Course 2023-2024 in Financial Risk Management
Lecture 10. Monte Carlo Simulation Methods

Thierry Roncalli*

*Amundi Asset Management[1]

*University of Paris-Saclay

September 2023

[1] The opinions expressed in this presentation are those of the authors and are not meant to represent the opinions or official positions of Amundi Asset Management.

General information

1. **Overview**
   The objective of this course is to understand the theoretical and practical aspects of risk management
2. **Prerequisites**
   M1 Finance or equivalent
3. **ECTS**
   4
4. **Keywords**
   Finance, Risk Management, Applied Mathematics, Statistics
5. **Hours**
   Lectures: 36h, Training sessions: 15h, HomeWork: 30h
6. **Evaluation**
   There will be a final three-hour exam, which is made up of questions and exercises
7. **Course website**
   `http://www.thierry-roncalli.com/RiskManagement.html`

Objective of the course

The objective of the course is twofold:

1. knowing and understanding the financial regulation (banking and others) and the international standards (especially the Basel Accords)
2. being proficient in risk measurement, including the mathematical tools and risk models

Class schedule

Course sessions

- September 15 (6 hours, AM+PM)
- September 22 (6 hours, AM+PM)
- September 19 (6 hours, AM+PM)
- October 6 (6 hours, AM+PM)
- October 13 (6 hours, AM+PM)
- October 27 (6 hours, AM+PM)

Tutorial sessions

- October 20 (3 hours, AM)
- October 20 (3 hours, PM)
- November 10 (3 hours, AM)
- November 10 (3 hours, PM)
- November 17 (3 hours, PM)

Class times: Fridays 9:00am-12:00pm, 1:00pm–4:00pm, University of Evry, Room 209 IDF

Agenda

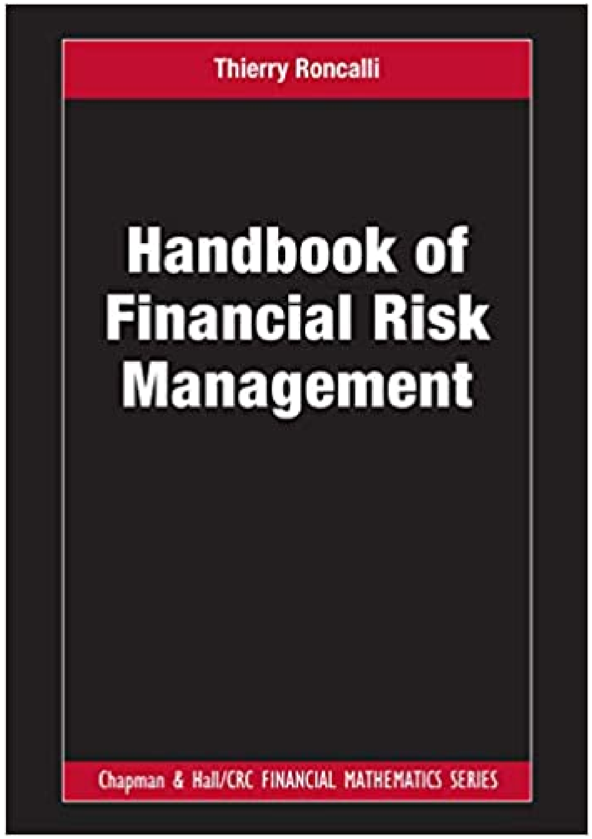
- Lecture 1: Introduction to Financial Risk Management
- Lecture 2: Market Risk
- Lecture 3: Credit Risk
- Lecture 4: Counterparty Credit Risk and Collateral Risk
- Lecture 5: Operational Risk
- Lecture 6: Liquidity Risk
- Lecture 7: Asset Liability Management Risk
- Lecture 8: Model Risk
- Lecture 9: Copulas and Extreme Value Theory
- Lecture 10: Monte Carlo Simulation Methods
- Lecture 11: Stress Testing and Scenario Analysis
- Lecture 12: Credit Scoring Models

Agenda

- Tutorial Session 1: Market Risk
- Tutorial Session 2: Credit Risk
- Tutorial Session 3: Counterparty Credit Risk and Collateral Risk
- Tutorial Session 4: Operational Risk & Asset Liability Management Risk
- Tutorial Session 5: Copulas, EVT & Stress Testing

Textbook

- Roncalli, T. (2020), *Handbook of Financial Risk Management*, Chapman & Hall/CRC Financial Mathematics Series.


Thierry Roncalli
Handbook of Financial Risk Management
Chapman & Hall/CRC FINANCIAL MATHEMATICS SERIES

Additional materials

- Slides, tutorial exercises and past exams can be downloaded at the following address:

  `http://www.thierry-roncalli.com/RiskManagement.html`

- Solutions of exercises can be found in the companion book, which can be downloaded at the following address:

  `http://www.thierry-roncalli.com/RiskManagementBook.html`

Agenda

- Lecture 1: Introduction to Financial Risk Management
- Lecture 2: Market Risk
- Lecture 3: Credit Risk
- Lecture 4: Counterparty Credit Risk and Collateral Risk
- Lecture 5: Operational Risk
- Lecture 6: Liquidity Risk
- Lecture 7: Asset Liability Management Risk
- Lecture 8: Model Risk
- Lecture 9: Copulas and Extreme Value Theory
- **Lecture 10: Monte Carlo Simulation Methods**
- Lecture 11: Stress Testing and Scenario Analysis
- Lecture 12: Credit Scoring Models

Uniform random numbers

The idea is to build a pseudorandom sequence $\mathcal{S}$ and repeat this sequence as often as necessary

Linear congruential generator

- The most famous and used algorithm is the linear congruential generator (LCG):

$$\begin{aligned} x_n &= (a \cdot x_{n-1} + c) \bmod m \\ u_n &= x_n / m \end{aligned}$$

  where:

  - a is the multiplicative constant
  - c is the additive constant
  - m is the modulus (or the order of the congruence)

- The initial number x_0 is called the seed
- $\{x_1, x_2, \ldots, x_n\}$ is a sequence of pseudorandom integer numbers $(0 \leq x_n < m)$
- $\{u_1, u_2, \ldots, u_n\}$ is a sequence of uniform random variates
- The maximum period is m

Linear congruential generator

Example #1

If we consider that $a = 3$, $c = 0$, $m = 11$ and $x_0 = 1$, we obtain the following sequence:

$$\{1, 3, 9, 5, 4, 1, 3, 9, 5, 4, 1, 3, 9, 5, 4, \ldots\}$$

The period length is only five, meaning that only five uniform random variates can be generated: 0.09091, 0.27273, 0.81818, 0.45455 and 0.36364

Linear congruential generator

The minimal standard LCG proposed by Lewis *et al.* (1969) is defined by $a = 7^5$, $c = 0$ and $m = 2^{31} - 1$

Its period length is equal to $m - 1 = 2^{31} - 2 \approx 2.15 \times 10^9$

Table: Simulation of 10 uniform pseudorandom numbers

n	x_n	u_n	x_n	u_n
0	1	0.000000	123 456	0.000057
1	16 807	0.000008	2 074 924 992	0.966212
2	282 475 249	0.131538	277 396 911	0.129173
3	1 622 650 073	0.755605	22 885 540	0.010657
4	984 943 658	0.458650	237 697 967	0.110687
5	1 144 108 930	0.532767	670 147 949	0.312062
6	470 211 272	0.218959	1 772 333 975	0.825307
7	101 027 544	0.047045	2 018 933 935	0.940139
8	1 457 850 878	0.678865	1 981 022 945	0.922486
9	1 458 777 923	0.679296	466 173 527	0.217079
10	2 007 237 709	0.934693	958 124 033	0.446161

Linear congruential generator

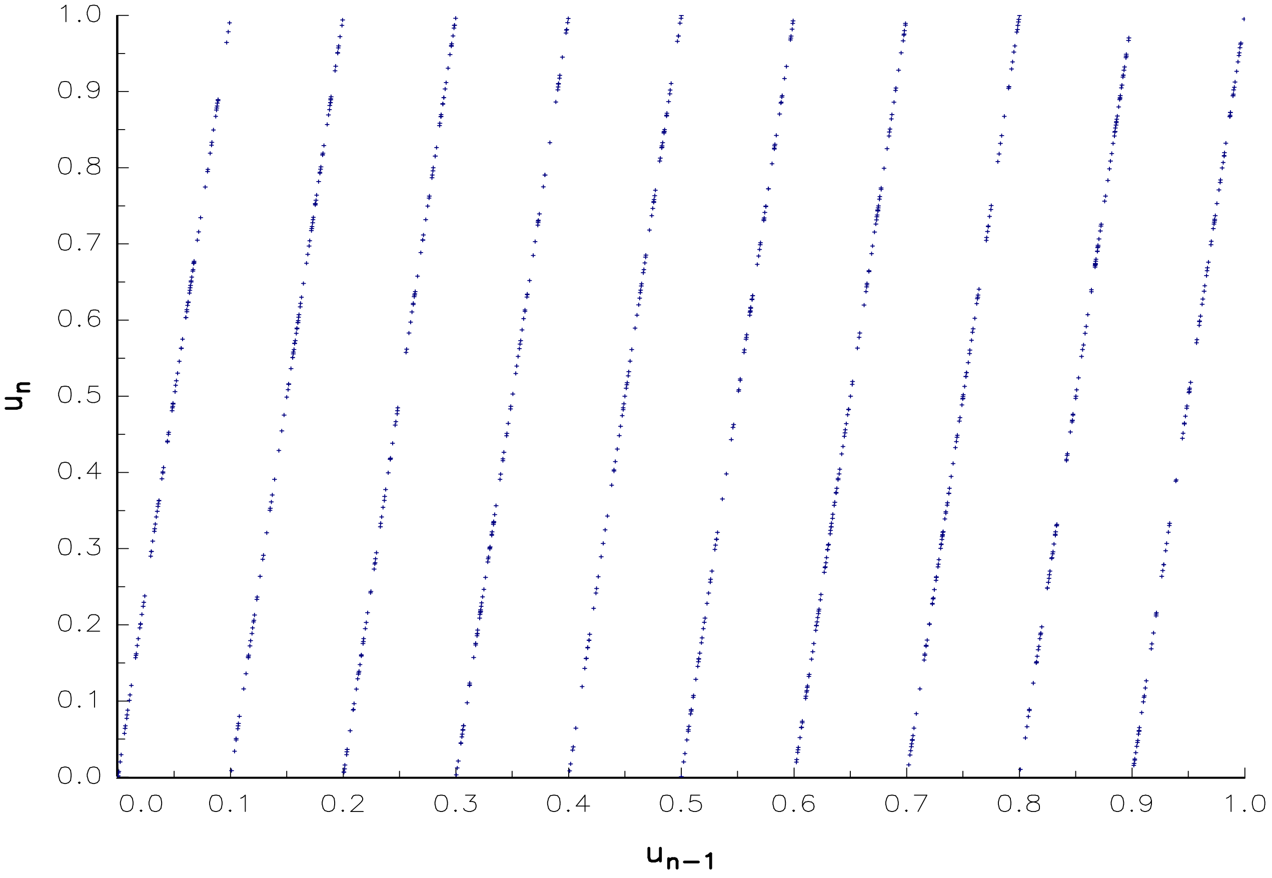


Figure: Lattice structure of the linear congruential generator

Multiple recursive generator

- We have

$$x_n = \left(\sum_{i=1}^{k} a_i \cdot x_{n-i} + c \right) \bmod m$$

- The famous MRG32k3a generator of L'Ecuyer (1999) uses two 32-bit multiple recursive generators:

$$\left\{ \begin{array}{l} x_n = (1403580 \cdot x_{n-2} - 810728 \cdot x_{n-3}) \bmod m_1 \\ y_n = (527612 \cdot y_{n-1} - 1370589 \cdot y_{n-3}) \bmod m_2 \end{array} \right.$$

  where $m_1 = 2^{32} - 209$ and $m_2 = 2^{32} - 22853$. The uniform random variate is then equal to:

$$u_n = \frac{x_n - y_n + \mathbb{1}\left\{x_n \leq y_n\right\} \cdot m_1}{m_1 + 1}$$

- The period length of this generator is equal to $2^{191} \approx 3 \times 10^{57}$

We now consider X a random variable whose distribution function is noted $\mathbf{F}$. There are many ways to simulate X, but all of them are based on uniform random variates

Method of inversion

Continuous random variables

- We assume that $\mathbf{F}$ is continuous
- Let $Y = \mathbf{F}(X)$ be the integral transform of X
- Its cumulative distribution function $\mathbf{G}$ is equal to:

$$\begin{aligned}
\mathbf{G}(y) &= \Pr\{Y \leq y\} \\
&= \Pr\{\mathbf{F}(X) \leq y\} \\
&= \Pr\{X \leq \mathbf{F}^{-1}(y)\} \\
&= \mathbf{F}\left(\mathbf{F}^{-1}(y)\right) \\
&= y
\end{aligned}$$

where $\mathbf{G}(0) = 0$ and $\mathbf{G}(1) = 1$

Method of inversion

Continuous random variables

- We deduce that $\mathbf{F}\left(X\right)$ has a uniform distribution $\mathcal{U}_{[0,1]}$:

$$\mathbf{F}\left(X\right) \sim \mathcal{U}_{[0,1]}$$

IIf U is a uniform random variable, then $\mathbf{F}^{-1}\left(U\right)$ is a random variable whose distribution function is $\mathbf{F}$:

$$U \sim \mathcal{U}_{[0,1]} \Rightarrow \mathbf{F}^{-1}\left(U\right) \sim \mathbf{F}$$

- To simulate a sequence of random variates $\left\{x_1, \ldots, x_n\right\}$, we can simulate a sequence of uniform random variates $\left\{u_1, \ldots, u_n\right\}$ and apply the transform $x_i \leftarrow \mathbf{F}^{-1}\left(u_i\right)$

Method of inversion

Continuous random variables

Example #2

If we consider the generalized uniform distribution $\mathcal{U}_{[a,b]}$, we have $\mathbf{F}(x) = (x-a)/(b-a)$ and $\mathbf{F}^{-1}(u) = a + (b-a)\,u$. The simulation of random variates x_i is deduced from the uniform random variates u_i by using the following transform:

$$x_i \leftarrow a + (b-a)\,u_i$$

Method of inversion

Continuous random variables

Example #3

In the case of the exponential distribution $\mathcal{E}(\lambda)$, we have $\mathbf{F}(x) = 1 - \exp(-\lambda x)$. We deduce that:

$$x_i \leftarrow -\frac{\ln(1 - u_i)}{\lambda}$$

Since $1 - U$ is also a uniform distributed random variable, we have:

$$x_i \leftarrow -\frac{\ln(u_i)}{\lambda}$$

Method of inversion

Continuous random variables

Example #4

In the case of the Pareto distribution $\mathcal{P}\left(\alpha, x_{-}\right)$, we have $\mathbf{F}\left(x\right) = 1 - \left(x/x_{-}\right)^{-\alpha}$ and $\mathbf{F}^{-1}\left(u\right) = x_{-}\left(1-u\right)^{-1/\alpha}$. We deduce that:

$$x_i \leftarrow \frac{x_{-}}{\left(1-u_i\right)^{1/\alpha}}$$

Method of inversion

Continuous random variables

- The method of inversion is easy to implement when we know the analytical expression of $\mathbf{F}^{-1}$
- When it is not the case, we use the Newton-Raphson algorithm:

  $$x_i^{m+1} = x_i^m + \frac{u_i - \mathbf{F}\left(x_i^m\right)}{f\left(x_i^m\right)}$$

  where x_i^m is the solution of the equation $\mathbf{F}\left(x\right) = u$ at the iteration m
- If we apply this algorithm to the Gaussian distribution $\mathcal{N}\left(0,1\right)$, we have:

  $$x_i^{m+1} = x_i^m + \frac{u_i - \Phi\left(x_i^m\right)}{\phi\left(x_i^m\right)}$$

Method of inversion

Discrete random variables

In the case of a discrete probability distribution $\{(x_1, p_1), (x_2, p_2), \ldots, (x_n, p_n)\}$ where $x_1 < x_2 < \ldots < x_n$, we have:

$$\mathbf{F}^{-1}(u) = \begin{cases} x_1 & \text{if} \quad 0 \leq u \leq p_1 \\ x_2 & \text{if} \quad p_1 < u \leq p_1 + p_2 \\ & \vdots \\ x_n & \text{if} \quad \sum_{k=1}^{n-1} p_k < u \leq 1 \end{cases}$$

Method of inversion

Discrete random variables

- We assume that:

x_i	1	2	4	6	7	9	10
p_i	10%	20%	10%	5%	20%	30%	5%
$\mathbf{F}(x_i)$	10%	30%	40%	45%	65%	95%	100%

- The inverse function is a step function
- If $u = 0.5517$, Then $X = \mathbf{F}^{-1}(u) = \mathbf{F}^{-1}(0.5517) = 7$

Method of inversion

Discrete random variables

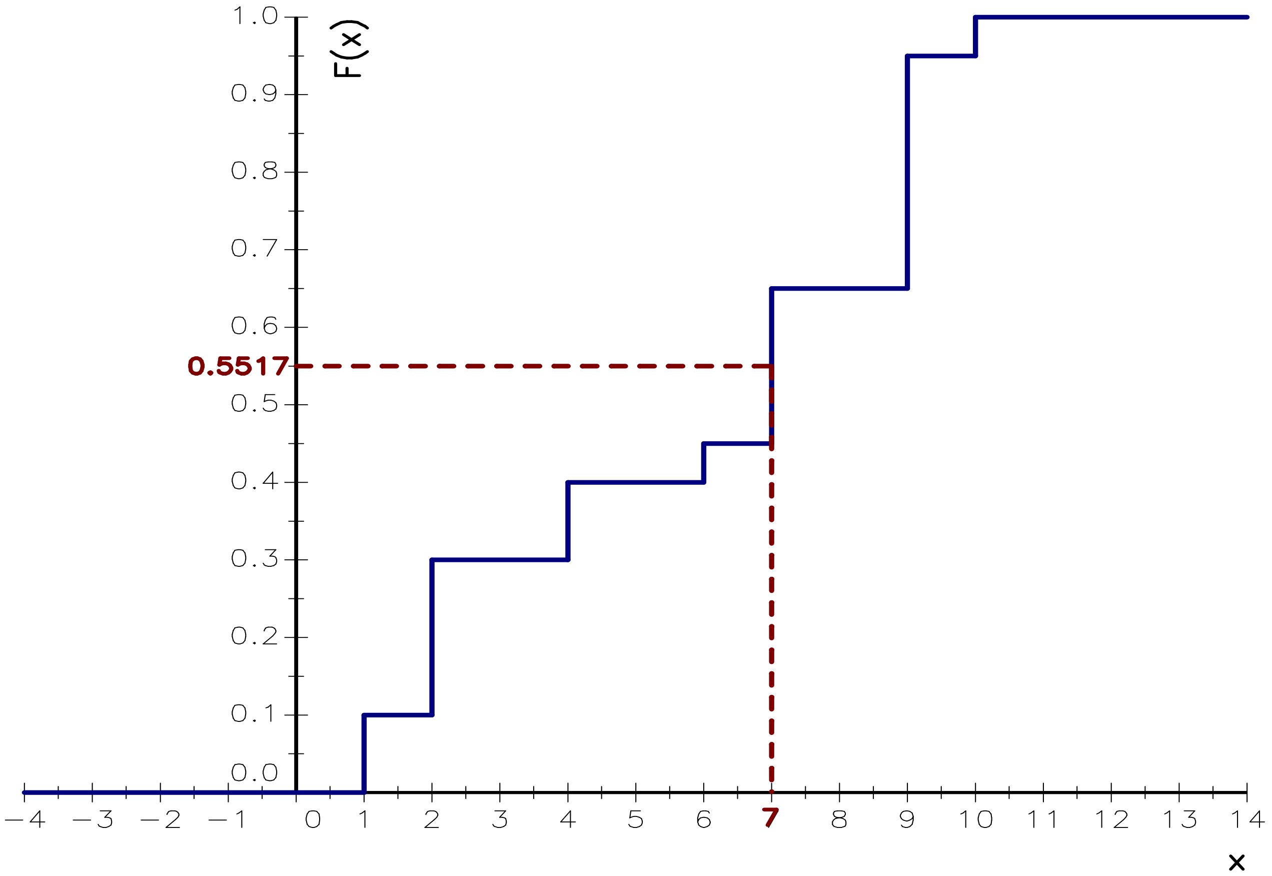


Figure: Inversion method when X is a discrete random variable

Method of inversion

Discrete random variables

Example #5

If we apply the method of inversion to the Bernoulli distribution $\mathcal{B}(p)$, we have:

$$x \leftarrow \begin{cases} 0 & \text{if} \quad 0 \leq u \leq 1 - p \\ 1 & \text{if} \quad 1 - p < u \leq 1 \end{cases}$$

or:

$$x \leftarrow \begin{cases} 1 & \text{if} \quad u \leq p \\ 0 & \text{if} \quad u > p \end{cases}$$

Method of inversion

Piecewise distribution functions

- A piecewise distribution function is defined as follows:

$$\mathbf{F}(x) = \mathbf{F}_m(x) \qquad \text{if } x \in \left]x_{m-1}^{\star}, x_m^{\star}\right]$$

where $x_m^{\star}$ are the knots of the piecewise function and:

$$\mathbf{F}_{m+1}(x_m^{\star}) = \mathbf{F}_m(x_m^{\star})$$

- In this case, the simulated value x_i is obtained using a search algorithm:

$$x_i \leftarrow \mathbf{F}_m^{-1}(u_i) \qquad \text{if } \mathbf{F}\left(x_{m-1}^{\star}\right) < u_i \leq \mathbf{F}\left(x_m^{\star}\right)$$

Method of inversion

Piecewise distribution functions

- We consider the piecewise exponential model
- The survival function has the following expression:

$$\mathbf{S}(t) = \mathbf{S}\left(t^{\star}_{m-1}\right) e^{-\lambda_m \left(t - t^{\star}_{m-1}\right)} \qquad \text{if } t \in \left]t^{\star}_{m-1}, t^{\star}_{m}\right]$$

- We know that $\mathbf{S}(\tau) \sim U$
- It follows that:

$$t_i \leftarrow t^{\star}_{m-1} + \frac{1}{\lambda_m} \ln \frac{\mathbf{S}\left(t^{\star}_{m-1}\right)}{u_i} \qquad \text{if } \mathbf{S}\left(t^{\star}_{m}\right) < u_i \leq \mathbf{S}\left(t^{\star}_{m-1}\right)$$

Method of inversion

Piecewise distribution functions

Example #6

We model the default time $\boldsymbol{\tau}$ with the piecewise exponential model and the following parameters:

$$\lambda = \begin{cases} 5\% & \text{if} \quad t \text{ is less or equal than one year} \\ 8\% & \text{if} \quad t \text{ is between one and five years} \\ 12\% & \text{if} \quad t \text{ is larger than five years} \end{cases}$$

Method of inversion

Piecewise distribution functions

We have $\mathbf{S}(0) = 1$, $\mathbf{S}(1) = 0.9512$ and $\mathbf{S}(5) = 0.6907$. We deduce that:

$$t_i \leftarrow \begin{cases} 0 + (1/0.05) \cdot \ln(1/u_i) & \text{if} \quad u_i \in [0.9512, 1] \\ 1 + (1/0.08) \cdot \ln(0.9512/u_i) & \text{if} \quad u_i \in [0.6907, 0.9512[\\ 5 + (1/0.12) \cdot \ln(0.6907/u_i) & \text{if} \quad u_i \in [0, 0.6907[\end{cases}$$

Table: Simulation of the piecewise exponential model

u_i	$t^\star_{m-1}$	$S\left(t^\star_{m-1}\right)$	λ_m	t_i
0.9950	0	1.0000	0.05	0.1003
0.3035	5	0.6907	0.12	11.8531
0.5429	5	0.6907	0.12	7.0069
0.9140	1	0.9512	0.08	1.4991
0.7127	1	0.9512	0.08	4.6087

Method of transformation

Let $\{Y_1, Y_2, \ldots\}$ be a vector of independent random variables. The simulation of the random variable $X = g(Y_1, Y_2, \ldots)$ is straightforward if we know how to easily simulate the random variables Y_i. We notice that the inversion method is a particular case of the transform method, because we have:

$$X = g(U) = \mathbf{F}^{-1}(U)$$

Method of transformation

- The Binomial random variable is the sum of n *iid* Bernoulli random variables:

$$\mathcal{B}\left(n,p\right)=\sum_{i=1}^{n}\mathcal{B}_i\left(p\right)$$

- We simulate the Binomial random variate x using n uniform random numbers:

$$x=\sum_{i=1}^{n}\mathbb{1}\left\{u_i\leq p\right\}$$

Method of transformation

To simulate the chi-squared random variable $\chi^2(\nu)$, we can use the following relationship:

$$\chi^2(\nu) = \sum_{i=1}^{\nu} \chi_i^2(1) = \sum_{i=1}^{\nu} \left(\mathcal{N}_i(0,1)\right)^2$$

Method of transformation

Box-Muller algorithm

if U_1 and U_2 are two independent uniform random variables, then X_1 and X_2 defined by:

$$\left\{ \begin{array}{l} X_1 = \sqrt{-2 \ln U_1} \cdot \cos \left(2\pi U_2\right) \\ X_2 = \sqrt{-2 \ln U_1} \cdot \sin \left(2\pi U_2\right) \end{array} \right.$$

are independent and follow the Gaussian distribution distribution $\mathcal{N}\left(0, 1\right)$

Method of transformation

- If N_t is a Poisson process with intensity λ, the duration T between two consecutive events is an exponential:

$$\Pr\left(T \leq t\right) = 1 - e^{-\lambda t}$$

- Since the durations are independent, we have:

$$T_1 + T_2 + \ldots + T_n = \sum_{i=1}^{n} E_i$$

  where $E_i \sim \mathcal{E}\left(\lambda\right)$

- Because the Poisson random variable is the number of events that occur in the unit interval of time, we also have:

$$X = \max\left\{n : T_1 + T_2 + \ldots + T_n \leq 1\right\} = \max\left\{n : \sum_{i=1}^{n} E_i \leq 1\right\}$$

Method of transformation

- We notice that:

$$\sum_{i=1}^{n} E_i = -\frac{1}{\lambda} \sum_{i=1}^{n} \ln U_i = -\frac{1}{\lambda} \ln \prod_{i=1}^{n} U_i$$

  where U_i are *iid* uniform random variables

- We deduce that:

$$X = \max \left\{ n : -\frac{1}{\lambda} \ln \prod_{i=1}^{n} U_i \leq 1 \right\} = \max \left\{ n : \prod_{i=1}^{n} U_i \geq e^{-\lambda} \right\}$$

Method of transformation

We can then simulate the Poisson random variable with the following algorithm:

1. set $n = 0$ and $p = 1$;
2. calculate $n = n + 1$ and $p = p \cdot u_i$ where u_i is a uniform random variate;
3. if $p \geq e^{-\lambda}$, go back to step 2; otherwise, return $X = n - 1$

Rejection sampling

Theorem

- $\mathbf{F}(x)$ and $\mathbf{G}(x)$ are two distribution functions such that $f(x) \leq cg(x)$ for all x with $c > 1$
- We note $X \sim \mathbf{G}$ and consider an independent uniform random variable $U \sim \mathcal{U}_{[0,1]}$
- Then, the conditional distribution function of X given that $U \leq f(X) / (cg(X))$ is $\mathbf{F}(x)$

Rejection sampling

Proof

Let us introduce the random variables B and Z:

$$B = \mathbb{1}\left\{U \leq \frac{f(X)}{cg(X)}\right\} \qquad \text{and} \qquad Z = X \,\Big|\, U \leq \frac{f(X)}{cg(X)}$$

We have:

$$\begin{aligned}
\Pr\{B = 1\} &= \Pr\left\{U \leq \frac{f(X)}{cg(X)}\right\} \\
&= \mathbb{E}\left[\frac{f(X)}{cg(X)}\right] = \int_{-\infty}^{+\infty} \frac{f(x)}{cg(x)} g(x)\,\mathrm{d}x \\
&= \frac{1}{c}\int_{-\infty}^{+\infty} f(x)\,\mathrm{d}x \\
&= \frac{1}{c}
\end{aligned}$$

Rejection sampling

Proof

The distribution function of Z is defined by:

$$\Pr\left\{Z \leq x\right\} = \Pr\left\{X \leq x \,\middle|\, U \leq \frac{f(X)}{cg(X)}\right\}$$

