



Dipartimento
di Economia e Finanza

Cattedra Probability

Monte Carlo Methods and Stochastic Processes in Option Pricing

Prof. Sara Biagini

RELATORE

Prof. Marco Perone Pacifico

CORRELATORE

Alessandro Lo Verde

CANDIDATO

Anno Accademico 2020/2021

Contents

1 A Toolbox for Stochastic Simulation: Fundamentals of Monte Carlo	5
1.1 Probability framework	5
1.1.1 The Laws of Large Numbers	6
1.1.2 Central Limit Theorem	8
1.2 Monte Carlo Method and Simulation	10
1.2.1 Definition and Principles	10
1.2.2 Confidence Intervals and Error Analysis	12
1.2.3 Efficiency of Monte Carlo Estimators	15
1.3 Generation of Uniform Pseudo Random Numbers	17
1.3.1 Linear Congruential Generator	18
1.3.2 Combined Generators and Other Methods	21
1.4 General Sampling Methods	22
1.4.1 Inversion Method	22
1.4.2 Acceptance rejection method	24
1.4.3 Gaussian Random Variables and Vectors	25
2 Modeling Option Prices: from Discrete Time to Continuous Time Pricing and First Simulations	30
2.1 Discrete Time Option Pricing	30
2.1.1 A Discrete Time Market Model	31
2.1.2 The Binomial Model	36
2.1.3 European Option Pricing	37
2.1.4 American Option Pricing	40
2.2 Continuous Time Option Pricing: Basic Theory and Models	41
2.2.1 Basic Notions	41
2.2.2 Price Models	42
2.2.3 Deriving BS differential equation	44
2.2.4 The Girsanov's Theorem	47
2.2.5 Martingale Approach, Risk neutral pricing and Monte Carlo .	48
2.3 Monte Carlo Evaluation of a European Option with Asian Payoff .	51
2.3.1 Simulation of the Trajectories: Euler and Exact Solution .	51
2.3.2 Numerical Solution	53
2.3.3 Analysis of the convergence	54

3 Option Pricing with Monte Carlo: Variance Reduction Techniques and Stochastic Volatility Model Application (SABR)	56
3.1 Variance Reduction Techniques: Improving the Monte Carlo estimate for a European Call Option with Asian Payoff	56
3.1.1 Antithetic Variates Technique	57
3.1.2 Antithetic Variates: Option pricing application	57
3.1.3 Control Variates Technique	60
3.1.4 Control Variates: Option pricing application	60
3.2 Beyond the Black and Scholes Model	63
3.2.1 Implied Volatility	63
3.2.2 Models for Local and Stochastic Volatility: a Brief Overview .	64
3.2.3 d-dimensional Brownian motion	66
3.3 Stochastic Volatility and Monte Carlo Methods	67
3.3.1 The SABR Model	67
3.3.2 Calibration of the Implied volatility Surface	69
3.3.3 Monte Carlo and SABR: Pricing of a Barrier Call Option down & out	71
A Appendices	77
A.1 Definitions	77
A.1.1 Expected value	77
A.1.2 L^P Spaces and norm	78
A.2 Convergences	78
A.2.1 Almost Sure Convergence: $X_n \xrightarrow{a.s.} X$	78
A.2.2 L^p Convergence: $X_n \xrightarrow{L^p} X$	78
A.2.3 Convergence in Probability: $X_n \xrightarrow{P} X$	79
A.2.4 Convergence in Distribution: $X_n \xrightarrow{d} X$	79
A.2.5 Relations between Convergences	79
A.3 Inequalities	80
A.3.1 Jensen's Inequality	80
A.3.2 p-Norms' Inequalities	80
A.3.3 Holder's Inequality	80
A.3.4 Cauchy Schwarz Inequality (Corollary of Holder)	81
A.3.5 Minkowsky's Inequality	81
A.3.6 Markov's Inequality	81
A.4 Characteristic Function	82
A.4.1 Characteristic Function: Normal case	82
A.4.2 Characteristic Function: Browninan motion case	82
A.5 Ito's Formula in the stochastic case	82
A.5.1 General Ito Formula	82
A.5.2 Geometric Brownian motion and Ito	83

Introduction

What does Monte Carlo and convergences have to do with finance? Asset pricing theory under certain circumstances says that the price of a derivative security can be usefully represented as an expected value [15]. A Monte Carlo method is a computational method that utilizes a huge quantity of random numbers; this method can be successfully implemented to get to a good estimate of these expectations.

The first chapter will introduce the basic notions useful for the development of the Monte Carlo Method and then it will present some important methods to generate pseudo uniform random numbers and random samples from basically any distribution. Once we can sample from different distributions, we will be able to generate random sample paths useful to simulate different stochastic processes.

In chapter two we will focus on options which give the buyer the right to buy (call) or sell (put) an underlying stock (S) at a certain strike price (K) at a fixed future date (T) called "maturity". If we are in discrete time ($t \in \mathbb{N}$), we will see S_t as a discrete time stochastic process, a sequence of realizations of random variables. If we are in continuous time ($t \in \mathbb{R}$), we will see it as a continuous time stochastic process, with a "stochastic differential equation" describing its dynamics over time. Once we have comprehensively introduced and explained the most important market models for option pricing, we will evaluate the price of a European option with an Asian payoff through a Monte Carlo method; in order to do so, we will have to introduce a discretization method to simulate a huge number of trajectories for the underlying price process. Finally we will study its convergence to the mean by increasing the number of simulations.

At the beginning of chapter three we will implement two variance reduction technique (antithetic and control variates) which will be useful to accelerate the convergence of the Monte-Carlo method without further increasing the number of simulated trajectories. The model of B & S illustrated in chapter two has been superseded in practice, since the issuers opted for more realistic models which will allow us to catch the "true" volatility surface of the market, which is not flat as reflected in the B & S model. The model chosen to calibrate the volatility surface was the stochastic volatility SABR model of Hagan, Kumar, Lesniewski and Woodward ([17]), since it is generally able to provide a very good fit to the implied volatility smile quoted in the market and its parameters have a direct and clear influence on the shape of the calibrated volatility surface.

Chapter 1

A Toolbox for Stochastic Simulation: Fundamentals of Monte Carlo

This chapter will present initially the basic theorems useful to develop the notion of Monte Carlo method and Monte Carlo simulation, which will be the topic of the second section. In order to price options in the next two chapters with Monte Carlo, it is fundamental to be able to generate random samples from the Gaussian distribution. The generators implemented into common programming languages are often based on the very algorithms and methods introduced in this chapter. So, in the third section we will introduce some important methods to generate uniform random numbers. Lastly we will implement those numbers in some famous algorithms to get random sample from basically any distribution.

1.1 Probability framework

Glasserman provides a useful introduction to the Monte Carlo by saying that "Monte Carlo methods are based on the analogy between probability and volume" [15]. A general notion of probability associates an event with a set of outcomes and defines the probability of an event as its volume or measure relatively to that of a set of possible outcomes. "Monte Carlo uses this identity in reverse, calculating the volume of a set by interpreting the volume as a probability" [15]. In practice, taking the simplest case like a coin toss, this procedure consists in sampling randomly from a universe of possible outcomes and then in considering the fraction of random draws that fall in a given set as an estimate of the set's volume. The law of large numbers ensures that this estimate converges to the correct value when the number of draws increases and it guarantees a kind of aggregate predictability when we deal with many typical random variables. The central limit theorem provides information about the likely magnitude of the error in the estimate after a finite number of draws. In this section we will present these two main theorems.

1.1.1 The Laws of Large Numbers

In probability theory a "weak law" is a theorem that tells us how a sequence of probabilities converges. In the example of independent fair coin tosses, the convergence in probability of the arithmetic mean of the sample to the theoretical mean of the population, means that the binomial probability of the sample mean over the first n tosses of a coin differing "much" from the theoretical mean should be small. In 1713, Jacob Bernoulli first proved the Weak Law of Large Numbers for the special case when the X_i are binomial random variables, known as "Bernoulli's Theorem. A "strong law" tells how the sequence of random variables *as a sample path* behaves in the limit. That is, among the infinitely many sequences (or paths) of coin tosses we select one "at random" and then evaluate the sequence of means along that path. The Strong Law of Large Numbers says that with probability 1 that sequence of means along that path will converge to the theoretical mean. A Strong Law is an experiment-by-experiment statement: it says that "almost every" sample mean approaches the population mean as the sample size increases.

Weak law of large numbers (WLLN)

Let $(X_n)_{n \in \mathbb{N}}$ be a sequence of i.i.d. random variables $X_i \in \mathbb{R}^d \forall i = 1, \dots, n$, defined in the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ which lives in $L^2(\Omega, \mathbb{P})$, with expected value $\mu := E[X_i]$ and variance $\sigma^2 := Var(X_i)$.

Let's define the arithmetic mean of the sequence as: $M_n = \frac{1}{n} \sum_{i=1}^n X_n$.

Then we have that:

$$E[(M_n - \mu)^2] = \frac{\sigma^2}{n} \quad (1.1)$$

If this is true, since :

$$\lim_{n \rightarrow +\infty} \mathbb{E}[|X_n - \mu|^2] = \lim_{n \rightarrow +\infty} \frac{\sigma^2}{n} = 0 \quad (1.2)$$

we have that M_n converges in $L^2(\Omega, \mathbb{P})$ norm to the constant r.v. μ :

$$M_n \xrightarrow{L^2} \mu$$

Combining (1.1) with the "Markov's inequality" (A.3.6), with $X = M_n - \mu$, we have that:

$$\lim_{n \rightarrow \infty} \mathbb{P}(|M_n - \mu| > \epsilon) \leq \lim_{n \rightarrow \infty} \frac{E[(M_n - \mu)^2]}{\epsilon^2} = \lim_{n \rightarrow \infty} \frac{\sigma^2}{n\epsilon^2} = 0 \quad (1.3)$$

So, by the definition of the convergence in probability (A.2.3):

$$M_n \xrightarrow{P} \mu$$

Lastly, from Theorem (A.2.5 - v), X_n converges to μ in distribution:

$$M_n \xrightarrow{d} \mu \quad (1.4)$$

Proof If we prove the statement (1.1), then all the other observations follow. By the definition of expected value, of arithmetic mean and by the property of linearity (A.3), we have that:

$$E[M_n] = E\left[\frac{1}{n} \sum_{i=1}^n X_i\right] = \frac{1}{n} \sum_{i=1}^n E[X_i] = \frac{1}{n}(E[X_1] + \dots + E[X_n]) = \frac{n\mu}{n} = \mu \quad (1.5)$$

Since $(X_n)_{n \in \mathbb{N}} \in \mathbb{R}$ are independent we observe that:

$$\begin{aligned} E[(M_n - \mu)^2] &= Var[M_n] = Var\left[\frac{1}{n} \sum_{i=1}^n X_i\right] = \\ &\frac{1}{n^2} \sum_{i=1}^n Var[X_i] = \frac{1}{n^2} \sum_{i=1}^n Var[X_1] + \dots + Var[X_n] = \frac{n\sigma^2}{n^2} = \frac{\sigma^2}{n} \end{aligned} \quad (1.6)$$

□

From Thm. (A.2.5 - iv), if: $M_n \xrightarrow{p} \mu \Rightarrow \exists$ a sub-sequence $(M_{n_k})_{k \in \mathbb{N}} \mid M_{n_k} \xrightarrow{a.s.} \mu$. In the next paragraph we will see that it is possible to prove that: $M_n \xrightarrow{a.s.} \mu$

Strong Law of Large numbers (SLLN)

Let $(X_n)_{n \in \mathbb{N}}$ be a sequence of i.i.d. random variables $X_i \in \mathbb{R}^d$, $\forall i = 1, \dots, n$, defined in the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ which lives in $L^2(\Omega, \mathbb{P})$, with expected value $\mu := E[X_i]$ and variance $\sigma^2 := Var(X_i)$. Let's define the arithmetic mean of the sequence as: $M_n = \frac{1}{n} \sum_{i=1}^n X_i$. The Strong Law of Large Numbers says that:

$$P\left(\lim_{n \rightarrow \infty} M_n = \mu\right) = 1 \quad (1.7)$$

or:

$$M_n \xrightarrow{a.s.} \mu$$

Proof A detailed proof can be found in [32]

Kolmogorov's Strong law of large numbers

The Russian mathematician Andrey Kolmogorov proved the Strong Law with weaker assumptions: Let $(X_n)_{n \in \mathbb{N}}$ be a sequence of i.i.d. random variables $X_i \in \mathbb{R}^d$, $\forall i = 1, \dots, n$, defined in the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ which lives in $L^1(\Omega, \mathbb{P})$, with expected value $\mu := E[X_i]$. Let's define the arithmetic mean of the sequence as: $M_n = \frac{1}{n} \sum_{i=1}^n X_i$. The Kolmogorov's Strong Law of Large Numbers says that:

$$P\left(\lim_{n \rightarrow \infty} M_n = \mu\right) = 1 \quad (1.8)$$

or:

$$M_n \xrightarrow{a.s.} \mu$$

Proof A detailed proof can be found in [22]

1.1.2 Central Limit Theorem

Let $(X_n)_{n \in \mathbb{N}}$ be a sequence of i.i.d. random variables $X_i \in \mathbb{R}^d$, $\forall i = 1, \dots, n$, defined in the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ which lives in $L^2(\Omega, \mathbb{P})$.

If we define with:

$$\mu := E[X_i] \quad \text{and} \quad \sigma^2 := \text{Var}(X_i)$$

respectively the finite mean and the variance of the population and with: $S_n = \sum_{i=1}^n X_i$, $M_n = \frac{S_n}{n}$ respectively the sum and the arithmetic mean of the n r.v. X_i , then from 1.5 and 1.6, we have that:

$$E[M_n] = \mu \quad \text{and} \quad \text{Var}(M_n) = \frac{\sigma^2}{n}$$

. Lastly, from the "Strong Law of Large Numbers" we have that $M_n \xrightarrow{a.s.} \mu$

Let's consider a normalized arithmetic mean \tilde{M}_n , defined as:

$$\tilde{M}_n := \frac{M_n - E[M_n]}{\sqrt{\text{Var}(M_n)}} = \frac{M_n - \mu}{\sqrt{\frac{\sigma^2}{n}}} = \frac{M_n - \mu}{\frac{\sigma}{\sqrt{n}}}$$

Then by multiplying by $\frac{n}{n}$, we have that:

$$\tilde{M}_n := \left(\frac{n}{n} \right) \frac{\frac{S_n}{n} - \mu}{\frac{\sigma}{\sqrt{n}}} = \frac{S_n - n\mu}{\sqrt{n}\sigma} = \frac{1}{\sqrt{n}} \sum_{i=1}^n \left(\frac{X_i - n\mu}{\sigma} \right) \quad (1.9)$$

The Central limit theorem states that, for every possible distribution of X_n , the sequence of the normalized arithmetic means, converge weakly or in distribution to a Gaussian standard normal random variable $Z \in \mathbb{R}^d$:

$$\tilde{M}_n \xrightarrow{d} Z \sim \mathcal{N}(0, 1) \quad (1.10)$$

Then, since $M_n = \mu + \tilde{M}_n \frac{\sigma}{\sqrt{n}}$ for n large enough:

$$M_n \simeq \mu + Z \frac{\sigma}{\sqrt{n}} \xrightarrow{d} \mathcal{N}(\mu, \frac{\sigma^2}{n}) \quad \text{for } n \gg 1 \quad (1.11)$$

and

$$S_n = M_n * n \simeq n\mu + \sqrt{n}Z\sigma \xrightarrow{d} \mathcal{N}(n\mu, n\sigma^2) \quad \text{for } n \gg 1 \quad (1.12)$$

The key observation is that absolutely nothing (except a finite variance) is assumed

about the distribution of the random variables $(X_n)_{n \in \mathbb{N}}$. Therefore, if we define a random variable as the sum of many i.i.d. random variables with finite variances, then one can infer that the random variable's distribution is approximately Gaussian. So we can use empirical data to estimate μ and σ^2 of the asymptotic distribution of X_n . One of the numerical methods to do inference on these parameters is the Monte-Carlo method that we will treat in the next section.

1.2 Monte Carlo Method and Simulation

1.2.1 Definition and Principles

A Monte Carlo (MC) method is a computational method that utilizes random numbers. There is no consensus on how Monte Carlo should be defined. For example, Sawilowsky [33] distinguishes between a Monte Carlo method, and a Monte Carlo simulation: a Monte Carlo method is a technique that can be used to solve a mathematical or statistical problem (ex. Determine the behaviour of repeated coin tosses by tossing a large number of times a coin and computing the ratio of heads vs tails); a Monte Carlo simulation uses repeated sampling to obtain the statistical properties of some phenomenon (or behaviour).

One of the first example of random simulation made for numerical computation is described by the experiment of the Buffon needle proposed by the count with the same name in 1733. Later results are from Von Neumann, Fermi and other great minds of the XX^{th} century (more on the history of Monte Carlo methods can be found in [16]).

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

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

The typical problem which can be solved by a Monte Carlo method is the numerical integration. Let's consider the following integral:

$$F(X) = \int_{[a,b]^d} f(x) dx \quad (1.13)$$

where $f : [a, b]^d \rightarrow \mathbb{R}$ is an integrable function and x is a $d \times 1$ vector of components:

$$\begin{bmatrix} x_1 & x_2 & \dots & x_d \end{bmatrix}$$

If the integral is in \mathbb{R}^d , it is always possible to make, as a preliminary step, a change of variable such that the domain of integration becomes $[a, b]^d$:

$$F(X) = \int_a^b \int_a^b \dots \int_a^b f(x_1, x_2, \dots, x_d) dx_1 dx_2 \dots dx_d$$

Crude Monte Carlo

We now illustrate a first approach to the Monte Carlo method to solve a problem of numerical integration using uniform random numbers.

Monte Carlo simulation is the evaluation of the definite integral (1.13) by identifying a random variable Y and a density p with support in $[a, b]^d$, and a function

g such that the expected value of $g(Y)$ is $F(X)$:

$$E[g(Y)] = \int_{[a,b]^d} g(y)p(y)dy = \int_{[a,b]^d} f(y)dy = F(X) \quad (1.14)$$

If, for instance, we take $Y = U_j \sim \mathcal{U}(0, 1)^d$, and $g(Y) = (b-a)^d f(Y)$, then we notice that the integral $F(X)$ is equal to the expected value $E[g(Y)] = E[g(U_j)] := \mu^f$, since:

$$\begin{aligned} \mu^f &= \int_{[a,b]^d} g(u_j)p(u_j)du_j = \int_a^b \dots \int_a^b g(u_{j,1}) * \dots * g(u_{j,d}) \left(\frac{1}{b-a}\right)^d du_{j,1}du_{j,2}...dx_{j,d} = \\ &= \left(\frac{1}{b-a}\right)^d \int_{[a,b]^d} g(u_j)du_j = \left(\frac{b-a}{b-a}\right)^d \int_{[a,b]^d} f(u_j)du_j = F(X) \end{aligned}$$

In this case we have that: $F(X) = E[g(U_j)] = (b-a)^d E[f(U_j)]$

The evaluation of $F(X)$ by Monte-Carlo method is typically based on generating 'n' i.i.d. random vectors (samples) of d components $(U_1, \dots, U_j, \dots, U_n)$ each one uniformly distributed on $[0, 1]^d$, where U_j is a d-dimensional uniform random vector $U_j \sim \mathcal{U}(0, 1)^d$.

Of course, the simulation of the d-components of each U_j can be made via the generation of 'd' i.i.d. random variables uniformly distributed on $[0, 1]$; in this way, we can construct the $n \times d$ matrix U which has as j-th row entries 'n' uniform random vectors of dimension 'd':

$$\begin{bmatrix} u_{1,1} & u_{1,2} & \dots & u_{1,d} \\ \dots & \dots & \dots & \dots \\ u_{j,1} & u_{j,2} & \dots & u_{j,d} \\ \dots & \dots & \dots & \dots \\ u_{n,1} & u_{n,2} & \dots & u_{n,d} \end{bmatrix} = \begin{bmatrix} U_1 \\ \dots \\ U_j \\ \dots \\ U_n \end{bmatrix}$$

So, in general, the Monte-Carlo estimator of $F(X)$, sampled from a multivariate uniform distribution is:

$$\hat{F}(X) = M[g(U_i)] = (b-a)^d M[f(U_i)] = (b-a)^d M_n^f := (b-a)^d \frac{1}{n} \sum_{i=1}^n f(U_i) \quad (1.15)$$

In our case with $X \sim \mathcal{U}(0, 1)^d$, we have that the Monte-Carlo estimator can be written as the arithmetic mean of the sample of $U_i \forall i = 1, \dots, n$:

$$\hat{F}(X) = M_n^f := \frac{1}{n} \sum_{i=1}^n f(u_i) \quad (1.16)$$

In every case, (1.8) assures that, if μ^f is finite, then M_n^f is a good estimator of μ^f (then of our $F(x)$ in our special case) because:

$$P\left(\lim_{n \rightarrow +\infty} M_n^f = \mu^f\right) = 1 \Rightarrow M_n^f \xrightarrow{a.s.} \mu^f \quad (1.17)$$

Moreover, if we suppose that f is square integrable ($\int_{[0,1]^d} f^2(x)dx < +\infty$) ; then by the central limit theorem (1.1.2), we have that:

$$\sqrt{n}(M_n^f - \mu^f) \xrightarrow{n \rightarrow +\infty} \mathcal{N}(0, \sigma^2) \quad (1.18)$$

with $\sigma^2 = \text{Var}(f(U_j)) \forall i = 1 \dots n$. We notice how the convergence rate (in distribution) does not depend on the dimension d , but on \sqrt{n} , which becomes especially interesting in problems where d is large (or even infinite). We will go deeper in the question of the convergence rate, accuracy and estimation of the error of the estimates in the following sections.

