Advanced Numerical Algorithms (MU4IN920)

Lecture 1: Monte Carlo Computations

Stef Graillat

Sorbonne Université

Course

Lecturer: Stef Graillat (office 26-00/313)
`stef.graillat@sorbonne-universite.fr`

Assignments, exams and grading

- 1 exam (50 %) + practicals (50 %)

Schedule

- 5 lectures: wednesday 8h30-10h30
- 5 tutorials: wednesday 10h45-12h45
- 5 practicals as a personal work in pairs
- Practicals due one week after
  → report written in LaTeX and submitted in one pdf file on Moodle

Aims of the course

Goals:

- Mathematical concept: mathematical definition of concepts and quantities
- Algorithm: how to efficiently calculate these quantities on a computer (via the use of MATLAB)?
- Problem solving: use concepts and algorithms to solve real-life problems

General course outline

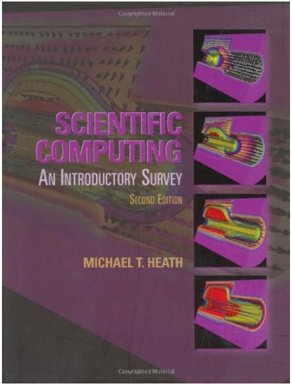
Advanced Numerical Algorithms

1. Monte Carle computations
2. Iterative methods for linear systems
3. Fast Fourier Tranforms
4. Error analysis in floating-point arithmetic
5. Numerical solutions of ordinary differential equations (EDO)

Main references

- **A First Course in Numerical Methods, Uri M. Ascher, Chen Greif, SIAM, 2011**
- **Scientific Computing, An Introductory Survey, Michael T. Heath, McGraw-Hill, 2002**
- **Scientific Computing with Case Studies, Dianne P. O'Leary, SIAM, 2009**
- **Introduction to Scientific Computing and Data Analysis, Mark H. Holmes, Springer, 2016**
- Mathématiques appliquées L3, sous la direction de Jacques-Arthur Weil et Alain Yger, Pearson, 2009
- Numerical Computing with MATLAB, Cleve Moler, SIAM, 2004
- MATLAB Guide, Desmond J. Higham, Nicholas J. Higham, 3e édition, SIAM, 2017
- Scientific Computing, An Introduction using Maple and MATLAB, Walter Gander, Martin Gander, Felix Kwok, Springer, 2014
- Numerical Recipes. The Art of Scientific Computing, William Press, Saul Teukolsky, William Vetterling et Brian Flannery, 3rd Edition, Cambridge University Press, 2007

Main references

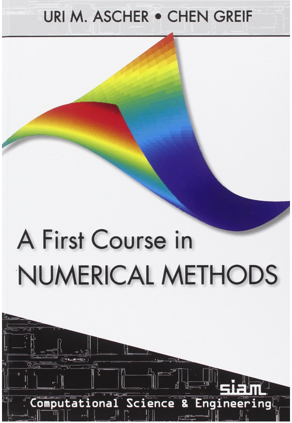


SCIENTIFIC COMPUTING

AN INTRODUCTORY SURVEY

SECOND EDITION

MICHAEL T. HEATH

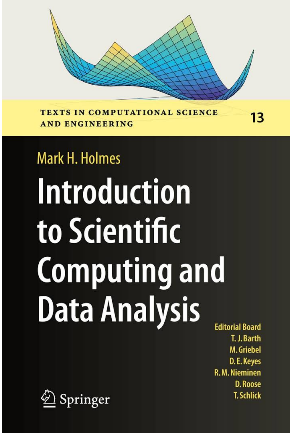


URI M. ASCHER • CHEN GREIF

A First Course in NUMERICAL METHODS

siam

Computational Science & Engineering



TEXTS IN COMPUTATIONAL SCIENCE AND ENGINEERING 13

Mark H. Holmes

Introduction to Scientific Computing and Data Analysis

Springer

Aims of this course

1. Basic probability and statistics
2. Principles of Monte Carlo methods
3. Monte Carlo methods for numerical integration

Outline of the lecture

1. Basic statistics: Random and pseudorandom numbers and their generation
2. Monte Carlo methods for numerical integration
3. Monte Carlo methods for optimization
4. Monte Carlo methods for counting
5. Introduction to derivatives in finance
6. Computing the price of a derivative (options)

Bibliography

- **Scientific Computing with Case Studies, Dianne P. O'Leary, SIAM, 2009** (the lecture is mainly based on this book and on the associated slides)
- Explorations in Monte Carlo Methods, R.W. Shonkwiler and F. Mendivil, Springer, 2009
- Monte Carlo Methods, M.H. Kalos and P.A. Whitlock, 2nd edition, Wiley-VCH, 2008
- Monte Carlo Strategies in Scientific Computing, Jun S. Liu, Springer, 2001
- An Introduction to Financial Option Valuation: Mathematics, Stochastics and Computation, Desmond J. Higham, Cambridge University Press, 2004
- Option (finance), Wikipedia, `https://en.wikipedia.org/wiki/Option_(finance)`

What is a Monte-Carlo method?