We deduce that:

$$\begin{aligned}
\Pr\left\{Z \leq x\right\} &= \frac{\Pr\left\{X \leq x, U \leq \dfrac{f(X)}{cg(X)}\right\}}{\Pr\left\{U \leq \dfrac{f(X)}{cg(X)}\right\}} = c\int_{-\infty}^{x}\int_{0}^{f(x)/(cg(x))} g(x)\,\mathrm{d}u\,\mathrm{d}x \\
&= c\int_{-\infty}^{x}\frac{f(x)}{cg(x)}g(x)\,\mathrm{d}x = \int_{-\infty}^{x} f(x)\,\mathrm{d}x \\
&= \mathbf{F}(x)
\end{aligned}$$

This proves that $Z \sim \mathbf{F}$

Rejection sampling

Acceptance-rejection algorithm

1. generate two independent random variates x and u from $\mathbf{G}$ and $\mathcal{U}_{[0,1]}$;
2. calculate v as follows:
$$v = \frac{f(x)}{cg(x)}$$
3. if $u \leq v$, return x ('*accept*'); otherwise, go back to step 1 ('*reject*')

Remark

The underlying idea of this algorithm is then to simulate the distribution function $\mathbf{F}$ by assuming that it is easier to generate random numbers from $\mathbf{G}$, which is called the proposal distribution. However, some of these random numbers must be '*rejected*', because the function $c \cdot g(x)$ '*dominates*' the density function $f(x)$

Rejection sampling

- The number of iterations N needed to successfully generate Z has a geometric distribution $\mathcal{G}(p)$, where $p = \Pr\{B = 1\} = c^{-1}$ is the acceptance ratio
- The average number of iterations is equal to:

$$\mathbb{E}[N] = 1/p = c$$

- To maximize the efficiency (or the acceptance ratio) of the algorithm, we have to choose the constant c such that:

$$c = \sup_x \frac{f(x)}{g(x)}$$

Rejection sampling

- We consider the normal distribution $\mathcal{N}(0,1)$
- We use the Cauchy distribution function as the proposal distribution:

$$g(x) = \frac{1}{\pi\left(1 + x^2\right)}$$

- We can show that:

$$\phi(x) \leq \frac{\sqrt{2\pi}}{e^{0.5}} g(x)$$

meaning that $c \approx 1.52$

- We have:

$$\mathbf{G}(x) = \frac{1}{2} + \frac{1}{\pi} \arctan x$$

and:

$$\mathbf{G}^{-1}(u) = \tan\left(\pi\left(u - \frac{1}{2}\right)\right)$$

Rejection sampling

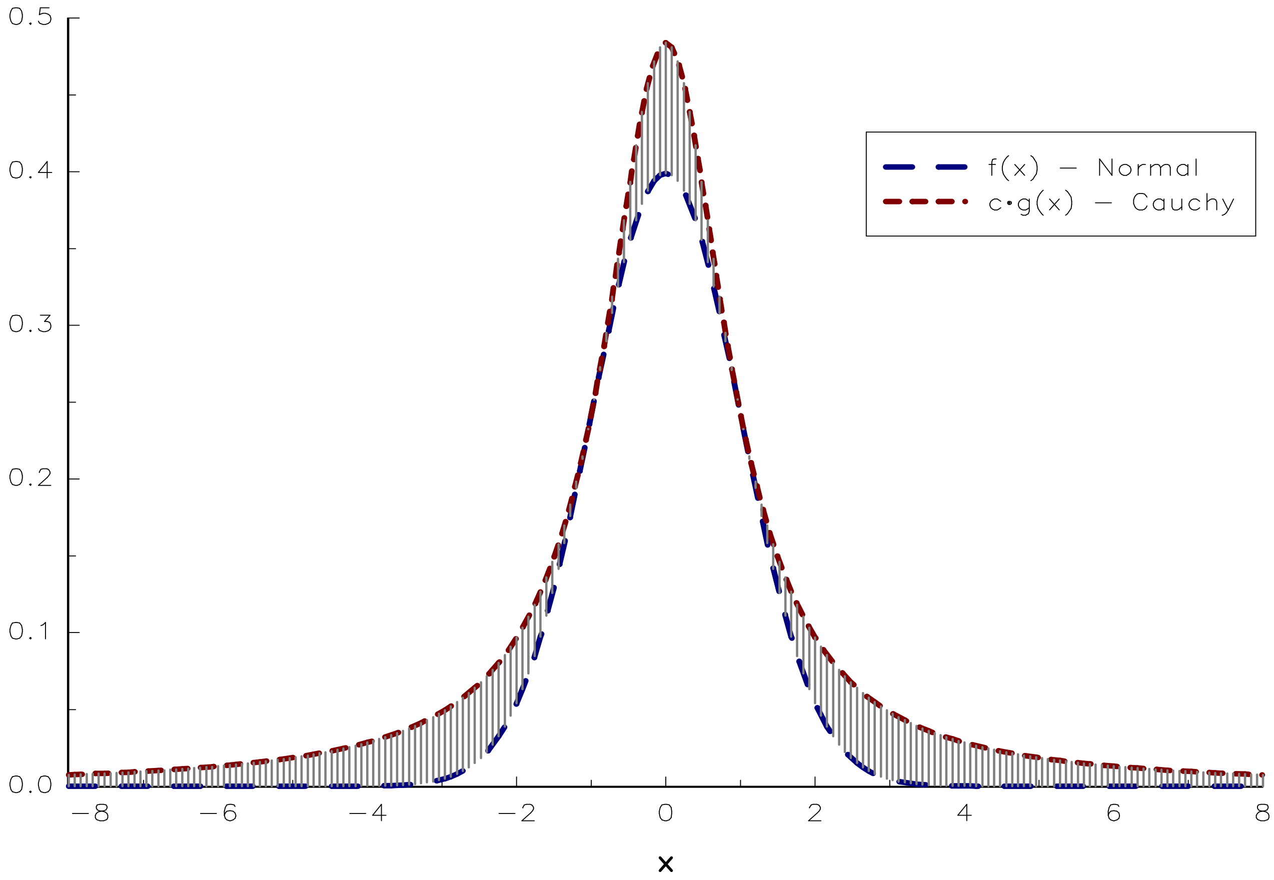


Figure: Rejection sampling applied to the normal distribution

Rejection sampling

Acceptance-rejection algorithm for simulating $\mathcal{N}(0,1)$

1. generate two independent uniform random variates u_1 and u_2 and set:

$$x \leftarrow \tan\left(\pi\left(u_1 - \frac{1}{2}\right)\right)$$

2. calculate v as follows:

$$v = \frac{e^{0.5}\phi(x)}{\sqrt{2\pi}g(x)} = \frac{(1+x^2)}{2e^{(x^2-1)/2}}$$

3. if $u_2 \leq v$, accept x; otherwise, go back to step 1

The acceptance ratio is $1/1.52 \approx 65.8\%$

Rejection sampling

Table: Simulation of the standard Gaussian distribution using the acceptance-rejection algorithm

u_1	u_2	x	v	test	z
0.9662	0.1291	9.3820	0.0000	reject	
0.0106	0.1106	−30.0181	0.0000	reject	
0.3120	0.8253	−0.6705	0.9544	accept	−0.6705
0.9401	0.9224	5.2511	0.0000	reject	
0.2170	0.4461	−1.2323	0.9717	accept	−1.2323
0.6324	0.0676	0.4417	0.8936	accept	0.4417
0.6577	0.1344	0.5404	0.9204	accept	0.5404
0.1596	0.6670	−1.8244	0.6756	accept	−1.8244
0.4183	0.3872	−0.2625	0.8513	accept	−0.2625
0.9625	0.0752	8.4490	0.0000	reject	

Rejection sampling

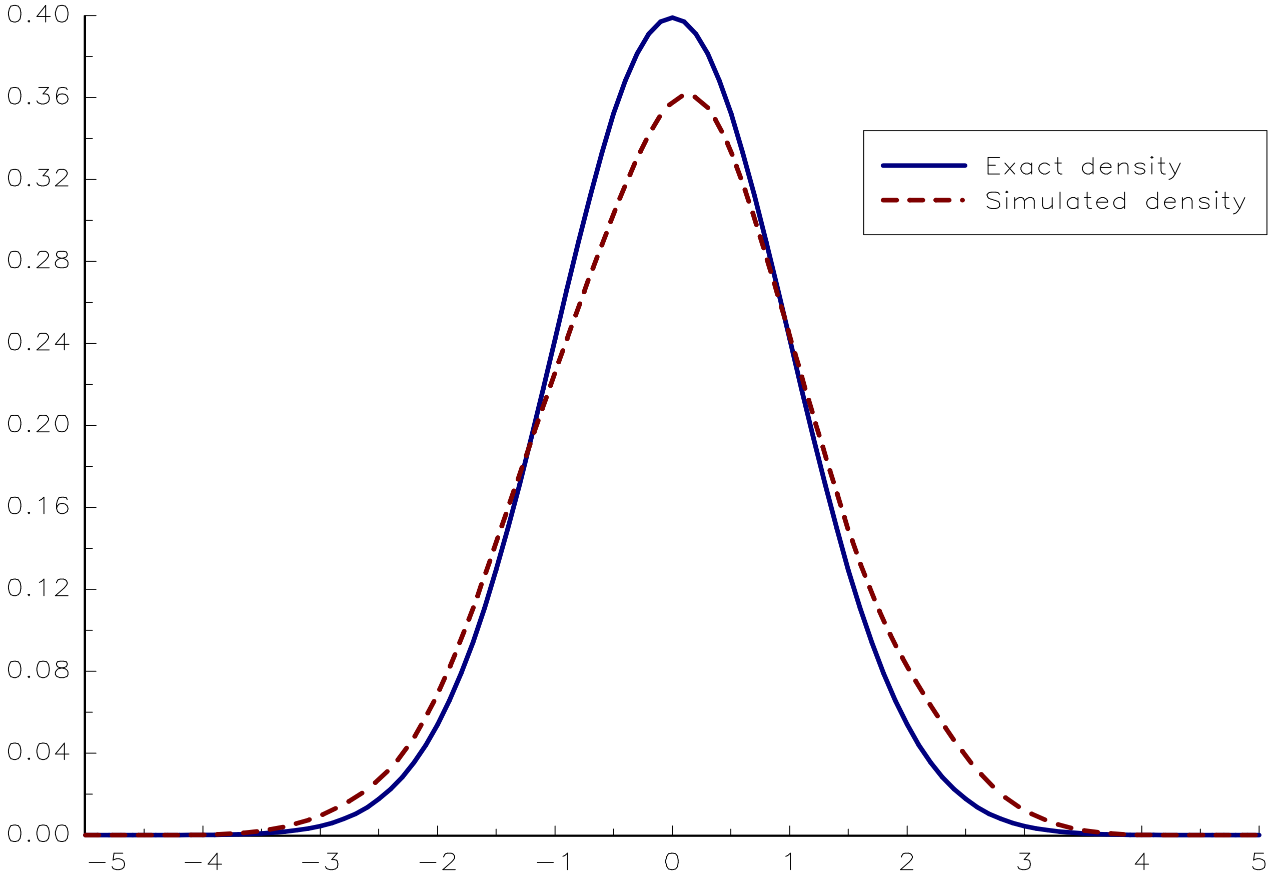


Figure: Comparison of the exact and simulated densities

Method of mixtures

- A finite mixture can be decomposed as a weighted sum of distribution functions:

$$\mathbf{F}(x) = \sum_{k=1}^{n} \pi_k \cdot \mathbf{G}_k(x)$$

  where $\pi_k \geq 0$ and $\sum_{k=1}^{n} \pi_k = 1$

- The probability density function is:

$$f(x) = \sum_{k=1}^{n} \pi_k \cdot g_k(x)$$

- To simulate the probability distribution **F**, we introduce the random variable B, whose probability mass function is defined by:

$$p(k) = \Pr\{B = k\} = \pi_k$$

  It follows that:

$$\mathbf{F}(x) = \sum_{k=1}^{n} \Pr\{B = k\} \cdot \mathbf{G}_k(x)$$

Method of mixtures

We deduce the following algorithm:

1. generate the random variate b from the probability mass function $p(k)$
2. generate the random variate x from the probability distribution $\mathbf{G}_b(x)$

Method of mixtures

The previous approach can be easily extended to continuous mixtures:

$$f(x) = \int_{\Omega} \pi(\omega) \, g(x; \omega) \, \mathrm{d}\omega$$

where $\omega \in \Omega$ is a parameter of the distribution $\mathbf{G}$

Method of mixtures

The negative binomial distribution is a gamma-Poisson mixture distribution:

$$\left\{ \begin{array}{l} \mathcal{NB}\left(r,p\right) \sim \mathcal{P}\left(\Lambda\right) \\ \Lambda \sim \mathcal{G}\left(r,\left(1-p\right)/p\right) \end{array} \right.$$

To simulate the negative binomial distribution, we simulate

1. the gamma random variate $g \sim \mathcal{G}\left(r,\left(1-p\right)/p\right)$
2. and then the Poisson random variable p, whose parameter λ is equal to g

Random vectors

The random vector $X = (X_1, \ldots, X_n)$ has a given distribution function $\mathbf{F}(x) = \mathbf{F}(x_1, \ldots, x_n)$

Method of conditional distributions

- If $X_1, \ldots, X_n$ are independent, we have:

$$\mathbf{F}(x_1, \ldots, x_n) = \prod_{i=1}^{n} \mathbf{F}_i(x_i)$$

To simulate X, we can then generate each component $X_i \sim \mathbf{F}_i$ individually, for example by applying the method of inversion

Method of conditional distributions

- If $X_1, \ldots, X_n$ are dependent, we have:

$$\begin{aligned}
\mathbf{F}\left(x_1, \ldots, x_n\right) & = \mathbf{F}_1\left(x_1\right) \mathbf{F}_{2 \mid 1}\left(x_2 \mid x_1\right) \mathbf{F}_{3 \mid 1,2}\left(x_3 \mid x_1, x_2\right) \times \cdots \times \\
& \quad \mathbf{F}_{n \mid 1, \ldots, n-1}\left(x_n \mid x_1, \ldots, x_{n-1}\right) \\
& = \prod_{i=1}^{n} \mathbf{F}_{i \mid 1, \ldots, i-1}\left(x_i \mid x_1, \ldots, x_{i-1}\right)
\end{aligned}$$

  where $\mathbf{F}_{i \mid 1, \ldots, i-1}\left(x_i \mid x_1, \ldots, x_{i-1}\right)$ is the conditional distribution of X_i given $X_1 = x_1, \ldots, X_{i-1} = x_{i-1}$

- This '*conditional*' random variable is denoted by $Y_i = X_i \mid X_1 = x_1, \ldots, X_{i-1} = x_{i-1}$

- The random variables $\left(Y_1, \ldots, Y_n\right)$ are independent

Method of conditional distributions

We obtain the following algorithm:

1. generate x_1 from $\mathbf{F}_1(x)$ and set $i = 2$
2. generate x_i from $\mathbf{F}_{i|1,\ldots,i-1}(x \mid x_1, \ldots, x_{i-1})$ given $X_1 = x_1, \ldots, X_{i-1} = x_{i-1}$ and set $i = i + 1$
3. repeat step 2 until $i = n$

Method of conditional distributions

$\mathbf{F}_{i|1,\ldots,i-1}\left(x \mid x_1,\ldots,x_{i-1}\right)$ is a univariate distribution function, which depends on the argument x and parameters $x_1,\ldots,x_{i-1}$. To simulate it, we can therefore use the method of inversion:

$$x_i \leftarrow \mathbf{F}_{i|1,\ldots,i-1}^{-1}\left(u_i \mid x_1,\ldots,x_{i-1}\right)$$

where $\mathbf{F}_{i|1,\ldots,i-1}^{-1}$ is the inverse of the conditional distribution function and u_i is a uniform random variate

Method of conditional distributions

Example #7

We consider the bivariate logistic distribution defined as:

$$\mathbf{F}\left(x_1, x_2\right) = \left(1 + e^{-x_1} + e^{-x_2}\right)^{-1}$$

Method of conditional distributions

We have $\mathbf{F}_1(x_1) = \mathbf{F}(x_1, +\infty) = \left(1 + e^{-x_1}\right)^{-1}$. We deduce that the conditional distribution of X_2 given $X_1 = x_1$ is:

$$\begin{aligned}\mathbf{F}_{2|1}(x_2 \mid x_1) &= \frac{\mathbf{F}(x_1, x_2)}{\mathbf{F}_1(x_1)} \\ &= \frac{1 + e^{-x_1}}{1 + e^{-x_1} + e^{-x_2}}\end{aligned}$$

We obtain:

$$\mathbf{F}_1^{-1}(u) = \ln u - \ln(1 - u)$$

and:

$$\mathbf{F}_{2|1}^{-1}(u \mid x_1) = \ln u - \ln(1 - u) - \ln\left(1 + e^{-x_1}\right)$$

Method of conditional distributions

We deduce the following algorithm:

1. generate two independent uniform random variates u_1 and u_2;
2. generate x_1 from u_1:

$$x_1 \leftarrow \ln u_1 - \ln (1 - u_1)$$

3. generate x_2 from u_2 and x_1:

$$x_2 \leftarrow \ln u_2 - \ln (1 - u_2) - \ln \left(1 + e^{-x_1}\right)$$

Because we have $(1 + e^{-x_1})^{-1} = u_1$, the last step can be replaced by:

3. generate x_2 from u_2 and u_1:

$$x_2 \leftarrow \ln \left(\frac{u_1 u_2}{1 - u_2}\right)$$

Method of conditional distributions

- The method of conditional distributions can be used for simulating uniform random vectors $(U_1, \ldots, U_n)$ generated by copula functions
- We have

$$\begin{aligned}\mathbf{C}(u_1, \ldots, u_n) &= \mathbf{C}_1(u_1)\,\mathbf{C}_{2|1}(u_2 \mid u_1)\,\mathbf{C}_{3|1,2}(u_3 \mid u_1, u_2) \times \cdots \times \\ &\quad \mathbf{C}_{n|1,\ldots,n-1}(u_n \mid u_1, \ldots, u_{n-1}) \\ &= \prod_{i=1}^{n} \mathbf{C}_{i|1,\ldots,i-1}(u_i \mid u_1, \ldots, u_{i-1})\end{aligned}$$

  where $\mathbf{C}_{i|1,\ldots,i-1}(u_i \mid u_1, \ldots, u_{i-1})$ is the conditional distribution of U_i given $U_1 = u_1, \ldots, U_{i-1} = u_{i-1}$
- By definition, we have $\mathbf{C}_1(u_1) = u_1$

Method of conditional distributions

We obtain the following algorithm:

1. generate n independent uniform random variates $v_1, \ldots, v_n$;
2. generate $u_1 \leftarrow v_1$ and set $i = 2$;
3. generate u_i by finding the root of the equation:

$$\mathbf{C}_{i|1,\ldots,i-1}\left(u_i \mid u_1, \ldots, u_{i-1}\right) = v_i$$

   and set $i = i + 1$;

4. repeat step 3 until $i = n$.

For some copula functions, there exists an analytical expression of the inverse of the conditional copula. In this case, the third step is replaced by:

3. generate u_i by the inversion method:

$$u_i \leftarrow \mathbf{C}_{i|1,\ldots,i-1}^{-1}\left(v_i \mid u_1, \ldots, u_{i-1}\right)$$

Method of conditional distributions

For any probability distribution, the conditional distribution can be calculated as follows:

$$\mathbf{F}_{i|1,\ldots,i-1}\left(x_i \mid x_1,\ldots,x_{i-1}\right) = \frac{\mathbf{F}\left(x_1,\ldots,x_{i-1},x_i\right)}{\mathbf{F}\left(x_1,\ldots,x_{i-1}\right)}$$

In particular, we have:

$$\begin{aligned}\partial_1\,\mathbf{F}\left(x_1,x_2\right) &= \partial_1\,\left(\mathbf{F}_1\left(x_1\right)\cdot\mathbf{F}_{2|1}\left(x_2 \mid x_1\right)\right) \\ &= f_1\left(x_1\right)\cdot\mathbf{F}_{2|1}\left(x_2 \mid x_1\right)\end{aligned}$$

For copula functions, the density $f_1\left(x_1\right)$ is equal to 1, meaning that:

$$\mathbf{C}_{2|1}\left(u_2 \mid u_1\right) = \partial_1\,\mathbf{C}\left(u_1,u_2\right)$$

We can generalize this result and show that the conditional copula given some random variables U_i for $i \in \Omega$ is equal to the cross-derivative of the copula function $\mathbf{C}$ with respect to the arguments u_i for $i \in \Omega$

Method of conditional distributions

- Archimedean copulas are defined as:

$$\mathbf{C}\left(u_1, u_2\right) = \varphi^{-1}\left(\varphi\left(u_1\right) + \varphi\left(u_2\right)\right)$$

  where $\varphi\left(u\right)$ is the generator function

- We have:

$$\varphi\left(\mathbf{C}\left(u_1, u_2\right)\right) = \varphi\left(u_1\right) + \varphi\left(u_2\right)$$

  and:

$$\varphi'\left(\mathbf{C}\left(u_1, u_2\right)\right) \cdot \frac{\partial\,\mathbf{C}\left(u_1, u_2\right)}{\partial\,u_1} = \varphi'\left(u_1\right)$$

- We deduce the following expression of the conditional copula:

$$\mathbf{C}_{2|1}\left(u_2 \mid u_1\right) = \frac{\partial\,\mathbf{C}\left(u_1, u_2\right)}{\partial\,u_1} = \frac{\varphi'\left(u_1\right)}{\varphi'\left(\varphi^{-1}\left(\varphi\left(u_1\right) + \varphi\left(u_2\right)\right)\right)}$$

- The calculation of the inverse function gives:

$$\mathbf{C}_{2|1}^{-1}\left(v \mid u_1\right) = \varphi^{-1}\left(\varphi\left(\varphi'^{-1}\left(\frac{\varphi'\left(u_1\right)}{v}\right)\right) - \varphi\left(u_1\right)\right)$$

Method of conditional distributions

We obtain the following algorithm for simulating Archimedean copulas:

1. generate two independent uniform random variates v_1 and v_2;
2. generate $u_1 \leftarrow v_1$;
3. generate u_2 by the inversion method:

$$u_2 \leftarrow \varphi^{-1}\left(\varphi\left(\varphi'^{-1}\left(\frac{\varphi'(u_1)}{v_2}\right)\right) - \varphi(u_1)\right)$$

Method of conditional distributions

Example #8

We consider the Clayton copula:

$$\mathbf{C}\left(u_1, u_2\right) = \left(u_1^{-\theta} + u_2^{-\theta} - 1\right)^{-1/\theta}$$

Method of conditional distributions

The Clayton copula is an Archimedean copula, whose generator function is:

$$\varphi(u) = u^{-\theta} - 1$$

We deduce that:

$$\begin{array}{rcl} \varphi^{-1}(u) & = & (1+u)^{-1/\theta} \\ \varphi'(u) & = & -\theta u^{-(\theta+1)} \\ \varphi'^{-1}(u) & = & (-u/\theta)^{-1/(\theta+1)} \end{array}$$

We obtain:

$$\mathbf{C}_{2|1}^{-1}(v \mid u_1) = \left(1 + u_1^{-\theta}\left(v^{-\theta/(\theta+1)} - 1\right)\right)^{-1//\theta}$$

Method of conditional distributions

Table: Simulation of the Clayton copula

Random uniform variates		Clayton copula			
		$\theta = 0.01$		$\theta = 1.5$	
v_1	v_2	u_1	u_2	u_1	u_2
0.2837	0.4351	0.2837	0.4342	0.2837	0.3296
0.0386	0.2208	0.0386	0.2134	0.0386	0.0297
0.3594	0.5902	0.3594	0.5901	0.3594	0.5123
0.3612	0.3268	0.3612	0.3267	0.3612	0.3247
0.0797	0.6479	0.0797	0.6436	0.0797	0.1704

Method of transformation

- To simulate a Gaussian random vector $X \sim \mathcal{N}(\mu, \Sigma)$, we consider the following transformation:

  $$X = \mu + A \cdot N$$

  where $AA^\top = \Sigma$ and $N \sim \mathcal{N}(\mathbf{0}, I)$
- Since Σ is a positive definite symmetric matrix, it has a unique Cholesky decomposition:

  $$\Sigma = PP^\top$$

  where P is a lower triangular matrix

Method of transformation

The decomposition $AA^\top = \Sigma$ is not unique. For instance, if we use the eigendecomposition:

$$\Sigma = U \Lambda U^\top$$

we can set $A = U \Lambda^{1/2}$. Indeed, we have:

$$\begin{aligned} AA^\top &= U \Lambda^{1/2} \Lambda^{1/2} U^\top \\ &= U \Lambda U^\top \\ &= \Sigma \end{aligned}$$

Method of transformation

To simulate a multivariate Student's t distribution $Y = (Y_1, \ldots, Y_n) \sim \mathbf{T}_n(\Sigma, \nu)$, we use the relationship:

$$Y_i = \frac{X_i}{\sqrt{Z/\nu}}$$

where the random vector $X = (X_1, \ldots, X_n) \sim \mathcal{N}(\mathbf{0}, \Sigma)$ and the random variable $Z \sim \chi^2(\nu)$ are independent

