MATH-414
Stochastic simulation

Prof. Fabio Nobile

A.Y. 2025-2026 – Fall semester

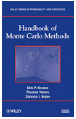
Course organization

- Lectures ex cathedra (theory – mostly on slides):
  - **Wednesday** 10:15 – 12:00, room **CE 15**
- Lab sessions (mostly computer based exercises):
  - **Thursday** 8:15 – 10:00, room **CE 1 105**
  - The esercise series are mainly computer based but the assigned room is not equipped with computers. We ask you to **bring your own laptop** (BYOD). If this is a problem, contact us.
  - **Answer poll on moodle** about the day/time of Lab sessions
- Assistants
  - **Eliott van Dieren** (`eliott.vandieren@epfl.ch`) – main assistant
  - Jakob Linnestad Sønstebø – student assistant
- All material available on the *moodle* web site

Course material

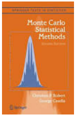
Lecture notes available on moodle

Reference books:

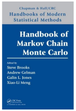


D.P. Kroese, T. Taimre, Z.I. Botev. *Handbook of Monte Carlo Methods*. John Wiley and Sons. 2011

S. Asmussen, P.W. Glynn. *Stochastic Simulation: Algorithms and Analysis*. Springer. 2007



C.P. Robert, G. Casella. *Monte Carlo statistical methods*. Springer. 2004



S. Brooks, A. Gelman, G.L Jones, X.L. Meng. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC. 2011

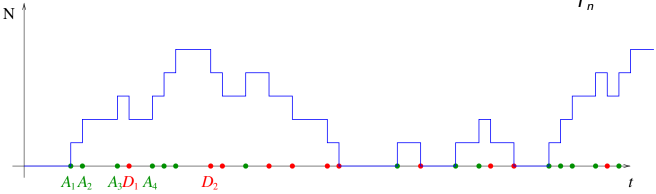
Motivating Examples

G/G/1 queueing model

Customers arrive at a (single) server, enter the queue and are served in a "first-in-first-out" (FIFO) queue discipline

- A_n: arrival time of the n-th customer
- W_n: waiting time of the n-th customer
- V_n: service time on the n-th customer (once s/he reaches the service)
- $D_n = A_n + W_n + V_n$: departure time of the n-th customer
- (*Lindley recursion*)

$$W_{n+1} = (D_n - A_{n+1})_+ = (W_n + V_n - \underbrace{(A_{n+1} - A_n)}_{T_n})_+$$



The process is fully characterized by the inter-arrival times $T_n = A_{n+1} - A_n$ and the service times V_n.

G/G/1 queueing model

In a G/G/1 queueing model the inter-arrival times $\{T_n\}_{n\geq 0}$ and the service times $\{V_n\}_{n>0}$ are independent random variables with General distribution (i.e. not exponential).

If the distribution of $\{T_n\}$ and $\{V_n\}$ is exponential (or Memoryless or Markovian), the queueing model is denoted M/M/1.

Interesting questions:

- What is the average waiting time $\mathbb{E}[W_\infty]$ at steady state (provided it exists)?
- What is the average number of customers in the queue $\mathbb{E}[Q(t)]$ at time t or at steady state, $\mathbb{E}[Q_\infty]$?
- What is the probability that the queue exceeds a critical length $\mathbb{P}(Q(\infty) > Q_{cr})$ at steady state?
- What is the probability that the waiting time of a customer exceeds a critical value $\mathbb{P}(W_\infty > W_{cr})$?

G/G/n queueing model

For the Markovian time homogeneous case M/M/1 there are theoretical answers to the above questions. However, for the general G/G/n case, this is not the case. Hence, one could try to give an (approximate) answer by simulation

- Generate many realizations of the queueing model;
- Use these realizations to estimate the average waiting time, queue length,etc. (Monte Carlo method)

Practical questions

- How can we simulate a random process on a computer?
- How many realizations do we need to have an accurate estimation of the above expectations? (i.e. how to control the accuracy of a Monte Carlo analysis)

Consider the problem of estimating $\mathbb{P}(Q(\infty) > Q_{cr})$ where Q_{cr} is a large value. This is likely a rare event. Assuming that the probability is of the order 10^{-3}, on average, 1 out of 1000 realizations will feature a long queue and if we want a reliable estimation of the probability we will probably need to run millions of realizations.

- Can we do better? i.e. can we improve the Monte Carlo method so that fewer realizations are needed (for instance by exploiting theoretical results available in the M/M/1 case)?

Computational finance – insurance risk

Cramér-Lundberg model for insurance risk:

- claims arrive according to a Poisson process $\{N(t),\ t > 0\}$ (interarrival times are independent and exponentially distributed)
- claim sizes $V_1, V_2, \ldots$ are iid random variables, independent of $\{N(t),\ t > 0\}$
- premiums come in at a continuous rate c

The amount by which claims exceed premiums at time t is

$$S(t) = \sum_{i=1}^{N(t)} V_i - ct$$

Interesting question: If x is the initial reserve and $\tau(x) = \inf\{t > 0 : \ S(t) > x\}$ is the first time at which S exceeds x (ruin has occurred): what is the probability of ruin?

$$\mathbb{P}(\tau(x) < \infty)$$

This problem can be recast to a M/G/1 queueing model.