Monte Carlo with known density p

Suppose that the integral we want to estimate is the Expected value of a certain $g(x)$:

$$F(X) = \int_{[a,b]^d} f(x)dx = \int_{[a,b]^d} g(x)p(x)dy = E[g(x)] \quad (1.19)$$

It may require some scaling to get the density to be over the support $(a, b)^d$ [14]. Moreover, suppose that we know the probability density p over the support, and that we can generate n random variates Y_i from the distribution with density p ; then our Monte Carlo estimate of $F(X)$ is:

$$\hat{F}(X) = M_n^f := \frac{1}{n} \sum_{i=1}^n g(y_i) \quad (1.20)$$

1.2.2 Confidence Intervals and Error Analysis

Non Asymptotic Error Estimates

Concerning the computational complexity, we can give a first estimate of the error of the Monte Carlo method directly by the Markov inequality: Given the assumptions and definitions presented in the first section for the WLLN (1.1), we define as the arithmetic mean $M_n^f := \frac{1}{n} \sum_{i=1}^n f(x_i)$ and $\mu^f = E[f(X)] = F[X]$ the i.i.d R.V. $X_i \sim \mathcal{U}(0, 1) \quad \forall i = 1, \dots, n$, then by Markov's inequality and the independence of the X (1.6), for every $\epsilon > 0$, we have that:

$$\mathbb{P}(|M_n^f - \mu^f| \geq \epsilon) \leq \frac{E[(M_n^f - \mu^f)^2]}{\epsilon^2} = \frac{\sigma^2}{n\epsilon^2} \quad (1.21)$$

The 1.21 gives us an "explicit estimate of the speed of convergence" [31], which can be seen clearly if we rewrite the inequality as:

$$\mathbb{P}(|M_n^f - \mu^f| \leq \epsilon) \geq p := 1 - \frac{\sigma^2}{n\epsilon^2} \quad (1.22)$$

Then for a fixed probability $p \in]0, 1[$ and a number of simulations n , the maximum approximation error is:

$$\epsilon = \frac{\sigma}{\sqrt{n(1-p)}} \quad (1.23)$$

So we have that:

$$\begin{aligned} \mathbb{P}(|M_n^f - \mu^f| \leq \frac{\sigma}{\sqrt{n(1-p)}}) &\geq p \\ \Rightarrow \mathbb{P}\left(\underbrace{\mu^f - \frac{\epsilon}{\sigma}}_{\sqrt{n(1-p)}} \leq M_n^f \leq \underbrace{(\mu^f + \frac{\epsilon}{\sigma})}_{\sqrt{n(1-p)}}\right) &\geq \underbrace{p}_{1 - \frac{\sigma^2}{n\epsilon^2}} \end{aligned} \quad (1.24)$$

First of all we note that, being the technique based on the generation of random numbers, the result and the error of the Monte Carlo method are random variables. Formula 1.22 gives an estimate of the error in terms of three parameters:

- i) n , the number of samples, i.e. how many random numbers we have generated;
- ii) ϵ , the maximum approximation error;
- iii) p , the minimum probability that the approximated value M_n belongs to the confidence interval $[\mu^f - \epsilon, \mu^f + \epsilon]$, with ϵ defined as in the (1.23) and p defined as in (1.22).

Let's reconsider the example of the computation of the integral (1.13) with the Monte Carlo Method; we have that $\sigma = \sqrt{\text{Var}[f(U_j)]} \quad \forall j = 1, \dots, n$ with $U \sim \mathcal{U}[0, 1]^d$. In this case the maximum error of the method can be estimated by:

$$\epsilon_M = \sqrt{\frac{\text{Var}(f(U_j))}{n(1-p)}} \quad (1.25)$$

We note that the error ϵ_m is of the order of $\sqrt{\frac{1}{n}}$ regardless of the dimension of the problem and is proportional to σ . So, from a computational point of view σ is a crucial parameter which influences significantly the efficiency of the approximation:

$$\sigma^2 = \text{Var}(f(U_j)) = E[(f(U_j))^2] - E[f(U_j)]^2, \quad \forall j = 1 \dots n$$

Typically σ is not known; nevertheless it can be written as a difference between expectations, so that we can evaluate this using the same random numbers used to evaluate the expected value μ^f :

$$\sigma_n^2 = \frac{1}{n-1} \left(\sum_{i=1}^n [(f(U_i))^2] - \overbrace{\sum_{i=1}^n f(U_i)^2}^{n(M_n^f)^2} \right), \quad \forall i = 1 \dots n \quad (1.26)$$

So, we know that the standard error of the mean is: $SEM = \sqrt{\text{Var}(M_n)} = \sigma_{M_n} = \frac{\sigma}{\sqrt{n}} \approx \frac{\sigma_n}{\sqrt{n}}$ but we know that this can be only a good approximation (we have assumptions on independence and on boundedness of the first moment), even though in generality is sufficiently accurate to estimate the error of the MC mean with

respect to the true population mean satisfactorily.

In absence of better approximations the Monte Carlo method allows us to construct an approximation of $E[f(X)]$ and a completely explicit error estimation, in terms of confidence intervals (as described before). Usually, in order to improve the effectiveness and reduce the error of estimation, variance-reduction methods are used. These techniques employ the specific features of the problem to reduce the value of σ_n and consequently increase the speed of convergence: for the description of such techniques, we refer to section 3.1.

Asymptotic Error Analysis

Now we show that the central limit theorem provides an estimate of the speed of convergence and of the error of distribution as $n \rightarrow \infty$.

Let $(Y_n)_{n \in \mathbb{N}}$ be a sequence of i.i.d. r.v. with finite expectation and variance such that $Y_i = f(X_i)$, where $X_i \sim U(0, 1) \forall i = 1, \dots, n$ (Monte Carlo estimation of $F(x)$ in the one dimensional case).

Indeed, by CLT - 1.1.2 we have that the standardized arithmetic mean of the Monte Carlo is:

$$\tilde{M}_n^f = \sqrt{n} \left(\frac{M_n^f - \mu^f}{\sigma} \right) \xrightarrow{d} Z \sim \mathcal{N}(0, 1) \quad (1.27)$$

and so, asymptotically for $n \rightarrow \infty$, for every $y = f(x) \in \mathbb{R}$, we have:

$$\sqrt{n} \left(\frac{M_n^f - \mu^f}{\sigma} \leq y \right) \approx \Phi(y) \quad (1.28)$$

where Φ is the standard normal distribution. Consequently, for every $y > 0$, we have that:

$$\mathbb{P}\left(\left(\mu^f - \frac{\sigma y}{\sqrt{n}}\right) \leq M_n^f \leq \left(\mu^f + \frac{\sigma y}{\sqrt{n}}\right)\right) \approx \overbrace{p}^{2\Phi(y)-1} \quad (1.29)$$

Therefore, for a fixed $p \in]0, 1[$, the distance between the exact value and the approximated one is with probability p less than:

$$\frac{\sigma}{\sqrt{n}} \Phi^{-1}\left(\frac{p+1}{2}\right) \quad (1.30)$$

For example, for $p \approx 95\%$, $\Phi^{-1}\left(\frac{p+1}{2}\right) \approx 1.96$. In practice these estimates give a more accurate and practical estimate than the non-asymptotic ones.

This fact can be justified rigorously by the Berry-Essen Theorem: it gives the speed of convergence in the central limit theorem, thus allowing us to obtain rigorous estimates for the confidence intervals.

In the next statement we assume, for the sake of simplicity, that $E[Y_j] = 0$ (we can always make this assumption satisfied by substituting Y with $Y - E[Y_j]$).

Berry Essent Theorem

Let $(Y_n)_{n \in \mathbb{N}}$ be a sequence of i.i.d. r.v. such that $E[Y_j] = 0$ and $\sigma^2 = \text{var}(Y_j)$, $\text{SKW} = E[|y|^3]$ are finite.

If Φ_n is the distribution function of $\frac{\sqrt{n}M_n^f}{\sigma}$ then, for every $y \in \mathbb{R}$:

$$|\Phi_n(y) - \Phi(y)| \leq \frac{\text{SKW}}{\sigma^3 \sqrt{n}} \quad (1.31)$$

For the proof we refer to [10].

Of course, in order to come back to our Monte Carlo Estimation framework, we can impose that $Y_i = f(X_i)$, where $X_i \sim U(0, 1) \forall i = 1, \dots, n$.

1.2.3 Efficiency of Monte Carlo Estimators

Part of this thesis is devoted to ways of improving Monte Carlo estimators. To discuss improvements, we first need to explain some criteria for comparing alternative estimators. Two considerations are particularly important regarding the efficiency in terms of optimization of the computing time and the possible bias .

Heuristic Efficiency Criterion

First, we introduce a heuristic criterion elaborated by Stéphane Crépey in [9] to compare the efficiency of various simulation schemes, with or without variance reduction. This criterion takes into account not only the accuracy (variance), but also the computation time required by the simulation for each scheme. The efficiency of a method \tilde{M} with respect to a method M is defined here as:

$$\mathcal{E} = \lim_{n, \tilde{n} \rightarrow +\infty} \frac{\sigma_{M_n}}{\tilde{\sigma}_{\tilde{M}_n}} \sqrt{\frac{t_n}{\tilde{t}_{\tilde{n}}}} \quad (1.32)$$

where σ_n and t_n are the standard error and the computation times of method M based on n simulation runs (respectively $\tilde{\sigma}_{\tilde{n}}$ and $\tilde{t}_{\tilde{n}}$ are the standard error and the computation times of the method \tilde{M} based on \tilde{n} simulation runs). Method \tilde{M} is considered to be more efficient than method M if $\mathcal{E} \geq 1$. For instance, $\mathcal{E} = 4$ means that for a given computation time method \tilde{M} is 4 times more accurate than method M , or that for a given accuracy method \tilde{M} is 4^2 times faster than method M .

Since by the strong WLLN (1.6), we know that:

$$\sigma_{M_n} = SE = \frac{\sigma}{\sqrt{n}} \quad \text{and} \quad \tilde{\sigma}_{\tilde{M}_n} = \tilde{SE} = \frac{\tilde{\sigma}}{\sqrt{\tilde{n}}}$$

and assuming computation times proportional to the sample sizes (so that $t_n = kn$, where k is a factor which expresses the complexity of the algorithm for method M , and likewise $\tilde{t}_{\tilde{n}} = \tilde{k}\tilde{n}$ for method \tilde{M}), we have that efficiency in the sense of the

criterion \mathcal{E} is asymptotically independent to the sample size (since both tend to infinity and disappear from the formula):

$$\mathcal{E} \sim_{n,\tilde{n} \rightarrow +\infty} \frac{\sigma}{\tilde{\sigma}} \sqrt{\frac{k}{\tilde{k}}} \quad (1.33)$$

Since:

$$\mathcal{E} = \lim_{n,\tilde{n} \rightarrow +\infty} \frac{\sigma_{M_n}}{\tilde{\sigma}_{M_n}} \sqrt{\frac{kn}{\tilde{k}\tilde{n}}} = \lim_{n,\tilde{n} \rightarrow +\infty} \frac{\sigma\sqrt{\tilde{n}}}{\tilde{\sigma}\sqrt{n}} \sqrt{\frac{kn}{\tilde{k}\tilde{n}}}$$

Bias

The efficiency discussed above rely on the fact the estimators compared are averages of unbiased replicators. Once we know the estimations are unbiased, then variability of the estimation and the computational effort are the most important factors to be considered. Reducing these two in order to converge to an incorrect value would be pointless. Given the definition that we gave before of arithmetic mean, we define an unbiased estimator of the average as:

$$E[M_n^f] = \mu^f \quad (1.34)$$

Nonetheless, a simulation estimator can be biased for all finite sample sizes but it can converge asymptotically to the mean with probability one as the number of replications increases (an example is the price of a standard call option). In this case we say that the estimator is not biased, but it is consistent, which means that it converges in probability to the true estimator: $M_n \xrightarrow{P} X$.

The sequence of R.V. we want to estimate with Monte Carlo comply with the conditions necessary to follow the LLN, then in the (1.5) we proved the unbiasedness of the Monte Carlo estimator and in the (1.2) the convergence in L^2 which proved in a sense the consistency of it.

1.3 Generation of Uniform Pseudo Random Numbers

As we said, Monte Carlo methods require a large amounts of random numbers, and it was their use that spurred the development of pseudo-random number generators, a series of apparently random numbers used to derive every simulation, which were far quicker to use than the tables of random numbers that had been previously used for statistical sampling. Usually, a random number generator, produces a finite sequence of numbers in the unit interval.

We define as a “Generator of genuinely random numbers” [15] a mechanism that produces a sequence of random variables U_1, U_2, \dots, U_n with these properties:

- (i) each U_i is Uniformly distributed between 0 and 1 ($U_i \sim \mathbb{U}(0, 1)$);
- (ii) Every U_i are mutually independent with the others.

An effective generator should produce values consistently with these two properties. If the generator is good then even for small segments of sequence generated, it is difficult to distinguish it from a realization of independent uniform variables. Operationally, we can construct a generic random number generator defining:

- A finite set X , called "State Space".
- An element $x_0 \in X$, called the "seed".
- A function $T : X \rightarrow X$, called the transition function.
- A function $G : X \rightarrow (0, \dots, m - 1)$.

Fixed a seed x_0 , the pseudo random numbers are computed as $U_i := G(x_i)$, given that $x_i = T(x_{i-1})$.

We can define the following criteria for "goodness of a random number generator":

1. Statistical uniformity (randomness): the sequence of random numbers U_1, U_2, \dots, U_n obtained using the random number generator are hardly discernible from a truly random sample. For most applications, one can use one of the many generators (ex. Mersenne Twister generator [28]) that comply with rigid theoretical properties and have survived over time to rigorous statistical tests.
2. Speed: in many applications a huge quantity of random numbers is needed. The generation of random numbers should not reduce the speed of the simulation.
3. Period length: since the sequence of random numbers is periodic and since in many applications we need a lot of random numbers, the period length should be sufficiently large.
4. Reproducibility: it is the capacity to rerun a simulation using exactly the same inputs used previously. Reproducibility is very useful for debugging codes, for instance. We should be always able to generate the same random sequence starting from the same seed x_0 .
5. Portability and jumping ahead: the generator and the sequence of random numbers generated should be portable to different computers and must guarantee the possibility to get x_{l+n} given x_l for n large, without generate all the states in between.

1.3.1 Linear Congruential Generator

A linear congruential generator produces a stream of numbers that appear to be independent realizations of a uniform process and takes the form:

$$x_{i+1} = f(x_i) \bmod m \quad (1.35)$$

$$u_{i+1} = g(x_{i+1})$$

in which each single number determines its successor by means of a simple linear function f followed by a modular reduction and each single random number u_{i+1} is determined by a deterministic function g of x_{i+1} .

The operation $y \bmod m$ returns the remainder of y after the division by m :

$$y \bmod m = y - \lfloor \frac{y}{m} \rfloor m \quad (1.36)$$

where the floor function $\lfloor . \rfloor$ is the greatest integer less than or equal to the argument. The general Linear Congruential Generator, sometimes called "Lehmer Sequence" (since it was proposed by Lehmer [25] in 1951), has a recurrence of the following form:

$$x_{i+1} = (ax_i + c) \bmod m \quad (1.37)$$

$$u_{x+i} = x_{i+1}/m$$

$$\text{with: } 0 \leq x_i < m - 1$$

where a is called the "multiplier", c is called the "increment", and m is called the "modulus" of the generator (all integers).

For $c \neq 0$, the generator is sometimes called a "mixed congruential generator". The seed for this generator is just a single integer x_0 . Each x_i is scaled into the unit interval $(0,1)$ dividing it by m : $u_i = \frac{x_i}{m}$. Since the result of the mod m operation is always an integer $\in [0, m-1]$, the output (re-scaled) values u_i are $\in [0, \frac{m-1}{m}] \rightarrow [0, 1]$.

Since x_i is determined by $m-1$ possible different values of the x_{i-1} , the maximum period or cycle length of the linear congruential generator is $m - 1$. For a random number generator to be useful in most practical simple applications, the period (modulus) must be of the order of 10^9 to 10^{15} [14]. Even if the period of such generators remains too short relatively to the computational effort required, it is often used as a basic element in other, more adequate generators and it is possible to set the fundamental properties used also for better generators. Anyway, in order to have full period (i.e. the number of distinct values generated by the seed is $m-1$) in the case $c \neq 0$, some conditions illustrated by Knuth in ([21]) must be satisfied:

- (i) c and m are relatively prime;
- (ii) every prime number that divides m divides $a-1$;
- (iii) $a - 1$ is divisible by 4 if m is divisible by 4 too.

As a consequence of this, it can be observed that if m is a power of 2 , the

generator has full period if is odd and $a = 4n + 1 \forall n \in \mathbb{N}$.

Often, c is taken to be 0, and, in this case, the generator is called a “multiplicative congruential generator”:

$$x_{i+1} = (ax_i) \bmod m \quad (1.38)$$

In order to have full period in these types of generators, when m is prime, we need $\forall x_0 \neq 0$:

- (i) $a^{m-1} - 1$ to be a multiple of m ;
- (ii) $a^j - 1$ to not be a multiple of m for $j = 1, \dots, m-2$.

In the table below we present some examples of combinations of moduli and multipliers recommended; in each case the modulus m is a large prime equal to $2^{31} - 1$ (which is the largest integer that can be represented in a 32-bit word).

modulus m	multiplier a	reference
$2^{31} - 1$	16807	Lewis, Goodman and Miller [26]
$2^{31} - 1$	39373	L'Ecuyer [23]
$2^{31} - 1$	742938285	Fishman and Moore [13]

Table 1.1: Parameters for linear congruential generators

In the simple example below we present the algorithm for the computation of the Linear Congruential Generator which is full period $= 2^{10} - 1$ (follows the conditions set by Knuth in ([21]) and has seed which varies with the time displayed by the clock of the PC:

Algorithm 1: Simple Linear Congruential Generator (MATLAB)

```

 $a = 5;$ 
 $c = 7;$ 
 $m = 2^8;$ 
 $O = \text{clock};$ 
 $x_0 = \text{floor}(10 * O(6));$ 
 $x_{m1} = \text{zeros}(\text{floor } m, 1);$ 
for  $k = 1 : m$ 
     $x_0 = \text{mod}((a * x_0 + c), m)$ 
     $x_m(k) = x_0;$ 
end
 $\text{disp} (\text{'The Random Numbers are'});$ 
 $u_{m1} = x_{m1}/m$ 

```

The random numbers of the full period linear congruential constructed with the algorithm (1) are represented in the upper plot of figure 1.1 ($c = 7$). If you do not satisfy one of the conditions set by Knuth [21] listed before, then a recursive structure appears directly in the sequence of random numbers. Using the same algorithm, the lower plot of figure 1.1 shows us what happens if we do not satisfy the first conditions of Knuth [21] (in this case we set $c = 8$ and $m = 2^8$ so that they are no more relatively prime): an equal sub-sequences of pseudo random numbers

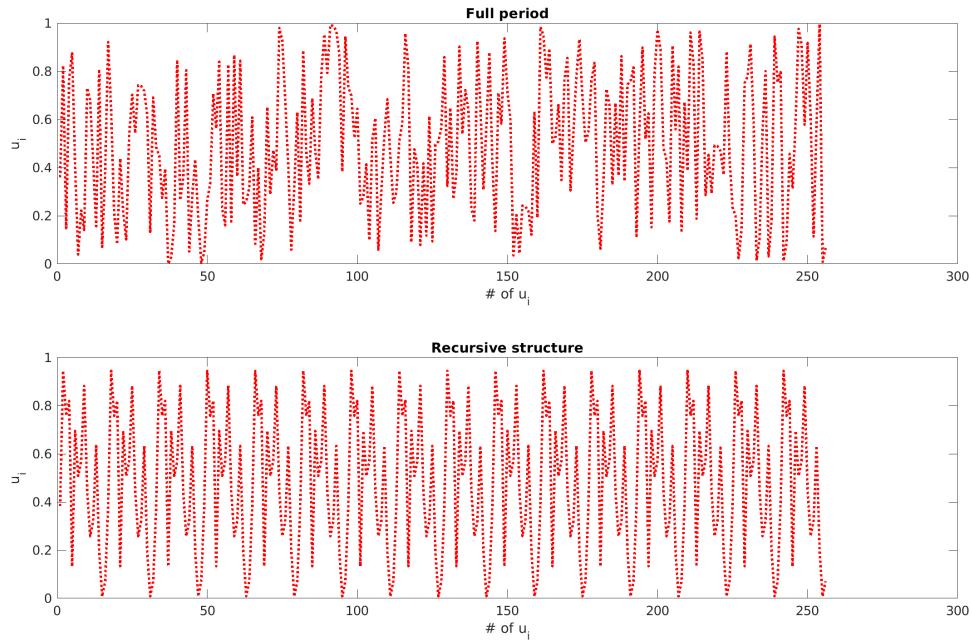


Figure 1.1: Linear Congruential Generator: full period vs recursive structure

repeats herself every $m/c = 32$ outputs. If a and m are properly chosen, the u_i s will “look like” they are randomly and uniformly distributed between 0 and 1. One way to evaluate a random number generator is to form points in $[0, 1]^d$ from consecutive output values and measure how uniformly they fill the space.