Method of transformation

- If $X = (X_1, \ldots, X_n) \sim \mathbf{F}$, then the probability distribution of the random vector $U = (U_1, \ldots, U_n)$ defined by:

$$U_i = \mathbf{F}_i (X)$$

  is the copula function $\mathbf{C}$ associated to $\mathbf{F}$

- To simulate the Normal copula with the matrix of parameters ρ, we simulate $N \sim \mathcal{N}(\mathbf{0}, I)$ and apply the transformation:

$$U = \Phi (P \cdot N)$$

  where P is the Cholesky decomposition of the correlation matrix ρ

- To simulate the Student's t copulawith the matrix of parameters ρ and ν degrees of freedom, we simulate $T \sim \mathbf{T}_n (\rho, \nu)$ and apply the transformation:

$$U_i = \mathbf{T}_v (T_i)$$

Method of transformation

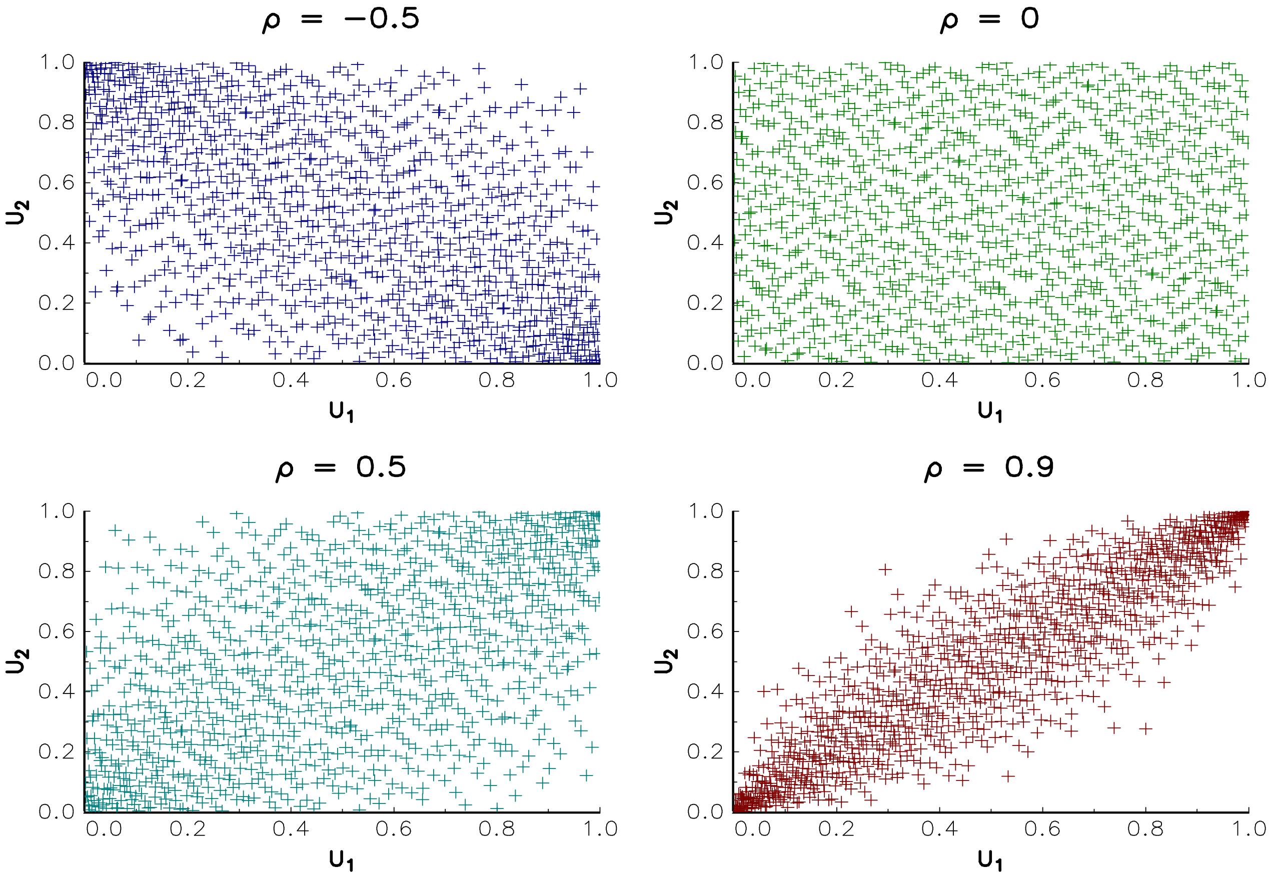


Figure: Simulation of the Normal copula

Method of transformation

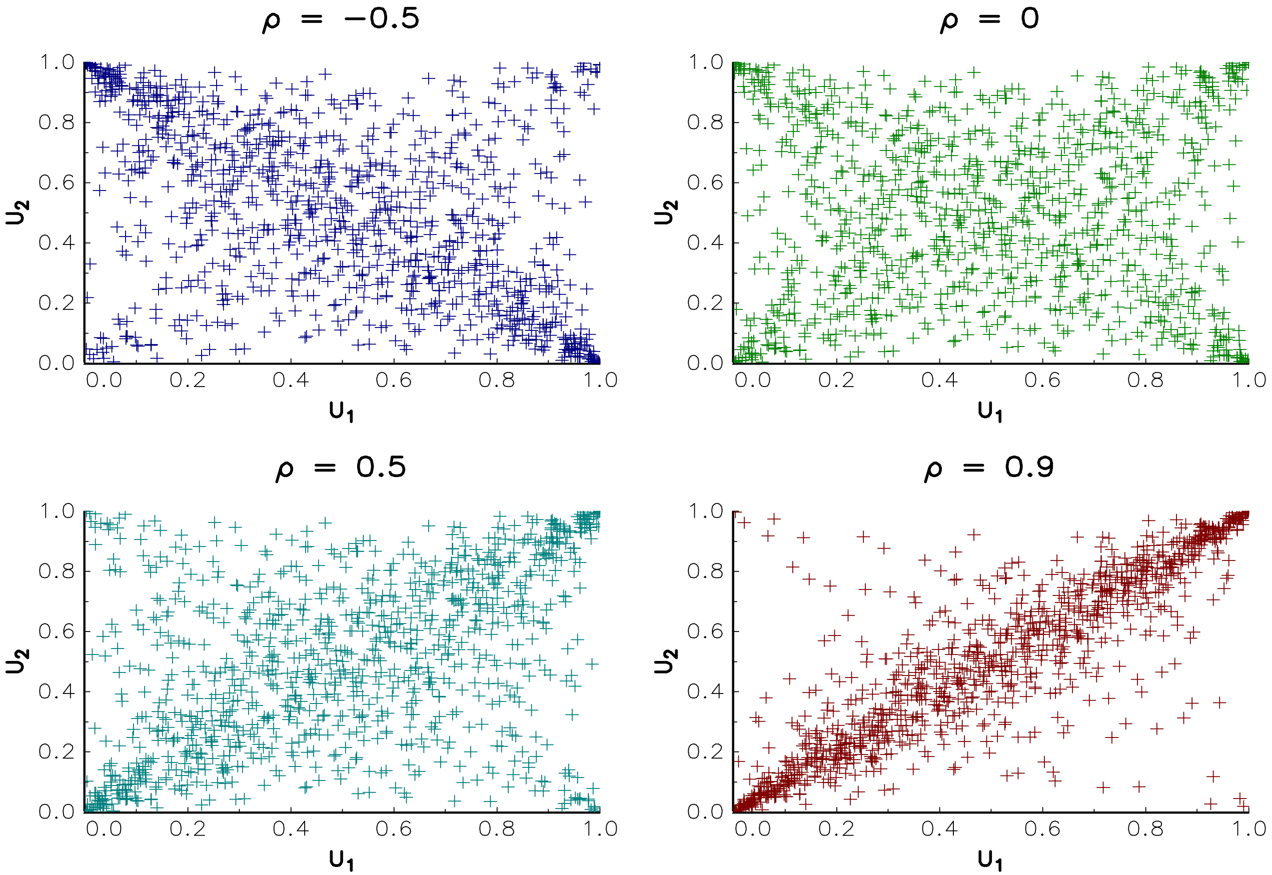


Figure: Simulation of the t_1 copula

Method of transformation

Frailty copulas are defined as:

$$\mathbf{C}\left(u_1, \ldots, u_n\right) = \psi\left(\psi^{-1}\left(u_1\right) + \ldots + \psi^{-1}\left(u_n\right)\right)$$

where $\psi\left(x\right)$ is the Laplace transform of a random variable X
They can be generated using the following algorithm:

1. simulate n independent uniform random variates $v_1, \ldots, v_n$;
2. simulate the frailty random variate x with the Laplace transform ψ;
3. apply the transformation:

$$\left(u_1, \ldots, u_n\right) \leftarrow \left(\psi\left(-\frac{\ln u_1}{x}\right), \ldots, \psi\left(-\frac{\ln u_n}{x}\right)\right)$$

Method of transformation

- The Clayton copula is a frailty copula where $\psi(x) = (1+x)^{-1/\theta}$ is the Laplace transform of the gamma random variable $\mathcal{G}(1/\theta, 1)$
- The algorithm to simulate the Clayton copula is:

$$\begin{cases} x \leftarrow \mathcal{G}(1/\theta, 1) \\ (u_1, \ldots, u_n) \leftarrow \left(\left(1 - \dfrac{\ln u_1}{x}\right)^{-1/\theta}, \ldots, \left(1 - \dfrac{\ln u_n}{x}\right)^{-1/\theta} \right) \end{cases}$$

Method of transformation

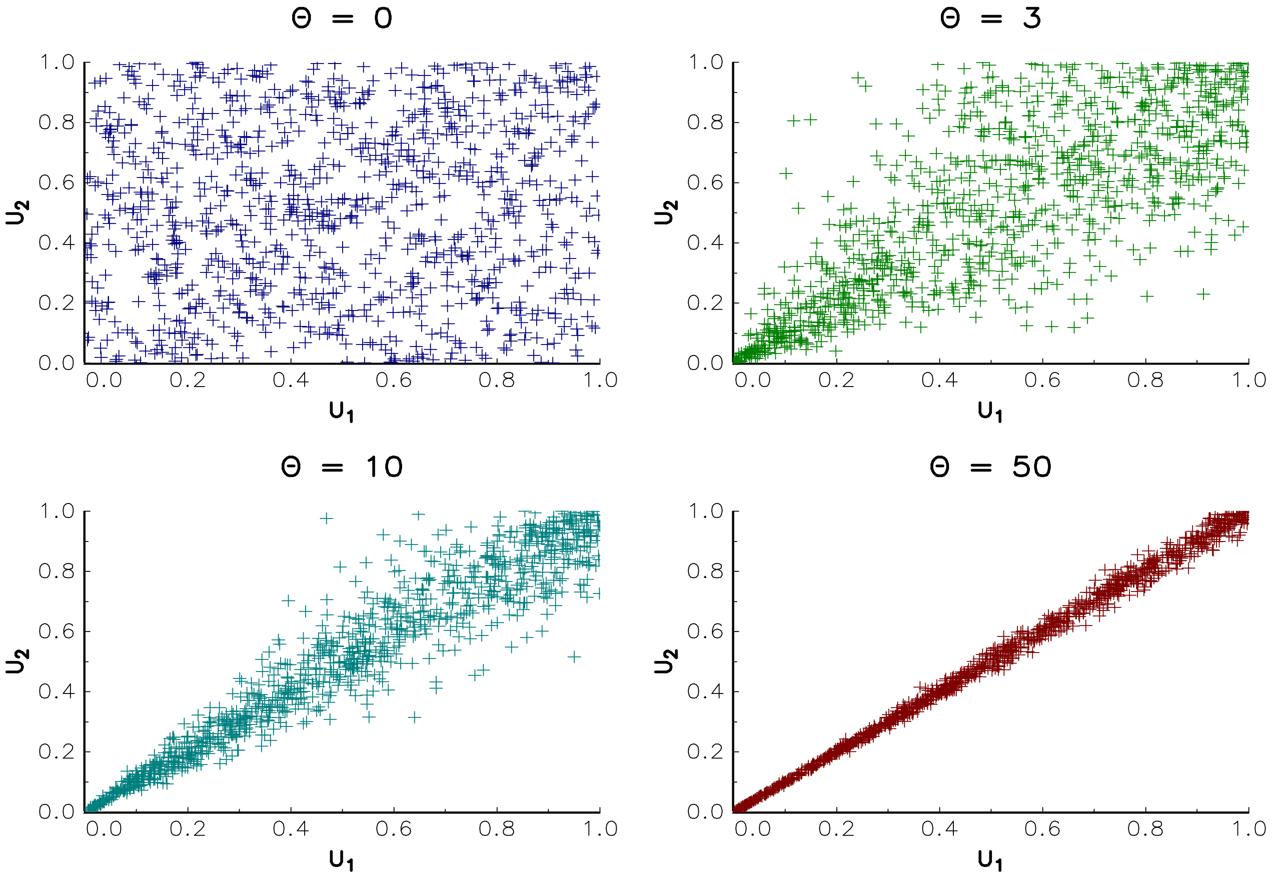


Figure: Simulation of the Clayton copula

Method of transformation

- We consider the multivariate distribution $\mathbf{F}(x_1, \ldots, x_n)$, whose canonical decomposition is defined as:

$$\mathbf{F}(x_1, \ldots, x_n) = \mathbf{C}(\mathbf{F}_1(x_1), \ldots, \mathbf{F}_n(x_n))$$

- If $(U_1, \ldots, U_n) \sim \mathbf{C}$, the random vector $(X_1, \ldots, X_n) = \left(\mathbf{F}_1^{-1}(U_1), \ldots, \mathbf{F}_n^{-1}(U_n)\right)$ follows the distribution function $\mathbf{F}$
- We deduce the following algorithm:

$$\left\{\begin{array}{l} (u_1, \ldots, u_n) \leftarrow \mathbf{C} \\ (x_1, \ldots, x_n) \leftarrow \left(\mathbf{F}_1^{-1}(u_1), \ldots, \mathbf{F}_n^{-1}(u_n)\right) \end{array}\right.$$

Method of transformation

- We assume that $\boldsymbol{\tau} \sim \mathcal{E}(5\%)$ and $\mathrm{LGD} \sim \mathcal{B}(2,2)$
- We also assume that the default time and the loss given default are correlated and the dependence function is a Clayton copula

Method of transformation

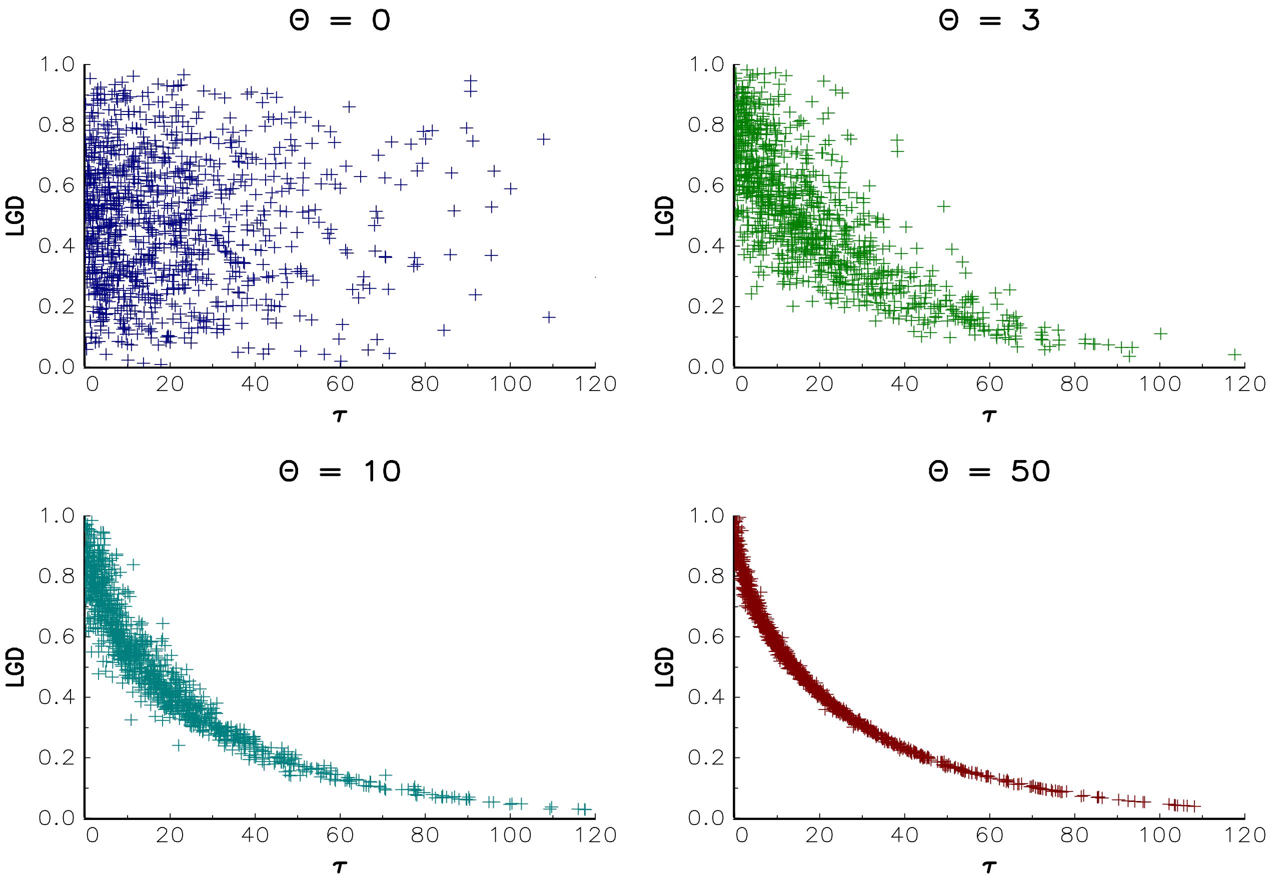


Figure: Simulation of the correlated random vector (τ, LGD)

Method of transformation

Remark

The previous algorithms suppose that we know the analytical expression $\mathbf{F}_i$ of the univariate probability distributions in order to calculate the quantile $\mathbf{F}_i^{-1}$. This is not always the case. For instance, in the operational risk, the loss of the bank is equal to the sum of aggregate losses:

$$L = \sum_{k=1}^{K} S_k$$

where S_k is also the sum of individual losses for the k^{th} cell of the mapping matrix. In practice, the probability distribution of S_k is estimated by the method of simulations

Method of transformation

The method of the empirical quantile function is implemented as follows:

1. for each random variable X_i, simulate m_1 random variates $x_{i,m}^{\star}$ and estimate the empirical distribution $\hat{\mathbf{F}}_i$;
2. simulate a random vector $(u_1, \ldots, u_n)$ from the copula function $\mathbf{C}(u_1, \ldots, u_n)$;
3. simulate the random vector $(x_1, \ldots, x_n)$ by inverting the empirical distributions $\hat{\mathbf{F}}_i$:

   $$x_i \leftarrow \hat{\mathbf{F}}_i^{-1}(u_i)$$

   we also have:

   $$x_i \leftarrow \inf \left\{ x \,\middle|\, \frac{1}{m_1} \sum\nolimits_{m=1}^{m_1} \mathbf{1}\left\{ x \leq x_{i,m}^{\star} \right\} \geq u_i \right\}$$

4. repeat steps 2 and 3 m_2 times

Method of transformation

- $X_1 \sim \mathcal{N}(0,1)$
- $X_2 \sim \mathcal{N}(0,1)$
- The dependence function of (X_1, X_2) is the Clayton copula with parameter $\theta = 3$

Method of transformation

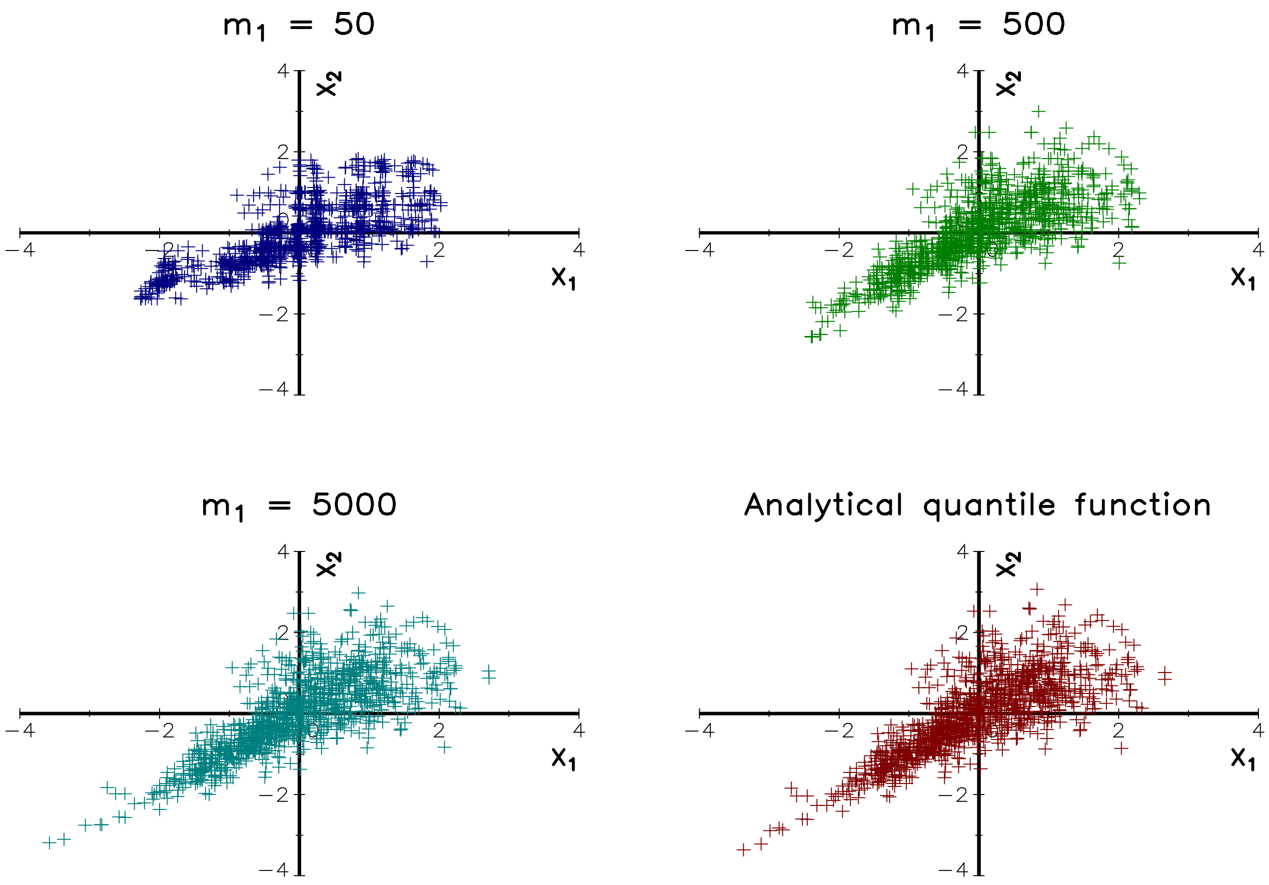


Figure: Convergence of the method of the empirical quantile function

Method of transformation

- $X_1 \sim \mathcal{N}(-1, 2)$, $X_2 \sim \mathcal{N}(0, 1)$, $Y_1 \sim \mathcal{G}(0.5)$ and $Y_2 \sim \mathcal{G}(1, 2)$ are four independent random variables
- Let $(Z_1 = X_1 + Y_1, Z_2 = X_2 \cdot Y_2)$ be the random vector
- The dependence function of Z is the t copula with parameters $\nu = 2$ and $\rho = -70\%$
- It is not possible to find an analytical expression of the marginal distributions of Z_1 and Z_2

Method of transformation

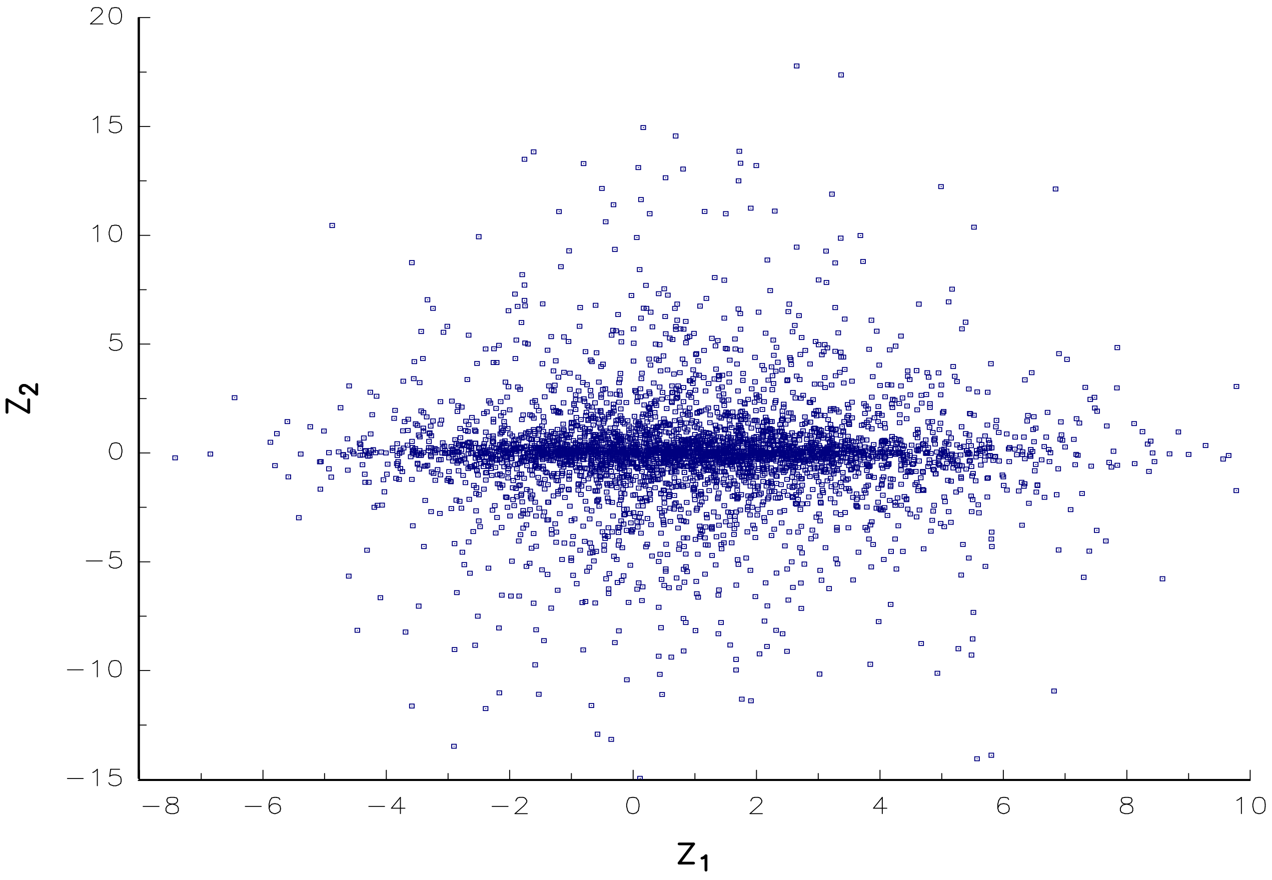


Figure: Simulation of the random variables Z_1 and Z_2

Method of transformation

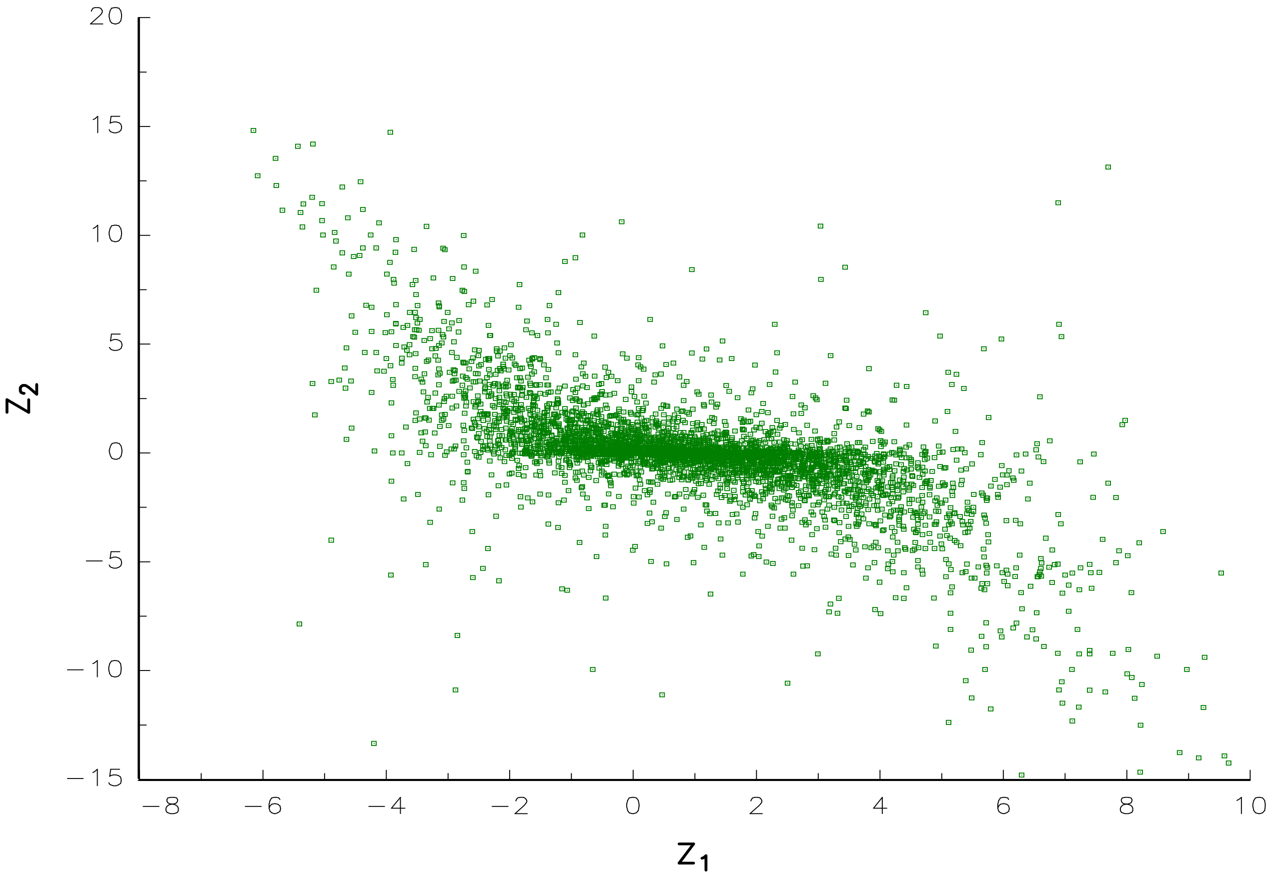


Figure: Simulation of the random vector (Z_1, Z_2)

Random matrices

- Orthogonal and covariance matrices
- Correlation matrices
- Wishart matrices

⇒ HFRM, Chapter 13, Section 13.1.4, pages 807-813

Brownian motion

- A Brownian motion (or a Wiener process) is a stochastic process $W(t)$, whose increments are stationary and independent:

$$W(t) - W(s) \sim \mathcal{N}(0, t-s)$$

- We have:

$$\left\{ \begin{array}{l} W(0) = 0 \\ W(t) = W(s) + \epsilon(s,t) \end{array} \right.$$

  where $\epsilon(s,t) \sim \mathcal{N}(0, t-s)$ are *iid* random variables

- To simulate $W(t)$ at different dates $t_1, t_2, \ldots$, we have:

$$W_{m+1} = W_m + \sqrt{t_{m+1} - t_m} \cdot \varepsilon_m$$

  where W_m is the numerical realization of $W(t_m)$ and $\varepsilon_m \sim \mathcal{N}(0,1)$ are *iid* random variables

- In the case of fixed-interval times $t_{m+1} - t_m = h$, we obtain the recursion:

$$W_{m+1} = W_m + \sqrt{h} \cdot \varepsilon_m$$

Geometric Brownian motion

- The geometric Brownian motion is described by the following SDE:

$$\left\{\begin{array}{l} \mathrm{d}X(t) = \mu X(t)\,\mathrm{d}t + \sigma X(t)\,\mathrm{d}W(t) \\ X(0) = x_0 \end{array}\right.$$

- Its solution is given by:

$$X(t) = x_0 \cdot \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma W(t)\right) = g(W(t))$$