Definition 1

In a Monte-Carlo method, the desired answer is formulated as a quantity in a stochastic model and estimated by random sampling of the model.

Nicholas Metropolis suggested using the name Monte Carlo in 1947 which refers to the Monte Carlo Casino in Monaco

Example:

- If we have a cube, with the sides numbered 1 to 6, we might toss the cube 120 times, observe which side comes up on top each time, and study whether the sides occur with approximately equal frequency.
- If we have a ‘‘black box’’ that takes a number between 0 and 1 as input and emits a number between 0 and 1 as an output, we could feed the box m numbers and observe the m outputs of the box and use the average of the observations as an estimate of the statistical mean of the process defined by the black box.

Applications of Monte Carlo methods

The development of the Monte Carlo methods mainly took place during the Second World War for research on the manufacture of the atomic bomb within the framework of the Manhattan Project.

Use of these probabilistic methods to solve partial differential equations in the framework of the Monte-Carlo N-Particle [MCNP].

Some current applications:

- Computation of multiple integral
- Particle filter: application in filtering problems arising in signal processing, image processing
- Finance: option valuation
- Bioinformatics: computing phylogenetic tree from genomic data
- Raytracing

Two basic principles

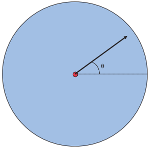
- There is an important difference between
  - Monte Carlo methods, which estimate quantities by random sampling, and
  - pseudo-Monte Carlo methods, which use samples that are more systematically chosen.

  In some sense, all practical computational methods are pseudo-Monte Carlo, since random number generators implemented on machines are generally not truly random. So the distinction between the methods is a bit fuzzy. But we'll use the term Monte Carlo for samples that are generated using pseudorandom numbers generated by a computer program.
- Monte Carlo methods are (at least in some sense) methods of last resort. They are generally quite expensive and only applied to problems that are too difficult to handle by deterministic (non-stochastic) methods.

Examples of how to generate random numbers

- Take n papers and number them 1 to n. Put them in a box, and draw one at random. After you record the resulting number, put the paper back in the box. You are taking random numbers that are uniformly distributed among the values $\{1, 2, \ldots, n\}$

- Make a spinner by anchoring a needle at the center of a circle. Draw a radius line on the circle. Spin the needle, and measure the angle it forms with the radius line. You obtain random numbers that are uniformly distributed on the interval $[0, 2\pi]$



Examples of how to generate random numbers (cont'd)

- If, on average, a radioactive substance emits α-particles every μ seconds, then the time between two successive emissions has the exponential distribution with mean μ
- The normal distribution is a good model in many situations:
  - Physical characteristics of plants and animals (height, weight, etc.)
  - Velocity distribution of molecules in a thermodynamic equilibrium (Maxwell-Boltzmann distribution)

Properties of samples from random distributions

Definition 2 (Mean)

The mean or average value or expected value of a set of samples $\{x_i\}$, $i = 1, \ldots, n$ *is*

$$\mu_n = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Definition 3 (Variance)

The variance of a set of samples $\{x_i\}$, $i = 1, \ldots, n$ *is*

$$\sigma_n^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu_n)^2$$

Properties of distributions

Let X be a real-valued random variable

Definition 4 (density)

We say that X has a probability density function $f : \Omega \to \mathbb{R}^+$ with $\Omega \subset \mathbb{R}$ if

$$\int_\Omega f(x)dx = 1$$

and

$$P(X \in B) = \int_B f(x)dx \quad \text{for } B \subset \Omega$$

Properties of random variables

Let X be a real-valued random variable with probability density function f

Definition 5 (Mean and variance)

The mean μ of a distribution and variance of σ^2 the distribution X are defined by

$$\begin{aligned} \mu &= \int_\Omega x f(x) dx \\ \sigma^2 &= \int_\Omega (x-\mu)^2 f(x) dx \end{aligned}$$