Let’s take for instance the pairs of consecutive (u_{i-1}, u_i) : even if we employ the best possible linear congruential generators (ex. using the parameters used by L’Ecuyer seen in table (1.1) (see Glasserman [15] pag. 47-49), their lattice structure (integer combination of a set of “basis vectors”) over the unit square will reveal their regular pattern and will help us distinguish them from genuinely random numbers.

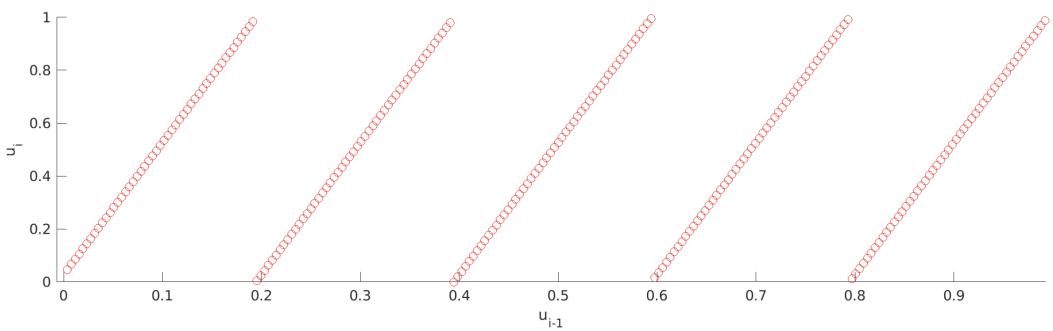


Figure 1.2: Lattice structure of the full period Linear Congruential Generator of Algorithm 1

An illustration of this regular pattern is revealed from the figure (1.2): The 2^8 distinct consecutive pairs generated by the full period algorithm (1.1), lie on just 5 parallel lines through the unit square.

1.3.2 Combined Generators and Other Methods

A first solution to the problems regarding the properties of the linear congruential generators is provided by methods that combine them with elementary operations. They seem to preserve the attractive computational features of these generators while extending their period and, attenuating their lattice structure.

Here we briefly illustrate the basic "Multiple Recursive Generator" Structure, which utilizes a higher order recursion of the form:

$$x_i = (ax_{i-1} + ax_{i-2} + \dots + ax_{i-k} + c) \bmod m \quad (1.39)$$

$$u_i = x_{i+1}/m$$

A seed for this generator consists of a stream of initial values $x_{k-1} + x_{k-2} + \dots + x_0$. The longest possible period becomes $m^k - 1$, since the vector $x_{i-1} + x_{i-2} + \dots + x_{i-k}$ can take up to m^k values (the period is shorter if the sequence reaches $(0, \dots, 0)$). Knuth [21] gives us the conditions in order to reach the maximal period for these sequences and L'Ecuyer [24] shows that a combined version of the MRG (first summing and then dividing them) produces a larger modulus and a less evident lattice structure while implementing smaller values and involving smaller computational effort.

Many methods have been proposed in literature to combine linear congruential generators:

- L'Ecuyer [24] and the combined version of the MRG,
- Eichenaucher-Herrman, Herrman, and Wegenkittl [11] for the "Inversive Congruential Generators";
- Tausworthe [34] for the "Feedback Shift Register".
- Matsumoto and Nishimura [28] and The Marsenne Twister generator (a refined version of the Feedback Shift with Period $2^{19937} - 1$ and optimal uniformity properties).

These and others advanced generators have been implemented largely in the algorithms provided by the most used softwares like R, Matlab, Python and allow us to obtain pseudo uniform random numbers which are statistically tested and have superior uniformity, longer periods, higher computational speed. In the next section we will see how to generate large random samples from a given probability distribution.

1.4 General Sampling Methods

We are assuming the availability of a sequence of $(U_n)_{n \in \mathbb{N}}$ independent and uniformly distributed on $(0, 1)$ random variables each one satisfying:

$$P(U_i \leq u) = \begin{cases} 0, & \text{for } u > 0 \\ u & \text{for } 0 \leq u \leq 1 \\ 1 & \text{for } u > 1 \end{cases} \quad (1.40)$$

Since most simulations involves random variables and vectors from distributions different form the uniform, we need to implement some algorithms that allow us to transform these uniforms into essentially any other distribution. We will see two classic methods to do it. Once we can sample from different distributions (in particular the Gaussian one) we will be able to generate random sample paths useful to simulate different stochastic processes.

1.4.1 Inversion Method

The goal of this method is to sample from a CDF F (continuous from the right, non-decreasing and $\in (0, 1)$) by generating a random variable X with the property:

$$P(X \leq x) = F(x) \quad \forall x$$

The method sets $U \sim \mathcal{U}[0, 1]$; then for strictly increasing F , the inverse of F is well defined and we can write:

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

For a non-decreasing CDF, we can set:

$$X = F^{-1}(u) = \inf\{x \mid F(x) \geq u\} \quad (1.41)$$

such that the inverse of F is well defined even if there are flat sections of the cumulative (in the flat section in which $F(x) = u$).

Proof We prove the assertion 1.4.1. By definition of F^{-1} we know that:

$$F^{-1}(u) \leq x \iff F(x) \geq u$$

Then we have that:

$$P(X \leq x) = P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x)$$

Exponential distribution Consider the exponential distribution with R.V. $X \sim \mathcal{E}(\lambda)$, with $\lambda > 0$. We have:

$$F(x) = 1 - e^{-\lambda x}$$

This function is invertible:

$$e^{-\lambda x} = 1 - F(x) \implies -\lambda X = \log(1 - F(x)) \implies F^{-1}(u) = \frac{1}{\lambda} \log(1 - u)$$

where the last equation follows by substituting the result obtained in the previous section 1.4.1). Since if U is normally distributed then $1 - U$ is uniformly distributed too, we can generate samples from the exponential distribution by:

$$F^{-1}(u) = X = -\frac{1}{\lambda} \log(U) \quad (1.42)$$

In the following algorithm (2) we present a simple example of inverse method applied to obtain samples from an exponential distribution using the (1.42). Once we can generate a sample from a R.V. $X \sim \mathcal{E}(\lambda)$ we can easily check that its arithmetic mean $M_x \approx \frac{1}{\lambda}$ for $n \rightarrow \infty$ and that it has the correct density and CDF:

Algorithm 2: Exponential random Variable with Inverse Method (MATLAB)

```

 $\lambda = 0.7;$ 
 $U = rand(10^7, 1);$ 
 $X = -(\frac{1}{\lambda} * \log(U));$ 
 $M_x = mean(X);$ 
 $f_x = \lambda * (exp(-\lambda * X));$ 
 $F_x = 1 - exp(-\lambda * X));$ 

```

```

figure
plot ( (sort (U)), (sort (X,'descendent') ) )
plot ( (sort (X)), (sort (f_x,'descendent') ) )
plot ( (sort (X)), (sort (F_x) ) )

```

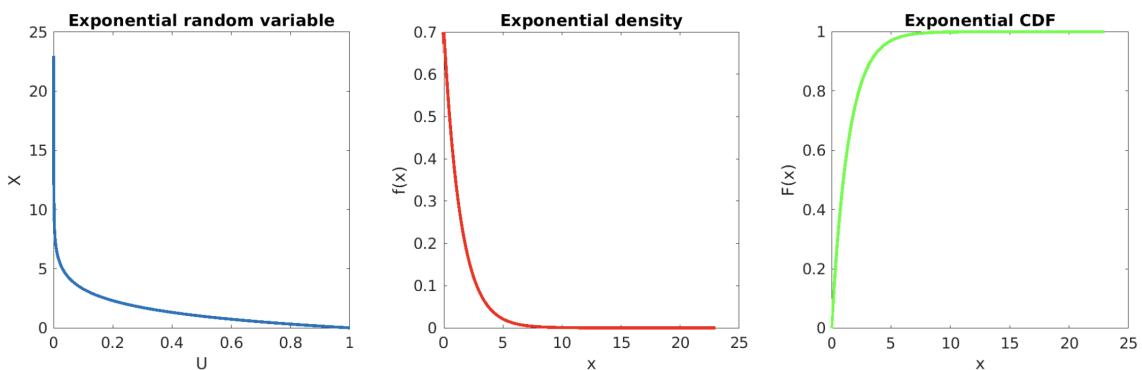


Figure 1.3: Inversion Method: Exponential Random variable

1.4.2 Acceptance rejection method

Finding an inverse CDF is not always possible. Moreover there are methods for generating a random variable distributed as F that are more efficient than the inverse transform method. One of these methods is the "acceptance-rejection method" introduced in 1951 by Von Neumann [36]. We start by assuming that the F has a probability density function $f(x)$ (continuous case). The basic idea is to find an alternative CDF G, with density function $g(x)$ such that it is close to $f(x)$ and for which we already have an efficient algorithm to generate random samples from it (for instance the inverse transform method or others).

In particular, we assume that the ratio $f(x)/g(x)$ is bounded by a constant $c > 0 \Rightarrow \sup_x \frac{f(x)}{g(x)} \leq c$. (Our goal is to have c as close to 1 as possible.) Here it is presented the procedure for generating X distributed as f with the Acceptance-Rejection method:

- 1 - Generate a r.v. Y distributed as G.
- 2 - Generate U (independent from Y).
- 3 - If $U \leq \frac{f(Y)}{cg(Y)}$, then set $X = Y$ ("accept") ; otherwise go back to 1 ("reject").

We notice that:

- $f(Y), g(Y)$, the ratio $\frac{f(Y)}{cg(Y)}$ are r.v. independent from U;
- The ratio is bounded between 0 and 1 $\Rightarrow 0 < \frac{f(Y)}{cg(Y)} \leq 1$;
- The number of times N that steps 1 and 2 need to be called (e.g., the number of iterations needed to successfully generate X) is itself a r.v. and it has a geometric distribution with "success" probability $p = P(U \leq \frac{f(Y)}{cg(Y)})$; $P(N = n) = (1 - p)^{n-1}p$; $n \geq 1$. and average $E(N) = 1/p$.
- In the end we obtain our X as having the conditional distribution of a Y given that the event $U \leq \frac{f(Y)}{cg(Y)}$ occurs.

Since U is uniform, Y has density $g(y)$ and that f is a density function, we have that p is equal to :

$$\begin{aligned} P(U \leq \frac{f(Y)}{cg(Y)}) &= E[\mathcal{I}_{[U \leq \frac{f(Y)}{cg(Y)}]}] = E[E[\mathcal{I}_{[U \leq \frac{f(Y)}{cg(Y)}]} \mid Y = y]] = E[P(U \leq \frac{f(Y)}{cg(Y)} \mid Y = y)] = \\ &= E[\frac{f(Y)}{cg(Y)}] = \int_{-\infty}^{+\infty} \frac{f(y)}{cg(y)} * g(y) dy = \frac{1}{c} \int_{-\infty}^{+\infty} f(y) dy = \frac{1}{c} \end{aligned} \quad (1.43)$$

So we have that $p = \frac{1}{c}$ and $E[N] = c$, the bounding constant, and we can now indeed see that it is desirable to choose our alternative density g so as to minimize this constant, that we take at its most efficient value of: $c = \sup_x \frac{f(x)}{g(x)}$. Of course the optimal function would be $g(x) = f(x)$ which is not what we have in mind since the whole point is to choose a different g (easy to simulate) in order to sample from f. The expected number of iterations of the algorithm required until an X is successfully generated is exactly the bounding constant $c = \sup_x \frac{f(x)}{g(x)}$.

Proof of the Acceptance/Rejection Method Now we need to prove that the distribution of Y given that $U \leq \frac{f(Y)}{cg(Y)}$ is equal to F:

$$P(Y \leq y \mid U \leq \frac{f(Y)}{cg(Y)}) = F(y)$$

From the (1.43), we have that: $P(U \leq \frac{f(Y)}{cg(Y)}) = p = \frac{1}{c}$.

Finally, we have that:

$$\begin{aligned} P(U \leq \frac{f(Y)}{cg(Y)} \mid Y \leq y) &= \frac{P(U \leq \frac{f(Y)}{cg(Y)}, Y \leq y)}{P(U \leq \frac{f(Y)}{cg(Y)})} = && [\text{Setting } \omega = Y] \\ &= P(U \leq \frac{f(\omega)}{cg(\omega)}, \omega \leq y) * c = \\ &= c \int_{-\infty}^y \int_0^{\frac{f(\omega)}{cg(\omega)}} g(\omega) d(u) d(\omega) \\ &= c \int_{-\infty}^y \frac{f(\omega)}{cg(\omega)} g(\omega) d(\omega) = \\ &= \frac{c}{c} \int_{-\infty}^y f(\omega) d(\omega) = F(y) \end{aligned}$$

1.4.3 Gaussian Random Variables and Vectors

The Gaussian distribution is fundamental in option pricing, in particular if we see at the assumption of log-normality of the returns in the Black and Scholes model for the pricing of European Options. Since a log normal r.v X is such that the log (X) = Y with Y distributed as a Gaussian, we can exponentiate Y to get a log normal. Let's see three methods that allow us to simulate normal random variables.

Acceptance Rejection method for Gaussian

Fishman [12] illustrate how we can create half normal random variables (has the distribution of the absolute value of a normal random variable). Glasserman [14] present a slightly different implementation to generate normal random variables. Here we present briefly the theory behind the method and an application with plot and algorithm in MATLAB. We define the double Exponential density (Laplace density) $g(y)$ on $(-\infty, +\infty)$ and the normal density $f(y)$ as:

$$g(y) = \frac{e^{-|y|}}{2} \quad \text{and} \quad f(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}$$

The ratio of the two densities is:

$$\frac{f(y)}{g(y)} = \sqrt{\frac{2}{\pi}} e^{-\frac{1}{2}y^2+|y|} \leq \sqrt{\frac{2e}{\pi}} \approx 1,3155 \equiv c$$

A sample from the double exponential random variable can be generated with the inverse algorithm (Algorithm 2) and then randomizing the sign.

The rejection test $u > \frac{f(y)}{cg(y)}$ must be implemented as:

$$u > \sqrt{\frac{2}{\pi}} e^{-\frac{1}{2}y^2 + |y|} * \sqrt{\frac{\pi}{2}} e^{-\frac{1}{2}} = e^{-\frac{1}{2}(|y|-1)}$$

Since f and g are two symmetric matrices around the y axis, we need to generate positive samples and determine randomly the sign only if the sample is accepted (we don't need an absolute value in the rejection test). The algorithm 3 illustrate the various steps of this sampling method. In Fig. 1.4 we observe that the normal density is dominated by the scaled exponential density $c*g(y)$.

Algorithm 3: Gaussian Acceptance Rejection

```

N = 106; c = 1.3155;
U1 = rand (N,1); U2 = rand (N,1); U3 = rand (N,1);
Y = - log(U1); Ys = sort (Y); gy = c * (exp(-Ys))/2;
Y0 = zeros(N, 1);
for i = 1:N
    Bou(i,1) = exp(-0.5*(Y(i,1)-1)2);
    if U2(i,1) ≤ Bou(i, 1)
        Y0(i, 1) = Y(i, 1);
    end
    if U3(i,1) ≤ 0.5
        Y2(i, 1) = -Y0(i, 1);
    else
        Y2(i, 1) = Y0(i, 1);
    end
end
X=Y2; Xs = sort(X); fx = exp(-(Xs.2)/2)/sqrt((2) * (π));

```

Figure

```

plot (Xs, fx, 'r', 'linewidth', 3) holdon
plot (-Ys, gy, 'b', 'linewidth', 3) holdon plot(Ys, gy, 'b', 'linewidth', 3)

```

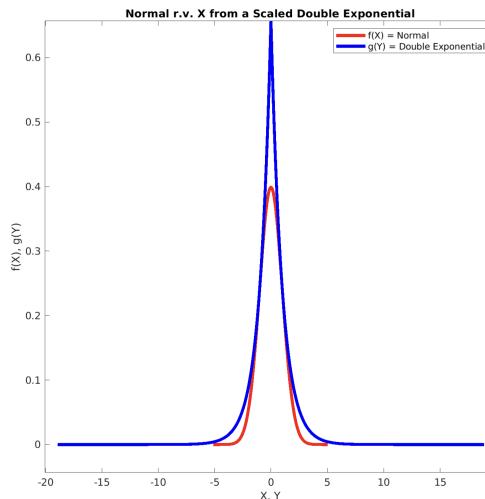


Figure 1.4: Acceptance Rejection Method for Gaussian Sampling

Box-Mueller pair: Sampling from the Bivariate Standard Normal

Let (U, V) be a bivariate uniform random vector, then $(Z_1, Z_2) = Z \sim \mathcal{N}(0, I_2)$:

$$Z_1 = \sqrt{-2 * \ln(U)} * \cos(2 * \pi * V) ; Z_2 = \sqrt{-2 * \ln(U)} * \sin(2 * \pi * V) \quad (1.44)$$

The properties which allow this sampling method are:

- (i) $R = Z_1^2 + Z_2^2 \sim \mathcal{E}(\frac{1}{2})$
- (ii) Given a radius \sqrt{R} of a circle centered in the origin, the point (Z_1, Z_2) is uniformly distributed around that circle.

So the algorithm consists in creating an exponential random variable R using the inverse method (thanks to the generation of the pseudo-uniform random variable U_1), and then by generating independently a random angle $V = 2\pi U_2 \sim \mathcal{U}(0, 2\pi)$ and using the trigonometric functions in order to compute the Gaussian pair (Z_1, Z_2) as represented in the Algorithm (4). For a formal proof involving test function see [9].

Algorithm 4: Box Mueller

```
N = 107; U1 = rand(N, 1); U2 = rand(N, 1);
for i = 1 :N
    R(i) = - 2*log(U1(i));
    V(i)= 2 * pi* U2(i);
    Z1(i) = sqrt(R(i))*cos(V(i));
    Z2(i) = sqrt(R(i))*sin(V(i));
end
```

```
figure
subplot(2,2,[3,4]); hist3([Z1',Z2'], 'nbins',[50,50])
subplot(2,2,1); histfit(Z1)
subplot(2,2,2); histfit(Z2)
```

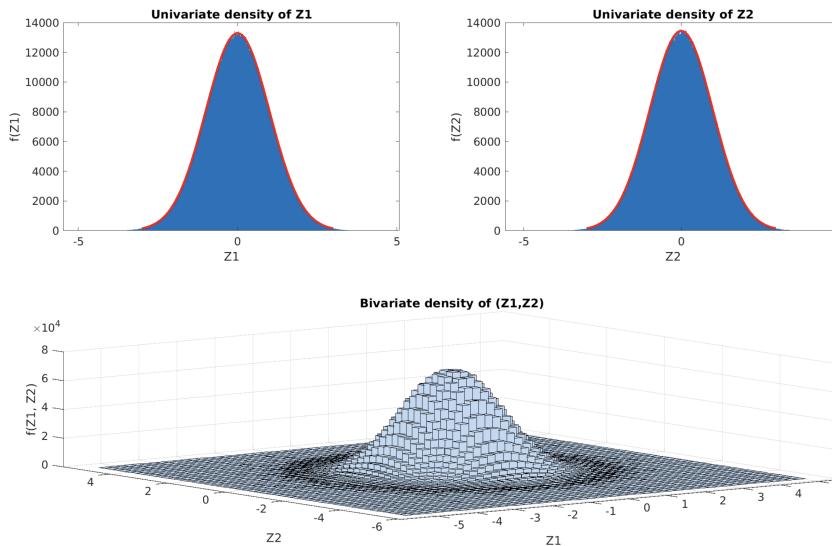


Figure 1.5: Standard Bivariate Density and Random Variables Generated with the Box-Mueller

Marsaglia-Bray Algorithm

The Marsaglia and Bray [27], illustrated in Algorithm 5, avoid the calculation of the trigonometric functions by implementing an acceptance-rejection method to sample points uniformly in the unit disc and then transforming these points into a couple of standard normal variables.

Algorithm 5: Marsaglia-Bray

```

N = 105; r1 = rand(N, 1); r2 = rand(N, 1); I = ones(N, 1);
U1 = (2*r1)-I; U2 = (2*r2)-I;
X = U1.2 + U2.2;
X1 = zeros(N,1) ;
for i = 1:N
    if X (i, 1) != I(i,1)
        X1(i,1) = X(i,1);
    end
end
R = sqrt(- 2*log(X1));
C1 = U1./sqrt(X1);
C2 = U2 ./sqrt(X1);
RE = R(R ~= inf);
C1 = C1(C1 ~= -inf);
C2 = C2(C2 ~= -inf);
C1a = C1(C1 ~= inf);
C2a = C2(C2 ~= inf);
Z1 = RE.*C1a; Z2 = RE.*C2a;
```

The first passage is performed in order to get a bivariate uniformly distributed (U_1, U_2) over the square $[-1, 1] \times [-1, 1]$. In the second step, after calculating $X = U_1^2 + U_2^2$, we start an if cycle accepting only those pairs for which: $X \leq 1$ and producing points uniformly distributed over the disc of radius 1 centered in the origin. Conditioning on acceptance, X is uniformly distributed between 0 and 1, so that:

- (i) $-2 \log(X) \sim \mathcal{E}(\frac{1}{2})$;
- (ii) $(\frac{U_1}{\sqrt{X}}, \frac{U_2}{\sqrt{X}})$ are uniformly distributed variables on the unit circle and independent from $(X | X = x \leq 1)$ and correspond respectively to the random cosine and the sine over it.