Geometric Brownian motion

1. Simulating the geometric Brownian motion $X(t)$ can be done by applying the transform method to the process $W(t)$
2. Another approach to simulate $X(t)$ consists in using the following formula:

$$X(t) = X(s) \cdot \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)(t-s) + \sigma\left(W(t) - W(s)\right)\right)$$

We have:

$$X_{m+1} = X_m \cdot \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)(t_{m+1} - t_m) + \sigma\sqrt{t_{m+1} - t_m} \cdot \varepsilon_m\right)$$

where $X_m = X(t_m)$ and $\varepsilon_m \sim \mathcal{N}(0,1)$ are *iid* random variables

3. If we consider fixed-interval times, the numerical realization becomes:

$$X_{m+1} = X_m \cdot \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)h + \sigma\sqrt{h} \cdot \varepsilon_m\right)$$

Geometric Brownian motion

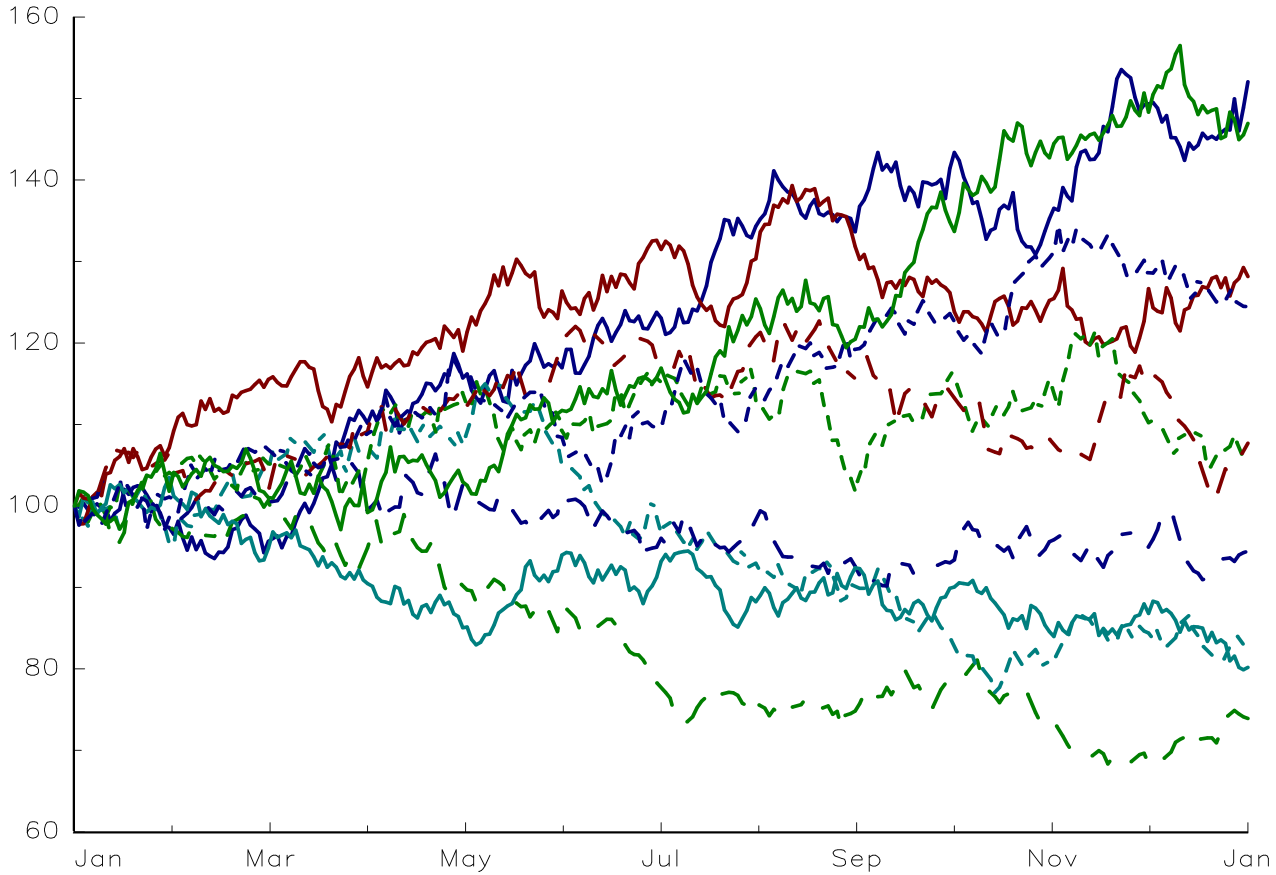


Figure: Simulation of the geometric Brownian motion

Ornstein-Uhlenbeck process

- The stochastic differential equation of the Ornstein-Uhlenbeck process is:

$$\left\{ \begin{array}{l} \mathrm{d}X(t) = a\left(b - X(t)\right)\mathrm{d}t + \sigma\,\mathrm{d}W(t) \\ X(0) = x_0 \end{array} \right.$$

- The solution of the SDE is:

$$X(t) = x_0 e^{-at} + b\left(1 - e^{-at}\right) + \sigma \int_0^t e^{a(\theta - t)}\,\mathrm{d}W(\theta)$$

- We also have:

$$X(t) = X(s)\,e^{-a(t-s)} + b\left(1 - e^{-a(t-s)}\right) + \sigma \int_s^t e^{a(\theta - t)}\,\mathrm{d}W(\theta)$$

where:

$$\int_s^t e^{a(\theta - t)}\,\mathrm{d}W(\theta) \sim \mathcal{N}\left(0, \frac{1 - e^{-2a(t-s)}}{2a}\right)$$

Ornstein-Uhlenbeck process

If we consider fixed-interval times, we obtain the following simulation scheme:

$$X_{m+1} = X_m e^{-ah} + b\left(1 - e^{-ah}\right) + \sigma \sqrt{\frac{1 - e^{-2ah}}{2a}} \cdot \varepsilon_m$$

where $\varepsilon_m \sim \mathcal{N}(0,1)$ are *iid* random variables

Ornstein-Uhlenbeck process

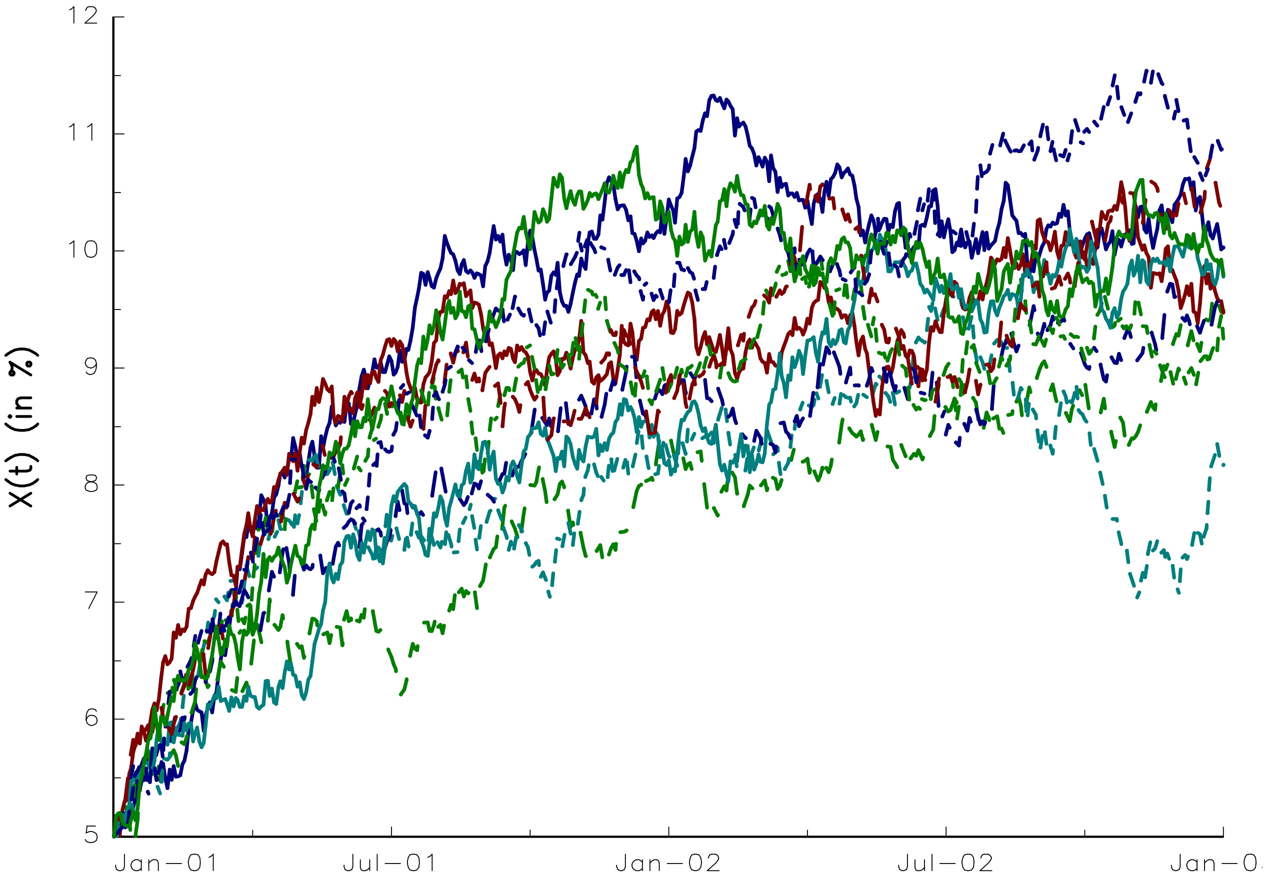


Figure: Simulation of the Ornstein-Uhlenbeck process

Stochastic differential equations without an explicit solution

- Let $X(t)$ be the solution of the following SDE:

$$\begin{cases} \mathrm{d}X(t) = \mu(t, X)\,\mathrm{d}t + \sigma(t, X)\,\mathrm{d}W(t) \\ X(0) = x_0 \end{cases}$$

- The Euler-Maruyama scheme uses the following approximation:

$$X(t) - X(s) \approx \mu(t, X(s)) \cdot (t - s) + \sigma(t, X(s)) \cdot (W(t) - W(s))$$

- If we consider fixed-interval times, the Euler-Maruyama scheme becomes:

$$X_{m+1} = X_m + \mu(t_m, X_m)\,h + \sigma(t_m, X_m)\sqrt{h} \cdot \varepsilon_m$$

where $\varepsilon_m \sim \mathcal{N}(0, 1)$ are *iid* random variables

Stochastic differential equations without an explicit solution

The fixed-interval Milstein scheme is:

$$\begin{aligned} X_{m+1} &= X_m + \mu\left(t_m, X_m\right) h + \sigma\left(t_m, X_m\right)\sqrt{h}\cdot\varepsilon_m + \\ & \frac{1}{2}\sigma\left(t_m, X_m\right)\partial_x\sigma\left(t_m, X_m\right) h\left(\varepsilon_m^2 - 1\right) \end{aligned}$$

Stochastic differential equations without an explicit solution

If we consider the geometric Brownian motion, the Euler-Maruyama scheme is:

$$X_{m+1} = X_m + \mu X_m h + \sigma X_m \sqrt{h} \cdot \varepsilon_m$$

whereas the Milstein scheme is:

$$\begin{aligned} X_{m+1} &= X_m + \mu X_m h + \sigma X_m \sqrt{h} \cdot \varepsilon_m + \frac{1}{2}\sigma^2 X_m h \left(\varepsilon_m^2 - 1\right) \\ &= X_m + \left(\mu - \frac{1}{2}\sigma^2\right) X_m h + \sigma X_m \sqrt{h} \left(1 + \frac{1}{2}\sigma \sqrt{h} \varepsilon_m\right) \varepsilon_m \end{aligned}$$

Stochastic differential equations without an explicit solution

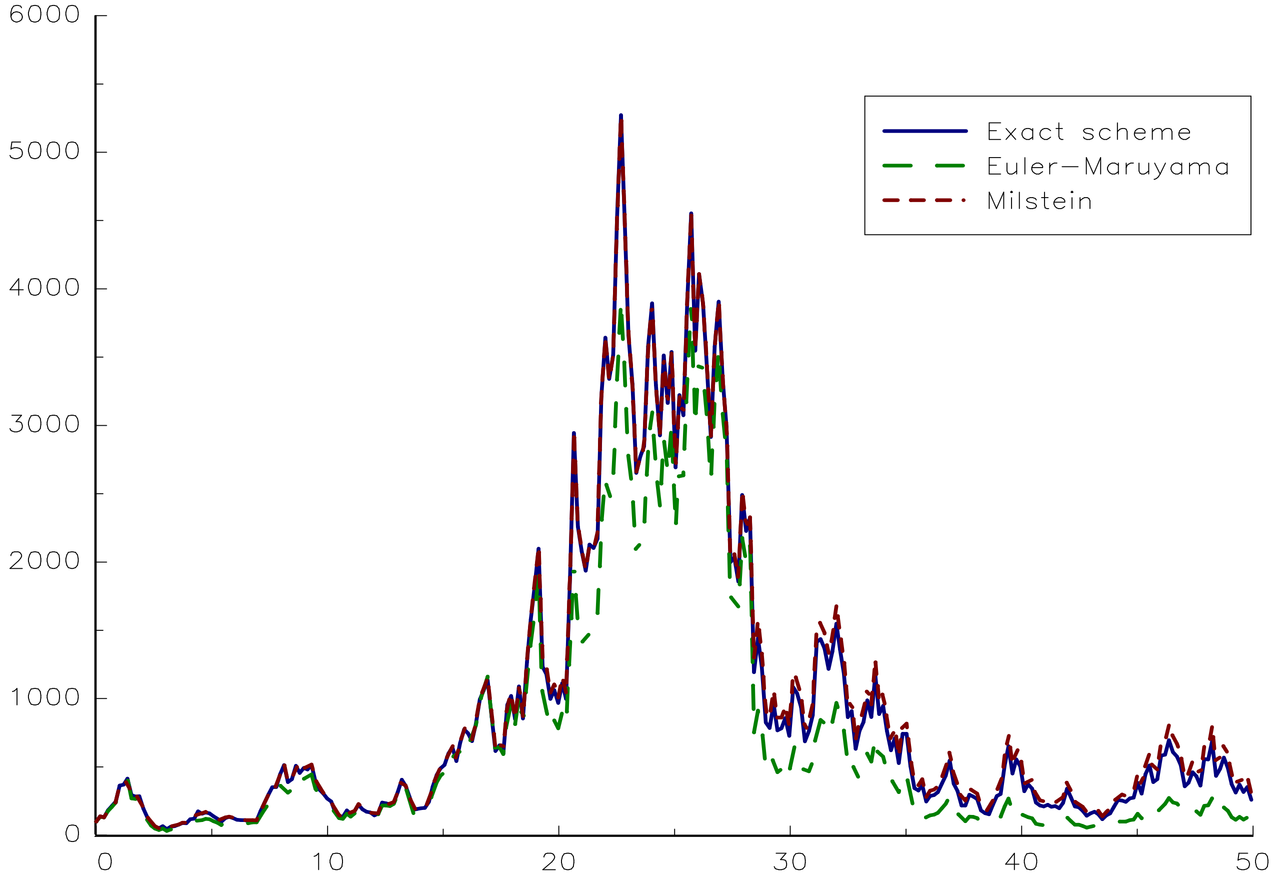


Figure: Comparison of exact, Euler-Maruyama and Milstein schemes (monthly discretization)

Stochastic differential equations without an explicit solution

When we don't know the analytical solution of $X(t)$, it is natural to simulate the numerical solution of $X(t)$ using Euler-Maruyama and Milstein schemes. However, it may be sometimes more efficient to find the numerical solution of $Y(t) = f(t, X(t))$ instead of $X(t)$ itself, in particular when $Y(t)$ is more regular than $X(t)$

Stochastic differential equations without an explicit solution

- By It's lemma, we have:

$$\begin{aligned} \mathrm{d}Y(t) &= \left(\partial_t f(t,X) + \mu(t,X)\,\partial_x f(t,X) + \frac{1}{2}\sigma^2(t,X)\,\partial_x^2 f(t,X)\right)\mathrm{d}t + \\ & \sigma(t,X)\,\partial_x f(t,X)\,\mathrm{d}W(t) \end{aligned}$$

- By using the inverse function $X(t) = f^{-1}(t, Y(t))$, we obtain:

$$\mathrm{d}Y(t) = \mu'(t,Y)\,\mathrm{d}t + \sigma'(t,Y)\,\mathrm{d}W(t)$$

  where $\mu'(t,Y)$ and $\sigma'(t,Y)$ are functions of $\mu(t,X)$, $\sigma(t,X)$ and $f(t,X)$

- We can then simulate the solution of $Y(t)$ using an approximation scheme and deduce the numerical solution of $X(t)$ by applying the transformation method:

$$X_m = f^{-1}(t_m, Y_m)$$

Stochastic differential equations without an explicit solution

Let us consider the geometric Brownian motion $X(t)$. The solution of $Y(t) = \ln X(t)$ is equal to:

$$\mathrm{d}Y(t) = \left(\mu - \frac{1}{2}\sigma^2\right)\mathrm{d}t + \sigma\,\mathrm{d}W(t)$$

We deduce that the Euler-Maruyama (or Milstein) scheme with fixed-interval times is:

$$Y_{m+1} = Y_m + \left(\mu - \frac{1}{2}\sigma^2\right)h + \sigma\sqrt{h}\cdot\varepsilon_m$$

It follows that:

$$\ln X_{m+1} = \ln X_m + \left(\mu - \frac{1}{2}\sigma^2\right)h + \sigma\sqrt{h}\cdot\varepsilon_m$$

Stochastic differential equations without an explicit solution

The CIR process is $\mathrm{d}X(t) = (\alpha + \beta X(t))\,\mathrm{d}t + \sigma\sqrt{X(t)}\,\mathrm{d}W(t)$. Using the transformation $Y(t) = \sqrt{X(t)}$, we obtain the following SDE:

$$\begin{aligned}
\mathrm{d}Y(t) &= \left(\frac{1}{2}\frac{(\alpha + \beta X(t))}{\sqrt{X(t)}} - \frac{1}{8}\frac{\sigma^2 X(t)}{X(t)^{3/2}}\right)\mathrm{d}t + \frac{1}{2}\frac{\sigma\sqrt{X(t)}}{\sqrt{X(t)}}\,\mathrm{d}W(t) \\
&= \frac{1}{2Y(t)}\left(\alpha + \beta Y^2(t) - \frac{1}{4}\sigma^2\right)\mathrm{d}t + \frac{1}{2}\sigma\,\mathrm{d}W(t)
\end{aligned}$$

We deduce that the Euler-Maruyama scheme of $Y(t)$ is:

$$Y_{m+1} = Y_m + \frac{1}{2Y_m}\left(\alpha + \beta Y_m^2 - \frac{1}{4}\sigma^2\right)h + \frac{1}{2}\sigma\sqrt{h}\cdot\varepsilon_m$$

It follows that:

$$X_{m+1} = \left(\sqrt{X_m} + \frac{1}{2\sqrt{X_m}}\left(\alpha + \beta X_m - \frac{1}{4}\sigma^2\right)h + \frac{1}{2}\sigma\sqrt{h}\cdot\varepsilon_m\right)^2$$

Poisson process

Let t_m be the time when the m^{th} event occurs. The numerical algorithm is then:

1. we set $t_0 = 0$ and $N(t_0) = 0$
2. we generate a uniform random variate u and calculate the random variate $e \sim \mathcal{E}(\lambda)$ with the formula:
$$e = -\frac{\ln u}{\lambda}$$
3. we update the Poisson process with:
$$t_{m+1} \leftarrow t_m + e \qquad \text{and} \qquad N(t_{m+1}) \leftarrow N(t_m) + 1$$
4. we go back to step 2

Mixed Poisson process (MPP)

The algorithm is initialized with a realization λ of the random intensity Λ

Non-homogenous Poisson process (NHPP)

- $\lambda(t)$ varies with time
- The inter-arrival times remain independent and exponentially distributed with:

$$\Pr\{T_1 > t\} = \exp(-\Lambda(t))$$

where T_1 is the duration of the first event and $\Lambda(t)$ is the integrated intensity function:

$$\Lambda(t) = \int_0^t \lambda(s)\,\mathrm{d}s$$

- It follows that:

$$\Pr\{T_1 > \Lambda^{-1}(t)\} = \exp(-t) \Leftrightarrow \Pr\{\Lambda(T_1) > t\} = \exp(-t)$$

Non-homogenous Poisson process (NHPP)

We deduce that if $\{t_1, t_2, \ldots, t_M\}$ are the occurrence times of the NHPP of intensity $\lambda(t)$, then $\{\Lambda(t_1), \Lambda(t_2), \ldots, \Lambda(t_M)\}$ are the occurrence times of the homogeneous Poisson process (HPP) of intensity one. Therefore, the algorithm is:

1. we simulate t'_m the time arrivals of the homogeneous Poisson process with intensity $\lambda = 1$
2. we apply the transform $t_m = \Lambda^{-1}(t'_m)$

Non-homogenous Poisson process (NHPP)

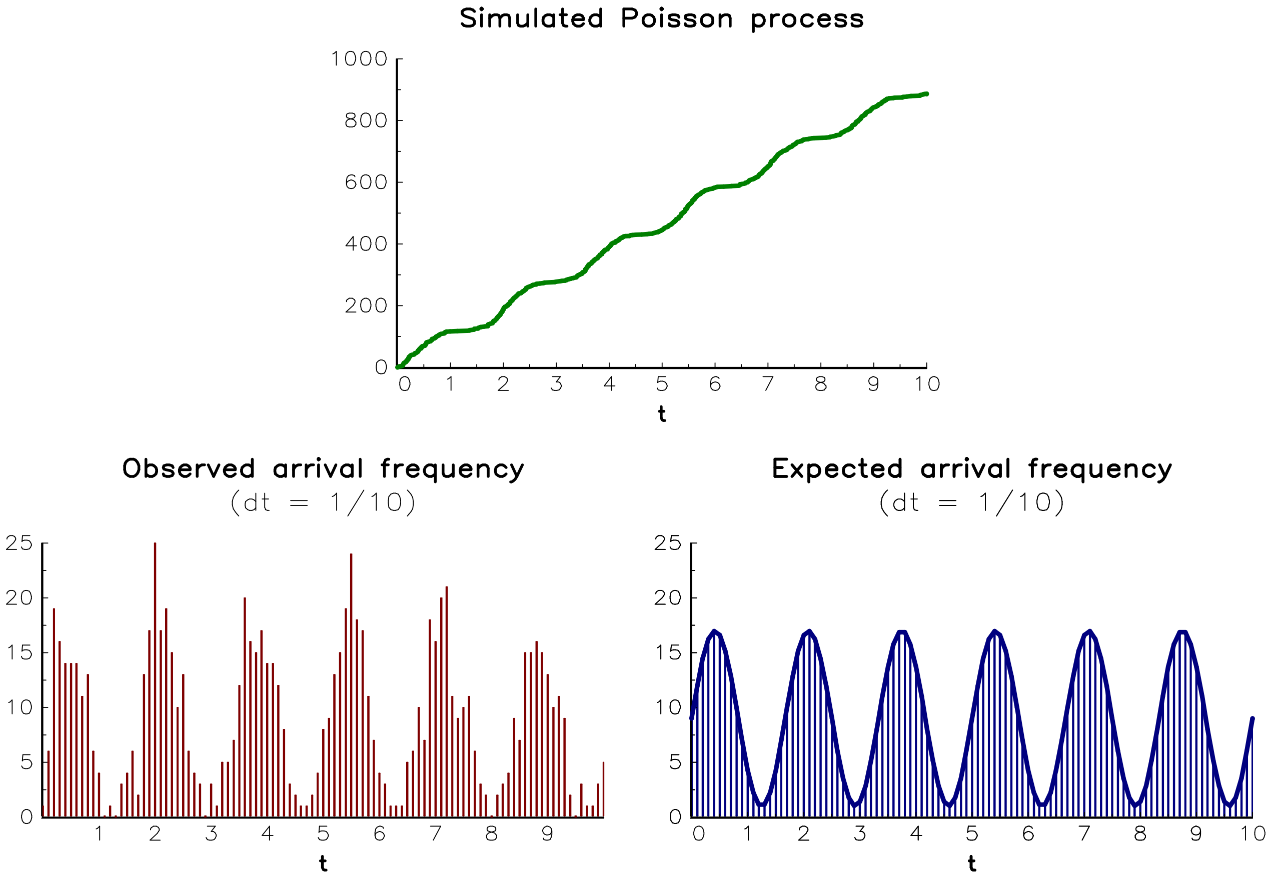


Figure: Simulation of a non-homogenous Poisson process with cyclical intensity

Multidimensional Brownian motion

- Let $W(t) = (W_1(t), \ldots, W_n(t))$, be a n-dimensional Brownian motion
- Each component $W_i(t)$ is a Brownian motion:

$$W_i(t) - W_i(s) \sim \mathcal{N}(0, t - s)$$

- We have:

$$\mathbb{E}[W_i(t) W_j(s)] = \min(t, s) \cdot \rho_{i,j}$$

where $\rho_{i,j}$ is the correlation between the two Brownian motions W_i and W_j

- We deduce that:

$$\left\{ \begin{array}{l} W(0) = \mathbf{0} \\ W(t) = W(s) + \epsilon(s, t) \end{array} \right.$$

where $\epsilon(s, t) \sim \mathcal{N}_n(\mathbf{0}, (t - s)\ \rho)$ are *iid* random vectors

Multidimensional Brownian motion

- It follows that the numerical solution is:

$$W_{m+1} = W_m + \sqrt{t_{m+1} - t_m} \cdot P \cdot \varepsilon_m$$

  where P is the Cholesky decomposition of the correlation matrix ρ and $\varepsilon_m \sim \mathcal{N}_n(0, I)$ are *iid* random vectors

- In the case of fixed-interval times, the recursion becomes:

$$W_{m+1} = W_m + \sqrt{h} \cdot P \cdot \varepsilon_m$$

Multidimensional Brownian motion

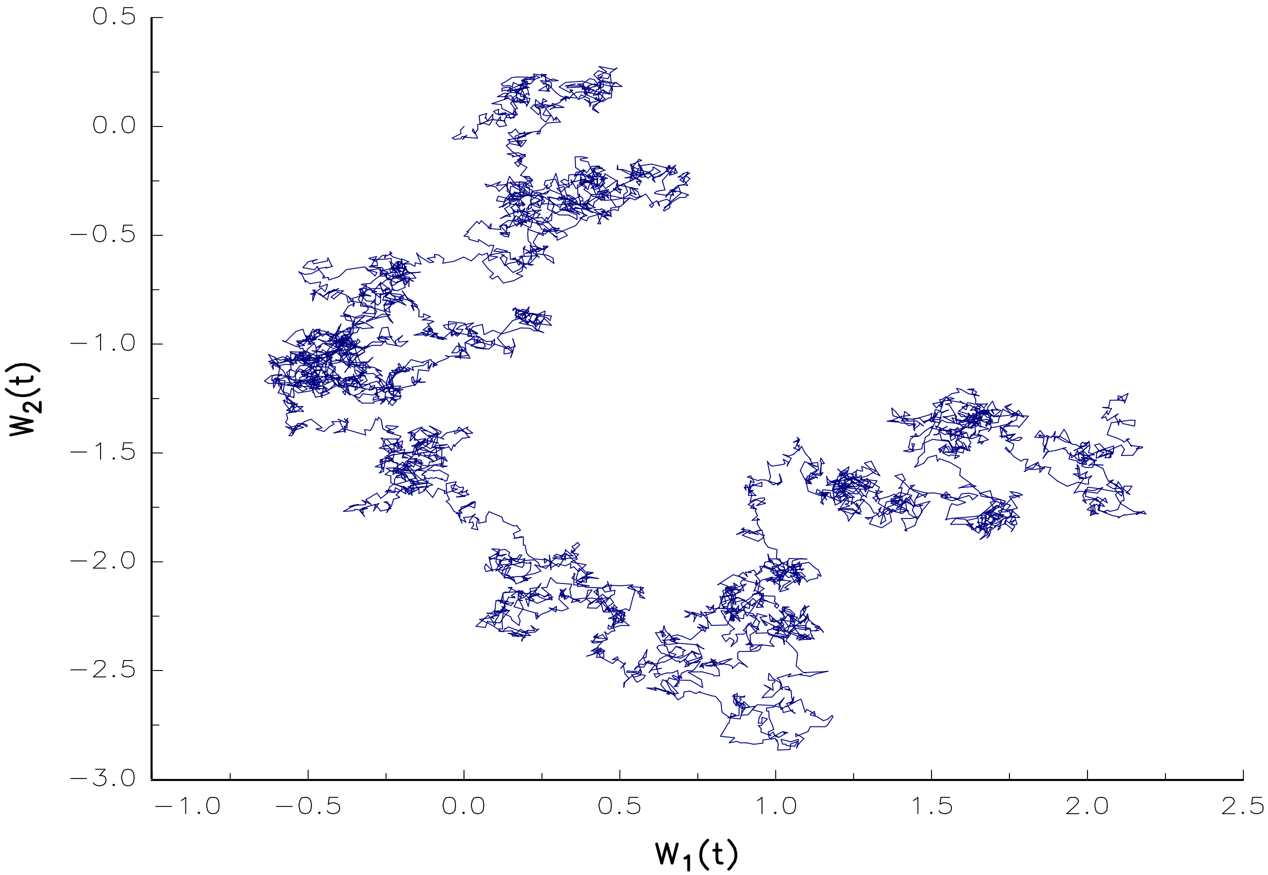


Figure: Brownian motion in the plane (independent case)