We also denote $\mathrm{E}[X]$ for μ and $\mathrm{Var}(X)$ for σ^2

Example distribution functions

- The uniform distribution over the interval $[0, m]$:

$$\begin{aligned} f(x) &= \frac{1}{m} \\ \mu &= \int_0^m \frac{x}{m} dx = \frac{m}{2} \\ \sigma^2 &= \int_0^m \frac{1}{m}\left(x - \frac{m}{2}\right)^2 dx = \frac{m^2}{12} \end{aligned}$$

- The exponential distribution with parameter μ over the interval $[0, +\infty[$:

$$\begin{aligned} f(x) &= \frac{1}{\mu} e^{-x/\mu} \\ \mu &= \mu \\ \sigma^2 &= \mu^2 \end{aligned}$$

Example distribution functions (cont'd)

- The normal distribution with parameters μ and σ over the interval $]-\infty,+\infty[$:

$$\begin{aligned} f(x) &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)} \\ \mu &= \mu \\ \sigma^2 &= \sigma^2 \end{aligned}$$

The world is normal

Central Limit Theorem

Let $f(x)$ be any distribution with mean μ and (finite) variance σ^2. Take a random sample of size n from $f(x)$, and call the mean of the sample x_n. Then the random variable x_n approaches the normal distribution.

More precisely, if we denote

$$y_n = \sqrt{n}\frac{\overline{x}_n - \mu}{\sigma},$$

then the distribution for y_n approaches the normal distribution with mean 0 and variance 1 as n increases.

Generating “random” numbers

- In principle, we could generate random samples as discussed above. For example, when we wanted a uniform distribution, we could build a spinner and play with it for a while, writing down our list of samples.
- In practice, when the random numbers are used in computer software, this does not work well. Because of this, we use pseudorandom numbers in computer software.
- Pseudo-random numbers are generated by the computer using a deterministic (i.e., reproducible) procedure and appear to be random, in the sense that the mean, variance, and other properties of sequences of n samples match what you would expect to obtain from a random process
- But the pseudorandom numbers actually cycle; i.e., if you ask for a long enough sequence, you will see periodicity.
- Random number generators on computers usually use a seed (a number) to determine where in the cycle to begin. Thus, if other people wanted to reproduce your results, they would simply use the same seed.

Generating “random” numbers (cont’d)

- It is fairly cheap to generate uniformly distributed pseudorandom numbers.
- Samples from other distributions are usually generated by taking two or more uniformly distributed pseudorandom numbers and manipulating them using function
- In MATLAB, use `rand` to generate random numbers that are uniformly distributed in the interval $(0, 1)$ and `randn` to generate random numbers that are normally distributed with mean 0 and variance 1

So now we know that standard software exists for generating pseudorandom numbers

From now on, we willll abbreviate by leaving the pseudo prefix off of random numbers

And we will switch our focus from what random numbers are to how random numbers help us solve problems in computational science

Monte Carlo methods for numerical integration

Problem: we want to estimate the value

$$I = \int_0^1 \cdots \int_0^1 f(x_1, \ldots, x_{10}) p(x_1, \ldots, x_{10}) dx_1 \cdots dx_{10} = \int_\Omega f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

Notation:

- $\mathbf{x} = [x_1, \ldots, x_{10}]$
- $\Omega = [0,1] \times \cdots \times [0,1]$
- usually $p(\mathbf{x})$ is a constant, equal to 1 divided by the volume of Ω but p may be a more general function.

  p need ot be a probability density function so it should be nonnegative and satisfy

  $$\int_\Omega p(\mathbf{x}) d\mathbf{x} = 1$$

→ How might we approach the problem of computing I?

Method 1: interpolation

Idea: interpolate $f(\mathbf{x})p(\mathbf{x})$ by a polynomial and then integrate it analytically.

For example, a polynomial of degree 2 in each variable would have terms of the form

$$x_1^{[]}x_2^{[]}x_3^{[]}x_4^{[]}x_5^{[]}x_6^{[]}x_7^{[]}x_8^{[]}x_9^{[]}x_{10}^{[]}$$

where the number in each box [] is 0, 1, or 2. So it has $3^{10} = 59049$ coefficients and we would need 59049 function values to determine these.

One needs to divide the region into small boxes so that a polynomial is a good approximation within each box.

If we divide the interval $[0,1]$ into 5 pieces, we make 5^{10} boxes, with 59049 function evaluations in each!

This method is expensive!