Once we have selected only the values of our interest (such that $X \leq 1$) we calculate the Gaussian pair of univariate standard normal $(Z1, Z2)$ as we basically did in the Box Mueller algorithm.

The Gaussian vector

A vector $\mathbf{X} = (X_1, \dots, X_m)$ is Gaussian if any linear combination of its components has the Gaussian distribution. A Gaussian vector \mathbf{X} is characterized by its mean μ and its covariance matrix Σ , and we write $\mathbf{X} \sim \mathcal{N}(\mu, \Sigma)$.

Generally, Gaussian vectors are generated by affine transforms of independent standard Gaussian random variables $Z \sim \mathcal{N}(0, I_d)$:

Let m and n be two non-zero integers, let Z be a n -dimensional standard Gaussian vector with the distribution $N(0, I_n)$, and $\mu \in \mathbb{R}^m$ and P be a $m \times n$ matrix. Then: $X = \mu + PZ \sim \mathcal{N}(\mu, \Sigma = PP')$ is an m -dimensional Gaussian vector with mean μ and variance-covariance matrix Σ .

So, since the covariance matrix Σ — symmetric non-negative definite matrix of size m — can always be decomposed non uniquely in the form $\Sigma = PP'$, if we know P , we can simulate any Gaussian vector X using the previous algorithms to generate a standard gaussian R.V. Z and then apply the transformation.

To calculate P , we can use the Choleski algorithm, which provides a lower triangular matrix (with $n = m$), with a computational cost with respect to the dimension proportional to m^3 .

In some cases (as illustrated by Gobet in [16]), it is possible to speed up the Choleski algorithm. If the Σ is an $m \times m$ matrix of the form:

$$\Sigma = \begin{bmatrix} 1 & \rho & \dots & \rho \\ \rho & 1 & \dots & \rho \\ \dots & \dots & \dots & \dots \\ \rho & \rho & \dots & 1 \end{bmatrix}$$

with $0 \leq \rho \leq 1$ it is possible to reduce the computational cost to the order m , by choosing a $m \times (m + 1)$ matrix P like this:

$$P = \begin{bmatrix} \sqrt{\rho} & \sqrt{1-\rho} & 0 & \dots & \dots & 0 \\ \sqrt{\rho} & 0 & \sqrt{1-\rho} & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & 0 \\ \sqrt{\rho} & 0 & \dots & \dots & 0 & \sqrt{1-\rho} \end{bmatrix}$$

When the variables have the Gaussian distribution, it is natural to model the dependence using the covariance matrix. Sometimes the modeling of the dependence cannot be reduced to a correlation coefficient only. In these cases, the dependence can be modeled intrinsically without taking the marginal distributions into account, using the notion of the copula (see [16]).

Chapter 2

Modeling Option Prices: from Discrete Time to Continuous Time Pricing and First Simulations

2.1 Discrete Time Option Pricing

In general, derivatives are contracts whose value depends on one or more primitive underlying asset (stocks, utilities, currencies...). In particular we will focus on options which give the right to buy (call) or sell (put) the underlying stock (S) at a certain "strike price" (K) and at a fixed future date (T) called "maturity".

What fluctuates is the price S_t . If we are in discrete time ($t \in \mathbb{N}$), we will see it as a discrete time stochastic process, a sequence of realizations of random variables. If we are in continuous time ($t \in \mathbb{R}$), we will see it as a continuous time stochastic process, with a "stochastic differential equation" describing its dynamics over time. Depending on whether it is only the price or also the interest rate, the volatility or other parameters that varies over time, we can differentiate the various classes of models (we will see in depth the classic B & S model but also the SABR model in chapter 3).

The payoff for a European Call (Put) is: $C_T = (S_T - K)^+$ ($P_T = (K - S_T)^+$); this is what is paid at maturity to the investor and represent a deterministic function of the final price.

There are also other "exotic options" which deviate from the classical European ones or that add some other condition to the payoff such as:

- Asian Options: highly used for high volatile markets (Ex. utilities). The payoff depends on different types of average calculated on the price observed at fixed discrete time intervals of the underlying. The most common is the arithmetic average: $X_t = (A_t - k)^+$ where: $A_t = \frac{1}{t} \int_0^t S_\mu d\mu$, but also other averages are used like a log-average or quadratic average (we will see an example in section 2.3).

- Options with Barrier: options whose payoff becomes null if the price of the underlying goes over or under a certain threshold before T (we will see them in

section 3.3).

These kind of contracts arise important problems for researchers and practitioners:

(i) Valuation or pricing: what is a fair price for these contracts? First of all we need a model for S (the approach is to model S). Then, the pricing problem can be solved through numerical methods like the Monte Carlo or finite differences methods;

(ii) Calibration of the unknown parameters of the model. The statistical theory (analyzing the historical market data and time series to get the rate r , the volatility ecc...) and the "implicit methods" (analyzing existing derivatives in the market to derive the parameters of a model) are the most common in practice for the estimation.

(iii) Hedging (the approach comes from Economic Principles/Rules): for example, how can the bank invest his money in order to cover the derivative contract at maturity? This is governed by economical laws; an hedging strategy will help us in deriving an equation to price derivatives under some assumptions.

(iv) Early Exercise: what is the optimal moment to exercise this right? In discrete time this doesn't represent a big problem, but in continuous time, stochastic optimization methods are required (we will treat this topic only in discrete time).

We will focus mainly on pricing and we will use some hedging framework to construct our model. The main references are Pascucci [31] and Bjork [2].

2.1.1 A Discrete Time Market Model

The Market Model

We consider a market model in which transactions take place only in discrete times: $t_k = \frac{T}{N}k$. The market consists of:

- A riskless bond B which has a deterministic dynamic of this type (r_n is the risk free rate at time n):

$$\begin{cases} B_0 = 1 \\ B_t = (1 + r_t)B_{t-1} \end{cases}$$

- d risky assets $= (S^1, \dots, S^d)$ that are stochastic processes $S_{t \in \mathbb{N}}(\omega)$, a sequence of random variables defined on a probability space (Ω, \mathcal{F}, P) which follows stochastic dynamic of this type:

$$\begin{cases} S_0^i \in \mathbb{R}^+ \\ S_t^i = S_{t-1}^i(1 + \mu_t^i), \quad t = 1, \dots, T \end{cases}$$

Where μ^i is the return from the i -th risky asset. Fixed $\omega \in \Omega$ and letting vary t , we observe the trajectory of the price, the single realization of the stochastic process. We will see how to simulate these trajectories in the next sections.

Value of Time

In this thesis we will initially use a discrete compounding law to deal with discrete time market models. Later on, for simplicity we will use a continuously compounding law to determine the value of the option and of the assets over time:

$$B_T = B_t e^{r(T-t)}$$

The risk free rate $r_t = r$ is assumed to be a constant (it could be also modelled as a stochastic process itself).

Filtration

Normally the probability space is defined with a filtration $\mathcal{F}_{t \in \mathbb{N}} = \sigma(S_0, \dots, S_t)$, a family (sequence) of sigma algebras which increase its size in time since the next one comprehend the previous one: $\mathcal{F}_t \supseteq \mathcal{F}_{t+1} \dots$. In an economic and financial context it can represent the information flow available at time t (market data, news...). We say that the filtration generated by the sequence of prices $S_{t \in \mathbb{N}}(\omega)$ is $\mathcal{F}_{t \in \mathbb{N}}^S = \sigma(S_0, \dots, S_t) = \{S_k \in H \mid 0 \leq k \leq t, H \in \mathcal{B}\}$, where \mathcal{B} is a Borel set.

Martingale properties

A Martingale measure of probability on (Ω, \mathcal{F}) for a certain market model, with numeraire B and a time interval $[0, T]$ is such that:

A) \mathbb{Q} is equivalent to \mathbb{P} ($Q \sim P$) such that: $Q(A) = 0 \Leftrightarrow P(A) = 0$ and $Q(B) = 1 \Leftrightarrow P(A) = 1$

B) The price (or prices) process is modelled as a Martingale in a measure Q ; $S_{t \in \mathbb{N}}$ is a stochastic process on a probability space $(\Omega, \mathcal{F}, P, \mathcal{F}_{t \in \mathbb{N}})$ such that:

- The best forecast for the price of tomorrow is the price today: $S_t = E^Q[S_{t+1} | \mathcal{F}_{t \in \mathbb{N}}]$
- It is constant in mean over time: $S_t = E^Q[S_{t+1} | \mathcal{F}_t] = E^Q[E^Q[S_{t+2} | \mathcal{F}_{t+1}] | \mathcal{F}_t] = E^Q[S_{t+2} | \mathcal{F}_t] \dots$

Investment Strategy

We define a strategy or portfolio as a stochastic process in \mathbb{R}^{d+1} :

$$(\alpha, \beta) = (\alpha_t^1, \dots, \alpha_t^d, \beta_t)_{t=1, \dots, N}$$

The value of the portfolio composed by d risky assets and of a risk-free asset will be:

$$V_t^{\alpha, \beta} = C_t = \sum_{i=1}^d \alpha_t^i S_t^i + \beta_t B_t \quad (2.1)$$

With initial value:

$$V_0^{\alpha, \beta} = C_0 = \sum_{i=1}^d \alpha_1^i S_0^i + \beta_1 B_0 \quad (2.2)$$

Now, for simplicity, we assume only one risky asset in the market and a risk free asset B. We make two hypothesis on the admissible strategies to build our model ([31]):

- i) (α_t, β_t) must be "predictable processes": they must be $\mathcal{F}_{t-1} = \sigma(S_0, S_1, \dots, S_{t-1})$ measurable (known at an instant before). So the strategies can be expressed as a deterministic function of the previous prices till the " $t - 1$ " moment.
- ii) The strategy must be "self-financing": at time t the wealth at our disposal is: $V_t^{\alpha, \beta} = \alpha_t S_t + \beta_t B_t$. The strategy is self financing if we can re-balance our portfolio in $t+1$ ($\forall t \in \mathbb{N}$) with a new strategy $(\alpha_{t+1}, \beta_{t+1})$ in such a way that we do not modify the overall value at time t of the portfolio:

$$V_t^{\alpha, \beta} = \alpha_t S_t + \beta_t B_t = \alpha_{t+1} S_t + \beta_{t+1} B_t \quad \forall t \in \mathbb{N} \quad (2.3)$$

In time " $t + 1$ ", the variation in the overall value will be driven only by the variation of the prices of the underlying and not by the fact that we have injected or withdrawn funds:

$$V_{t+1}^{\alpha, \beta} - V_t^{\alpha, \beta} = \alpha_{t+1}(S_{t+1} - S_t) + \beta_{t+1}(B_{t+1} - B_t) \quad \forall t \in \mathbb{N}$$

The total variation in the value of the option will be the sum of the single variation in all the discrete instants of time:

$$V_T^{\alpha, \beta} - V_0^{\alpha, \beta} = \sum_{j=0}^T \alpha_j (S_j - S_{j-1}) + \beta_j (B_j - B_{j-1}) \quad (2.4)$$

If we express with $\tilde{S}_t = \frac{S_t}{B_t}$, the discounted prices, we can express the variation of the portfolio as:

$$\tilde{V}_{t+1}^{\alpha, \beta} - \tilde{V}_t^{\alpha, \beta} = \alpha_{t+1}(\tilde{S}_{t+1} - \tilde{S}_t) + \beta_{t+1}\left(\frac{B_{t+1}}{B_{t+1}} - \frac{B_t}{B_t}\right) = \alpha_{t+1}(\tilde{S}_{t+1} - \tilde{S}_t) \quad \forall t \in \mathbb{N}$$

Such that the discounted value of the self financing portfolio is:

$$\tilde{V}_T^{\alpha, \beta} = V_0^{\alpha, \beta} + \sum_{j=0}^T \alpha_j (\tilde{S}_j - \tilde{S}_{j-1}) \quad (2.5)$$

First fundamental Theorem of Asset Pricing: Arbitrage-Free (correct) Market Model

For a fixed observation date t and maturity T, a market with risk-free asset B is free from arbitrages (for a fixed time horizon) if and only if there exists at least one probability measure $Q = Q_{t,T}$ such that the price at time t of any traded security delivered at T is given by the expected payoff under Q discounted by B. Such a measure is called "risk neutral".

Second fundamental theorem of Asset Pricing: Complete Market Model

For a fixed observation date t and maturity T , an arbitrage-free market with risk free asset B is complete if and only if there exists a unique probability measure Q such that at time t , the value of any traded security delivered at T is given by the expected payoff under Q discounted by B .

$$E^Q\left[\frac{X_T}{B^T}\right] = X_t \quad (2.6)$$

The discounted price of an option is a martingale

Remembering the definition of "martingale"; we assume that the discounted prices $\tilde{S}_t = S_t * e^{-r(T-t)}$ are martingales in a measure Q :

$$\tilde{S}_t = E^Q[\tilde{S}_{t+1} | \mathcal{F}_t]$$

If this is true, then also the $\tilde{V}_t \forall (\alpha, \beta) \in \mathcal{A}$ (family of admissible strategies: predictable and self-financing) is a martingale too.

Proof

Since \tilde{S}_t is a martingale:

$$\begin{aligned} E^Q[\tilde{V}_{t+1} - \tilde{V}_t | \mathcal{F}_t] &= E^Q[\alpha_{t+1}(\tilde{S}_{t+1} - \tilde{S}_t) | \mathcal{F}_t] \\ &= \alpha_{t+1}(E^Q[\tilde{S}_{t+1} | \mathcal{F}_t] - E^Q[\tilde{S}_t | \mathcal{F}_t]) \\ &= 0 \end{aligned}$$

This means that also \tilde{V}_t is a martingale since:

$$\begin{aligned} E^Q[\tilde{V}_{t+1} - \tilde{V}_t | \mathcal{F}_t] &= E^Q[\tilde{V}_{t+1} | \mathcal{F}_t] - E^Q[\tilde{V}_t | \mathcal{F}_t] \\ &= E^Q[\tilde{V}_{t+1} | \mathcal{F}_t] - V_t = 0 \\ \Rightarrow E^Q[\tilde{V}_{t+1} | \mathcal{F}_t] &= V_t \end{aligned}$$

Arbitrage

It is an investment which produces a profit without risk. Using the terminology introduced before, it is an admissible strategy (α, β) (in the real measure p) such that:

- i) $V_0^{\alpha, \beta} = 0$;
- ii) $V_T^{\alpha, \beta} \geq 0$;
- iii) $P(V_T^{\alpha, \beta} > 0) > 0$.

If in our model with arbitrages exists a martingale measure Q , then for all $(\alpha, \beta) \in \mathcal{A}$:

$$E^Q[\tilde{V}_T | \mathcal{F}_T] = V_0$$

Since our self financing portfolio is crafted as an arbitrage portfolio by buying and selling short S and B, we have that: $V_0 = 0$. Then since the arbitrage should have a positive expected value (from the (iii) condition), but an initial value which is zero, this is clearly an absurd (in contradiction with \tilde{V}_t being a martingale).

So we should assume a model free from arbitrage.

Non arbitrage principle

In a discrete time model two admissible strategies $(\alpha, \beta)(\alpha', \beta')$ are defined such that their final value are \mathbb{P} -a.s. equal: $V_T^{(\alpha, \beta)} = V_T^{(\alpha', \beta')}$; if a \mathbb{Q} measure martingale exists, since \mathbb{P} and \mathbb{Q} are equivalent, then they are also \mathbb{Q} -a.s equal.

We can express the non arbitrage principle with the fact that if the values of the two portfolios are \mathbb{Q} -a.s. equal at time T, then they must be \mathbb{Q} -a.s. equal at any previous time $t < T$, otherwise arbitrages would be possible (so called "free lunch"): $V_T = V'_T \Rightarrow V_t = V'_t, \forall t \leq T$.

We observe that if $V_T = V'_T$ \mathbb{Q} -a.s., then also their expected values must be equal: $E^Q[V_T] = E^Q[V'_T]$. Since V and V' are martingales (given a filtration \mathcal{F}_t) or more heuristically by the second FTAP presented before their prices at any time $t < T$ are calculated as expected values discount by B, under a unique measure \mathbb{Q} , otherwise arbitrages would be possible; so if $V_T = V'_T$ then also the discounted value of these portfolio (which can be seen as prices of a derivative with payoff V_T) must be equal:

$$\tilde{V}_t = E^Q[\tilde{V}_T] = E^Q[\tilde{V}'_T] = \tilde{V}'_t \quad \forall t < T$$

All options must be replicable in a correct and complete market

It can be proved that in a complete and correct market model all the derivatives' payoffs must be replicable, which means that can be created by already existing assets a portfolio which guarantees the same payoff: $X_T = V_T^{\alpha, \beta}$

In an incomplete market all the derivatives become not replicable.

If the model is both correct (free from arbitrage opportunities) and complete, for the second FTAP there will be a unique risk neutral probability measure \mathbb{Q} and for every derivative there will be a replicating strategy for its payoff. So for the non-arbitrage principle there exists a risk neutral price (expected value of the discounted payoff) which coincides with the initial value of a self-financing (replicating) portfolio:

$$X_0 = E^Q[\tilde{X}_T] = E^Q[\tilde{V}_T^{\alpha, \beta}] = V_0^{\alpha, \beta}$$

Put-Call Parity

From the "non-arbitrage principle", we can derive also the "Put-Call Parity". If we calculate the price of a call, for example with a Monte Carlo method, we can derive the price of the put by simply applying this equation and vice versa.

If we define:

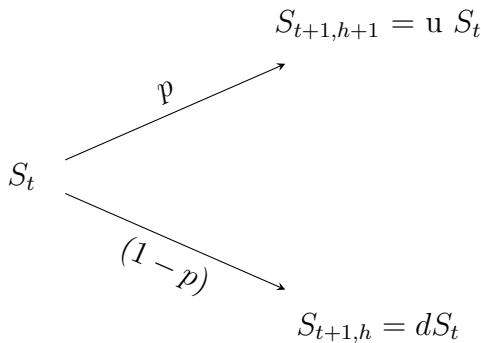
$$\begin{cases} X_t = S_t + P_t \\ X_T = S_T - \text{Max}\{K - S_T, 0\} \Rightarrow \text{Max}\{S_T, K\} = X_T \end{cases}$$

$$\begin{cases} Y_t = C_t + Ke^{-r(T-t)} \\ Y_T = \max\{S_T - K, 0\} + K \Rightarrow \text{Max}\{S_t, K\} = X_T \end{cases}$$

$$\begin{aligned} \text{So: } X_T &= Y_T \Rightarrow X_t = Y_t \quad \forall t \leq T \\ &\Rightarrow S_t + P_t = C_t + Ke^{-r(T-t)} \end{aligned}$$

2.1.2 The Binomial Model

The evolution of the prices of the risky asset between t and $t+1$ in a binomial model is described by the following scheme:



where:

- $S_{t+1} = (1 + \mu_{t+1})S_t$ and the return is a random variable with bernoulli distribution: $1 + \mu_{t+1} \sim p\delta_u + (1 - p)\delta_d$ with $0 < d < 1 + r < u$.
- "u" is the upward change if the price goes up and "d" is the downward change in the price if the price drops.
- "h" is the number of upward changes in the price till a certain time step.

The bank account evolution is $B_{t+1} = (1 + r)B_t$.

If an EMM Q exists then $S_t > 0$ is \mathcal{F}_t measurable and we are able to derive the constant measure "q" uniquely:

$$\begin{aligned} \tilde{S}_t &= E^Q[\tilde{S}_{t+1} | \mathcal{F}_t] \\ \frac{S_t}{B_t} &= E^Q\left[\frac{S_t(1 + \mu_{t+1})}{B_{t+1}} | \mathcal{F}_t\right] \\ \frac{S_t}{B_t} &= \frac{S_t}{B_t(1 + r)} E^Q[(1 + \mu_{t+1}) | \mathcal{F}_t] \\ 1 + r &= E^Q[(1 + \mu_{t+1}) | \mathcal{F}_t] \\ 1 + r &= u(q_t) + d(1 - q_t) \end{aligned}$$

where: $q_t = Q(1 + \mu_{t+1} = u | \mathcal{F}_t)$ is the probability in the martingale measure conditioned to \mathcal{F}_t .

Solving for q_t , we have that:

$$q_t = \frac{1 + r - d}{u - d} \quad (2.7)$$

Since r , u , d are constants, then $q_t = Q(1 + \mu_{t+1} = u | \mathcal{F}_t) = q$ so the probability of an upward change in the price q :

- it doesn't depend from t ;
- it doesn't depend from ω the event;

So, it can be proved that the returns μ_{t+1} and \mathcal{F}_t are independent in Q , such that the conditioned probability q is equal to the unconditioned probability. $q_t = Q(1 + \mu_{t+1} = u | \mathcal{F}_t) = Q(1 + \mu_{t+1} = u) = q$.