Multidimensional Brownian motion

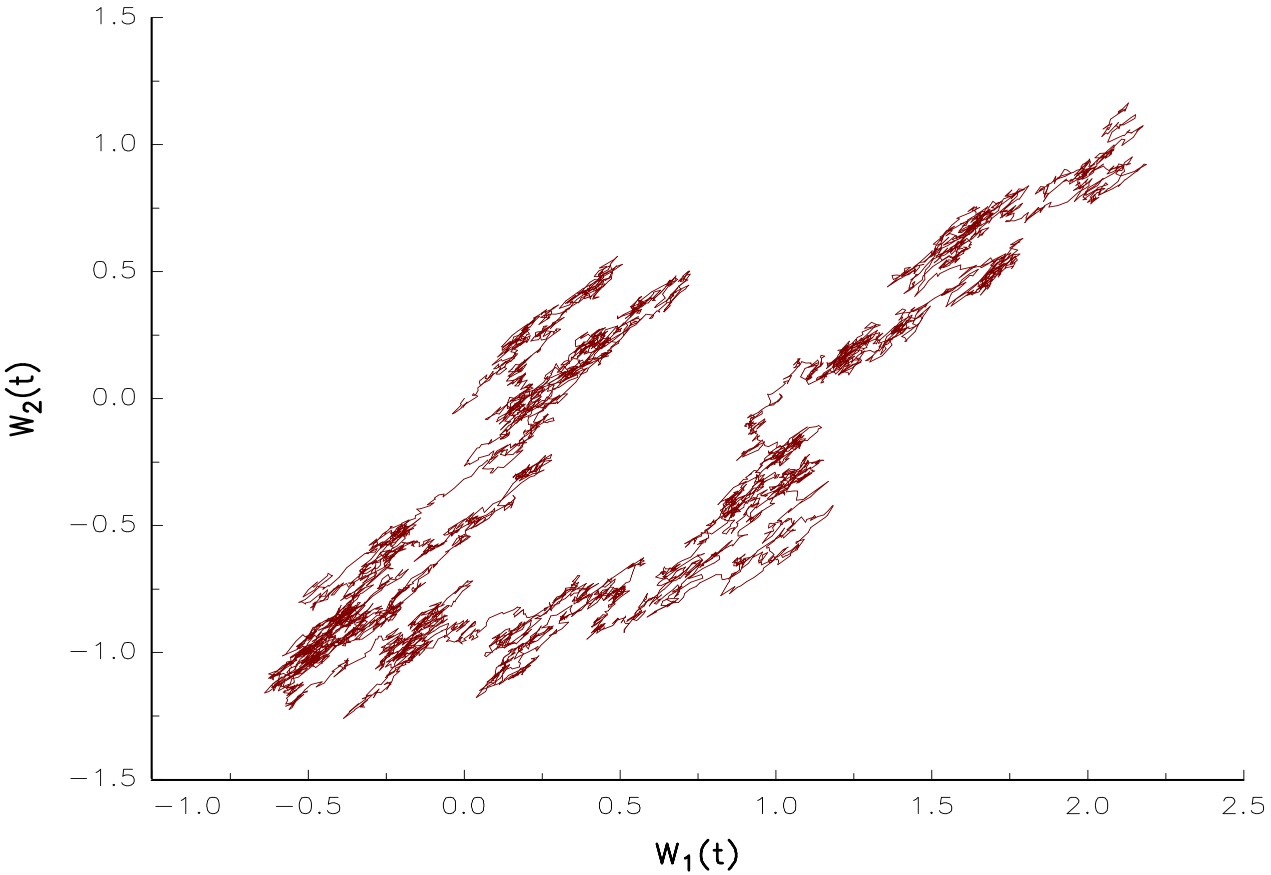


Figure: Brownian motion in the plane ($\rho_{1,2} = 85\%$)

Multidimensional geometric Brownian motion

- We consider the multidimensional geometric Brownian motion:

$$\left\{\begin{array}{l} \mathrm{d}X(t) = \mu \odot X(t)\,\mathrm{d}t + \mathrm{diag}(\sigma \odot X(t))\,\mathrm{d}W(t) \\ X(0) = x_0 \end{array}\right.$$

  where $X(t) = (X_1(t), \ldots, X_n(t))$, $\mu = (\mu_1, \ldots, \mu_n)$, $\sigma = (\sigma_1, \ldots, \sigma_n)$ and $W(t) = (W_1(t), \ldots, W_n(t))$ is a n-dimensional Brownian motion with $\mathbb{E}\left[W(t)\,W(t)^\top\right] = \rho\,t$

- If we consider the j^{th} component of $X(t)$, we have:

$$\mathrm{d}X_j(t) = \mu_j X_j(t)\,\mathrm{d}t + \sigma_j X_j(t)\,\mathrm{d}W_j(t)$$

- The solution of the multidimensional SDE is a multivariate log-normal process with:

$$X_j(t) = X_j(0) \cdot \exp\left(\left(\mu_j - \frac{1}{2}\sigma_j^2\right)t + \sigma_j W_j(t)\right)$$

  where $W(t) \sim \mathcal{N}_n(0, \rho\,t)$

Multidimensional geometric Brownian motion

- We deduce that the exact scheme to simulate the multivariate GBM is:

$$\left\{\begin{array}{l} X_{1,m+1} = X_{1,m} \cdot \exp\left(\left(\mu_1 - \frac{1}{2}\sigma_1^2\right)\left(t_{m+1} - t_m\right) + \sigma_1\sqrt{t_{m+1} - t_m} \cdot \varepsilon_{1,m}\right) \\ \vdots \\ X_{j,m+1} = X_{j,m} \cdot \exp\left(\left(\mu_j - \frac{1}{2}\sigma_j^2\right)\left(t_{m+1} - t_m\right) + \sigma_j\sqrt{t_{m+1} - t_m} \cdot \varepsilon_{j,m}\right) \\ \vdots \\ X_{n,m+1} = X_{n,m} \cdot \exp\left(\left(\mu_n - \frac{1}{2}\sigma_n^2\right)\left(t_{m+1} - t_m\right) + \sigma_n\sqrt{t_{m+1} - t_m} \cdot \varepsilon_{n,m}\right) \end{array}\right.$$

where $(\varepsilon_{1,m}, \ldots, \varepsilon_{n,m}) \sim \mathcal{N}_n\left(\mathbf{0}, \rho\right)$

Euler-Maruyama and Milstein schemes

- We consider the general SDE:

$$\left\{ \begin{array}{l} \mathrm{d}X\left(t\right) = \mu\left(t, X\left(t\right)\right)\,\mathrm{d}t + \sigma\left(t, X\left(t\right)\right)\,\mathrm{d}W\left(t\right) \\ X\left(0\right) = x_0 \end{array} \right.$$

  where $X\left(t\right)$ and $\mu\left(t, X\left(t\right)\right)$ are $n \times 1$ vectors, $\sigma\left(t, X\left(t\right)\right)$ is a $n \times p$ matrix and $W\left(t\right)$ is a $p \times 1$ vector

- We assume that $\mathbb{E}\left[W\left(t\right) W\left(t\right)^\top\right] = \rho\, t$, where ρ is a $p \times p$ correlation matrix

Euler-Maruyama and Milstein schemes

- The corresponding Euler-Maruyama scheme is:

$$X_{m+1} = X_m + \mu\left(t_m, X_m\right) \cdot \left(t_{m+1} - t_m\right) + \sigma\left(t_m, X_m\right)\sqrt{t_{m+1} - t_m} \cdot \varepsilon_m$$

  where $\varepsilon_m \sim \mathcal{N}_p\left(0, \rho\right)$

- In the case of a diagonal system, we retrieve the one-dimensional scheme:

$$X_{j,m+1} = X_{j,m} + \mu_j\left(t_m, X_{j,m}\right) \cdot \left(t_{m+1} - t_m\right) + \sigma_{j,j}\left(t_m, X_{j,m}\right) \cdot \sqrt{t_{m+1} - t_m \varepsilon_{j,m}}$$

  However, the random variables $\varepsilon_{j,m}$ and $\varepsilon_{j',m}$ may be correlated

Euler-Maruyama and Milstein schemes

We consider the Heston model:

$$\left\{ \begin{array}{l} \mathrm{d}X(t) = \mu X(t)\,\mathrm{d}t + \sqrt{v(t)}X(t)\,\mathrm{d}W_1(t) \\ \mathrm{d}v(t) = a(b - v(t))\,\mathrm{d}t + \sigma\sqrt{v(t)}\,\mathrm{d}W_2(t) \end{array} \right.$$

where $\mathbb{E}[W_1(t)\,W_2(t)] = \rho\,t$. By applying the fixed-interval Euler-Maruyama scheme to $(\ln X(t), v(t))$, we obtain:

$$\ln X_{m+1} = \ln X_m + \left(\mu - \frac{1}{2}v_m\right)h + \sqrt{v_m h}\cdot\varepsilon_{1,m}$$

and:

$$v_{m+1} = v_m + a(b - v_m)\,h + \sigma\sqrt{v_m h}\cdot\varepsilon_{2,m}$$

Here, $\varepsilon_{1,m}$ and $\varepsilon_{2,m}$ are two standard Gaussian random variables with correlation ρ

Euler-Maruyama and Milstein schemes

The multidimensional version of the Milstein scheme is:

$$\begin{aligned} X_{j,m+1} &= X_{j,m} + \mu_j \left(t_m, X_m\right) \left(t_{m+1} - t_m\right) + \sum_{k=1}^{p} \sigma_{j,k} \left(t_m, X_m\right) \Delta W_{k,m} + \\ & \sum_{k=1}^{p} \sum_{k'=1}^{p} \mathcal{L}^{(k)} \sigma_{j,k'} \left(t_m, X_m\right) \mathcal{I}_{(k,k')} \end{aligned}$$

where $\Delta W_{k,m} = W_k \left(t_{m+1}\right) - W_k \left(t_m\right)$ and:

$$\mathcal{L}^{(k)} f \left(t, x\right) = \sum_{k''=1}^{n} \sigma_{k'',k} \left(t_m, X_m\right) \frac{\partial f \left(t, x\right)}{\partial x_{k''}}$$

and:

$$\mathcal{I}_{(k,k')} = \int_{t_m}^{t_{m+1}} \int_{t_m}^{s} \mathrm{d}W_k \left(t\right) \mathrm{d}W_{k'} \left(s\right)$$

Euler-Maruyama and Milstein schemes

In the case of a diagonal system, the Milstein scheme may be simplified as follows:

$$\begin{aligned} X_{j,m+1} &= X_{j,m} + \mu_j\left(t_m, X_{j,m}\right)\left(t_{m+1} - t_m\right) + \sigma_{j,j}\left(t_m, X_{j,m}\right)\Delta W_{j,m} + \\ & \quad \mathcal{L}^{(j)}\sigma_{j,j}\left(t_m, X_{j,m}\right)\mathcal{I}_{(j,j)} \end{aligned}$$

where:

$$\begin{aligned} \mathcal{I}_{(j,j)} &= \int_{t_m}^{t_{m+1}} \int_{t_m}^{s} \mathrm{d}W_j\left(t\right)\,\mathrm{d}W_j\left(s\right) \\ &= \int_{t_m}^{t_{m+1}} \left(W_j\left(s\right) - W_j\left(t_m\right)\right)\mathrm{d}W_j\left(s\right) \\ &= \frac{1}{2}\left(\left(\Delta W_{j,m}\right)^2 - \left(t_{m+1} - t_m\right)\right) \end{aligned}$$

Euler-Maruyama and Milstein schemes

We deduce that the Milstein scheme is:

$$\begin{aligned} X_{j,m+1} &= X_{j,m} + \mu_j\left(t_m, X_{j,m}\right)\left(t_{m+1} - t_m\right) + \\ & \sigma_{j,j}\left(t_m, X_{j,m}\right)\sqrt{t_{m+1} - t_m}\varepsilon_{j,m} + \\ & \frac{1}{2}\sigma_{j,j}\left(t_m, X_{j,m}\right)\partial_{x_j}\,\sigma_{j,j}\left(t_m, X_{j,m}\right)\left(t_{m+1} - t_m\right)\left(\varepsilon_{j,m}^2 - 1\right) \end{aligned}$$

Euler-Maruyama and Milstein schemes

If we apply the fixed-interval Milstein scheme to the Heston model, we obtain:

$$\ln X_{m+1} = \ln X_m + \left(\mu - \frac{1}{2} v_m \right) h + \sqrt{v_m h} \cdot \varepsilon_{1,m}$$

and:

$$v_{m+1} = v_m + a \left(b - v_m \right) h + \sigma \sqrt{v_m h} \cdot \varepsilon_{2,m} + \frac{1}{4} \sigma^2 h \left(\varepsilon_{2,m}^2 - 1 \right)$$

Here, $\varepsilon_{1,m}$ and $\varepsilon_{2,m}$ are two standard Gaussian random variables with correlation ρ

Euler-Maruyama and Milstein schemes

Remark

The multidimensional Milstein scheme is generally not used, because the terms $\mathcal{L}^{(k)}\sigma_{j,k'}\left(t_m, X_m\right)\mathcal{I}_{(k,k')}$ are complicated to simulate. For the Heston model, we obtain a very simple scheme, because we only apply the Milstein scheme to the process $v\left(t\right)$ and not to the vector process $\left(\ln X\left(t\right), v\left(t\right)\right)$

Euler-Maruyama and Milstein schemes

If we also apply the Milstein scheme to $\ln X\left(t\right)$, we obtain:

$$\ln X_{m+1} = \ln X_m + \left(\mu - \frac{1}{2} v_m\right) h + \sqrt{v_m h} \cdot \varepsilon_{1,m} + A_m$$

where:

$$\begin{array}{rcl} A_m & = & \displaystyle\sum_{k=1}^{2} \sum_{k'=1}^{2} \left(\sum_{k''=1}^{2} \sigma_{k'',k}\left(t_m, X_m\right) \frac{\sigma_{1,k'}\left(t_m, X_m\right)}{\partial\, x_{k''}} \right) \mathcal{I}_{(k,k')} \\ & = & \sigma \sqrt{v\left(t\right)} \cdot \dfrac{1}{2\sqrt{v\left(t\right)}} \cdot \mathcal{I}_{(2,1)} \\ & = & \dfrac{\sigma}{2} \cdot \mathcal{I}_{(2,1)} \end{array}$$

Euler-Maruyama and Milstein schemes

Let $W_2(t) = \rho W_1(t) + \sqrt{1-\rho^2} W^\star(t)$ where $W^\star(t)$ is a Brownian motion independent from $W_1(t)$. It follows that:

$$\begin{aligned}
\mathcal{I}_{(2,1)} &= \int_{t_m}^{t_{m+1}} \int_{t_m}^{s} \mathrm{d}W_2(t)\,\mathrm{d}W_1(s) \\
&= \int_{t_m}^{t_{m+1}} \left(\rho W_1(s) + \sqrt{1-\rho^2} W^\star(s)\right) \mathrm{d}W_1(s) - \\
&\quad \int_{t_m}^{t_{m+1}} \left(\rho W_1(t_m) + \sqrt{1-\rho^2} W^\star(t_m)\right) \mathrm{d}W_1(s) \\
&= \rho \int_{t_m}^{t_{m+1}} \left(W_1(s) - W_1(t_m)\right) \mathrm{d}W_1(s) + \\
&\quad \sqrt{1-\rho^2} \int_{t_m}^{t_{m+1}} \left(W^\star(s) - W^\star(t_m)\right) \mathrm{d}W_1(s)
\end{aligned}$$

and:

$$\mathcal{I}_{(2,1)} = \frac{1}{2}\rho\left(\left(\Delta W_{1,m}\right)^2 - \left(t_{m+1} - t_m\right)\right) + B_m$$

Euler-Maruyama and Milstein schemes

We finally deduce that the multidimensional Milstein scheme of the Heston model is:

$$\ln X_{m+1} = \ln X_m + \left(\mu - \frac{1}{2} v_m\right) h + \sqrt{v_m h} \cdot \varepsilon_{1,m} + \frac{1}{4} \rho \sigma h \left(\varepsilon_{1,m}^2 - 1\right) + B_m$$

and:

$$v_{m+1} = v_m + a \left(b - v_m\right) h + \sigma \sqrt{v_m h} \cdot \varepsilon_{2,m} + \frac{1}{4} \sigma^2 h \left(\varepsilon_{2,m}^2 - 1\right)$$

where B_m is a correction term defined by:

$$B_m = \sqrt{1 - \rho^2} \int_{t_m}^{t_{m+1}} \left(W^\star \left(s\right) - W^\star \left(t_m\right)\right) \mathrm{d}W_1 \left(s\right)$$

A basic example

- Suppose we have a circle with radius r and a $2r \times 2r$ square of the same center. Since the area of the circle is equal to πr^2, the numerical calculation of π is equivalent to compute the area of the circle with $r = 1$
- In this case, the area of the square is 4, and we have:

$$\pi = 4 \frac{\mathcal{A}\left(\text{circle}\right)}{\mathcal{A}\left(\text{square}\right)}$$

- To determine π, we simulate n_S random vectors (u_s, v_s) of uniform random variables $\mathcal{U}_{[-1,1]}$ and we obtain:

$$\pi = \lim_{n_S \rightarrow \infty} 4 \frac{n_c}{n}$$

where n_c is the number of points (u_s, v_s) in the circle:

$$n_c = \sum_{s=1}^{n_S} \mathbb{1}\left\{u_s^2 + v_s^2 \leq r^2\right\}$$

A basic example

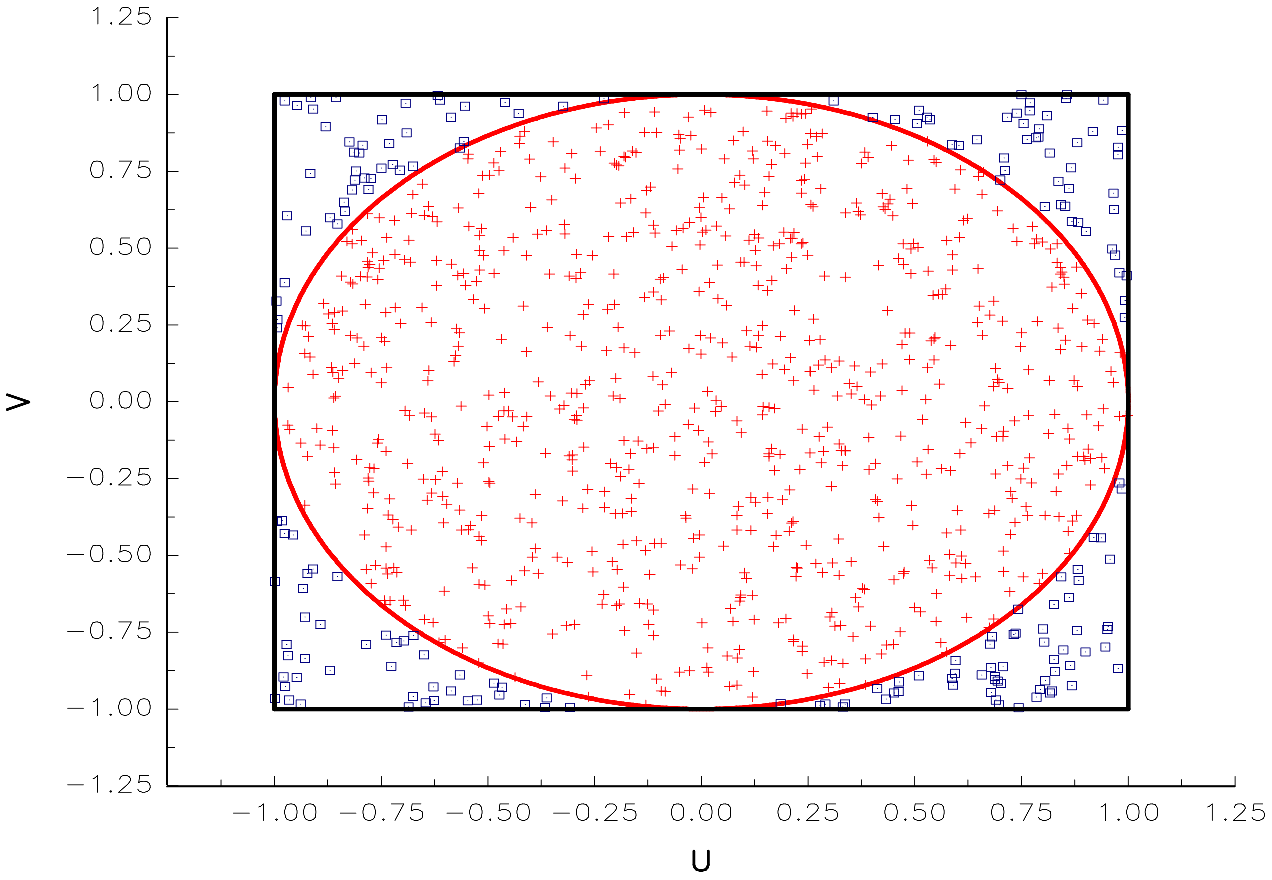


Figure: Computing π with 1 000 simulations

Theoretical framework

- We consider the multiple integral:

$$I = \int \cdots \int_{\Omega} \varphi \left(x_1, \ldots, x_n \right) \,\mathrm{d}x_1 \cdots \mathrm{d}x_n$$

- Let $X = (X_1, \ldots, X_n)$ be a uniform random vector with probability distribution $\mathcal{U}_{[\Omega]}$, such that Ω is inscribed within the hypercube $[\Omega]$
- The pdf is:

$$f \left(x_1, \ldots, x_n \right) = 1$$

- We deduce that:

$$\begin{aligned} I &= \int \cdots \int_{[\Omega]} \mathbb{1} \left\{ \left(x_1, \ldots, x_n \right) \in \Omega \right\} \cdot \varphi \left(x_1, \ldots, x_n \right) \,\mathrm{d}x_1 \cdots \mathrm{d}x_n \\ &= \mathbb{E} \left[\mathbb{1} \left\{ \left(X_1, \ldots, X_n \right) \in \Omega \right\} \cdot \varphi \left(X_1, \ldots, X_n \right) \right] \\ &= \mathbb{E} \left[h \left(X_1, \ldots, X_n \right) \right] \end{aligned}$$

where:

$$h \left(x_1, \ldots, x_n \right) = \mathbb{1} \left\{ \left(x_1, \ldots, x_n \right) \in \Omega \right\} \cdot \varphi \left(x_1, \ldots, x_n \right)$$

Theoretical framework

- Let $\hat{I}_{n_S}$ be the random variable defined by:

$$\hat{I}_{n_S} = \frac{1}{n_S} \sum_{s=1}^{n_S} h\left(X_{1,s}, \ldots, X_{n,s}\right)$$

where $\left\{X_{1,s}, \ldots, X_{n,s}\right\}_{s \geq 1}$ is a sequence of *iid* random vectors with probability distribution $\mathcal{U}_{[\Omega]}$

- Using the strong law of large numbers, we obtain:

$$\begin{aligned} \lim_{n_s \to \infty} \hat{I}_{n_s} &= \mathbb{E}\left[h\left(X_1, \ldots, X_n\right)\right] \\ &= \int \cdots \int_{\Omega} \varphi\left(x_1, \ldots, x_n\right) \mathrm{d}x_1 \cdots \mathrm{d}x_n \end{aligned}$$

- Moreover, the central limit theorem states that:

$$\lim_{n_s \to \infty} \sqrt{n_S} \left(\frac{\hat{I}_{n_s} - I}{\sigma\left(h\left(X_1, \ldots, X_n\right)\right)} \right) = \mathcal{N}\left(0, 1\right)$$

Theoretical framework

- When n_S is large, we can deduce the following confidence interval:

$$\left[\hat{I}_{n_S} - c_\alpha \cdot \frac{\hat{S}_{n_S}}{\sqrt{n_S}}, \hat{I}_{n_S} + c_\alpha \cdot \frac{\hat{S}_{n_S}}{\sqrt{n_S}}\right]$$

where α is the confidence level, $c_\alpha = \Phi^{-1}\left(\left(1+\alpha\right)/2\right)$ and $\hat{S}_{n_S}$ is the usual estimate of the standard deviation:

$$\hat{S}_{n_S} = \sqrt{\frac{1}{n_S - 1}\sum_{s=1}^{n_S} h^2\left(X_{1,s}, \ldots, X_{n,s}\right) - \hat{I}_{n_s}}$$

Theoretical framework

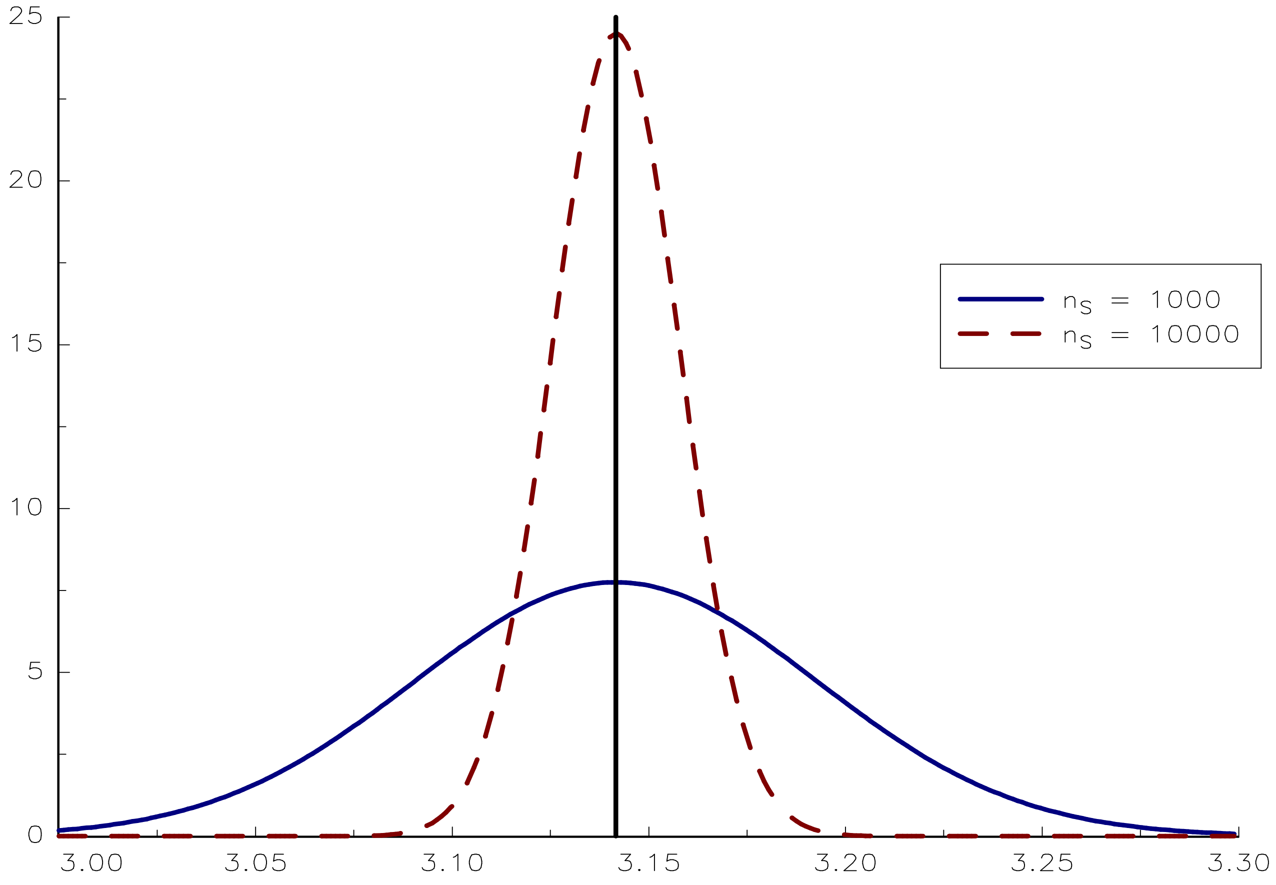


Figure: Density function of $\hat{\pi}_{n_S}$

Extension to the calculation of mathematical expectations

- Let $X = (X_1, \ldots, X_n)$ be a random vector with probability distribution $\mathbf{F}$. We have:

$$\begin{aligned}\mathbb{E}\left[\varphi\left(X_1, \ldots, X_n\right)\right] &= \int \cdots \int \varphi\left(x_1, \ldots, x_n\right) \,\mathrm{d}\mathbf{F}\left(x_1, \cdots, x_n\right) \\ &= \int \cdots \int \varphi\left(x_1, \ldots, x_n\right) f\left(x_1, \cdots, x_n\right) \,\mathrm{d}x_1 \cdots \mathrm{d}x_n \\ &= \int \cdots \int h\left(x_1, \ldots, x_n\right) \,\mathrm{d}x_1 \cdots \mathrm{d}x_n\end{aligned}$$

  where f is the density function

- The Monte Carlo estimator of this integral is:

$$\hat{I}_{n_S} = \frac{1}{n_S} \sum_{s=1}^{n_S} \varphi\left(X_{1,s}, \ldots, X_{n,s}\right)$$

  where $\left\{X_{1,s}, \ldots, X_{n,s}\right\}_{s \geq 1}$ is a sequence of *iid* random vectors with probability distribution $\mathbf{F}$