Method 2: product rules

Some functions $f(\mathbf{x})p(\mathbf{x})$ can be well approximated by a separable function

$$f(\mathbf{x})p(\mathbf{x}) \approx f_1(x_1)f_2(x_2)\cdots f_{10}(x_{10})$$

In that case we can approximate our integral by

$$I \approx \int_0^1 f_1(x_1)dx_1 \cdots \int_0^1 f_{10}(x_{10})dx_{10}$$

This is an efficient method but it is quite rare to be able to separate $f(\mathbf{x})p(\mathbf{x})$

Method 3: use a 1-d method

If we have a function `quad` that integrates functions of a single variable, then we can use quad to compute

$$\int_0^1 g(x_1)dx_1$$

where

$$g(z) = \int_0^1 \cdots \int_0^1 f(z, x_2, \ldots, x_{10}) p(z, x_2, \ldots, x_{10}) dx_2 \cdots dx_{10}$$

as long as we can evaluate $g(z)$

But $g(z)$ is just an integration, so we can evaluate it using `quad`, too!

We end up with 10 nested calls to `quad` so this is very expensive!

Monte Carlo integration

Idea:

- Generate n points $\{\mathbf{z}^{(i)}\}$ that are randomly distributed with probability density function p
- Then

$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}^{(i)})$$

is an approximation to the mean value of f in the region Ω, and therefore the value of the integral is

$$I \approx \mu_n \int_\Omega p(\mathbf{x})dx_1 \cdots dx_{10} = \mu_n$$

Monte Carlo method

Monte Carlo method relies on the strong law of large numbers:

Let $(X_n)_{n\in\mathbb{N}}$ be an infinite sequence of independent and identically distributed (i.i.d.) random variables with expected value $E[X_1] = E[X_2] = \cdots$ then

$$\frac{X_1 + \cdots + X_n}{n} \to E[X_1] \text{ when } n \to +\infty$$

if X is a random variable with density function p and $f : \mathbb{R} \to \mathbb{R}$ then

$$E[f(X)] = \int_\Omega f(x)p(x)dx$$

As a consequence of $(X_n)_{n\in\mathbb{N}}$ be an infinite sequence of independent and identically distributed (i.i.d.) random variables with density function p, then

$$\frac{f(X_1) + \cdots + f(X_n)}{n} \to E[f(X_1)] = \int_\Omega f(x)p(x)dx \text{ quand } n \to +\infty$$

Error estimate

- The expected value of this estimate is the true value of the integral. We call it an unbiased estimator.
- For large n, the distribution of the estimate approximate a normal distribution of variance variance σ^2/n where

$$\sigma^2 = \int_\Omega (f(\mathbf{x}) - \mu_n)^2 p(\mathbf{x}) d\mathbf{x}$$

- Indeed, we have $\mu_n - I \approx \frac{\sigma}{\sqrt{n}} Z$ where Z follows a normal distribution with mean 0 and variance 1
- We note that the variance is a constant independent of the dimension of the integration

Variance-reduction by importance sampling

Suppose that we want to estimate

$$I = \int_\Omega f(\mathbf{x})d\mathbf{x}$$

where Ω is a region of $\mathbb{R}^{10}$ with volume equal to 1

Method 1: Our Monte Carlo estimate of this integral involves taking uniformly distributed samples from Ω and taking the average value of $f(\mathbf{x})$ at these samples

Method 2: Let us choose a function $p(x)$ satisfying $p(\mathbf{x}) > 0$ for all $\mathbf{x} \in \Omega$ so that

$$\int_\Omega p(\mathbf{x})d\mathbf{x} = 1$$

Then

$$I = \int_\Omega \frac{f(\mathbf{x})}{p(\mathbf{x})} p(\mathbf{x})d\mathbf{x}$$

We can get a Monte Carlo estimate of this integral by taking samples from the distribution with probability density $p(\mathbf{x})$ and taking the average value of $f(\mathbf{x})/p(\mathbf{x})$ at these samples.

Variance-reduction by importance sampling

When will Method 2 be better than Method 1?

Recall that the variance of our estimate is proportional to

$$\sigma^2 = \int_\Omega \left(\frac{f(\mathbf{x})}{p(\mathbf{x})} - I \right)^2 p(\mathbf{x}) d\mathbf{x}$$

with

$$I = \int_\Omega f(\mathbf{x}) d\mathbf{x}$$

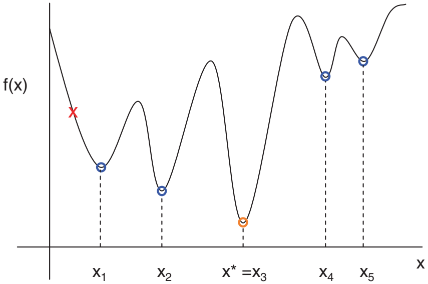
If we chose p so that $f(\mathbf{x})/p(\mathbf{x})$ is close to constant, then σ is close to 0!