Under a martingale measure, the returns must be independent, even if in the real probability are dependent. So we have uniquely identified the measure.

If this is not true a simple arbitrage could be implemented.

The price of the risky asset at time 't' in the risk neutral measure should be equal to:

$$\tilde{S}_t = \frac{(qS_t u + (1 - q)S_t d)}{1 + r} = \frac{(qS_{t+1,h+1} + (1 - q)S_{t+1,h})}{1 + r} \quad (2.8)$$

2.1.3 European Option Pricing

Risk neutral price

So we have defined a discrete binomial model free from arbitrage (complete and correct) on a probability space $(\Omega, \mathcal{F}, P, (\mathcal{F}_t)_{t \in \mathbb{N}})$, in which a Q martingale measure exists (objective for all the market participants and correspondent to the implied probability in the prices).

The European call option is defined as the random variable "payoff" X which depends only on the final value of this contract: $X_T = (S_T - K)^+$.

If the payoff is replicable by an admissible strategy and exists a martingale measure Q , the risk neutral price of the derivative (expected value of the discounted payoff under a measure Q) will be:

$$E^Q\left[\frac{X}{B_T}\right] = E^Q[\tilde{V}_T^{\alpha, \beta}] = V_0^{\alpha, \beta}$$

Which is equal to the initial value of the strategy (α, β) .

In an arbitrage free and complete market (in which every financial instrument is replicable) the arbitrage price and the risk neutral price coincide: they are determined by the quoted price X_0 observable in the market. This arbitrage price does not depend on the subjective estimation of the probability p of the event of a rise or fall of the underlying. So normally we can compute the price of an option with its risk neutral price as an expected value of the payoff at T under the risk neutral

probability q and we know that this is a rational price, since it coincides with the initial value of a replicating portfolio/strategy (arbitrage-free price). In this model we have already solved the hedging problem: we can find the strategy which allow us to perfectly replicate the option payoff (no gains).

So from the second FTAP in our binomial model the price is defined as follows (if $\exists! Q$ mg), then the price of a EU call option will be:

$$\begin{aligned} V_0 &= E^Q\left[\frac{1}{(1+r)^T}(S_T - K)^+\right] = \frac{1}{(1+r)^T}E^Q[(S_T - K)^+] \\ &= \frac{1}{(1+r)^T} \sum_{h=0}^T (u^h d^{T-h} S_0 - K)^+ Q(S_T = u^h d^{T-h} S_0) \\ &= \frac{1}{(1+r)^T} \sum_{h=0}^T (u^h d^{T-h} S_0 - K)^+ \binom{T}{h} q^h (1-q)^{T-h} \end{aligned}$$

With: $q = \frac{1+r-d}{u-d}$

This price expresses the value that a risk neutral investor assigns to the risky asset C: the future discounted expected profit from the option.

Price of the European derivative

We have two equal methods to calculate V_0 and all the prices of the binomial tree for an option given S_0 , K , r , σ and maturity T . Once we are given u , d , which can be calculated thanks to a simple first order Taylor approximation (see formulas in the figure below), we have the risk neutral martingale measure q , from the formula 2.7. Suppose we want to calculate a Put European option in a Binomial model. The first two steps are in common for both methods:

- 1) we calculate the binomial tree for the undelying price S_t with the following binomial scheme described in section 2.1.2
- 2) we calculate the final payoff for a put option (h is the number of increasing price steps) in the various scenarios:

$$V_T = (K - u^h d^{T-h} S_0)^+ \quad (2.9)$$

Martingale method

Since we are in a binomial model, the price of our option will be given by the discounted value of the two possible states of the world which can exist one step forward, as in formula 2.8. In our case, the price of the call option at each time step can be calculated as follows:

$$V_{t-1,h} = \frac{q \cdot V_{t,h+1} + (1-q) \cdot V_{t,h}}{B_t} \quad \text{dove } n, h = 0, 1, 2, \dots \quad (2.10)$$

By using this formula iteratively, starting from $t = T$ (in which V_T is known from step 2) to $t = 0$, it is possible to get easily V_0 , as shown in the example below

implemented in Excel (fig. 2.1) for a put option with fixed risk free rate, K and S_0 , a continuously compounded regime for the bond and a maturity $T=1$ divided in 5 time steps.

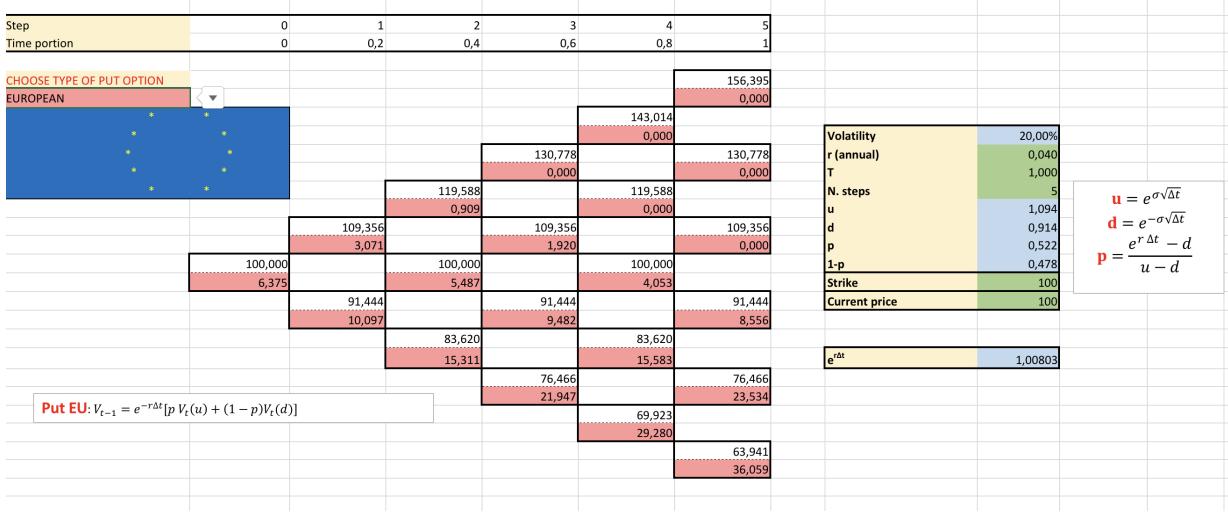


Figure 2.1: Simulation of 5 step binomial model to price a EU Put option

Self-financing Condition Method (Non arbitrage price)

An alternative approach is based on deriving the value of the option from the no arbitrage condition presented before. In order to price correctly a financial instrument it is sufficient to determine an investment strategy with the same final value (payoff). We consider an investment, which consists in holding a number α of shares of the risky assets and a number of β of bonds. The value of the portfolio composed by one risky assets and one risk-free asset will be:

$$V_t^{\alpha, \beta} = C_t = \alpha_t S_t + \beta_t B_t \quad (2.11)$$

We can calculate the strategy (α, β) function of S_{t-1} by solving the following linear system for each time step and going iteratively backward:

$$\begin{cases} \alpha_{t,h} S_{t,h+1} + \beta_{t,h} B_t = V_{t,h+1} \\ \alpha_{t,h} S_{t,h} + \beta_{t,h} B_t = V_{t,h} \end{cases}$$

for which the solution for alpha and beta is:

$$\hat{\alpha}_{t,h} = \frac{V_{t,h+1} - V_{t,h}}{S_{t,h+1} - S_{t,h}} = \frac{V_{t,h+1} - V_{t,h}}{S_{t,h}(u - d)} \quad (2.12)$$

$$\hat{\beta}_{t,h} = \frac{V_{t,h+1} - \tilde{\alpha}_{t,h} S_{t,h+1}}{B_t} = B_t \frac{V_{t,h} S_{t,h+1} - V_{t,h+1} S_{t,h}}{S_{t,h+1} - S_{t,h}} \quad (2.13)$$

Once we have found the process (α, β) at maturity, we can use the self-financing condition backward to get the value of the option in all the previous time steps:

$$V_{t-1} = \hat{\alpha}_t S_{t-1} + \hat{\beta}_t B_{t-1} \quad (2.14)$$

Following these steps iteratively will drive us simply to V_0 .

2.1.4 American Option Pricing

The American Put option payoff is defined as a stochastic process depending from all the trajectory of the price and is equal to: $X_{t \in \mathbb{N}} = (K - S_t)^+$ (it depends from the time of the exercise). In the binomial model the pricing becomes very easy since in the single time step we are dealing with European options (this is true only because we are in discrete time and not in continuous time); the price is given by calculating iteratively this quantity from $t = T-1$ to 0 (in T , $V_T = X_T$, calculated as in 2.9):

$$V_t = \max\{X_t, E_t = \frac{E_t^Q[V_{t+1} | \mathcal{F}_t]}{B_t}\} \quad (2.15)$$

where: X_t is the payoff at time t that we receive if we exercise the call and E_t is the discounted expected value of the option with "maturity" in the next period ($t+1$) (called "continuation value").

In case of exercise of the option, the contract is over since it is more convenient to exercise the option than remaining with the expected value of the contract. Here we present a simple implementation in excel with the same data as for the European Put option example before. We can observe how the price of the American option is always greater or equal to the one of the European one, since we have the opportunity of early exercise (here we would already use it in some scenarios at step 4; in these cases we should end our contract before the expiration). These higher potential payoffs increase the value of all the previous option values till the price at time 0:

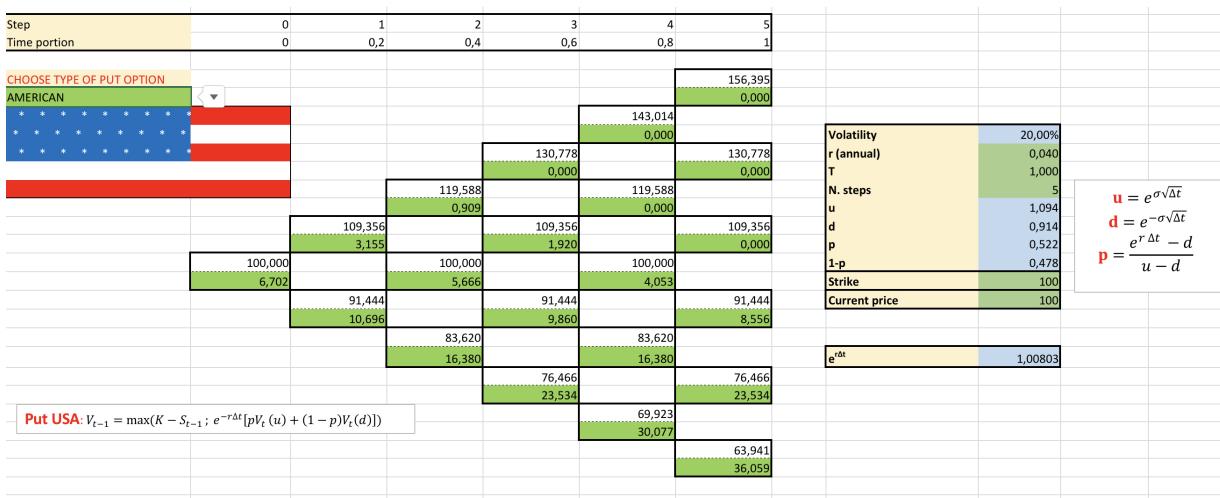


Figure 2.2: Simulation of 5 step binomial model to price an American Put option

2.2 Continuous Time Option Pricing: Basic Theory and Models

It is possible to prove that most of the results valid for a discrete time model, are valid also for a continuous time one, which better approximates the situation in the reality. Moreover, by sending the number of N time step to infinity, the price of a European option calculated with the binomial model will converge to the one calculated with a continuous time model like the one of Black and Scholes.

In this section we will provide in synthesis the theoretical framework and the main results obtained in their work by Merton [29] and Black & Scholes [3]. The approach followed will be a more rigorous and probabilistic one, in which the absence of arbitrage is a consequence of the self financing property and of the martingale property, not the starting point as in other heuristic approaches: if a strategy is adapted and self financing, then it cannot reasonably generate a risk free profit greater than the bond (i.e. there cannot be an arbitrage opportunity).

2.2.1 Basic Notions

Price of a stock as a Stochastic process in Continuous Time

A stochastic process on a probability space (ω, \mathcal{F}, P) is defined as:

$$S = S_t(\omega) : [0, T] \times \Omega \Rightarrow \mathbb{R} \quad (2.16)$$

$$(t, \omega) \Rightarrow S_t(\omega)$$

If we fix t: S_t is the price at time t and is a random variable on (ω, \mathcal{F}, P) .

If we fix ω (event): $t \Rightarrow S_t(\omega)$ is a trajectory of the stock price (a function of $[0, T] \Rightarrow \mathbb{R}$); to each ω we associate a different possible trajectory of the price in time.

Filtration and Adapted Processes

The filtration is an increasing family of sigma algebras $\mathcal{F}_{t \in [0, T]}$:

$$\mathcal{F}_t \subseteq \mathcal{F}_s \subseteq \mathcal{F}_T \quad \text{with } t \leq s \leq T$$

S is an "adapted process" to a filtration $\mathcal{F}_{t \in [0, T]}$ if $S_t \in m\mathcal{F}_t \quad \forall t \in \mathbb{R}$.

Brownian motion

We call Real Brownian motion a stochastic process $W = W_{t \geq 0}$ in \mathbb{R} defined on a filtered probability space $(\omega, \mathcal{F}, P, \mathcal{F}_{t \geq 0})$ which verifies the following properties:

- i) $W_0 = 0$;
- ii) W is an adapted stochastic process to a filtration $\mathcal{F}_{t \geq 0}$;

- iii) W is a continuous process (almost every trajectory is a continuous function);
- iv) The increment $W_t - W_s$ for $0 \leq s < t$ must be independent from the filtration of s (\mathcal{F}_s). A consequence of this is that the increments are independent from the previous increments, since they are independent from the past information (the filtration contains also other information so to be more precise, it is an even stronger condition than solely independence from previous past increments).
- v) $W_t - W_s \sim \mathcal{N}(0, t-s)$ for $0 \leq s < t$. In particular: $W_t \sim \mathcal{N}(0, t)$ (the variance of the brownian motion increase with time: $Var(W_t) = E[W_t^2] = t$; it is the distance squared of W_t from W_0).

vi) W is a martingale which means that is constant in mean (the unconditional mean is zero) and the best expectation for tomorrow's value, given the information till t ($\mathcal{F}_{t \in [0, T]}$), is the value at time t :

$$E[W_{T_1} | \mathcal{F}_t] = E[W_T | \mathcal{F}_t] = W_t \text{ with } 0 \leq t \leq T_1 \leq T.$$

Proof: By the Holder's inequality (A.3.3) we have that W is integrable:

$$E[|W_t|]^2 \leq E[W_t^2] = t$$

We notice that the expected value of W in a point in the future T is equal to the last observed value:

$$\begin{aligned} E[W_T | \mathcal{F}_t] &= E[W_T - W_t + W_t | \mathcal{F}_t] \\ &= E[W_T - W_t | \mathcal{F}_t] + E[W_t | \mathcal{F}_t] \\ &= W_t + E[W_T - W_t] \\ &= W_t \end{aligned}$$

Which follows from the fact that $W_T - W_t$ is independent from \mathcal{F}_t , the expected value of the Brownian increments is zero and that W_t is an \mathcal{F}_t -adapted process.

2.2.2 Price Models

Bachelier

First model for the price of a stock is from Bachelier (see [1]):

$$S_t = S_0(1 + \mu t) + \sigma W_t$$

In this framework, S_t is distributed normally $\sim \mathcal{N}(S_0(1 + \mu t), \sigma^2 t)$ and $\mu \in \mathbb{R}$. The first part is the drift part, the second part is the noise, the diffusion part which doesn't change the deterministic direction of the trajectory (the drift). This model is not desirable for our task since prices can take negative values and that is not possible in stock markets.

Geometric Brownian motion

The most reasonable choice has been to model stock prices with a Geometric Brownian motion ($S_t \sim \mathcal{GBM}(\mu, \sigma^2)$):

$$S_t = S_0 e^{(\mu - \frac{1}{2}\sigma^2)t + \sigma W_t} \quad \text{with } \mu, \sigma \in \mathbb{R}, \sigma > 0 \quad (2.17)$$

This formula follows from the application of the Ito formula (see A.5) to the following stochastic differential equation:

$$dS_t = \mu S_t dt + \sigma S_t dW_t \quad \text{with } \mu, \sigma \in \mathbb{R}, \sigma > 0 \quad (2.18)$$

In this framework, S_t is distributed log-normally, since $\frac{S_t}{S_0} \sim \mathcal{LN}([\mu - \frac{1}{2}\sigma^2]t, \sigma^2)$ and $\mu \in \mathbb{R}$.

Its expected value is:

$$\begin{aligned} E[S_t] &= S_0 e^{(\mu - \frac{1}{2}\sigma^2)t} E[e^{\sigma W_t}] \\ &= S_0 e^{(\mu - \frac{1}{2}\sigma^2)t} \int_{\mathbb{R}} e^{\sigma x} \phi(t, x) dx \\ &= S_0 e^{(\mu - \frac{1}{2}\sigma^2)t} \varphi_{W_t}(\eta) \\ &= S_0 e^{(\mu - \frac{1}{2}\sigma^2)t} \varphi_{W_t}(\frac{\sigma}{i}) \\ &= S_0 e^{(\mu - \frac{1}{2}\sigma^2)t} e^{-\frac{t(\frac{\sigma}{i})^2}{2}} \\ &= S_0 e^{(\mu - \frac{1}{2}\sigma^2)t + \frac{\sigma^2}{2}t} \\ &= S_0 e^{\mu t} \end{aligned} \quad (2.19)$$

with $\phi(t, x) = \frac{1}{\sqrt{2\pi t}} e^{-\frac{x^2}{2t}}$, $W_t \sim \mathcal{N}(0, t)$, $\varphi_{W_t}(\eta)$ characteristic function of W_t (see appendix A.4) and $\eta i = \sigma$.

A martingale Brownian exponential can be obtained by standardizing the expected value of the exponential of the sigma times the Brownian motion $\forall \sigma \geq 0$; as we have seen before $E[e^{\sigma W_t}] = e^{\frac{\sigma^2}{2}t}$ then:

$$E[e^{\sigma W_t - \frac{\sigma^2}{2}t}] = 1$$

Recalling the properties of a martingale in section (2.1.1) X is a martingale if: $X_t = E[X_T | \mathcal{F}_t]$ with $t \leq T$.

So $X_t = e^{\sigma W_t - \frac{\sigma^2}{2}t}$ is a martingale since:

$$\begin{aligned} E[X_T | \mathcal{F}_t] &= X_t \\ E[e^{\sigma W_T - \frac{\sigma^2}{2}T} | \mathcal{F}_t] &= e^{\sigma W_t - \frac{\sigma^2}{2}t} \\ E[e^{\sigma(W_T - W_t)} | \mathcal{F}_t] &= e^{\frac{\sigma^2}{2}(T-t)} \\ E[e^{\sigma(W_T - W_t)}] &= e^{\frac{\sigma^2}{2}(T-t)} \\ e^{\frac{\sigma^2}{2}(T-t)} &= e^{\frac{\sigma^2}{2}(T-t)} \end{aligned} \quad (2.20)$$

This is true since $W_T - W_t \sim \mathcal{N}(0, T - t)$ is independent from \mathcal{F}_t , and that W_t is an \mathcal{F}_t -adapted process.

We can see that if $r = \mu$ then the discounted price of a stock is a martingale too, since:

$$\begin{aligned}\tilde{S}_t &= e^{-rt} S_t \\ &= e^{-rt} S_0 e^{(\mu - \frac{1}{2}\sigma^2)t + \sigma W_t} \\ &= S_0 e^{(r - \frac{\sigma^2}{2})t + \sigma W_t} \\ &= S_0 X_t\end{aligned}$$

So $\mu = r$, becomes the condition in order to have a martingale as a discounted price:

$$\begin{aligned}\tilde{S}_t &= e^{-rt} S_0 e^{(\mu - \frac{1}{2}\sigma^2)t + \sigma W_t} \\ &= S_0 e^{(r - \frac{\sigma^2}{2})t + \sigma W_t}\end{aligned}$$

Here the exponent is distributed as: $(r - \frac{\sigma^2}{2})t + \sigma W_t \sim \mathcal{N}((r - \frac{\sigma^2}{2})t, \sigma^2 t)$.

2.2.3 Deriving BS differential equation

Assumptions

In deriving our formula for the value of an option in terms of the price of the stock, we will assume the "ideal conditions" in the market of Black and Scholes [3]:

- a) The short-term interest rate is known and is constant through time.
- b) The stock price follows a random walk in continuous time with a variance rate proportional to the square of the stock price. Thus the distribution of possible stock prices at the end of any finite interval is log-normal. The variance rate of the return on the stock is constant.
- c) The stock pays no dividends or other distributions.
- d) The option is "European," that is, it can only be exercised at maturity.
- e) There are no transaction costs in buying or selling the stock or the option.
- f) It is possible to borrow any fraction of the price of a security to buy it or to hold it, at the short-term interest rate.
- g) There are no penalties to short selling. A seller who does not own a security will simply accept the price of the security from a buyer, and will agree to settle with the buyer on some future date by paying him an amount equal to the price of the security on that date.