Extension to the calculation of mathematical expectations

- The price of the look-back option with maturity T is given by:

$$\mathcal{C} = e^{-rT}\mathbb{E}\left[\left(S(T) - \min_{0\leq t\leq T} S(t)\right)^{+}\right]$$

- The price $S(t)$ of the underlying asset is given by the following SDE:

$$\mathrm{d}S(t) = rS(t)\,\mathrm{d}t + \sigma S(t)\,\mathrm{d}W(t)$$

  where r is the interest rate and σ is the volatility of the asset
- For a given simulation s, we have:

$$S_{m+1}^{(s)} = S_m^{(s)} \cdot \exp\left(\left(r - \frac{1}{2}\sigma^2\right)(t_{m+1} - t_m) + \sigma\sqrt{t_{m+1} - t_m} \cdot \varepsilon_m^{(s)}\right)$$

  where $\varepsilon_m^{(s)} \sim \mathcal{N}(0,1)$ and $T = t_M$
- The Monte Carlo estimator of the option price is then equal to:

$$\widehat{\mathcal{C}} = \frac{e^{-rT}}{n_S}\sum_{s=1}^{n_S}\left(S_M^{(s)} - \min_m S_m^{(s)}\right)^{+}$$

Extension to the calculation of mathematical expectations

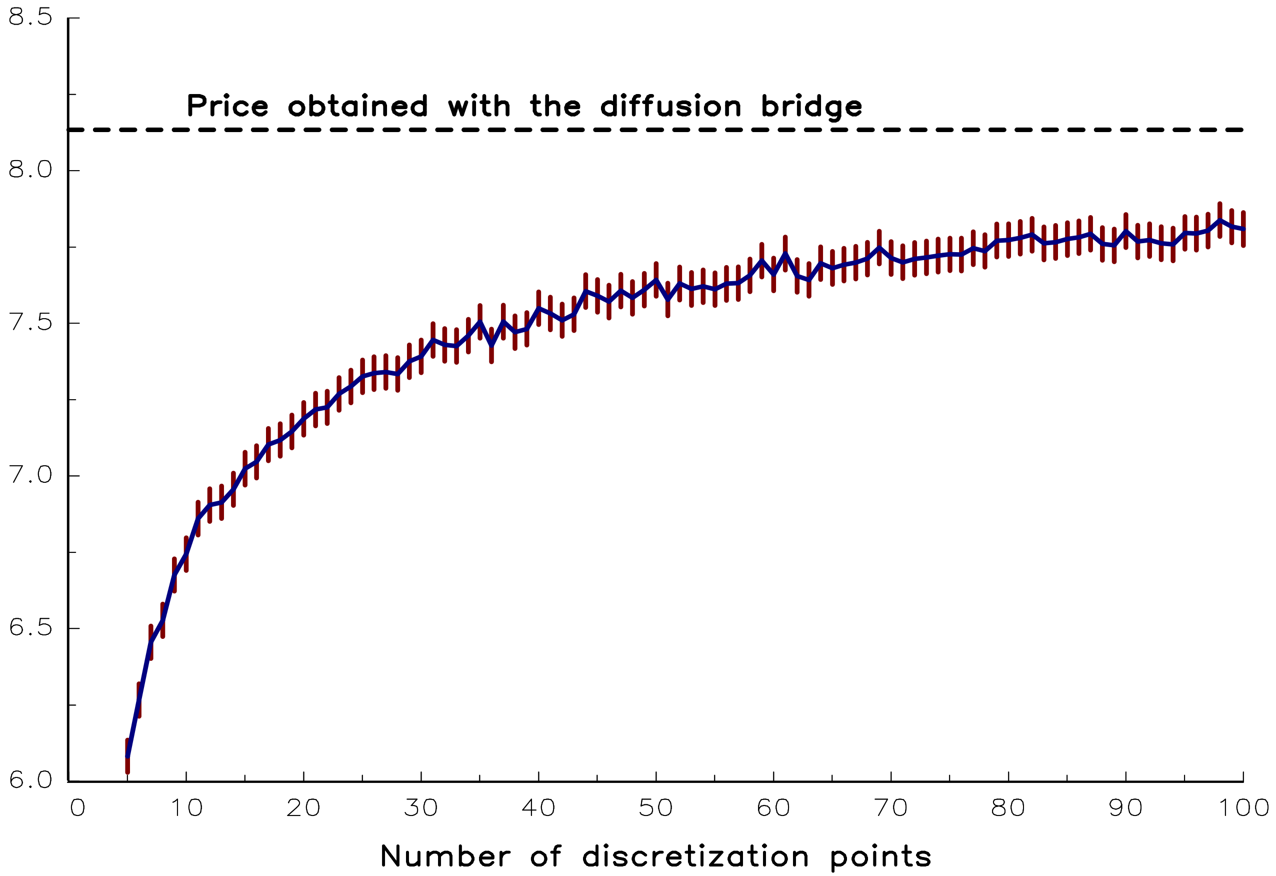


Figure: Computing the look-back option price

Extension to the calculation of mathematical expectations

- Let us consider the following integral:

$$I = \int \cdots \int h(x_1, \ldots, x_n) \, \mathrm{d}x_1 \cdots \mathrm{d}x_n$$

- We can write it as follows:

$$I = \int \cdots \int \frac{h(x_1, \ldots, x_n)}{f(x_1, \cdots, x_n)} f(x_1, \cdots, x_n) \, \mathrm{d}x_1 \cdots \mathrm{d}x_n$$

  where $f(x_1, \cdots, x_n)$ is a multidimensional density function

- We deduce that:

$$I = \mathbb{E}\left[\frac{h(X_1, \ldots, X_n)}{f(X_1, \ldots, X_n)}\right]$$

- This implies that we can compute an integral with the MC method by using any multidimensional distribution function

Extension to the calculation of mathematical expectations

If we apply this result to the calculation of π, we have:

$$\begin{aligned} \pi &= \iint_{x^2+y^2\leq 1} \mathrm{d}x\,\mathrm{d}y = \iint \mathbb{1}\left\{x^2+y^2\leq 1\right\}\,\mathrm{d}x\,\mathrm{d}y \\ &= \iint \frac{\mathbb{1}\left\{x^2+y^2\leq 1\right\}}{\phi(x)\,\phi(y)}\phi(x)\,\phi(y)\,\mathrm{d}x\,\mathrm{d}y \end{aligned}$$

We deduce that:

$$\pi = \mathbb{E}\left[\frac{\mathbb{1}\left\{X^2+Y^2\leq 1\right\}}{\phi(X)\,\phi(Y)}\right]$$

where X and Y are two independent standard Gaussian random variables. We can then estimate π by:

$$\hat{\pi}_{n_S} = \frac{1}{n_S}\sum_{s=1}^{n_S}\frac{\mathbb{1}\left\{x_s^2+y_s^2\leq 1\right\}}{\phi(x_s)\,\phi(y_s)}$$

where x_s and y_s are two independent random variates from the probability distribution $\mathcal{N}(0,1)$

Extension to the calculation of mathematical expectations

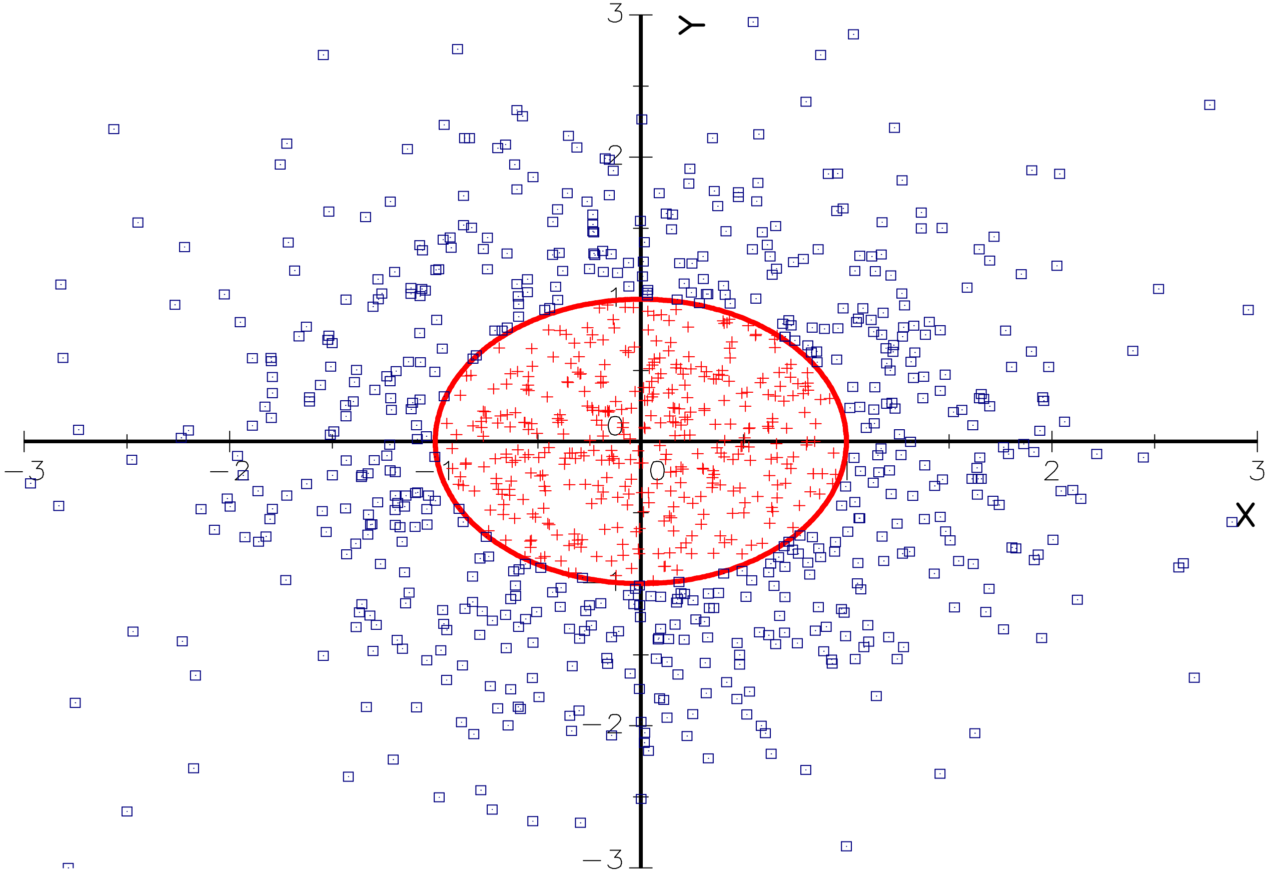


Figure: Computing *pi* with normal random numbers

Variance reduction

- We consider two unbiased estimators $\hat{I}_{nS}^{(1)}$ and $\hat{I}_{nS}^{(2)}$ of the integral I, meaning that $\mathbb{E}\left[\hat{I}_{nS}^{(1)}\right] = \mathbb{E}\left[\hat{I}_{nS}^{(2)}\right] = I$
- We say that $\hat{I}_{nS}^{(1)}$ is more efficient than $\hat{I}_{nS}^{(2)}$ if the inequality $\operatorname{var}\left(\hat{I}_{nS}^{(1)}\right) \leq \operatorname{var}\left(\hat{I}_{nS}^{(2)}\right)$ holds for all values of n_S that are larger than $n_S^{\star}$
- Variance reduction is then the search of more efficient estimators

Antithetic variates

- We have:

$$I = \mathbb{E}\left[\varphi\left(X_1, \ldots, X_n\right)\right] = \mathbb{E}\left[Y\right]$$

  where $Y = \varphi\left(X_1, \ldots, X_n\right)$ is a one-dimensional random variable
- It follows that:

$$\hat{I}_{n_S} = \bar{Y}_{n_S} = \frac{1}{n_S}\sum_{s=1}^{n_S} Y_s$$

- We now consider the estimators $\bar{Y}_{n_S}$ and $\bar{Y}'_{n_S}$ based on two different samples and define $\bar{Y}^{\star}$ as follows:

$$\bar{Y}^{\star} = \frac{\bar{Y}_{n_S} + \bar{Y}'_{n_S}}{2}$$

Antithetic variates

- We have:

$$\mathbb{E}\left[\bar{Y}^{\star}\right] = \mathbb{E}\left[\frac{\bar{Y}_{ns} + \bar{Y}'_{ns}}{2}\right] = \mathbb{E}\left[\bar{Y}_{ns}\right] = I$$

and:

$$\begin{aligned} \operatorname{var}\left(\bar{Y}^{\star}\right) &= \operatorname{var}\left(\frac{\bar{Y}_{ns} + \bar{Y}'_{ns}}{2}\right) \\ &= \frac{1}{4}\operatorname{var}\left(\bar{Y}_{ns}\right) + \frac{1}{4}\operatorname{var}\left(\bar{Y}'_{ns}\right) + \frac{1}{2}\operatorname{cov}\left(\bar{Y}_{ns}, \bar{Y}'_{ns}\right) \\ &= \frac{1 + \rho\left\langle \bar{Y}_{ns}, \bar{Y}'_{ns}\right\rangle}{2}\operatorname{var}\left(\bar{Y}_{ns}\right) \\ &= \frac{1 + \rho\left\langle Y_s, Y'_s\right\rangle}{2}\operatorname{var}\left(\bar{Y}_{ns}\right) \end{aligned}$$

where $\rho\left\langle Y_s, Y'_s\right\rangle$ is the correlation between Y_s and Y'_s

Antithetic variates

- Because we have $\rho \left\langle Y_s, Y_s' \right\rangle \leq 1$, we deduce that:

$$\operatorname{var}\left(\bar{Y}^{\star}\right) \leq \operatorname{var}\left(\bar{Y}_{ns}\right)$$

- If we simulate the random variates Y_s and Y_s' independently, $\rho \left\langle Y_s, Y_s' \right\rangle$ is equal to zero and the variance of the estimator is divided by 2
- However, the number of simulations have been multiplied by two. The efficiency of the estimator has then not been improved

Antithetic variates

- The underlying idea of antithetic variables is therefore to use two perfectly dependent random variables Y_s and Y_s':

$$Y_s' = \psi(Y_s)$$

  where ψ is a deterministic function

- This implies that:

$$\bar{Y}_{n_S}^{\star} = \frac{1}{n_S}\sum_{s=1}^{n_S} Y_s^{\star}$$

  where:

$$Y_s^{\star} = \frac{Y_s + Y_s'}{2} = \frac{Y_s + \psi(Y_s)}{2}$$

- It follows that:

$$\rho\left\langle \bar{Y}_{n_S}, \bar{Y}'_{n_S} \right\rangle = \rho\left\langle Y, Y' \right\rangle = \rho\left\langle Y, \psi(Y) \right\rangle$$

Antithetic variates

- Minimizing the variance $\mathrm{var}\left(\bar{Y}^{\star}\right)$ is then equivalent to minimize the correlation $\rho\left\langle Y,\psi\left(Y\right)\right\rangle$
- We also know that the correlation reaches its lower bound if the dependence function between Y and $\psi\left(Y\right)$ is equal to the lower Frchet copula:

$$\mathbf{C}\left\langle Y,\psi\left(Y\right)\right\rangle=\mathbf{C}^{-}$$

- However, $\rho\left\langle Y,\psi\left(Y\right)\right\rangle$ is not necessarily equal to -1 except in some special cases

Antithetic variates

- We consider the one-dimensional case with $Y = \varphi(X)$
- If we assume that φ is an increasing function, it follows that:

$$\mathbf{C}\langle Y, \psi(Y)\rangle = \mathbf{C}\langle \varphi(X), \psi(\varphi(X))\rangle = \mathbf{C}\langle X, \psi(X)\rangle$$

- To obtain the lower bound $\mathbf{C}^{-}$, X and $\psi(X)$ must be countermonotonic:

$$\psi(X) = \mathbf{F}^{-1}(1 - \mathbf{F}(X))$$

  where $\mathbf{F}$ is the probability distribution of X

- For instance, if $X \sim \mathcal{U}_{[0,1]}$, we have $X' = 1 - X$. In the case where $X \sim \mathcal{N}(0,1)$, we have:

$$X' = \Phi^{-1}(1 - \Phi(X)) = \Phi^{-1}(\Phi(-X)) = -X$$

Antithetic variates

Example #9

We consider the following functions:

1. $\varphi_1(x) = x^3 + x + 1$
2. $\varphi_2(x) = x^4 + x^2 + 1$
3. $\varphi_3(x) = x^4 + x^3 + x^2 + x + 1$

Antithetic variates

For each function, we want to estimate $I = \mathbb{E}\left[\varphi\left(\mathcal{N}\left(0,1\right)\right)\right]$ using the antithetic estimator:

$$\bar{Y}^{\star}_{n_S} = \frac{1}{n_S}\sum_{s=1}^{n_S}\frac{\varphi\left(X_s\right)+\varphi\left(-X_s\right)}{2}$$

where $X_s \sim \mathcal{N}\left(0,1\right)$

- Let $X \sim \mathcal{N}\left(0,1\right)$. We have $\mathbb{E}\left[X^2\right] = 1$, $\mathbb{E}\left[X^{2m}\right] = \left(2m-1\right)\mathbb{E}\left[X^{2m-2}\right]$ and $\mathbb{E}\left[X^{2m+1}\right] = 0$ for $m \in \mathbb{N}$
- We obtain the following results:

$\varphi\left(x\right)$	$\varphi_1\left(x\right)$	$\varphi_2\left(x\right)$	$\varphi_3\left(x\right)$
$\mathbb{E}\left[\varphi\left(X_s\right)\right]$ or $\mathbb{E}\left[\varphi\left(-X_s\right)\right]$	1	5	5
$\operatorname{var}\left(\varphi\left(X_s\right)\right)$ or $\operatorname{var}\left(\varphi\left(-X_s\right)\right)$	22	122	144
$\operatorname{cov}\left(\varphi\left(X_s\right),\varphi\left(-X_s\right)\right)$	-22	122	100
$\rho\left\langle\varphi\left(X_s\right),\varphi\left(-X_s\right)\right\rangle$	-1	1	$^{25}/_{36}$

Antithetic variates

To understand these numerical results, we must study the relationship between $\mathbf{C}\left\langle X, X'\right\rangle$ and $\mathbf{C}\left\langle Y, Y'\right\rangle$. Indeed, we have:

$$\left\{\mathbf{C}\left\langle X, X'\right\rangle = \mathbf{C}^{-} \Rightarrow \mathbf{C}\left\langle Y, Y'\right\rangle = \mathbf{C}^{-}\right\} \Leftrightarrow \varphi'\left(x\right) \geq 0$$

Antithetic variates

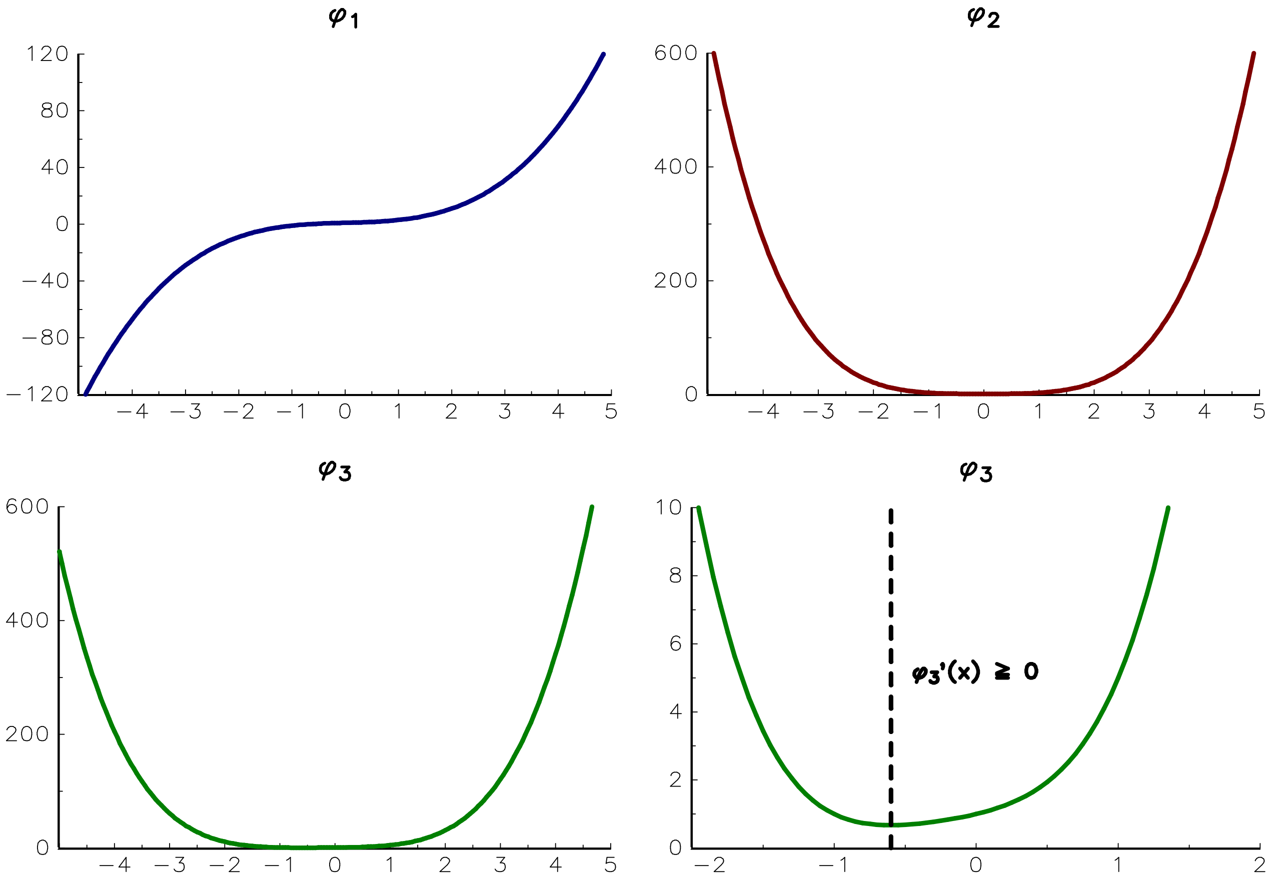


Figure: Functions $\varphi_1(x)$, $\varphi_2(x)$ and $\varphi_3(x)$

Application to the geometric Brownian motion

- In the Gaussian case $X \sim \mathcal{N}(0,1)$, the antithetic variable is:

$$X' = -X$$

- As the simulation of $Y \sim \mathcal{N}\left(\mu, \sigma^2\right)$ is obtained using the relationship $Y = \mu + \sigma X$, we deduce that the antithetic variable is:

$$Y' = \mu - \sigma X = \mu - \sigma \frac{(Y-\mu)}{\sigma} = 2\mu - Y$$

- If we consider the geometric Brownian motion, the fixed-interval scheme is:

$$X_{m+1} = X_m \cdot \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right) h + \sigma\sqrt{h} \cdot \varepsilon_m\right)$$

whereas the antithetic path is given by:

$$X'_{m+1} = X'_m \cdot \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right) h - \sigma\sqrt{h} \cdot \varepsilon_m\right)$$

Application to the geometric Brownian motion

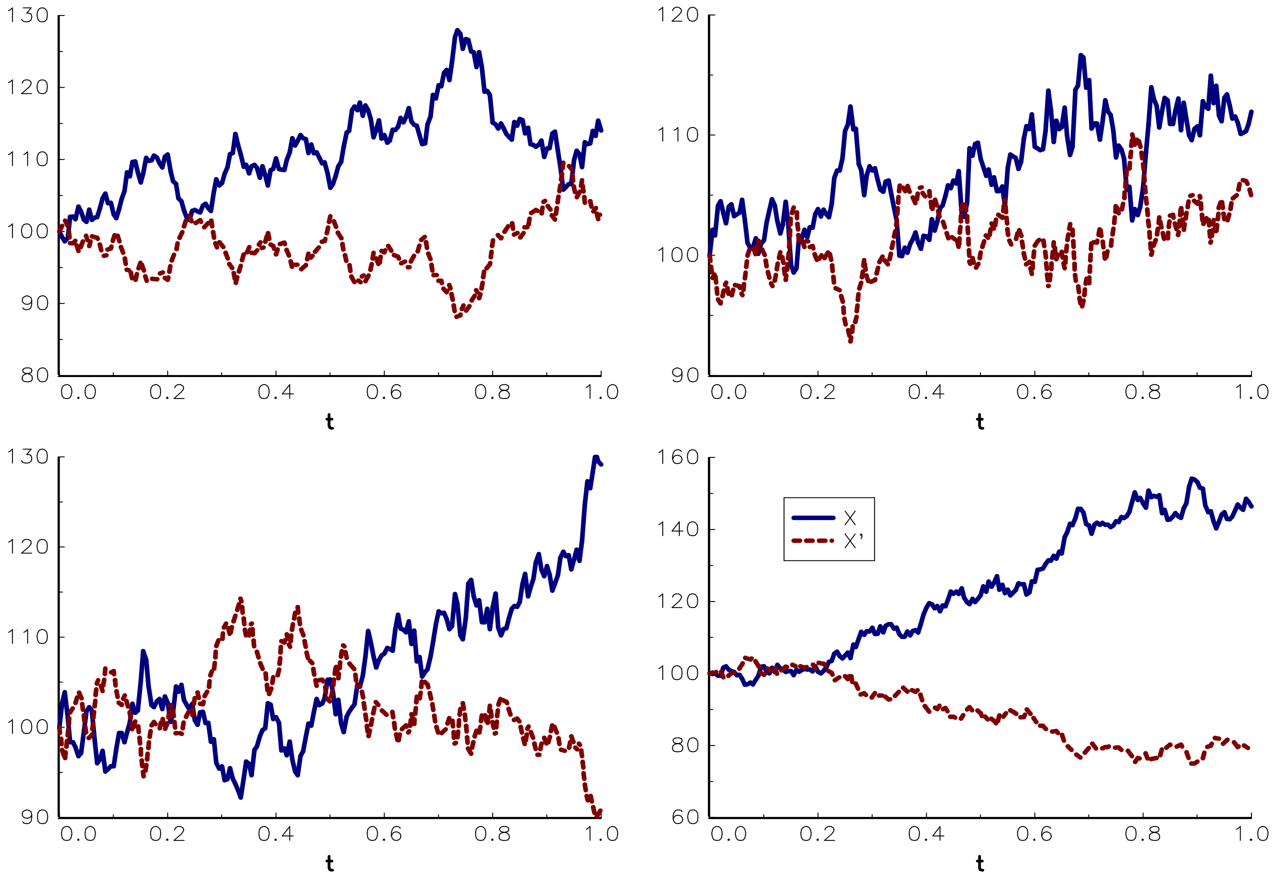


Figure: Antithetic simulation of the GBM process

Application to the geometric Brownian motion

- In the multidimensional case, we recall that:

$$X_{j,m+1} = X_{j,m} \cdot \exp\left(\left(\mu_j - \frac{1}{2}\sigma_j^2\right) h + \sigma_j \sqrt{h} \cdot \varepsilon_{j,m}\right)$$

  where $\varepsilon_m = (\varepsilon_{1,m}, \ldots, \varepsilon_{n,m}) \sim \mathcal{N}_n(\mathbf{0}, \rho)$

- We simulate ε_m by using the relationship $\varepsilon_m = P \cdot \eta_m$ where $\eta_m \sim \mathcal{N}_n(\mathbf{0}, I_n)$ and P is the Cholesky matrix satisfying $PP^\top = \rho$
- The antithetic trajectory is then:

$$X'_{j,m+1} = X'_{j,m} \cdot \exp\left(\left(\mu_j - \frac{1}{2}\sigma_j^2\right) h + \sigma_j \sqrt{k} \cdot \varepsilon'_{j,m}\right)$$

  where:

$$\varepsilon'_m = -P \cdot \eta_m = -\varepsilon_m$$

- We verify that $\varepsilon'_m = (\varepsilon'_{1,m}, \ldots, \varepsilon'_{n,m}) \sim \mathcal{N}_n(\mathbf{0}, \rho)$

