how many times you hit each boundary point.
- Compute the fraction of times you have ended at each boundary points p_i .
- Suppose that the goal of the game is to compute his average value $U(A)$ for all this walks.
- The average reward is

$$U(A) = \sum g_i \mathbb{P}(p_i)$$

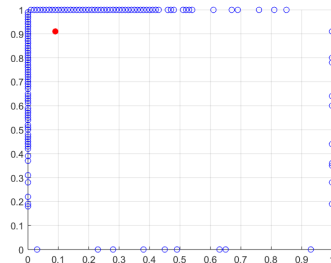
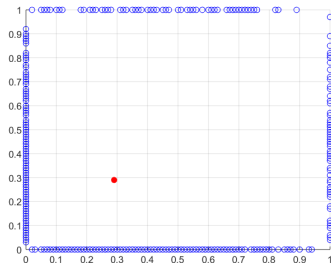
where g_i is value of function $g(x, y)$ at boundary point p_i ,
 $\mathbb{P}(p_i)$ is probability that random walk finish at point



- Probability is estimated by the fraction of times.

Results of simulation process

The ending points for $n = 100$ simulation random walk started at the red point.

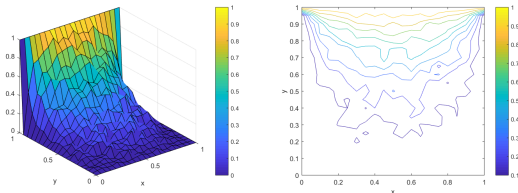


The relation between the MCSs and solving of the Laplace problem.

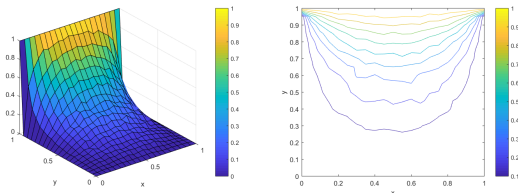
- Average value $U(A)$ is clearly the average of the four average values of the four neighbours.
- $U(A) = \frac{1}{4} (U(A_N) + U(A_S) + U(A_W) + U(A_E))$
- $U(A)$ corresponds to $u_{i;j}$ in the finite difference equations
$$u_{i;j} = \frac{1}{4} (u_{i-1;j} + u_{i+1;j} + u_{i;j-1} + u_{i;j+1}) \quad \text{for } (i;j) \text{ interior point}$$
$$u_{i;j} = g_{i;j} \quad \text{for } (i;j) \text{ boundary point}$$
- $U(A)$ will approximate the true solution of the PDE at the point A .

The Monte Carlo solution of PDEs

The step for grid 0.025 and number of simulations $s = 100$



The step for grid 0.025 and number of simulations $s = 500$



Conclusion

The Monte Carlo methods are useful for explain not only probabilistic and statistics problems but also for geometric interpretations problems from mathematical analysis.

Thank you for your attention.