The model

The model is composed by one risky asset S and by one risk-free asset r :

$$\begin{cases} dS_t = \mu S_t dt + \sigma S_t dW_t \text{ with: } \mu, \sigma \in \mathbb{R}, \sigma > 0, S(0) = S_0 \\ dB_t = r B_t dt \text{ with: } B(0) = B_0 \end{cases}$$

Where μ is the risky return and σ is the volatility parameter of the risky asset.
We know that the solutions of these two differential equations are:

$$\begin{cases} S_t = S_0 e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t} \text{ with: } \mu, \sigma \in \mathbb{R}, \sigma > 0, S(0) = S_0 \\ B_t = B_0 e^{rt} \text{ with: } B(0) = B_0 \end{cases}$$

Where the first one follows from the Ito's Lemma application (see A.5.2) and the second is the solution to a first order homogeneous differential equation.

Admissible and Markovian strategies

An investment strategy is composed by two stochastic processes (α_t, β_t) on the probability space $(\Omega, \mathcal{F}, \mathbb{P}, (\mathcal{F}_t)_{t \geq 0})$ on which also a Brownian motion is defined. α_t is the number of risky assets, β_t is the number of non assets.

The value of the strategy will be:

$$V_t = \alpha_t S_t + \beta_t B_t$$

In general we assume "admissible strategies" (see also 2.1.1):

1) α_t and β_t are "adapted processes" or \mathcal{F}_t measurable, which means that depends on everything has happened till time t .

2) The strategy must be "self-financing":

It is the continuous time version of the condition in discrete time:

$$V_t - V_{t-1} = \alpha_t (S_t - S_{t-1}) + \beta_t (B_t - B_{t-1})$$

The strategy is self-financing if the stochastic differential of V_t depends only on the differential of S_t and B_t (not on $d\alpha$, $d\beta_t$):

$$dV_t = \alpha_t dS_t + \beta_t dB_t$$

A strategy is "Markovian" if we can express α_t and β_t (stochastic processes) as functions (regular functions, such that we can use ITO) of t and S_t : $\alpha_t = \alpha(t, S_t)$ $\beta_t = \beta(t, S_t)$

For a Markovian strategy, its value can be written as:

$$V_t = \alpha(t, S_t) S_t + \beta(t, S_t) B_t = F(t, S_t)$$

Where V_t is the stochastic process and $F(t, S_t)$ is the value function, which can be seen as a deterministic function, since V is function only of t and S_t (Markovian).

The self financing condition and the B & S differential equation

Let's rewrite the value function:

$$F(t, S_t) = \alpha(t, S_t) S_t + \beta(t, S_t) B_0 e^{rt} \text{ with } t \geq 0, x \in \mathbb{R}.$$

For a Markovian strategy, the self-financing condition can be written as:

$$dV_t = \alpha(t, S_t) dS_t + \beta(t, S_t) dB_t$$

If the Markovian strategy is self-financing we have two implications:

a) F solves the differential equation of Black and Scholes:

$$\partial_t F(t, S_t) + \partial_S F(t, S_t)S_t + \frac{1}{2}\partial_{SS}F(t, S_t)\sigma^2 S_t^2 = F(t, S_t)r \text{ With } t > 0, X > 0$$

$$\text{b)} \alpha(t, S_t) = \partial_S F(t, S_t) \text{ With } t > 0, X > 0$$

We prove these two statements below.

If we calculate the stochastic differential of $V_t = F(t, S_t)$ with Ito (A.5), we calculate the effect of a variation of S_t on the value of the portfolio F, remembering that $dS_t = \mu S_t dt + \sigma S_t dW_t$ we get:

$$\begin{aligned} dV_t &= dF(t, S_t) \stackrel{\text{ITO}}{=} \partial_t F(t, S_t)dt + \partial_S F(t, S_t)dS_t + \frac{1}{2}\partial_{SS}F(t, S_t)d\langle S_t \rangle \\ &= \partial_t F(t, S_t)dt + \partial_S F(t, S_t)(\mu S_t dt + \sigma S_t dW_t) + \frac{1}{2}\partial_{SS}F(t, S_t)\sigma^2 S_t^2 dt \\ &= [\partial_t F(t, S_t) + \partial_S F(t, S_t)\mu S_t + \frac{1}{2}\partial_{SS}F(t, S_t)\sigma^2 S_t^2]dt + \partial_S F(t, S_t)\sigma S_t dW_t \end{aligned}$$

Let's rewrite the self-financing condition by substituting the differential equations of the B & S model:

$$\begin{aligned} dV_t &= dF(t, S_t) = \alpha(t, S_t)dS_t + \beta(t, S_t)dB_t \\ &= \alpha(t, S_t)(\mu S_t dt + \sigma S_t dW_t) + \beta(t, S_t)(rB_t dt) \\ &= [\alpha(t, S_t)\mu S_t + \beta(t, S_t)rB_t]dt + \alpha(t, S_t)\sigma S_t dW_t \end{aligned}$$

So we impose that the differential equation of F developed with Ito and the differential equation derived from the self-financing condition coincide in the B & S model coincide (the drift and the diffusive part must be equal):

$$\begin{cases} \partial_t F(t, S_t) + \partial_S F(t, S_t)\mu S_t + \frac{1}{2}\partial_{SS}F(t, S_t)\sigma^2 S_t^2 = \alpha(t, S_t)\mu S_t + \beta(t, S_t)rB_t dt , (\mu, \sigma \in \mathbb{R}, \sigma > 0) \\ \sigma S_t(\partial_S F)(t, S_t) = \alpha(t, S_t)\sigma S_t \\ F(t, S_t) = \alpha(t, S_t)S_t + \beta(t, S_t)B_t \end{cases}$$

$$\begin{cases} \partial_t F(t, S_t) + \partial_S F(t, S_t)\mu S_t + \frac{1}{2}\partial_{SS}F(t, S_t)\sigma^2 S_t^2 = \partial_S F(t, S_t)\mu S_t + \beta(t, S_t)rB_t dt , (\mu, \sigma \in \mathbb{R}, \sigma > 0) \\ \partial_S F(t, S_t) = \alpha(t, S_t) \\ F(t, S_t) - \partial_S F(t, S_t)S_t = \beta(t, S_t)B_t \end{cases}$$

$$\begin{cases} \partial_t F(t, S_t) + \frac{1}{2}(\partial_{SS}F)(t, S_t)\sigma^2 S_t^2 = [F(t, S_t) - \partial_S F(t, S_t)S_t]r (\mu, \sigma \in \mathbb{R}, \sigma > 0) \\ \alpha(t, S_t) = \partial_S F(t, S_t) \\ F(t, S_t) - \partial_S F(t, S_t)S_t = \beta(t, S_t)B_t \end{cases}$$

$$\begin{cases} F(t, S_t)r = \partial_t F(t, S_t) + r\partial_S F(t, S_t)S_t + \frac{1}{2}(\partial_{SS}F)(t, S_t)\sigma^2 S_t^2 (\mu, \sigma \in \mathbb{R}, \sigma > 0) \\ \alpha(t, S_t) = \partial_S F(t, S_t) \\ \beta(t, S_t)B_t = F(t, S_t) - \partial_S F(t, S_t)S_t \end{cases}$$

The first equation obtained is the stochastic partial differential equation of B &

S: if the strategy is self-financing, then its value function satisfies this equation. It has not a unique solution, unless we associate a final condition to it.

Black And Scholes: Replication condition of the final payoff

Given a European call option of the form (with payoff) $X_T = \varphi(S_T) = (S_T - K)^+$

In a complete and correct market X is replicable, so there is an admissible Markovian strategy $(\alpha(t, S_t), \beta(t, S_t))$ such that the final value of this strategy is equal to the value of the payoff at time T: $V_T = X_T$.

Admissibility of the strategy and replicability of the final payoff finally brought us to this system of equations:

$$\begin{cases} L_{BS} = \partial_t F(t, S_t) + r \partial_S F(t, S_t) S_t + \frac{1}{2} (\partial_{SS} F)(t, S_t) \sigma^2 S_t^2 - r F(t, S_t) = 0 \\ \text{with: } \mu, \sigma \in \mathbb{R}, \sigma > 0, S_t > 0, 0 \leq t < T \\ F(T, S_T) = \varphi(S_T) \text{ with: } S_T > 0 \end{cases}$$

Where the function F is the function value $V_t = F(t, S_t)$ of an admissible strategy which replicates the derivative in time T. By the non-arbitrage principle we know that $V_T = X_T \rightarrow V_0 = X_0$ so the value of the self financing strategy will be exactly the value of the call option in 0:

$$V_0 = F(0, S_0)$$

The function F can be calculated solving a Cauchy problem in the variable S: we can lead back our equation to the heat equation through a change of variables $S_t = e^y$, whose solution has been found in the XIX century by Fourier. Indeed by solving the Cauchy problem of the Black-Scholes partial differential equation together with the boundary condition (final payoff) we can derive the well known Black-Scholes formulas for the pricing of a European derivative (see [3], [29]).

Sometimes, when the option is not European or has a more complicated payoff, these kind of problems don't have an analytical solution; however an approximated solution can be found by implementing numerical methods like finite differences methods (Ex. Euler backward scheme), where the final payoff will represent the terminal boundary condition.

We don't go into further details discussing the solution of the B & S equation.

Anyway we notice that the L_{BS} equation doesn't depends on μ , so the price of the derivative will not depend on any rate of return of the risky asset, but only on the risk free return, which is what we expected from the option value in order to be a martingale.

2.2.4 The Girsanov's Theorem

It explains the relation between changing the drift of an Ito process and changing the measure of probability. A detailed proof can be found in Bjork [2].

Let W be a Brownian motion on a probability space $(\Omega, \mathcal{F}, \mathbb{P}, (\mathcal{F}_t)_{t \geq 0})$.

Consider the process \tilde{W}_t obtained by adding a drift to the process W_t :

$$\tilde{W}_t = W_t + \int_0^t \lambda_s ds$$

where $\lambda \in L^2$ is an adapted process.

In differential notation is:

$$d\tilde{W}_t = dW_t + \lambda dt$$

The theorem states that it exists a measure of probability Q on the same space (Ω, \mathcal{F}) such that \tilde{W} is a Brownian motion in $(\Omega, \mathcal{F}, Q, (\mathcal{F}_t)_{t \geq 0})$. We are basically re-balancing the probabilities of the trajectories.

So W is a BM in \mathbb{P} iff \tilde{W} is a BM in Q .

What is the relation between the expected values in the two measures and how to pass from a measure to another?

Fixed T (maturity of the derivative) and $X \in m\mathcal{F}_T$. If we have the expected value in P of a payoff X :

$$E^Q[X] = E^P[XZ]$$

where Z is called the Radon - Nikodym of Q with respect to Q :

$$Z = \frac{dQ}{dP} = e^{-\int_0^T \lambda_s dW_s - \frac{1}{2} \int_0^T \lambda_s^2 ds}$$

This is called derivative since we need it for algebraically equate the two expected values:

$$E^Q[X] = \int_{\Omega} X dQ = \int_{\Omega} X \left(\frac{dQ}{dP} \right) dP = \int_{\Omega} X Z dP$$

2.2.5 Martingale Approach, Risk neutral pricing and Monte Carlo

In the real measure \mathbb{P} the B & S model is:

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

and \tilde{S}_t is not martingale (I don't expect that this is constant in mean), with $\mu > r$.

Let's rewrite the previous equation in another way:

$$dS_t = rS_t dt + \sigma S_t (dW_t + \frac{\mu-r}{\sigma} dt)$$

If we call $d\tilde{W}_t$ the new Brownian motion in which we added a drift $\lambda = \frac{\mu-r}{\sigma}$ we have that:

$$d\tilde{W}_t = dW_t + \frac{\mu-r}{\sigma} dt$$

λ is called Market price of risk (excess return on volatility).

$t = e^{-rt} S_t$ is a Q -MgSo applying the Girsanov's theorem, we have that it exists another measure Q in which $d\tilde{W}_t$ is a Brownian motion.

In Q , we have that:

$$dS_t = rS_t dt + \sigma S_t d\tilde{W}_t$$

In which S_t is a geometric Brownian motion.

In this framework Q is called martingale measure or risk neutral measure. We already proved in section 2.2.2 that the discounted value of the underlying price $\tilde{S}_t = e^{-rt} S_t$ is a Q -Mg; now we want to prove also that the discounted value of the self financing strategy (so by the non-arbitrage principle of the option replicated

too) at time t $\tilde{V}_t = e^{-rt} F(t, S_t)$ is a \mathbb{Q} -Mg. This result can be used to find the risk neutral price of the option, which is nothing else than an expected value that can be calculated using numerical methods like Monte Carlo.

Since the BM in \mathbb{Q} has exactly the same properties of the BM in \mathbb{P} , we can simplify the notation writing simply our risk neutral dynamic imposing $r = \mu$:

$$dS_t = rS_t dt + \sigma S_t dW_t$$

we know that its expected value for the equation 2.19 is:

$$E[S_t] = S_0 e^{rt}$$

If we define the discounted value of the price as $\tilde{S}_t = e^{-rt} S_t$ the expected value becomes:

$$E[\tilde{S}_t] = S_0$$

By the Ito formula, the stochastic differential of \tilde{S}_t is:

$$\begin{aligned} d\tilde{S}_t &= d(e^{-rt} S_t) = -re^{-rt} S_t dt + e^{-rt} dS_t \\ &= -re^{-rt} S_t dt + e^{-rt} (rS_t dt + \sigma S_t dW_t) \\ &= -r\tilde{S}_t dt + r\tilde{S}_t dt + \sigma \tilde{S}_t dW_t \\ &= \sigma \tilde{S}_t dW_t \end{aligned}$$

So we proved once again that \tilde{S} is a martingale (we obtained a null drift), if and only if we took r as the risky return.

Now we can apply the Ito formula in \mathbb{Q} to the discounted value of a strategy (with a certain parameter of the drift (r), such that \tilde{S} is a martingale, and $d\tilde{W}_t$ is a BM for girsanov):

$$d(e^{-rt} F(t, S_t)) = e^{-rt} L_{BS} F(t, S_t) dt + \dots d\tilde{W}_t$$

The B & S operator appears in the drift only applying Ito in the risk neutral measure!! If F is the value function of a self financing strategy which replicates a certain derivative, then the B & S operator must be $L_{BS} = 0$ (satisfying the B & S equation), which means that we put to zero the drift; this means that $\tilde{V}_t = e^{-rt} F(t, S_t)$ (discounted value of the value function of the strategy) is a martingale in \mathbb{Q} .

So a very easy and practical way in general to determine the pricing equation of a derivative is to apply Ito in the risk neutral measure and then, by doing so, the Black and Scholes operator will appear in the drift part; in order to satisfy the B & S equation and the self financing condition we put to zero the drift, getting possibly the value function of the derivative without doing all the passages made before to get to the B & S differential equation (section 2.2.3).

So, since V is a \mathbb{Q} -Martingale and the replication condition says that in a complete market exists a self financing strategy such $V_T = F(T, S_T) = \varphi(S_T)$, the value in 0 of the strategy (which means the price of the option replicated) is equal

to the expected value in \mathbb{Q} of the discounted value of the payoffs we get in T:

$$\begin{aligned} V_0 &= E^{\mathbb{Q}}[\tilde{V}_T] \\ &= e^{-rT} E^{\mathbb{Q}}[V_T] \\ &= e^{-rT} E^{\mathbb{Q}}[\varphi(S_T)] \end{aligned}$$

So V_0 gives us the risk neutral price of the derivative, given the payoff in T. If an analytical solution cannot be found to solve the differential equation expressing the dynamic of the option price, this expected value can be calculated with a Monte Carlo simulation, knowing that also in \mathbb{Q} , S is log normal (\tilde{W} in geometric Brownian motion \mathbb{Q}) and can be simulated. $E^{\mathbb{Q}}[\varphi(S_T)] \approx \frac{1}{M} \sum_{k=1}^M \varphi(S_T^{(k)})$

We just need to calculate the arithmetic mean of the payoffs of the option simulated, discounted at time 0. In the measure \mathbb{Q} , we need to take as drift parameter the risk free rate 'r', such that S and V are martingales:

$$dS_t = rS_t dt + \sigma S_t d\tilde{W}_t$$

From the point of view of the simulation, it doesn't change anything, except that we are taking r as drift parameter. So we need to simulate \tilde{W} as a normal: $\tilde{W} = \sqrt{t}Z$ with $Z \sim N(0, 1)$

In the case of a simple Geometric Brownian Motion, an explicit solution is known:
 $S_t = S_0 e^{\sigma \tilde{W}_t + (r - \frac{\sigma^2}{2})t}$

So we can discretize the trajectory of the underlying S explicitly from the solution of its stochastic differential equation (the calculators work only with discrete numbers):

$$S_{t+\Delta} = S_t e^{(r - \frac{1}{2}\sigma^2)\Delta_t + \sigma\sqrt{\Delta_t}z_t} \quad (2.21)$$

If an analytical solution is not known for the dynamic of the underlying stock price, a Euler Method can be adopted, an iterative method which discretize directly the stochastic differential equation:

$$S_{t+\Delta} = S_t + rS_t\Delta_t + \sigma S_t \Delta_{\tilde{W}} \quad (2.22)$$

where:

$$\Delta_{\tilde{W}} = \tilde{W}_{t+\Delta} - \tilde{W}_t = \sqrt{\Delta_t}z_{t+\Delta}$$

2.3 Monte Carlo Evaluation of a European Option with Asian Payoff

Let's consider an arbitrage free market in which there are only two titles whose dynamic in the risk neutral probability measure is:

$$dB_t = rB_t dt$$

$$dS_t = rS_t dt + \sigma S_t dW_t$$

- where $B(t)$ is the bank account and it is a deterministic function of t , yielding a constant risk free rate equal to r ;
- $W(t)$ is a standard Brownian motion, and $\mu = r$ is a constant determining the slope of the drift term.

In such a market we want to evaluate a European option with maturity $T = 1$, which has an "Asian" payoff:

$$\phi_T = (Q_T - K)^+$$

where K is the pre-determined strike price and

$$Q_T^2 = \frac{1}{T} \int_0^T S_u^2 du$$

In this section we will implement a Euler Monte Carlo Method for the evaluation of a Call option (the procedure for a put is exactly the same), having fixed some of the parameters and simulated the trajectories. We will also comment the convergence of the method by changing the number of simulations.

2.3.1 Simulation of the Trajectories: Euler and Exact Solution

Fixed some standard parameters:

$$- S_0 = 100 - K = 100 - \sigma = 0.2 - r = 0.01$$

The starting point in the application of Monte Carlo simulation is the sample path generation, given a stochastic differential equation describing the dynamics of a stock price. The simplest discretization approach, known as Euler scheme, yields the following discrete-time model for the stock price for a given time increment Δ_t is (we have r , since we are dealing with the risk neutral measure):

$$\begin{aligned} S_{t+\Delta_t} &= S_t + rS_t \Delta_t + \sigma S_t \Delta W \\ &= S_t + rS_t \Delta_t + \sigma S_t \sqrt{\Delta_t} z_t \end{aligned} \tag{2.23}$$

Where $z_t \sim i.i.d.\mathcal{N}(0, 1)$, $\Delta W \sim \mathcal{N}(0, \Delta_t)$, S_0 has to be given and $S_{t+\Delta}|S_t$ is

normal distributed.

Algorithm 6: Geometric Brownian motion approximation with Euler Scheme (MATLAB)

```

vol = 0.2; r = 0.01; S0 = 100; T = 1; m = 20; t = 100; dt = 1/t
tt = linspace(0, T, t);
SE = zeros(m, t);
SE(:, 1) = S0
for i = 1 : t - 1
    dW = sqrt(dt) * randn(1, m);
    SE(:, i + 1) = SE(:, i) + SE(:, i) * r * (dt) + SE(:, i). * vol. * dW';
end
figure
plot (SE')

```

Analogously, we can simulate the trajectories following the exact solution provided by the Ito's Lemma (appendix A.5) ($X_t = \log(S_t)$):

$$\begin{aligned}
 dX_t &= \frac{\partial \log((S_t)}{\partial (S_t)} dS_t + \frac{1}{2} \frac{\partial^2 \log(S_t)}{\partial S_t^2} \sigma^2 S_t^2 dt \\
 &= \frac{1}{S_t} (r S_t dt + \sigma S_t dW_t) - \frac{1}{2} \frac{1}{S_t^2} \sigma_t^2 dt \\
 &= (r - \frac{1}{2} \sigma^2) dt + \sigma dW_t
 \end{aligned} \tag{2.24}$$

In discrete time we have that the log trajectory is:

$$\Delta X = (r - \frac{1}{2} \sigma^2) \Delta t + \sigma \Delta W \tag{2.25}$$

So the final trajectory for the price S_t at time t will be:

$$S_{t+\Delta} = S_t e^{(r - \frac{1}{2} \sigma^2) \Delta t + \sigma \sqrt{\Delta t} z_t} \tag{2.26}$$

Where $z_t \sim i.i.d. \mathbb{N}(0, 1)$, $\Delta W \sim \mathbb{N}(0, \Delta_t)$, S_0 has to be given and $S_{t+\Delta}|S_t$ is log normal.