Application to the geometric Brownian motion

In the Black-Scholes model, the price of the spread option with maturity T and strike K is given by:

$$\mathcal{C} = e^{-rT}\mathbb{E}\left[\left(S_1\left(T\right) - S_2\left(T\right) - K\right)^+\right]$$

where the prices $S_1\left(t\right)$ and $S_2\left(t\right)$ of the underlying assets are given by the following SDE:

$$\left\{\begin{array}{l} \mathrm{d}S_1\left(t\right) = rS_1\left(t\right)\mathrm{d}t + \sigma_1 S_1\left(t\right)\mathrm{d}W_1(t) \\ \mathrm{d}S_2\left(t\right) = rS_2\left(t\right)\mathrm{d}t + \sigma_2 S_2\left(t\right)\mathrm{d}W_2(t) \end{array}\right.$$

and $\mathbb{E}\left[W_1\left(t\right)W_2\left(t\right)\right] = \rho\, t$

Application to the geometric Brownian motion

- To calculate the option price using Monte Carlo methods, we simulate the bivariate GBM $S_1(t)$ and $S_2(t)$ and the MC estimator is:

$$\hat{\mathcal{C}}_{\mathrm{MC}} = \frac{e^{-rT}}{n_S} \sum_{s=1}^{n_S} \left(S_1^{(s)}(T) - S_2^{(s)}(T) - K \right)^+$$

  where $S_j^{(s)}(T)$ is the s^{th} simulation of the terminal value $S_j(T)$

- For the AV estimator, we obtain:

$$\hat{\mathcal{C}}_{\mathrm{AV}} = \frac{e^{-rT}}{n_S} \sum_{s=1}^{n_S} \frac{\left(S_1^{(s)}(T) - S_2^{(s)}(T) - K \right)^+ + \left(S_1'^{(s)}(T) - S_2'^{(s)}(T) - K \right)^+}{2}$$

  where $S_j'^{(s)}(T)$ is the antithetic variate of $S_j^{(s)}(T)$

Application to the geometric Brownian motion

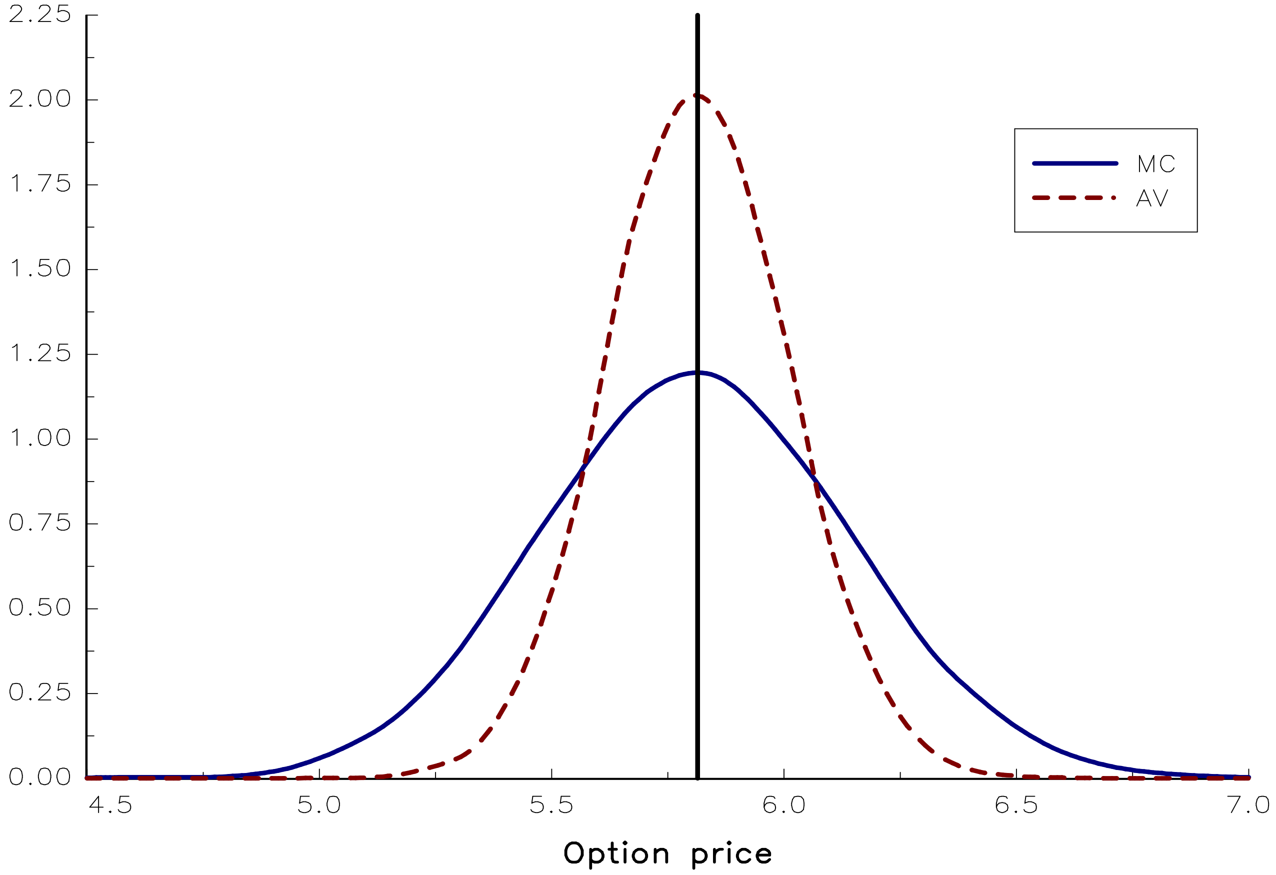


Figure: Probability density function of $\widehat{\mathcal{C}}_{\mathrm{MC}}$ and $\widehat{\mathcal{C}}_{\mathrm{AV}}$ ($n_S = 1\,000$)

Control variates

- Let $Y = \varphi(X_1, \ldots, X_n)$ and V be a random variable with known mean $\mathbb{E}[V]$
- We define Z as follows: $Z = Y + c \cdot (V - \mathbb{E}[V])$
- We deduce that:

$$\begin{array}{rcl} \mathbb{E}[Z] & = & \mathbb{E}[Y + c \cdot (V - \mathbb{E}[V])] \\ & = & \mathbb{E}[Y] + c \cdot \mathbb{E}[V - \mathbb{E}[V]] \\ & = & \mathbb{E}[\varphi(X_1, \ldots, X_n)] \end{array}$$

and:

$$\begin{array}{rcl} \mathrm{var}(Z) & = & \mathrm{var}(Y + c \cdot (V - \mathbb{E}[V])) \\ & = & \mathrm{var}(Y) + 2 \cdot c \cdot \mathrm{cov}(Y, V) + c^2 \cdot \mathrm{var}(V) \end{array}$$

Control variates

- It follows that:

$$\begin{aligned}\operatorname{var}(Z) \leq \operatorname{var}(Y) &\Leftrightarrow 2 \cdot c \cdot \operatorname{cov}(Y, V) + c^2 \cdot \operatorname{var}(V) \leq 0 \\ &\Rightarrow c \cdot \operatorname{cov}(Y, V) \leq 0\end{aligned}$$

- In order to obtain a lower variance, a necessary condition is that c and $\operatorname{cov}(Y, V)$ have opposite signs
- The minimum is obtained when $\partial_c \operatorname{var}(Z) = 0$ or equivalently when:

$$c^\star = -\frac{\operatorname{cov}(Y, V)}{\operatorname{var}(V)} = -\beta$$

Control variates

- The optimal value $c^{\star}$ is then equal to the opposite of the beta of Y with respect to the control variate V. In this case, we have:

$$Z = Y - \frac{\operatorname{cov}(Y, V)}{\operatorname{var}(V)} \cdot (V - \mathbb{E}[V])$$

and:

$$\operatorname{var}(Z) = \operatorname{var}(Y) - \frac{\operatorname{cov}^2(Y, V)}{\operatorname{var}(V)} = \left(1 - \rho^2 \langle Y, V \rangle\right) \cdot \operatorname{var}(Y)$$

- This implies that we have to choose a control variate V that is highly (positively or negatively) correlated with Y in order to reduce the variance

Control variates

Example

We consider that $X \sim \mathcal{U}_{[0,1]}$ and $\varphi(x) = e^x$. We would like to estimate:

$$I = \mathbb{E}[\varphi(X)] = \int_0^1 e^x \,\mathrm{d}x$$

Control variates

- We set $Y = e^X$ and $V = X$
- We know that $\mathbb{E}[V] = 1/2$ and $\operatorname{var}(V) = 1/12$
- It follows that:

$$\begin{aligned}
\operatorname{var}(Y) &= \mathbb{E}\left[Y^2\right] - \mathbb{E}^2[Y] \\
&= \int_0^1 e^{2x}\,\mathrm{d}x - \left(\int_0^1 e^x\,\mathrm{d}x\right)^2 \\
&= \left[\frac{e^{2x}}{2}\right]_0^1 - \left(e^1 - e^0\right)^2 \\
&= \frac{4e - e^2 - 3}{2} \\
&\approx 0.2420
\end{aligned}$$

Control variates

- We have:

$$\begin{aligned}
\operatorname{cov}(Y, V) &= \mathbb{E}[VY] - \mathbb{E}[V]\,\mathbb{E}[Y] \\
&= \int_0^1 x e^x \,\mathrm{d}x - \frac{1}{2}\left(e^1 - e^0\right) \\
&= \left[x e^x\right]_0^1 - \int_0^1 e^x \,\mathrm{d}x - \frac{1}{2}\left(e^1 - e^0\right) \\
&= \frac{3 - e}{2} \\
&\approx 0.1409
\end{aligned}$$

- If we consider the VC estimator Z defined by:

$$\begin{aligned}
Z &= Y - \frac{\operatorname{cov}(Y, V)}{\operatorname{var}(V)} \cdot (V - \mathbb{E}[V]) \\
&= Y - (18 - 6e) \cdot \left(V - \frac{1}{2}\right)
\end{aligned}$$

Control variates

- We have $\beta \approx 1.6903$
- We obtain:

$$\begin{aligned} \operatorname{var}(Z) &= \operatorname{var}(Y) - \frac{\operatorname{cov}^2(Y, V)}{\operatorname{var}(V)} \\ &= \frac{4e - e^2 - 3}{2} - 3 \cdot (3 - e)^2 \\ &\approx 0.0039 \end{aligned}$$

- We conclude that we have dramatically reduced the variance of the estimator, because we have:

$$\frac{\operatorname{var}\left(\hat{I}_{\mathrm{CV}}\right)}{\operatorname{var}\left(\hat{I}_{\mathrm{MC}}\right)} = \frac{\operatorname{var}(Z)}{\operatorname{var}(Y)} = 1.628\%$$

Control variates

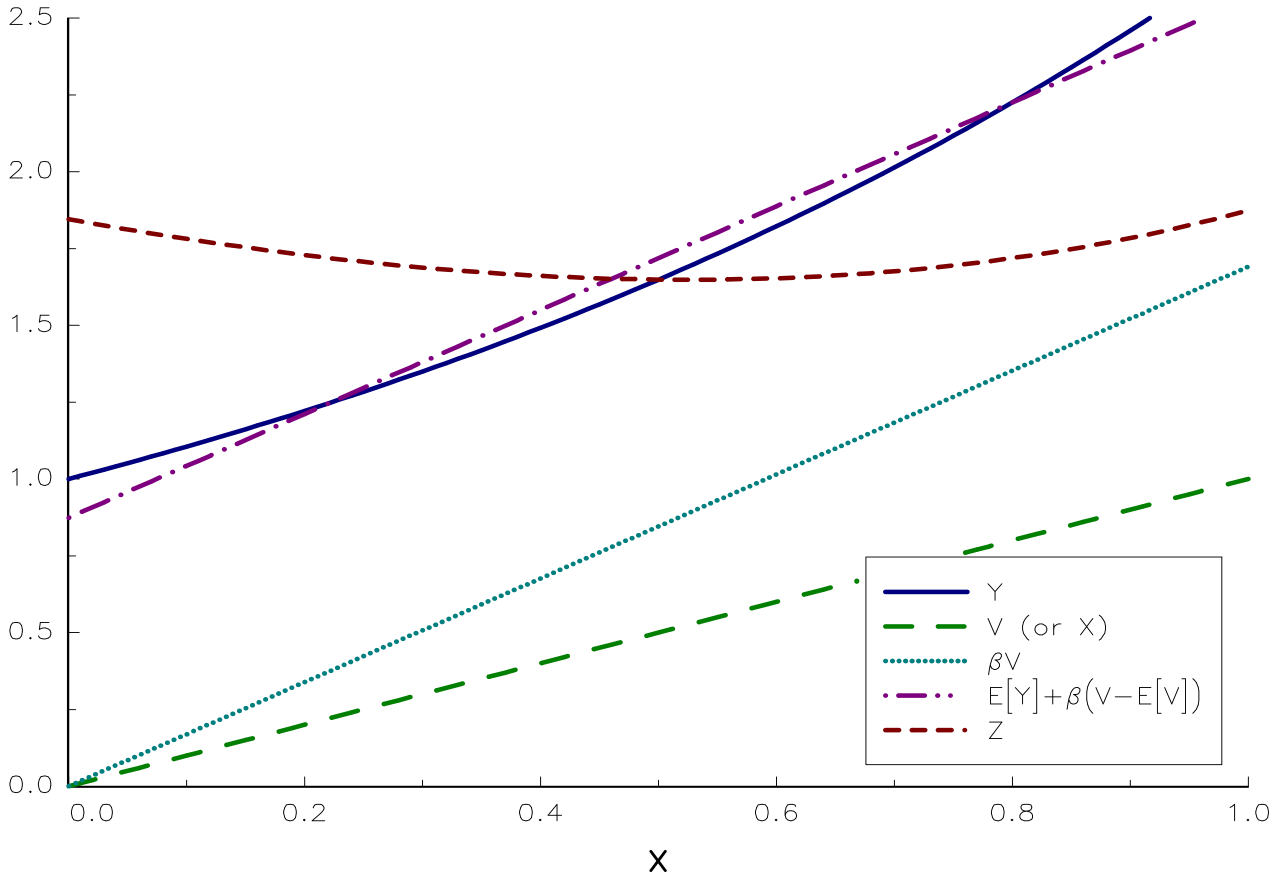


Figure: Understanding the variance reduction in control variates

Control variates

- $\hat{Y}$ is the conditional expectation of Y with respect to V:

$$\mathbb{E}\left[Y \mid V\right] = \mathbb{E}\left[Y\right] + \beta\left(V - \mathbf{E}\left[V\right]\right)$$

- This is the best linear estimator of Y
- The residual U of the linear regression is then equal to:

$$U = Y - \hat{Y} = \left(Y - \mathbb{E}\left[Y\right]\right) - \beta\left(V - \mathbf{E}\left[V\right]\right)$$

- The CV estimator Z is a translation of the residual in order to satisfy $\mathbb{E}\left[Z\right] = \mathbb{E}\left[Y\right]$:

$$Z = \mathbb{E}\left[Y\right] + U = Y - \beta\left(V - \mathbf{E}\left[V\right]\right)$$

- By construction, the variance of the residual U is lower than the variance of the random variable Y. We conclude that:

$$\operatorname{var}\left(Z\right) = \operatorname{var}\left(U\right) \leq \operatorname{var}\left(Y\right)$$

Control variates

We can therefore obtain a large variance reduction if the following conditions are satisfied:

- the control variate V largely explains the random variable Y
- the relationship between Y and V is almost linear

Control variates

The price of an arithmetic Asian call option is given by:

$$\mathcal{C} = e^{-rT}\mathbb{E}\left[\left(\bar{S} - K\right)^{+}\right]$$

where K is the strike of the option and $\bar{S}$ denotes the average of $S(t)$ on a given number of fixing dates[2] $\{t_1, \ldots, t_{n_F}\}$:

$$\bar{S} = \frac{1}{n_F}\sum_{m=1}^{n_F} S(t_m)$$

We can estimate the option price using the Black-Scholes model

[2] We have $t_{n_F} = T$.

Control variates

We can also reduce the variance of the MC estimator by considering the following control variates:

1. the terminal value $V_1 = S(T)$ of the underlying asset;
2. the average value $V_2 = \bar{S}$;
3. the discounted payoff of the call option $V_3 = e^{-rT}(S(T) - K)^+$;
4. the discounted payoff of the geometric Asian call option $V_4 = e^{-rT}\left(\tilde{S} - K\right)^+$ where:

$$\tilde{S} = \left(\prod_{m=1}^{n_F} S(t_m)\right)^{1/n_F}$$

Control variates

For these control variates, we know the expected value

- In the first case, we have:

$$\mathbb{E}\left[S\left(T\right)\right] = S_0 e^{rT}$$

- In the first case, we have:

$$\mathbb{E}\left[\bar{S}\right] = \frac{S_0}{n_F}\sum_{m=1}^{n_F} e^{rt_m}$$

Control variates

- The expected value of the third control variate is the Black-Scholes formula of the European call option:

$$\tilde{S} = \left(\prod_{m=1}^{n_F} S_0 e^{\left(r - \frac{1}{2}\sigma^2\right) t_m + \sigma W(t_m)}\right)^{1/n_F} = S_0 \cdot \exp\left(\left(r - \frac{1}{2}\sigma^2\right)\bar{t} + \sigma \bar{W}\right)$$

where:

$$\bar{t} = \frac{1}{n_F} \sum_{m=1}^{n_F} t_m$$

and:

$$\bar{W} = \frac{1}{n_F} \sum_{m=1}^{n_F} W\left(t_m\right)$$

- Because $\tilde{S}$ has a log-normal distribution, we deduce that the expected value of the fourth control variate is also given by a Black-Scholes formula

Control variates

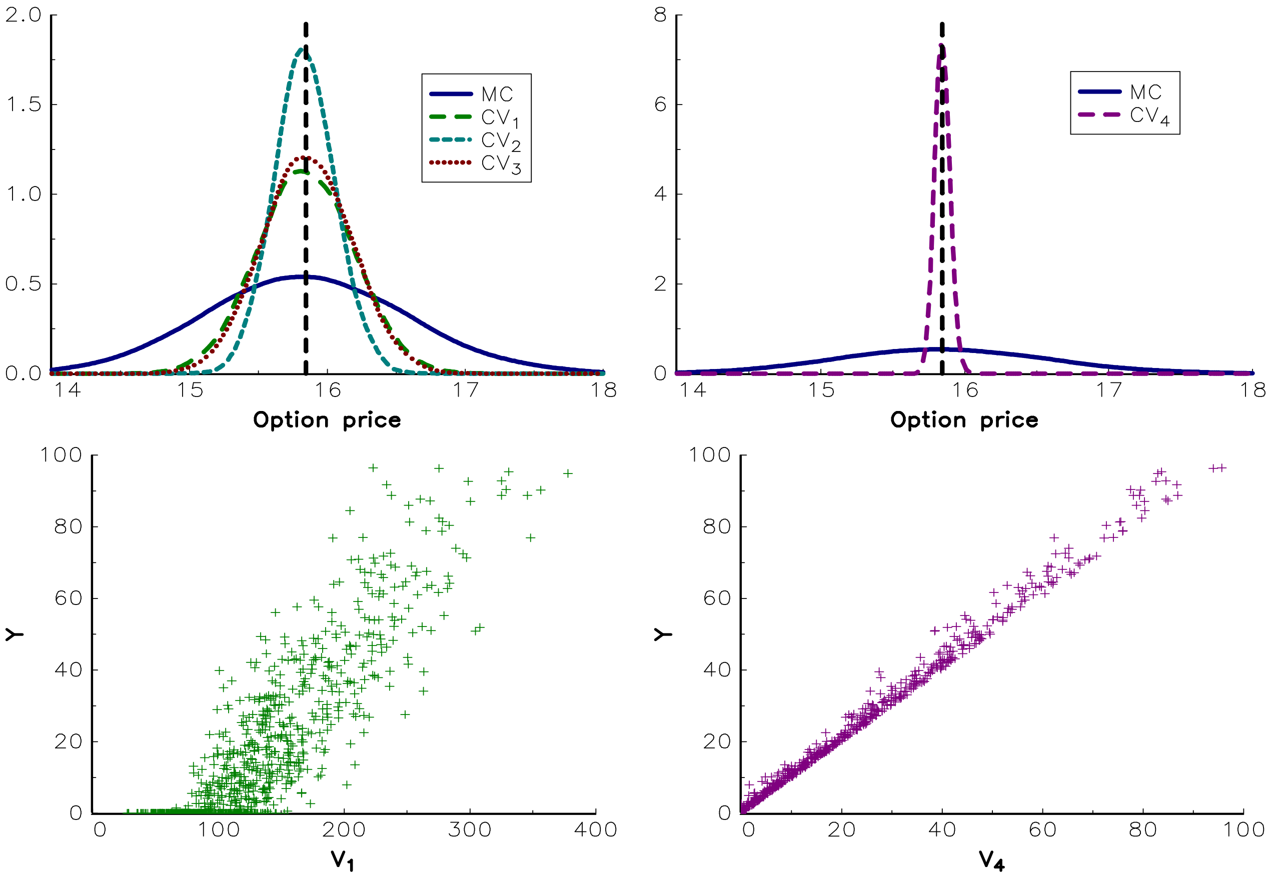


Figure: CV estimator of the arithmetic Asian call option

Control variates

- The previous approach can be extended in the case of several control variates:

$$Z = Y + \sum_{i=1}^{n_{CV}} c_i \cdot (V_i - \mathbb{E}[V_i]) = Y + c^\top (V - \mathbb{E}[V])$$

where $c = (c_1, \ldots, c_{n_{CV}})$ and $V = (V_1, \ldots, V_{n_{CV}})$

- We can show that the optimal value of c is equal to:

$$c^\star = -\operatorname{cov}(V, V)^{-1} \cdot \operatorname{cov}(V, Y)$$

- Minimizing the variance of Z is equivalent to minimize the variance of U:

$$U = Y - \hat{Y} = Y - (\alpha + \beta^\top V)$$

- We deduce that $c^\star = -\beta$. It follows that

$$\operatorname{var}(Z) = \operatorname{var}(U) = \left(1 - R^2\right) \cdot \operatorname{var}(Y)$$

where R^2 is the R-squared coefficient of the linear regression $Y = \alpha + \beta^\top V + U$

Control variates

Table: Linear regression between the Asian call option and the control variates

$\hat{\alpha}$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	R^2	$1 - R^2$
-51.482	0.036	0.538			90.7%	9.3%
-24.025	-0.346	0.595	0.548		96.5%	3.5%
-4.141	0.069		0.410		81.1%	18.9%
-38.727		0.428	0.174		92.9%	7.1%
-1.559	-0.040	0.054	0.111	0.905	99.8%	0.2%

Importance sampling

- Let $X = (X_1, \ldots, X_n)$ be a random vector with distribution function $\mathbf{F}$
- We have:

$$\begin{aligned} I &= \mathbb{E}\left[\varphi\left(X_1, \ldots, X_n\right) \mid \mathbf{F}\right] \\ &= \int \cdots \int \varphi\left(x_1, \ldots, x_n\right) f\left(x_1, \ldots, x_n\right) \mathrm{d}x_1 \cdots \mathrm{d}x_n \end{aligned}$$

where $f\left(x_1, \ldots, x_n\right)$ is the probability density function of X

Importance sampling

- It follows that:

$$\begin{aligned} I &= \int \cdots \int \left(\varphi\left(x_1, \ldots, x_n\right) \frac{f\left(x_1, \ldots, x_n\right)}{g\left(x_1, \ldots, x_n\right)} \right) g\left(x_1, \ldots, x_n\right) \,\mathrm{d}x_1 \cdots \mathrm{d}x_n \\ &= \mathbb{E}\left[\varphi\left(X_1, \ldots, X_n\right) \frac{f\left(X_1, \ldots, X_n\right)}{g\left(X_1, \ldots, X_n\right)} \middle| \mathbf{G} \right] \\ &= \mathbb{E}\left[\varphi\left(X_1, \ldots, X_n\right) \mathcal{L}\left(X_1, \ldots, X_n\right) \mid \mathbf{G} \right] \end{aligned}$$

where $g\left(x_1, \ldots, x_n\right)$ is the probability density function of $\mathbf{G}$ and $\mathcal{L}$ is the likelihood ratio:

$$\mathcal{L}\left(x_1, \ldots, x_n\right) = \frac{f\left(x_1, \ldots, x_n\right)}{g\left(x_1, \ldots, x_n\right)}$$

- The values taken by $\mathcal{L}\left(x_1, \ldots, x_n\right)$ are also called the importance sampling weights

Importance sampling

- Using the vector notation, the relationship becomes:

$$\mathbb{E}\left[\varphi\left(X\right) \mid \mathbf{F}\right] = \mathbb{E}\left[\varphi\left(X\right)\mathcal{L}\left(X\right) \mid \mathbf{G}\right]$$

- It follows that:

$$\mathbb{E}\left[\hat{I}_{\mathrm{MC}}\right] = \mathbb{E}\left[\hat{I}_{\mathrm{IS}}\right] = I$$

  where $\hat{I}_{\mathrm{MC}}$ and $\hat{I}_{\mathrm{IS}}$ are the Monte Carlo and importance sampling estimators of I

- We also deduce that:

$$\operatorname{var}\left(\hat{I}_{\mathrm{IS}}\right) = \operatorname{var}\left(\varphi\left(X\right)\mathcal{L}\left(X\right) \mid \mathbf{G}\right)$$

Importance sampling

- It follows that:

$$\begin{aligned}
\operatorname{var}\left(\hat{I}_{\mathrm{IS}}\right) &= \mathbb{E}\left[\varphi^2\left(X\right)\mathcal{L}^2\left(X\right) \mid \mathbf{G}\right] - \mathbb{E}^2\left[\varphi\left(X\right)\mathcal{L}\left(X\right) \mid \mathbf{G}\right] \\
&= \int \varphi^2\left(x\right)\mathcal{L}^2\left(x\right)g\left(x\right)\,\mathrm{d}x - I^2 \\
&= \int \varphi^2\left(x\right)\frac{f^2\left(x\right)}{g^2\left(x\right)}g\left(x\right)\,\mathrm{d}x - I^2 \\
&= \int \varphi^2\left(x\right)\frac{f^2\left(x\right)}{g\left(x\right)}\,\mathrm{d}x - I^2
\end{aligned}$$

Importance sampling

- If we compare the variance of the two estimators $\hat{I}_{\mathrm{MC}}$ and $\hat{I}_{\mathrm{IS}}$, we obtain:

$$\begin{aligned}\mathrm{var}\left(\hat{I}_{\mathrm{IS}}\right)-\mathrm{var}\left(\hat{I}_{\mathrm{MC}}\right) &= \int\varphi^2(x)\frac{f^2(x)}{g(x)}\,\mathrm{d}x-\int\varphi^2(x)f(x)\,\mathrm{d}x \\ &= \int\varphi^2(x)\left(\frac{f(x)}{g(x)}-1\right)f(x)\,\mathrm{d}x \\ &= \int\varphi^2(x)\left(\mathcal{L}(x)-1\right)f(x)\,\mathrm{d}x\end{aligned}$$

- The difference may be negative if the weights $\mathcal{L}(x)$ are small ($\mathcal{L}(x)\ll 1$) because the values of $\varphi^2(x)f(x)$ are positive
- The importance sampling approach changes then the importance of some values x by transforming the original probability distribution $\mathbf{F}$ into another probability distribution $\mathbf{G}$

Importance sampling

- The first-order condition is:

$$-\varphi^2(x) \cdot \frac{f^2(x)}{g^2(x)} = \lambda$$

where λ is a constant

- We have:

$$\begin{array}{rcl} g^\star(x) & = & \arg\min \operatorname{var}\left(\hat{I}_{\mathrm{IS}}\right) \\ & = & \arg\min \int \varphi^2(x) \frac{f^2(x)}{g(x)} \,\mathrm{d}x \\ & = & c \cdot |\varphi(x)| \cdot f(x) \end{array}$$

where c is the normalizing constant such that $\int g^\star(x) \,\mathrm{d}x = 1$

- A good choice of the IS density $g(x)$ is then an approximation of $|\varphi(x)| \cdot f(x)$ such that $g(x)$ can easily be simulated

Importance sampling

Remark

In order to simplify the notation and avoid confusions, we consider that $X \sim \mathbf{F}$ and $Z \sim \mathbf{G}$ in the sequel. This means that $\hat{I}_{\mathrm{MC}} = \varphi(X)$ and $\hat{I}_{\mathrm{IS}} = \varphi(Z) \mathcal{L}(Z)$