Computational finance – option pricing

The evolution of the price $S(t)$ of an asset is often modeled (under the risk-neutral probability measure) by a stochastic differential equation as

$$S(t) = e^{X(t)}, \qquad dX(t) = rdt + \sigma dW$$

where $\{W(t)\}_{t \geq 0}$ is a Wiener process (Brownian motion) or more generally a Lévy process (combination of Brownian motion and a jump process)

A call option is a contract that gives the holder the right (but not the obligation) to buy a certain amount of a given asset at the price K (strike) at time T (maturity).
Then the correct price of the option is given by

$$\text{option price} = e^{-rT}\mathbb{E}[(S(T) - K)_+]$$

For problems where an exact solution is not available, the option price can be estimated by simulation.

Computational statistics – likelihood ratio test

Let $\mathbf{X} = (X_1, \ldots, X_n)$ be a random sample from a population with probability density $f(x|\theta)$, where θ is a parameter. The likelihood function is $L(\theta|\mathbf{x}) = \prod_{i=1}^n f(x_i|\theta)$.

We want to test the hypothesis $H_0 : \theta = \theta_0$ using a likelihood ratio test

$$\lambda(\mathbf{x}) = \frac{L(\theta_0|\mathbf{x})}{L(\hat{\theta}|\mathbf{x})}$$

where $\hat{\theta}$ is the maximum likelihood estimator.

Then, one may use the test statistic $T(\mathbf{x}) = -2\log\lambda(\mathbf{x})$ such that H_0 is rejected at level α if $T(\mathbf{x}) > t_\alpha$ where t_α is chosen so that $\mathbb{P}_{\theta_0}(T \leq t_\alpha) = 1 - \alpha$.

The explicit computation of t_α is often not feasible so one may use simulation to estimate t_α.

Computational statistics – Bayesian inference

Let $\mathbf{X} = (X_1, \ldots, X_n)$ be a random sample from a population with probability density $f(x|\theta)$, where θ is an unknown parameter to be estimated from the sample.

In Bayesian inference, the parameter θ is interpreted as a random variable itself, following a prior probability distribution with density $\pi(\theta)$, which represents our subjective belief about θ.

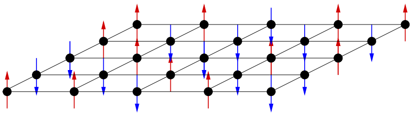
When information is available (sample $\mathbf{X}$), the prior distribution is updated into the posterior distribution $\pi_{post}(\theta)$ using Bayes' rule

$$\pi_{post}(\theta) = \pi(\theta|\mathbf{x}) = \frac{L(\theta|\mathbf{x})\pi(\theta)}{\int L(\theta|\mathbf{x})\pi(\theta)d\theta} \propto L(\theta|\mathbf{x})\pi(\theta)$$

The major question is how to sample from the posterior distribution or compute the posterior mean $\mathbb{E}_{\pi_{post}}[\theta]$ or other related quantities.

Computational physics – Ising model

Consider the configuration of atoms in the figure, where the atom in the (i,j)-position of the lattice can have spin in either of the two states $s_{ij} = +1$ (up) or $s_{ij} = -1$ (down).



Each atom interacts with its neighbors. The total energy of the system $S = \{s_{ij}\}$ is given by

$$H(S) = -\sum_{i,j} s_{ij}(s_{i-1,j} + s_{i+1,j} + s_{i,j-1} + s_{i,j+1})$$

and the probability of finding the system in a given state S is given by the Boltzmann distribution

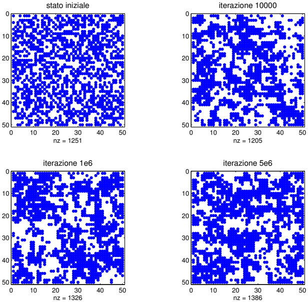
$$p(S) = \frac{1}{Z}\exp\{-H(S)/kT\}$$

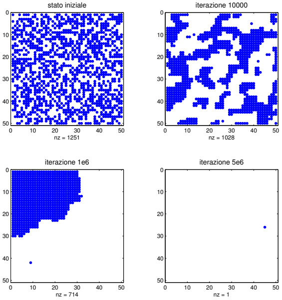
where Z is the normalization constant (partition function), T the temperature and k the Boltzmann constant.

Computational physics – Ising model

One might be interested in computing several quantities as e.g. the average magnetic moment $\mathbb{E}[M]$ where $M(S) = |\sum_{ij} s_{ij}|$.

Given the huge number of possible states of the system, a direct computation of such an expectation is unfeasible. Hence, one can resort to simulation.





Contents of the course

- Random variable generation
- Simulation of random processes
- Monte Carlo method; output analysis
- Variance reduction techniques (antithetic variables, control variables, importance sampling, stratification, ...)
- Quasi Monte Carlo methods
- Markov Chain Monte Carlo methods (Metropolis-Hasting, Gibbs sampler)
- Derivative estimation and Stochastic optimization

Exercise sessions

- The exercises will be mostly computer based.
- The text of the exercises will be available on the moodle web page one day before the exercise session.
- Solutions of some (but not all) of the exercises will be made available on moodle one week after the exercise session.
- The software chosen for this course is Python.

Exam

- The final grade is based on a project (40% of the grade) and a written exam (60% of the grade)
- A list of projects will be given in the second half of the semester. A report has to be handed in early January (precise date will be communicated later).
- The project can be done in group of up to three students.
- The written exam will contain exercises similar to those of the exercise series, with a mix of theoretical questions and computer based ones. It will be in a computer room.

Questions?