The two methods are very similar: the Euler is less precise but faster, the "exact one" is closer to the effective solution of the SDE but more computationally demanding.

The next step is to calculate the quadratic mean of the price at each step, for different simulations (the algorithms for both the simulations are exposed in 7).

The quadratic mean at each point in time t in discrete time will be calculated as follows:

$$Q_t = \sqrt{\frac{\sum_{s=1}^t S_s^2}{t}}$$

Algorithm 7: Geometric Brownian motion approximation Ito's Formula and quadratic mean trajectories (MATLAB)

```

vol = 0.2; r = 0.01; S0 = 100; T = 1; m = 20; t = 100; dt = 1/t
S = zeros(m,t);
S(:,1) = S0
for i = 1 : t - 1
    dW = sqrt(dt) * randn(1,m);
    S(:, i + 1) = S(:, i). * exp((r - 0.5 * vol^2) * (dt) + vol * dW');
    Q(:, i + 1) = sqrt(mean(S(:, 1 : i + 1)^2, 2));
end
figure
plot (S')
plot (Q')

```

Here we present the algorithm and a plot of the trajectories of S and of their quadratic mean Q at each point in time.

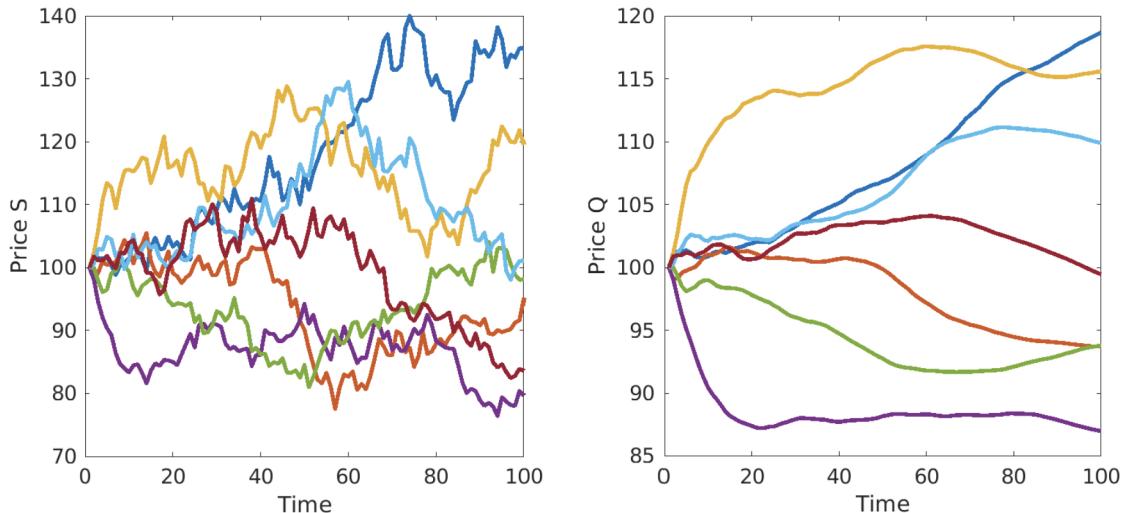


Figure 2.3: Simulation of 7 Trajectories of the Brownian motion: S Vs Q

2.3.2 Numerical Solution

The risk neutral price of the European Call and Put option is:

$$C = e^{-rT} E^Q[(Q_T - K)^+] \quad \text{and} \quad P = e^{-rT} E^Q[(K - Q_T)^+]$$

So the expected value can be calculated over a sample of M simulations of the final payoff as:

$$C = e^{-rT} \frac{\sum_{i=1}^M (Q_{T,i} - K)^+}{M} \quad \text{and} \quad P = \frac{\sum_{i=1}^M (K - Q_{T,i})^+}{M}$$

Finally we repeat the procedure n times in order to have a sequence of different results for the price to build the Monte Carlo mean price and the confidence bands.

Here we present the algorithm which has been employed to get to the results and the parameters used for the simulation.

Algorithm 8: European Call option pricing with Asian payoff with Monte Carlo (MATLAB)

```

 $vol = 0.2; r = 0.01; S_0 = 100; K = 100; T = 1; m = 10^6; t = 360; dt = 1/t$ 
 $S = zeros(m, t);$ 
 $S(:, 1) = S_0;$ 
 $Q = zeros(m, nsim);$ 
for  $j = 1 : nsim$ 
    for  $i = 1 : t - 1$ 
         $dW = sqrt(dt) * randn(1, m);$ 
         $S(:, i + 1) = S(:, i). * exp((r - 0.5 * vol^2) * (dt) + vol * dW');$ 
    end
     $Q_T(:, j) = sqrt(mean(S(:, 1 : end)^2, 2));$ 
     $payoffC = Q_T - K;$ 
     $C = exp(-r * T) * mean(max(payoffC, 0));$ 
end
 $Cmean = mean(C)$ 
 $Cstddev = std(C)$ 
 $ErrC = C - Cmean;$ 

```

The price of the call with a confidence interval of the 95% with 10^6 simulation of the trajectories, over 10^2 number of price simulated will be:

$$Call = 4,9934 +_- 1,96 * 0.0069$$

2.3.3 Analysis of the convergence

The convergence of the price to its true mean is reached by increasing the number of simulations by the WLLN. The asymptotic behaviour of the Monte Carlo mean follows what we illustrated in section 1.2.2: by increasing the number of simulation M by a factor of 100, we decrease the standard error of the Monte Carlo mean by a factor of 10 since: $\sigma_M = SE(M_M) = \frac{\sigma}{\sqrt{M}}$. Here we present the results for the standard error, calculated as standard deviation of 100 prices obtained through MC simulation with 3 different number of simulated trajectories:

$M = 10^2$	$M = 10^4$	$M = 10^6$
$C = 5,1602$	$C = 4,9859$	$C = 4,9934$
$SE = 0,7784$	$SE = 0,0819$	$SE = 0,0069$

In section 3.1 we will present a method to decrease even more the St. error with a rate higher with respect to the simulations added (antithetic variates technique and control variates technique).

Here we present a graph 2.4 in which we can appreciate the decrease in the MC error for a different number of simulated trajectories taken into exam (10^2 , 10^4 , 10^6).

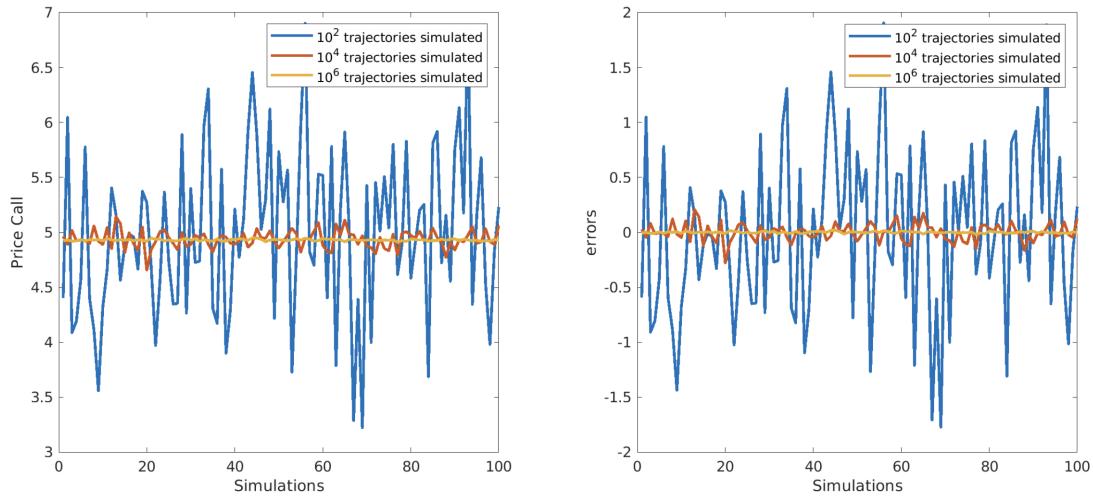


Figure 2.4: Prices and magnitude of the errors (differences with respect to the mean) over a different number of simulated trajectories (100 trials)

Here we present a graph 2.5 of the evolution of the convergence of the price to the MC mean and of the MC error $\hat{C} - \bar{C}$ and standard error $\sigma_M = SE \approx \frac{\sigma}{\sqrt{M}}$, varying the number of simulated trajectories from 10^2 to 10^3 .

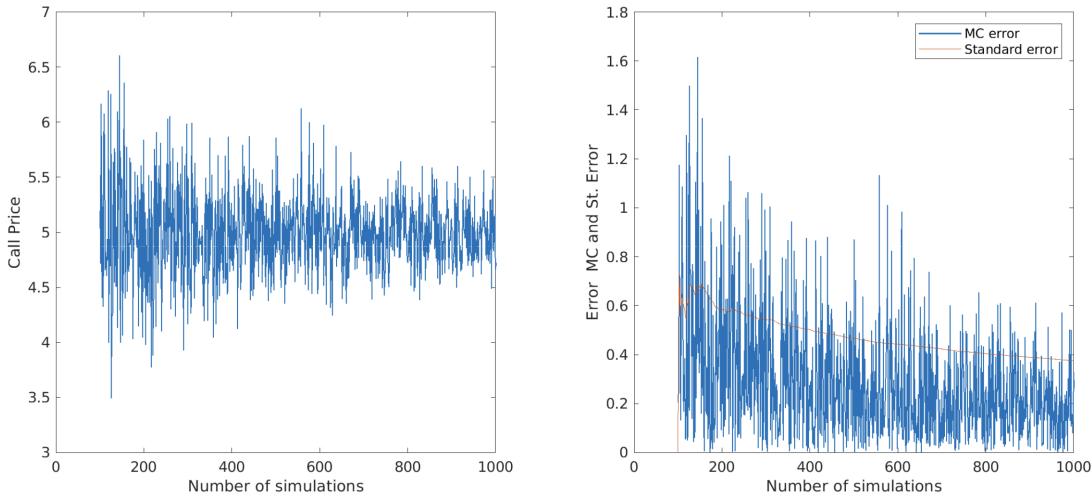


Figure 2.5: Convergence evolution from 10^2 simulations to 10^3 simulations

Chapter 3

Option Pricing with Monte Carlo: Variance Reduction Techniques and Stochastic Volatility Model Application (SABR)

3.1 Variance Reduction Techniques: Improving the Monte Carlo estimate for a European Call Option with Asian Payoff

In 1.1.2 we saw that the confidence intervals for controlling the error in Monte-Carlo computations have a length proportional to the standard deviation of the sampled random variable X . The main disadvantage of the standard Monte Carlo simulation analyzed in 1.2.1 is its convergence rate in $\frac{\sigma}{\sqrt{n}}$.

Reducing the variance (either by modifying the simulation procedure, or transforming the problem) may help to accelerate the convergence of the Monte-Carlo method. There are several methods which can be used to improve the convergence speed. Decreasing the variance by a factor 10 is equivalent, asymptotically, to increase the number of simulations (and hence the computational time) by a factor of 10 to achieve a given accuracy. We can improve the constant factor $\sigma(f; X)^2 = \text{var}(f(X))$ instead of trying to increase the convergence rate with more n . The idea is basically this: we define a Random Variable Y and functions g such that:

$$E[g(y)] = E[f(X)]$$

and of course:

$$\text{Var}(g(y)) < \text{Var}(f(X))$$

Once provided that the generation of samples from $g(Y)$ is not much more expensive than the generation of samples $f(X)$, we can decrease the computational work, by

simulating samples of $g(Y)$.

3.1.1 Antithetic Variates Technique

The antithetic variates variance reduction technique derives from this two facts:

1. if U has the uniform distribution, then $1 - U$ has still the uniform distribution;
2. if $X \sim N(0, I_d)$ considering a d-dimensional normal distribution, then $-X$ has the same normal distribution.

Therefore, if X has one of these distribution, the transformation does not change the expected value $E[f(X)]$. In general we can take into consideration any transformed \tilde{X} which has the same distribution of X and such that a realization of \tilde{X} can be computed as a deterministic function of the variable X .

Given X and \tilde{X} having the same law, we define the antithetic Monte Carlo estimate as:

$$E_N^{anti}[f(X)] = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i) + f(\tilde{X}_i)}{2} \quad (3.1)$$

To estimate the computing time, assume that the simulation of $\frac{f(X_i) + f(\tilde{X}_i)}{2}$ takes at most two times the computer time as the simulation of $f(X_i)$. Therefore, to compute $E_N^{anti}[f(X)]$ we need at most the time for computing $E_{2N}[f(X)]$ (since we are simply doubling the number of simulation produced). The use of antithetic variates is reasonable if the standard error of $E_N^{anti}[f(X)]$ is smaller than the one for $E_{2N}[f(X)]$, that is:

$$\begin{aligned} \frac{var\left(\frac{f(X)+f(\tilde{X})}{2}\right)}{M} &< \frac{var(f(X))}{2M} \\ 2 \frac{\frac{1}{4}var(f(X)) + \frac{1}{4}var(f(\tilde{X})) + \frac{2}{4}cov(f(X_i), f(\tilde{X}))}{M} &< \frac{var(f(X))}{M} \\ \frac{var(f(X))}{M} + \frac{cov(f(X_i), f(\tilde{X}))}{M} &< \frac{var(f(X))}{M} \end{aligned}$$

So the antithetic variates technique can speed up the Monte Carlo only if the covariance is negative: $cov(f(X_i), f(\tilde{X})) < 0$ (in our example we will take exactly X and $-X$ normally distributed).

3.1.2 Antithetic Variates: Option pricing application

A possible application of the previous techinque can be done on the exercise seen in section 2.3 (European option with asian payoff): in a simulation driven by independent standard normal random variables, antithetic variates can be implemented by pairing a sequence Z_1, Z_2, \dots of i.i.d. $N(0,1)$ variables with the sequence $-Z_1, -Z_2, \dots$ of i.i.d. $iV(0,1)$ variables. If the Z_i are used to simulate the increments of a Brownian path, then the $-Z_i$ simulate the increments of the reflection of the path about the origin. It seems intuitive that running a pair of simulations using the original path

and then its reflection combined may result in lower variance, without changing the expected value.

Here we present the algorithm which has been employed to get to the results and the parameters used for the simulation.

Algorithm 9: European Call option pricing with Asian payoff with Monte Carlo and antithetic variates (MATLAB)

```

 $S_0 = 100; K = 100; \sigma = 0.2; r = 0.01; m = 10^6; nsim = 100; t = 360; dt =$ 
 $1/t$ 
 $S = zeros(m, t);$ 
 $S(:, 1) = S_0;$ 
 $Q = zeros(m, nsim);$ 
 $for j = 1 : nsim$ 
 $for i = 1 : t - 1$ 
 $dW = sqrt(dt) * randn(1, m);$ 
 $S(:, i + 1) = S(:, i). * exp((r - 0.5 * vol^2) * (dt) + vol * dW');$ 
 $SA(:, i + 1) = S(:, i). * exp((r - 0.5 * vol^2) * (dt) - vol * dW');$ 
 $S = [S; SA];$ 
 $end$ 
 $Q_T(:, j) = sqrt(mean(S(:, 1 : end)^2, 2));$ 
 $payoffC = Q_T - K;$ 
 $C = exp(-r * T) * mean(max(payoffC, 0));$ 
 $end$ 
 $Cmean = mean(C)$ 
 $Cstdev = std(C)$ 
 $ErrC = C - Cmean;$ 

```

The price of the call with a confidence interval of the 95% with 10^6 simulation of the trajectories, over 10^2 of MC simulations done will be:

$$Call = 4,9934 \pm 1,96 * 0.0069$$

The convergence of the price to its true mean is reached by increasing the number of simulations by the WLLN and by doubling the number of simulations with the antithetic variates technique. The asymptotic behaviour of the Monte Carlo will improve thanks to the variance reduction technique. Here we present the results for the standard error with different number of simulations. It seems that implementing an antithetic variates technique results always in a lower standard error with respect to a standard Monte Carlo with twice the number of simulated trajectories with respect to the ones used in the antithetic variates estimate:

$M = 10^2; 2M = 2 * 10^2$	$M = 10^4; 2M = 2 * 10^4$	$M = 10^6; 2M = 2 * 10^6$
$SE(Msim) = 0,7784$	$SE(Msim) = 0,0819$	$SE(Msim) = 0,0069$
$SE(2Msim) = 0,4994$	$SE(2Msim) = 0,0590$	$SE(2Msim) = 0,0053$
$SE_antithetic(Msim) = 0,4232$	$SE_antithetic(Msim) = 0,0390$	$SE_antithetic(Msim) = 0,0041$

In fig. 3.1 we can appreciate the decrease in the variability of the price and in the MC error between the Monte Carlo with $2 * M = 10^4$ simulated trajectories

and Monte Carlo with antithetic variates with only 10^4 simulated trajectories (the different trajectories and respective prices have been simulated 100 times). We are in fact assuming that implementing an antithetic variates technique can take at most twice the time and computational effort than the simple MC, so in order to have an advantage, we should see, as we saw in the table before, a sensible improvement.

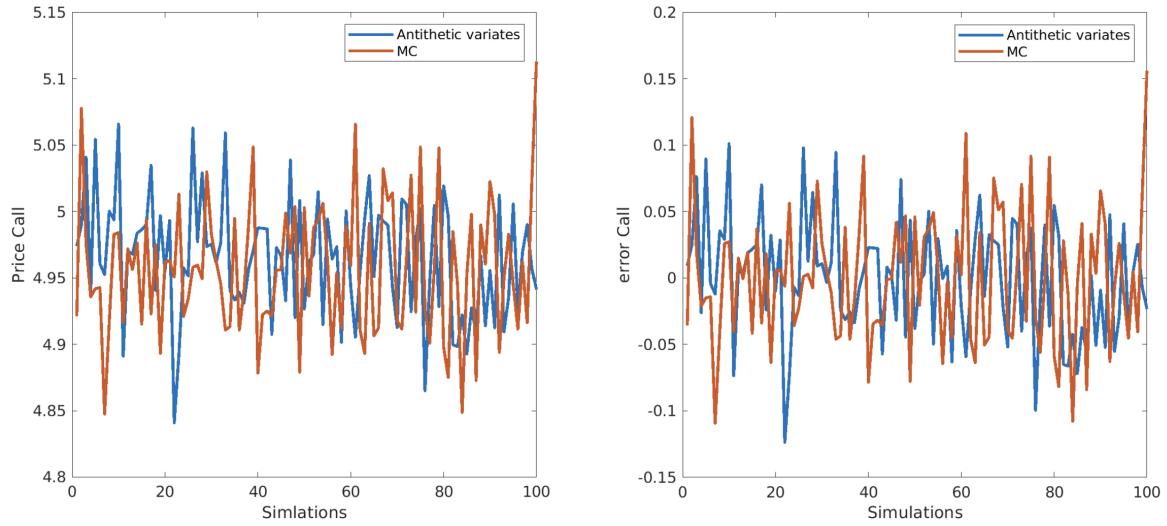


Figure 3.1: Prices and magnitude of the errors (differences with respect to the mean) on $M = 10^6$ simulated trajectories for AV and $2 * M = 2 * 10^6$ (100 trials)

Here we present a graph (fig. 3.2) of the evolution of the convergence of the price to the MC mean and of the MC error $\hat{C} - \bar{C}$ and standard error $\sigma_M = SE(M_M) = \frac{\sigma}{\sqrt{M}}$, varying the number of simulated trajectories from 10^2 to 10^3 in the simple MC case and in the AV case.