We try to get a density function p close to:

$$f/I.$$

Monte Carlo methods for continuous optimization

- Monte Carlo methods provide a good means for generating starting points for optimization problems that are non-convex.
- A single starting point may result in an algorithm converging to a local minimum rather than a global one.
- If we take a set of random samples over the domain of the function we are trying to minimize, we increase the probability that we will eventually get the global minimum.



Introduction to derivatives in finance

A financial option is a derivative product that gives the buyer the right to buy a financial instrument:

- to buy (call option, also called *call*),
- or to sell (put option, also called *put*),

a given quantity of a financial asset (stock, bond, stock index, currency, commodity, etc.), called underlying asset

- at a price that is usually specified in advance (exercise price or strike),
- at a given expiration date (option known as European),
- or during the entire period until expiration (option known as American).

This right itself is traded, on a specialized options market (managed by an exchange, or over-the-counter), for a certain price, called premium.

Introduction to derivatives in finance (cont'd)

A derivative is a financial instrument:

- whose value fluctuates according to changes in the rate or price of a product called the underlying;
- that requires little or no initial net investment;
- whose settlement takes place at a future date.

This is a contract between two parties, a buyer and a seller, which fixes future financial flows based on those of an underlying real or theoretical asset usually financial.

Introduction to derivatives in finance (cont'd)

The options described above are called vanilla options.

There are more complex options called exotic options

For example, there are the Asian options:

- The expiration value of an Asian option is derived from the average price of the underlying in a given time interval. It is lower than that of a vanilla option because the average value of an underlying is less volatile than its value at a given time.

Use of options

The options can be used:

- to cover the risk of a fall or rise in the price of the underlying asset (for example, an oil producer may choose to buy puts in order to protect himself from a fall in the price),
- for speculate on the fall or rise of the underlying (this is why they are distributed as compensation under the name of stock options),
- for speculate on volatility.

Use of options (cont'd)

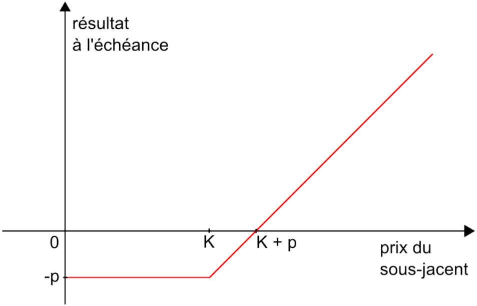
In the absence of a specific hedge and in the worst case scenario, the buyer of an option will have a loss limited to the premium he has paid. On the other hand, his theoretical maximum gain is unlimited (or limited to the strike price minus the premium for a put whose underlying cannot have a negative price).

Symmetrically, the maximum gain for the seller of an option is limited to the premium he receives. His loss can be unlimited or limited (seller of a put whose underlying cannot have a negative price). This is a very risky speculative strategy.

If the option has not been exercised by the expiration date, it is said to be abandoned.

Price of a call

- K: the exercise price of the option
- S: the price of the underlying asset
- p: the option premium
- R: the result at maturity



Price of a call

We denote $(X)_+ = \max(0, X)$

- Let us note (S_t) the (random) price of the (underlying) stock considered at time t.
- If $S_T > K$, then the call holder exercises his right, and buys the stock at price K and resells it immediately at price S_T; his gain is then $R = (S_T - K) - p = (S_T - K)_+ - p$
- If, on the other hand, $S_T < K$, then the holder does not exercise his right (he does not buy the share), and his gain is then $R = -p = (S_T - K)_+ - p$.

In both cases, the gain is on average $E[(S_T - K)_+] - p$ where $E[\cdot]$ denotes the expectation of the random variable.

Price of a call

- For the problem to be fair, the expectation of the payoff is zero, so that $p = E[(S_T - K)_+]$.
- It remains to model the price of the shares. We use the following model: between instants t_i and t_{i+1} the price of the stock varies by

$$S_{t_{i+1}} - S_{t_i} = Y_i S_{t_i}$$

  where the Y_i are iid variables of centered normal distribution $\mathcal{N}(0,1)$.
- The central limit theorem allows to deduce that $S_T \approx e^Z$ with Z of law $\mathcal{N}(0,1)$. The previous formula was introduced by Black and Scholes which earned them the Nobel Prize in Economics.

Price of a call

Fischer Black (1938 - 1995)

Myron Scholes (1941 -)

Conclusion

- Monte Carlo methods are methods of last resort
- They require the generation of random numbers with good properties
- They are used in many fields