Importance sampling

- We consider the estimation of the probability $p = \Pr\left\{X \geq 3\right\} \approx 0.1350\%$ when $X \sim \mathcal{N}\left(0,1\right)$
- We have:

$$\varphi\left(x\right) = \mathbb{1}\left\{x \geq 3\right\}$$

- Importance sampling with $Z \sim \mathcal{N}\left(\mu_z, \sigma_z^2\right)$, $\mu_z = 3$ and $\sigma_z = 1 \Rightarrow$ the probability $\Pr\left\{Z \geq 3\right\}$ is equal to 50%

Importance sampling

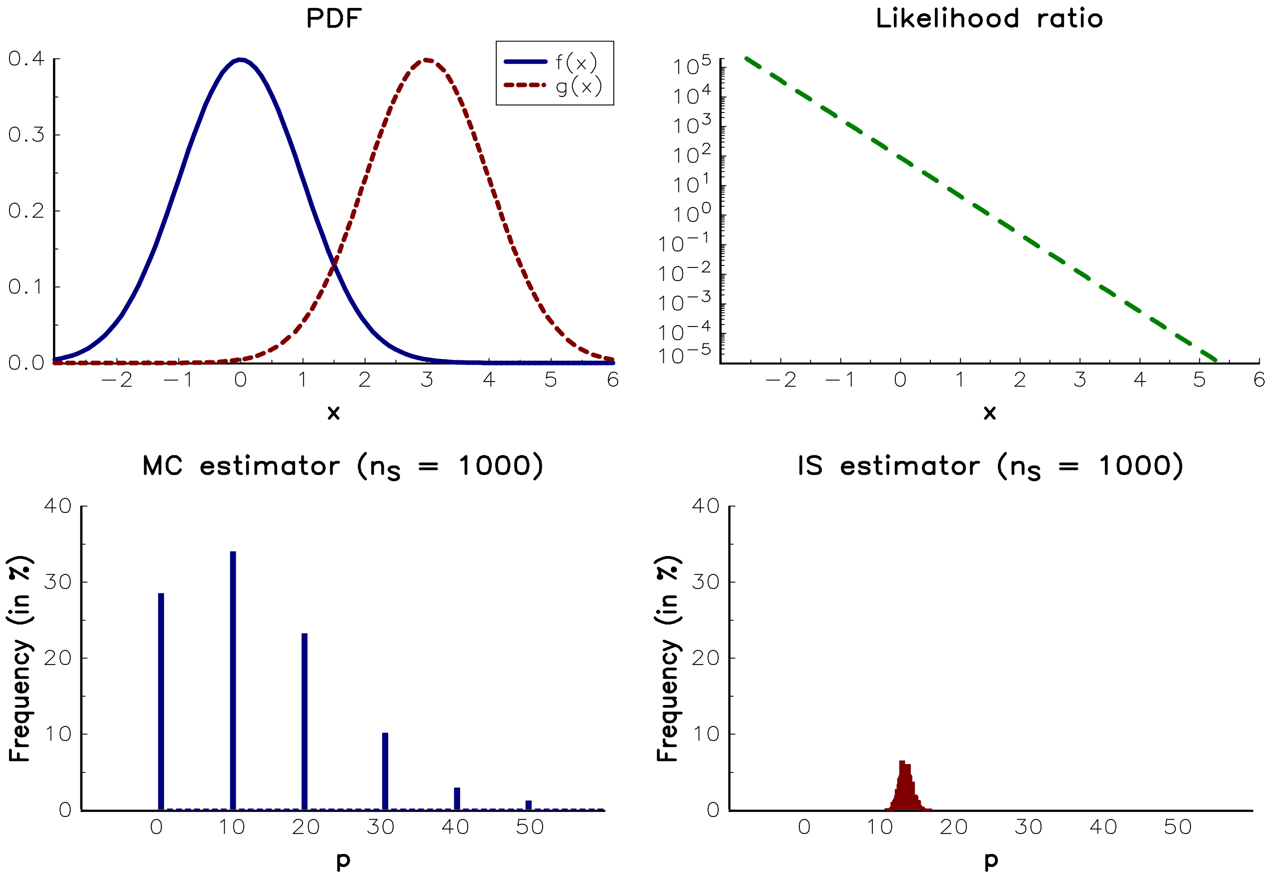


Figure: Histogram of the MC and IS estimators ($n_S = 1\,000$)

Importance sampling

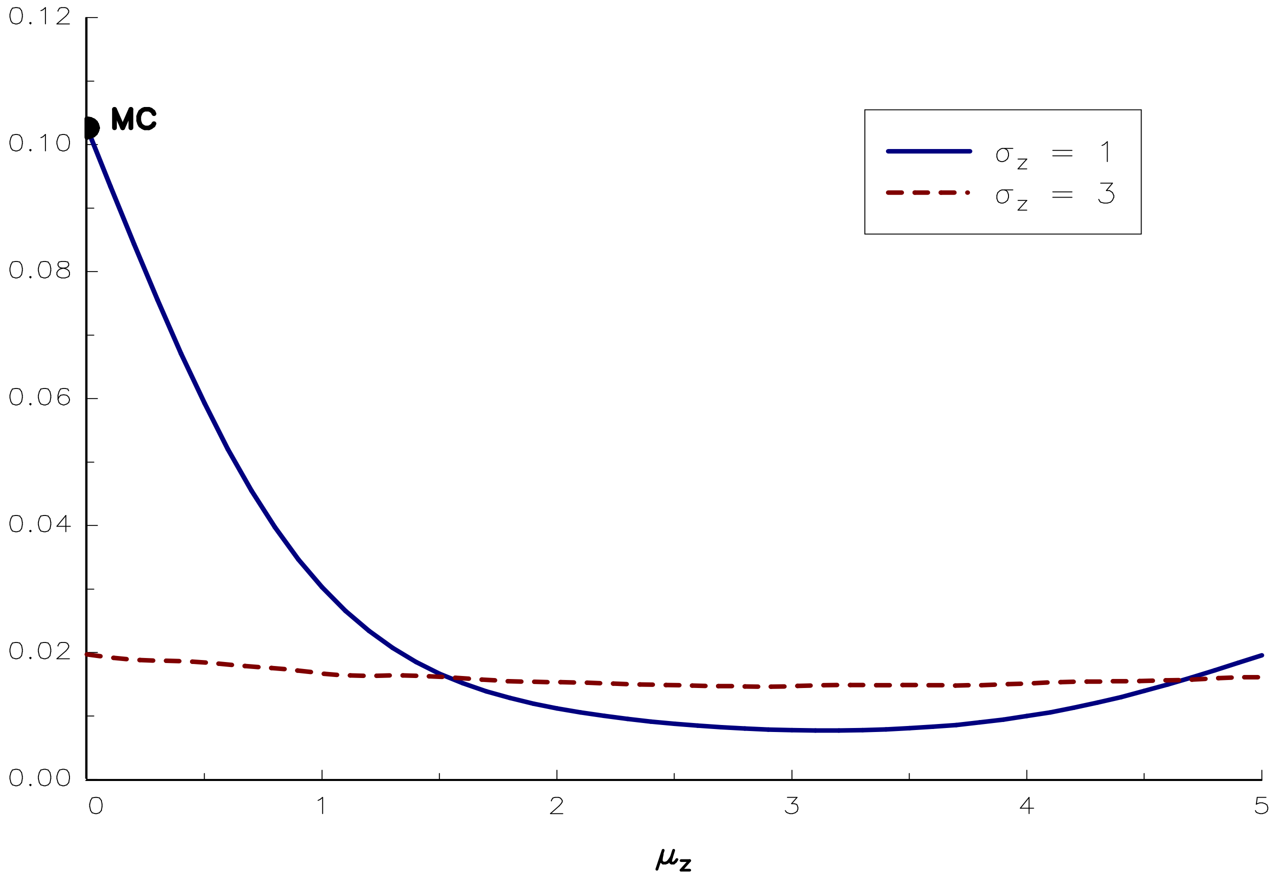


Figure: Standard deviation (in %) of the estimator $\hat{p}_{\mathrm{IS}}$ ($n_S = 1\,000$)

Importance sampling

- We consider the pricing of the put option:

$$\mathcal{P} = e^{-rT}\mathbb{E}\left[(K - S(T))^{+}\right]$$

- We can estimate the option price by using the Monte Carlo method with:

$$\varphi(x) = e^{-rT}(K - x)^{+}$$

- In the case where $K \ll S(0)$, the probability of exercise $\Pr\{S(T) \leq K\}$ is very small
- Therefore, we have to increase the probability of exercise in order to obtain a more efficient estimator

Importance sampling

- In the case of the Black-Scholes model, the density function of $S(T)$ is equal to:

$$f(x) = \frac{1}{x\sigma_x}\phi\left(\frac{\ln x - \mu_x}{\sigma_x}\right)$$

  where $\mu_x = \ln S_0 + \left(r - \sigma^2/2\right) T$ and $\sigma_x = \sigma\sqrt{T}$

- We consider the IS density $g(x)$ defined by:

$$g(x) = \frac{1}{x\sigma_z}\phi\left(\frac{\ln x - \mu_z}{\sigma_z}\right)$$

  where $\mu_z = \theta + \mu_x$ and $\sigma_z = \sigma_x$

Importance sampling

- For instance, we can choose θ such that the probability of exercise is equal to 50%. It follows that:

$$\begin{aligned}
\Pr\left\{Z \leq K\right\} = \frac{1}{2} & \Leftrightarrow & \Phi\left(\frac{\ln K - \theta - \mu_x}{\sigma_x}\right) = \frac{1}{2} \\
& \Leftrightarrow & \theta = \ln K - \mu_x \\
& \Leftrightarrow & \theta = \ln \frac{K}{S_0} - \left(r - \frac{1}{2}\sigma^2\right) T
\end{aligned}$$

Importance sampling

- We deduce that:

$$\mathcal{P} = \mathbb{E}\left[\varphi\left(S\left(T\right)\right)\right] = \mathbb{E}\left[\varphi\left(S'\left(T\right)\right)\cdot\mathcal{L}\left(S'\left(T\right)\right)\right]$$

where:

$$\mathcal{L}\left(x\right) = \frac{\dfrac{1}{x\sigma_x}\phi\left(\dfrac{\ln x - \mu_x}{\sigma_x}\right)}{\dfrac{1}{x\sigma_z}\phi\left(\dfrac{\ln x - \mu_z}{\sigma_z}\right)} = \exp\left(\frac{\theta^2}{2\sigma_x^2} - \left(\frac{\ln x - \mu_x}{\sigma_x}\right)\cdot\frac{\theta}{\sigma_x}\right)$$

and $S'\left(T\right)$ is the same geometric Brownian motion than $S\left(T\right)$, but with another initial value:

$$S'\left(0\right) = S\left(0\right)e^{\theta} = Ke^{-\left(r-\sigma^2/2\right)T}$$

Importance sampling

Example #10

We assume that $S_0 = 100$, $K = 60$, $r = 5\%$, $\sigma = 20\%$ and $T = 2$. If we consider the previous method, the IS process is simulated using the initial value $S'(0) = Ke^{-\left(r-\sigma^2/2\right)T} = 56.506$, whereas the value of θ is equal to -0.5708

Importance sampling

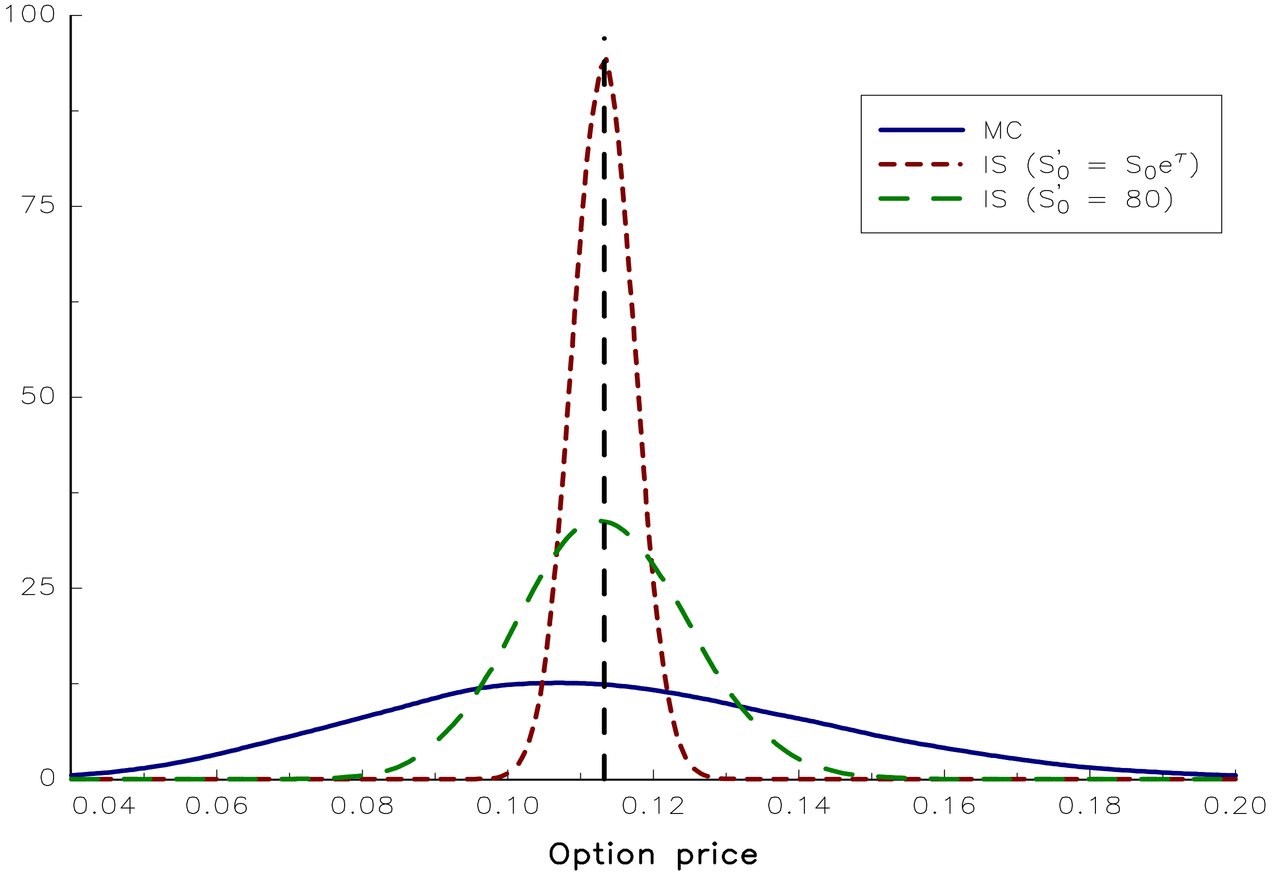


Figure: Density function of the estimators $\hat{\mathcal{P}}_{\mathrm{MC}}$ and $\hat{\mathcal{P}}_{\mathrm{IS}}$ ($n_S = 1\,000$)

Quasi-Monte Carlo simulation methods

- We consider the following Monte Carlo problem:

$$I = \int \cdots \int_{[0,1]^n} \varphi \left(x_1, \ldots, x_n \right) \mathrm{d}x_1 \cdots \mathrm{d}x_n$$

- Let X be the random vector of independent uniform random variables. It follows that $I = \mathbb{E} \left[\varphi \left(X \right) \right]$
- The Monte Carlo method consists in generating uniform coordinates in the hypercube $[0,1]^n$
- Quasi-Monte Carlo methods use non-random coordinates in order to obtain a more nicely uniform distribution

Quasi-Monte Carlo simulation methods

A low discrepancy sequence $\mathcal{U} = \{u_1, \ldots, u_{n_S}\}$ is a set of deterministic points distributed in the hypercube $[0,1]^n$

Quasi-Monte Carlo simulation methods

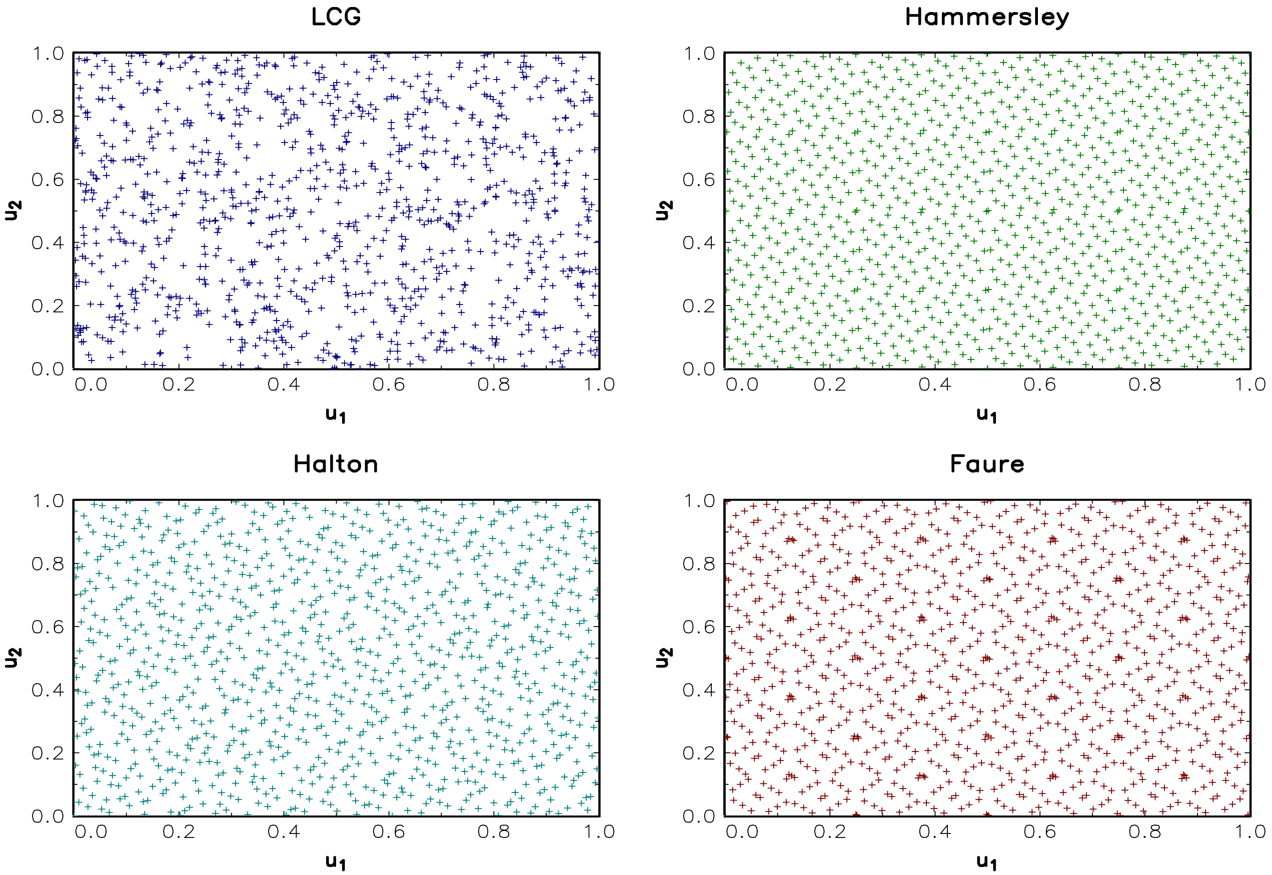


Figure: Comparison of different low discrepancy sequences

Quasi-Monte Carlo simulation methods

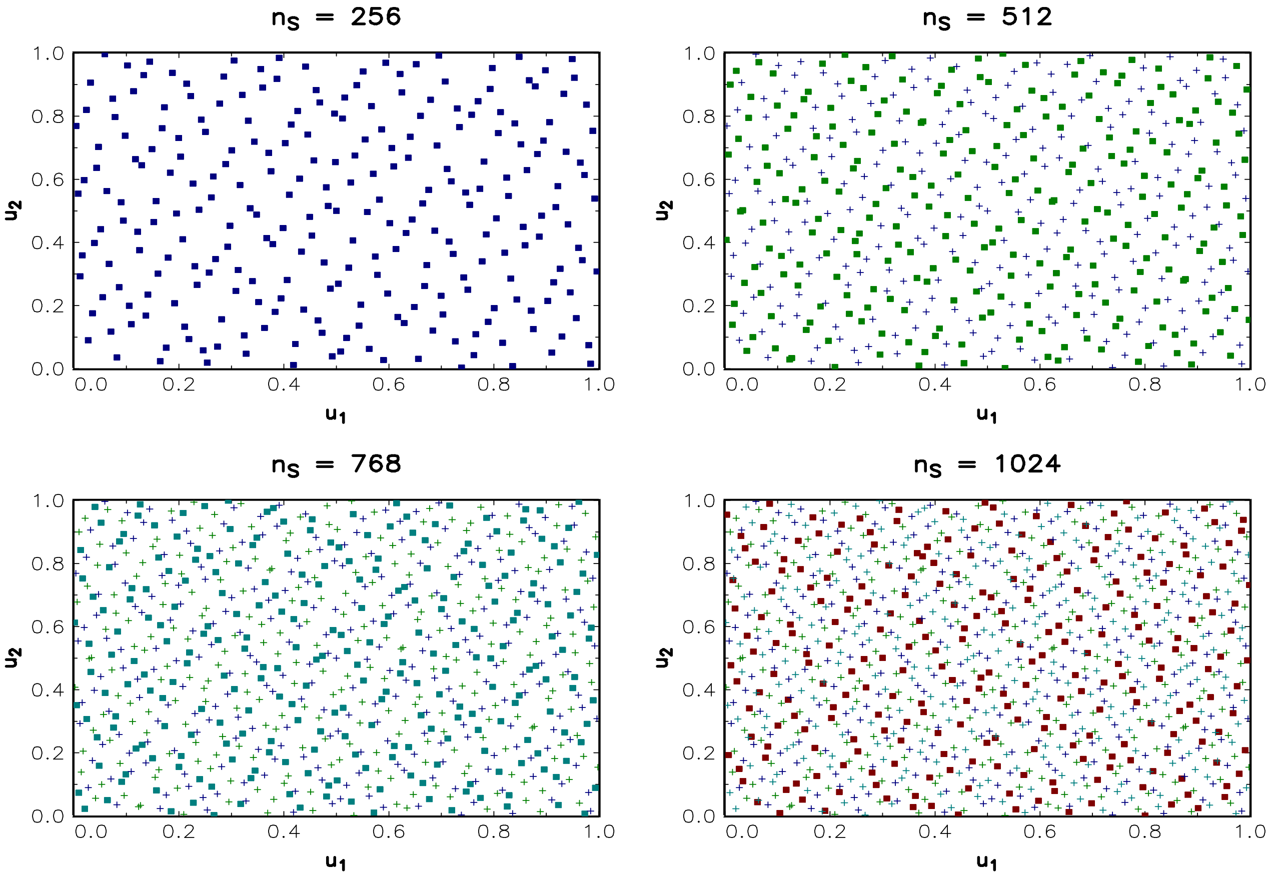


Figure: The Sobol generator

Quasi-Monte Carlo simulation methods

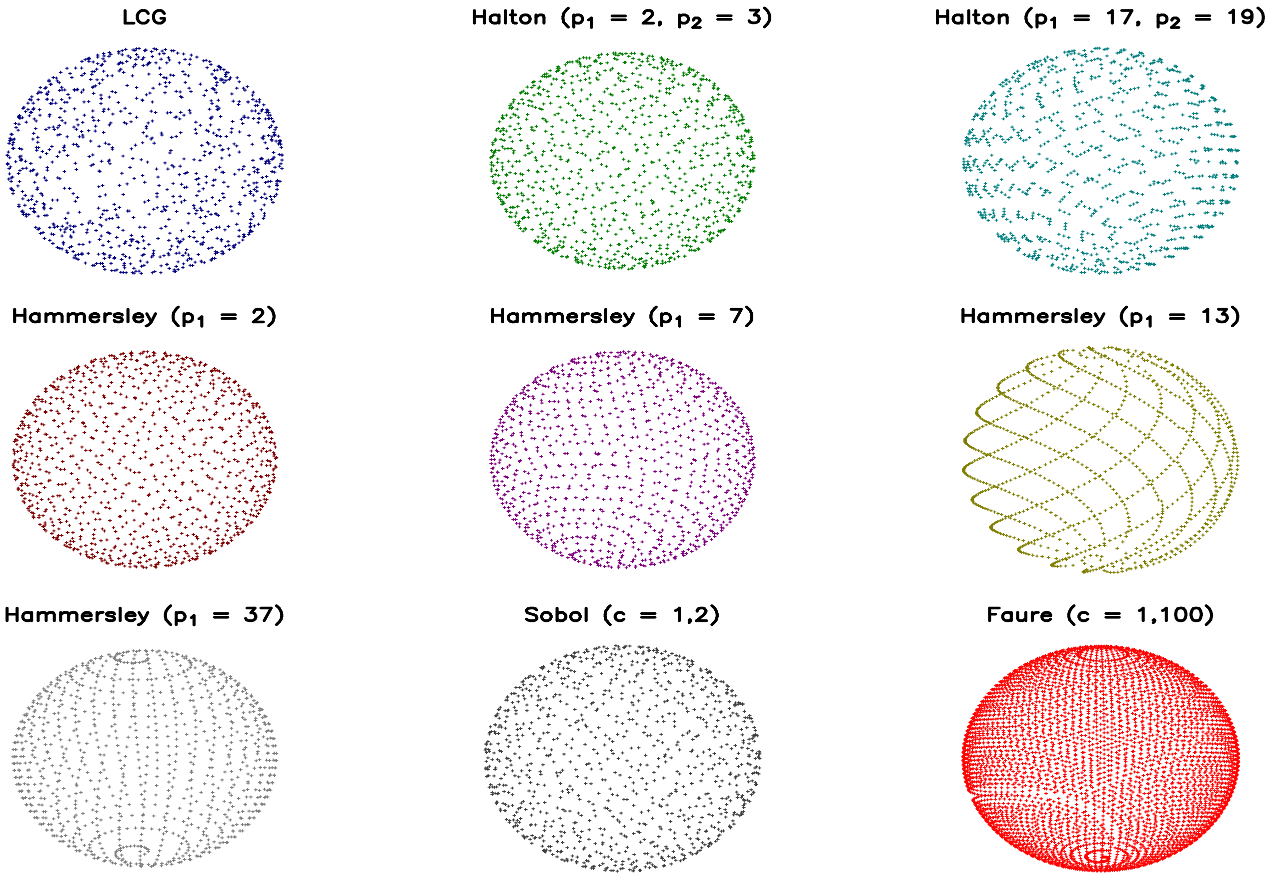


Figure: Quasi-random points on the unit sphere

Quasi-Monte Carlo simulation methods

Example #11

We consider a spread option whose payoff is equal to $(S_1(T) - S_2(T) - K)^+$. The price is calculated using the Black-Scholes model, and the following parameters: $S_1(0) = S_2(0) = 100$, $\sigma_1 = \sigma_2 = 20\%$, $\rho = 50\%$ and $r = 5\%$. The maturity T of the option is set to one year, whereas the strike K is equal to 5. The true price of the spread option is equal to 5.8198.

Quasi-Monte Carlo simulation methods

Table: Pricing of the spread option using quasi-Monte Carlo methods

n_S	10^2	10^3	10^4	10^5	10^6	5×10^6
LCG (1)	4.3988	5.9173	5.8050	5.8326	5.8215	5.8139
LCG (2)	6.1504	6.1640	5.8370	5.8219	5.8265	5.8198
LCG (3)	6.1469	5.7811	5.8125	5.8015	5.8142	5.8197
Hammersley (1)	32.7510	26.5326	21.5500	16.1155	9.0914	5.8199
Hammersley (2)	32.9082	26.4629	21.5465	16.1149	9.0914	5.8199
Halton (1)	8.6256	6.1205	5.8493	5.8228	5.8209	5.8208
Halton (2)	10.6415	6.0526	5.8544	5.8246	5.8208	5.8207
Halton (3)	8.5292	6.0575	5.8474	5.8235	5.8212	5.8208
Sobol	5.7181	5.7598	5.8163	5.8190	5.8198	5.8198
Faure	5.7256	5.7718	5.8157	5.8192	5.8197	5.8198

Exercises

- Exercise 13.4.1 – Simulating random numbers using the inversion method
- Exercise 13.4.6 – Simulation of the bivariate Normal copula
- Exercise 13.4.7 – Computing the capital charge for operational risk

References

DEVROYE, L. (1986)
Non-Uniform Random Variate Generation, Springer-Verlag.

RONCALLI, T. (2020)
Handbook of Financial Risk Management, Chapman and Hall/CRC Financial Mathematics Series, Chapter 13.