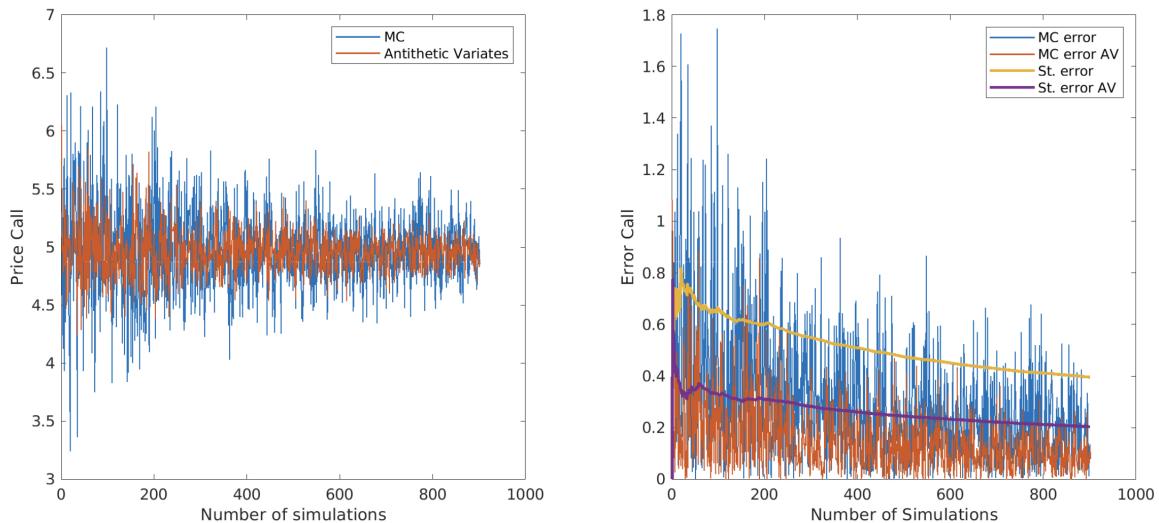


Figure 3.2: Convergence evolution from 10^2 simulations to 10^3 simulations

3.1.3 Control Variates Technique

We consider a second possible approach in order to reduce the variance. It exploits information about the errors in estimates of known quantities to reduce the error in an estimate of an unknown quantity [15]. Suppose that we can define a random variable Y and a function g such that we know the exact value of $E[g(Y)]$. Notice that we can choose $Y = X$. Therefore:

$$E[f(X)] = E[f(X) - \beta(g(Y) - E[g(Y)])] \quad \text{with } \beta \in \mathbb{R}$$

We can define the following Monte-Carlo estimator for $E[f(X)]$:

$$E_N^{control}[f(X)] := \frac{1}{N} \sum (f(X_i) - \beta g(Y_i)) + \beta E[g(Y)] \quad (3.2)$$

where (X_i, Y_i) are independent realizations of (X, Y) . We may assume that the simulation of $E_N^{control}[f(X)]$ takes at most twice the time of simulation of $E_N[f(X)]$. Since β is arbitrarily chosen, we can define it as that parameter minimizing:

$$\text{var}(f(X) - \beta g(Y))$$

So we have that:

$$\text{var}(f(X) - \beta g(Y)) = \text{var}(f(X)) - 2\beta \text{cov}(f(X), g(Y)) + \beta^2 \text{var}(g(Y))$$

can be minimized choosing:

$$\beta = \frac{\text{cov}(f(X), g(Y))}{\text{var}(g(Y))}$$

The higher the correlation between $f(X)$ and $g(Y)$, the higher the improvement of calculation time, where ρ is the correlation coefficient between them.

3.1.4 Control Variates: Option pricing application

One of the first implementations of the control variates technique in option pricing estimation comes from Boyle [4], which uses it in a binomial framework. Kemna and Vorst [20] analyze a specific case on Asian options (building a MC estimator for an arithmetic mean Asian option, using as control variates a geometric mean Asian option). A similar application of the one done by Kemna and Vorst can be done on the exercise seen in section 2.3. Suppose C_s is a quadratic mean Asian option price as we saw in the previous examples and C_g be the geometric Asian option price. They represent two functions of the price of the same underlying S_t (so we don't have to double the simulations to be made), whose risk neutral price is represented

by these two functions, which can be computed through a simple MC method:

$$\hat{C}_s = f(S_t) = E^Q[e^{-rT}(\sqrt{\frac{\sum_{s=1}^T S_s^2}{T}} - K)^+] = \frac{e^{-rT} \sum_{i=1}^M (\sqrt{\frac{\sum_{s=1}^T S_{s,i}^2}{T}} - K)^+}{M}$$

$$\hat{C}_g = g(S_t) = E^Q[e^{-rT}((\prod_{s=1}^T S_s)^{\frac{1}{T}} - K)^+] = \frac{e^{-rT} \sum_{i=1}^M ((\prod_{s=1}^T S_{s,i})^{\frac{1}{T}} - K)^+}{M}$$

Once we have estimated our price given a certain number of M trajectories of S_t simulated, we can repeat the MC simulation n times, adjusting every time the simple \hat{C}_s estimator according to the difference between the known value of the explicit solution existing for the geometric mean Asian option price C_g (the explicit formula can be found in [20] and [19]) and the MC estimator of the Asian option \hat{C}_g . So we can build our MC estimator as follows:

$$C_{CVs} = \frac{\sum_{i=1}^n \hat{C}_s + \beta(C_g - \hat{C}_g)}{n} \quad (3.3)$$

choosing beta, such that it minimizes the variance of the MC estimator:

$$\hat{\beta} = \frac{cov(\hat{C}_s, \hat{C}_g)}{var(\hat{C}_g)} \quad (3.4)$$

The known error $C_g - \hat{C}_g$ is thus used as a control in the estimation of C_{CVs} , adjusted for the value of the $\hat{\beta}$ estimated.

The control variates method is efficient if the covariance between \hat{C}_s and \hat{C}_g is high (for more on beta and variance estimators in the control variate method see [35]). We implemented the technique also for a simple European call option, but the beta was lower for the geometric mean Asian option, respectively $\beta_{EU} = 0,51$ $\beta_g = 1,01$. The latter seems to be more efficient in adjusting the estimates and improving the standard error of the estimation.

Using the geometric mean option as control variate, the price of the call with a confidence interval of the 95% with 10^6 simulation of the trajectories, over 10^2 number of price simulated is:

$$Call = 4,9987 \pm 1,96 * 0.0039$$

The convergence of the price to its true mean is reached by increasing the number of simulations by the WLLN and by correcting the estimates employing two control variates: the price of an European call and the price of an Asian call with geometric mean payoff. Here we present the results for the call price and the standard error with different number of simulated trajectories M and for n = 100 samples simulated (SE will give us a measure of the dispersion around the true mean of the different mean prices obtained with each Monte Carlo simulation).

$M = 10^2$	$M = 10^4$	$M = 10^6$
$C = 5,1602$	$C = 4,9859$	$C = 4,9934$
$SE = 0,7784$	$SE = 0,0819$	$SE = 0,0069$
$C \text{ (} CV_{simpleuro} \text{) } = 4,9586$	$C \text{ (} CV_{simpleuro} \text{) } = 5,0051$	$C \text{ (} CV_{simpleuro} \text{) } = 4,9995$
$SE \text{ (} CV_{simpleuro} \text{) } = 0,4077$	$SE \text{ (} CV_{simpleuro} \text{) } = 0,0455$	$SE \text{ (} CV_{simpleuro} \text{) } = 0,0047$
$C \text{ (} CV_{geomean} \text{) } = 4,9401$	$C \text{ (} CV_{geomean} \text{) } = 4,9973$	$C \text{ (} CV_{geomean} \text{) } = 4,9987$
$SE \text{ (} CV_{geomean} \text{) } = 0.4006$	$SE \text{ (} CV_{geomean} \text{) } = 0,0406$	$SE \text{ (} CV_{geomean} \text{) } = 0,0039$

Here we present the algorithm (10) employed to get to the results

Algorithm 10: European Call option pricing with asian payoff with Monte Carlo and control variates (EU and geometric) (MATLAB)

```

 $S_0 = 100; K = 100; \sigma = 0.2; r = 0.01; m = 10^6; nsim = 100;$ 
 $t = 360; dt = 1/t$ 
 $S = zeros(m, t); S(:, 1) = S_0;$ 
for  $j = 1 : nsim$ 
    for  $i = 1 : t - 1$ 
         $dW = sqrt(dt) * randn(1, m);$ 
         $S(:, i + 1) = S(:, i). * exp((r - 0.5 * vol^2) * (dt) + vol * dW');$ 
    end
     $Q_{Ts}(:, j) = sqrt(mean(S(:, 1 : end)^2, 2));$ 
     $Q_{Tg}(:, j) = geomean(S1(:, 1 : end)')$ 
     $Q_{Te}(:, j) = S1(:, end)$ 
     $payoffC_s = Q_{Ts} - K;$ 
     $payoffC_g = Q_{Tg} - K;$ 
     $payoffC_e = Q_{Te} - K;$ 
     $C_s = exp(-r * T) * mean(max(payoffC_s, 0));$ 
     $C_g = exp(-r * T) * mean(max(payoffC_g, 0));$ 
     $C_e = exp(-r * T) * mean(max(payoffC_e, 0));$ 
end
 $Call_g = asiansensbykv(RateSpec, StockSpec, 'call', K, Settle, ExDates)$ 
 $[Call_e, Put] = blsprice(S0, K, r, T, vol)$ 
 $MCov_g = cov(C_s, C_g)$ 
 $MCov_e = cov(C_s, C_e)$ 
 $beta_g = MCov_g(1, 2)/MCov_g(2, 2)$ 
 $beta_e = MCov_e(1, 2)/MCov_e(2, 2)$ 
 $CALLCV_{sg} = C_s + beta * (Call_g - C_g);$ 
 $CALLCVmean_{sg} = mean(CALLCV_{sg})$ 
 $CALLCVerror_{sg} = CALLCV_{sg} - CALLCVmean_{sg};$ 
 $CALLCVstdev_{sg} = std(CALLCV_{sg})$ 
 $CALLCV_{se} = C_s + beta * (Call_e - C_e)$ 
 $CALLCVmean_{se} = mean(CALLCV_{se})$ 
 $CALLCVerror_e = CALLCV_{se} - CALLCVmean_{se};$ 
 $CALLCVstdev_{se} = std(CALLCV_{se})$ 

```

3.2 Beyond the Black and Scholes Model

3.2.1 Implied Volatility

Having fixed all the other parameters, the value of the option is increasing with respect to the volatility parameter.

If we fix a σ we have a correspondent price (all the other parameters being equal) and viceversa, having fixed a price, we can have the implied volatility to a certain observed price.

Suppose we observe the price of a call option (empirical price) $\bar{C}(r, T, K, S_0)$. We can determine the implied volatility $\bar{\sigma}(r, T, K, S_0)$ such that $\rightarrow C_{BS}(r, T, K, S_0, \bar{\sigma}) = \bar{C}(r, T, K, S_0)$, where C_{BS} is the well known price formula of a European call option that can be found in [3].

In the real market, the prices of the derivatives are often given in terms of implied volatility. B & S is the language in which are expressed the prices. The implied volatility is always related to the B & S model; we can derive it from a quoted price using the B & S, which transforms volatility in prices and viceversa.

Why often prices are expressed in terms of implied volatility of B & S?

Because the implied volatility brings all the contracts in a unique measure unit. For example, if we take the price of two European call options: $C_{BS}^1 = (r, S_0, T_1, K_1; \sigma)$ and $C_{BS}^2 = (r, S_0, T_2, K_2; \sigma)$, they are not comparable, but their implied volatilities are comparable: the higher the volatility, the more expansive is the contract.

Implied Volatility Surface

Let's fix r and S_0 .

The implied volatility surface will be the the surface which represents graphically (with a certain approximation) the implied volatilities varying K and T . Where the surface is higher there are more expansive derivatives.

If a bank uses the Black and Scholes model to produce those prices (quotes), all the implied volatilities would be equal. In reality that's not the case. The volatility surface observed on the same asset is empirically not flat but is higher for more extreme strikes (deeply at or out of the money options) and tends to vary its shape for different maturities.

Normally, fixed a T and varying K , we observe a smile in the form of the volatility curve, whose exact shape (convexity, slope...) depends on the specific market (ex. for utilities we have strange behaviours) to which the underlying belongs.

This "smile" shape is often explained with some intuitive arguments:

- Typically, the more liquid and traded derivatives are the ones in the neighborhood of S_0 , so we have a lower implied volatility for a strike close to that value. Additionally, we often observe that the implied volatilities of the out of the money options are usually higher, since higher is the risk perceived for these derivatives.

- The implied volatility for extreme strikes is higher since extremal events are more likely than hypothesized by the log normality assumption of B & S. In a sense, BS underestimates the probability of extreme events (the probability is normal, the probability of the tales is little). The tails of the empirical distribution of the underlying S in reality are much more thicker.

The hypothesis of BS are no longer believed or seen as too simplistic; the banks opted for more complicated models, with non constant volatility surfaces.

We have to go beyond BS because a geometric Brownian Motion with constant parameter cannot catch the true volatility surface of the market.

3.2.2 Models for Local and Stochastic Volatility: a Brief Overview

These models are extensions of Black and Scholes: they are able to reproduce the volatility surface of the market, to quote other derivatives.

BS extensions

For our purpose we need models with "non constant volatility" or "Volatility Models":

- Local volatility models: we assume that the volatility parameter and/or the drift parameter are not constant. The coefficients are no more linear, but are functions. Typically they are expressions which depend on some parameters, typically t and S_t :

$$\begin{cases} dB_t = rB_t dt \\ dS_t = \mu(t, S_t) dt + \sigma(t, S_t) S_t dW_t \end{cases}$$

The solution of the differential equation exists and is unique if $\mu(t, S_t)$ and $\sigma(t, S_t)$ are Lipschitz limited functions. S will be the solution of the stochastic differential equation.

Using the same arguments we used for the BS model, we can sometimes derive an analytical formula for the value of the option written on the underlying or we can approximate the solution through a finite difference method:

- a) Use Girsanov's Theorem to get a drift equal to the risk free bond: $rS_t = rS_t dt + \sigma(t, S_t) S_t d\tilde{W}_t$ with \tilde{W} Brownian motion in \mathbb{Q} . What changes is that in this case $\lambda_t = \frac{\mu(t, S_t) - r}{\sigma(t, S_t)}$. In this way the discounted value of the strategy (the option price) is a martingale.

We notice again that with the Girsanov's theorem we change the drift but the diffusive part remains unchanged basically (it can estimate it in one measure or the other one, is the same).

- b) Calculate the stochastic differential in \mathbb{Q} of a discounted hedging strategy with Ito:

$$d(e^{-rt} f(t, S_t)) = e^{-rt} (Lf(t, s))_{BS\ operator} dt + \sigma(t, S_t) \partial f(t, S_t) d\tilde{W}_t$$

The BS operator $Lf(t, s) = \frac{\sigma(t,s)^2 s^2}{2} \partial_{ss} + rs \partial_s + \partial_t - r$ in this case is different since it has a variable coefficient σ , so the Cauchy problem formed with the final condition doesn't have often an explicit solution but can be solved through some change of variables transformed in the heat equation which can be solved numerically with a finite differences method.

$$\begin{cases} Lf(t, s) = 0 \\ f(t, s) = \varphi(s) \end{cases}$$

Anyway, we can alternatively proceed to calculate the risk neutral price of the option as the expected value in \mathbb{Q} of the payoff of the option, since a self-financing strategy $f(t, S_T)$ which can replicate the final payoff $\varphi(S_T)$ of the option in a complete market exists and its discounted value $e^{-rt} f(t, S_t)$ is a Q-mg (since as we saw in section 2.2.5, it is enough that the condition $Lf(t, s) = 0$ of null drift can be satisfied). We will say that the market is complete if it exists a unique martingale measure identified with the Girsanov's theorem. If the market is incomplete it doesn't mean we cannot identify a martingale measure, but it means that the Q-mg is not unique (there are infinitely many of them). By the non-arbitrage principle, the discounted price of the derivative X_0 must be equal to:

$$X_0 = f(0, S_0) = E^{\mathbb{Q}}[e^{-rt} f(T, S_T)] = E^{\mathbb{Q}}[e^{-rt} \varphi(S_T)]$$

In this way we can calculate the price with the Monte Carlo methods, but we must be able to simulate S_t . We can simulate the trajectories iteratively with a Euler discretization method ($\mu(t, S_t) = r$ is taken constant):

$$\begin{cases} S_{tk} = S_{tk-1} + r S_{tk-1} (t_k - t_{k-1}) + \sigma(t_{k-1}, S_{tk-1}) S_{tk-1} (\tilde{W}_{tk} - \tilde{W}_{tk-1}) \\ S_{t_0} = S_0 \end{cases}$$

An example of local volatility model is the CEV, which presents an inverse relation between price of the underlying and volatility:

$$\sigma(t, S_t) = \sigma_0 S_t^{-\beta} \text{ with } \beta < 0, \sigma_0 > 0 \text{ (if } \beta = \frac{1}{2} \text{ CIR model)}.$$

- Stochastic volatility models: the volatility parameter is a stochastic process which is not a function only of t and S_t (for example it can depend on other Brownian motion or jumps; typically the volatility solves another differential equation):

$$dS_t = \mu dt + \sigma_t S_t dW_t \tag{3.5}$$

The passages to find the pricing equation or to compute a risk neutral price estimated with Monte Carlo are similar to the local volatility models, but we need to specify a differential equation for the volatility too, which like the SABR model analyzed in section 3.3, will depend on another Brownian motion. So, in order to

work with these kind of models we need to introduce some theoretical knowledge on d-dimensional Brownian motions.

3.2.3 d-dimensional Brownian motion

A d-dimensional brownian motion $W = (W^1, \dots, W^d)$ is a stochastic process with 'd' components defined on a filtered probability space $(\Omega, \mathcal{F}, P, \mathcal{F}_{t>0})$ which verifies the following properties:

- 1) $W_0 = (0, \dots, 0)$ is the vector with null components;
 - 2) W is a continuous and adapted process
 - 3) $W_t - W_s$ is independent from \mathcal{F}_s with $t > s$ (it has independent increments)
 - 4) $W_t - W_s \sim \mathcal{N}(0, (t-s)\mathbb{I}_d)$ (it has increments distributed multi- normally);
- where the multi normal distribution is a distribution on $(\mathbb{R}^d, \mathcal{B}_d)$ with density:

$$\Gamma(x) = \frac{1}{\sqrt{(2\pi)^d \det(C)}} e^{-\frac{1}{2} \langle C^{-1}(x-\mu), (x-\mu) \rangle}$$

with $\mu \in \mathbb{R}^d$ and C a $d \times d$ symmetric positive definite matrix (all eigenvalues positive or quadratic form positive: $\langle C_y, y \rangle > 0, \in \mathbb{R}^d / \{0\}$).

The characteristic function of the multi normal is: $\varphi_x(\eta) = e^{i \langle \mu, \eta \rangle - \frac{1}{2} \langle C\eta, \eta \rangle}$

Observations:

i) W^i is a Brownian motion 1-dim (the components of the d-dimensional BM are mono-dim BM).

ii) W_t^i, W_t^j are independent if $i \neq j$ (null covariances and diagonal cov. matrix).

So a d-dimensional BM is a vector of independent mono-dimensional BM.

We can consider d as the "degrees" of randomness and N as the number of processes i want to consider(prices, volatility, rates..).

Correlated d-dim Brownian motions

Sometimes we can use a correlated d-dimensional Brownian motion (B_t):

$$W_t \sim \mathcal{N}(0, (t)\mathbb{I}_d)$$

$B_t := A * W_t$ with A $N \times d$ matrix such that:

$$B_t \sim \mathcal{N}(0, (t)AA')$$

In our Stochastic volatility model (SABR) we will use two Brownians (B^1 and B^2 are mono dimensional BM), one for the price and one for the volatility (n=d=2), with covariance: $Cov(B_t^1, B_t^2) = \rho t$ (ρ is the correlation).:

$$\begin{cases} B_t^1 = \sqrt{1 - \rho^2} W_t^1 + \rho W_t^2 & \text{with } -1 \leq \rho \leq 1 \\ B_t^2 = W_t^2 \end{cases}$$

where: $W = (W^1, W^2)$ is a bi-dimensional BM;

$$A = \begin{bmatrix} \sqrt{1 - \rho^2} & \rho \\ 0 & 1 \end{bmatrix}, \quad AA' = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

3.3 Stochastic Volatility and Monte Carlo Methods

3.3.1 The SABR Model

We have introduced in Chapter 2 the Bachelier and BS Models. However, as pointed out by Crispoldi [8], even if they are intuitive, simple and their parameters can be quickly calibrated to obtain a price in agreement with the market, these simple models cannot be calibrated to more than one volatility per expiry.

On the other hand, the downside of the local volatility models is that we cannot use a Markovian model based on a single Brownian motion to manage the "smile risk" (the risk coming from the different shapes that a volatility surface can take).

Rather than making the model non-Markovian, or basing it on stochastic processes other than the Brownian motions, Hagan, Kumar, Lesniewski and Woodward (we will refer to them with Hagan[17]) chose to develop a two factor model. In selecting the second factor, they notes that most markets experience both relatively quiescent and relatively chaotic periods. This suggests a non constant volatility, which can be described as a random function of time. Volatility tends to vary during time, but also to persist around a certain level in close periods of time. The "volatility" α is now itself a stochastic process.

The model they came up with is the "stochastic- $\alpha \beta \rho$ model," which has become known as the SABR model. The SABR model is originally described by [17] with the following set of stochastic differential equations and initial conditions:

$$\begin{cases} dF_t = \alpha_t F_t \beta dB_t^1 & \text{with: } F(0) = f \\ d\alpha_t = \nu \alpha_t dB_t^2 & \text{with: } \alpha(0) = \alpha_0 \end{cases}$$

where F_t is the forward price, α is the volatility, β is the elasticity coefficient, ν is the volatility of the volatility process and, under the forward (real) measure, the two brownian motions are correlated in this way:

$$dB_t^1 dB_t^2 = \rho dt \quad (3.6)$$

Where ρ is the correlation coefficient between the two Brownian motions.

The main critique against stochastic volatility models is that they are "models for incomplete markets", since stochastic volatility risk cannot be completely hedged. This critique is firmly rejected since Hagan says that it can be hedged by buying or selling options on the asset to neutralize it, as we sell and buy the underlying F and a risk free bond to neutralize the risk of changes in the underlying price (so called "Delta hedging"). For some other pitfalls regarding the forward process and the risk neutral probability density of the SABR see [8].

The success of the SABR model instead, can be summarized in the following points, summed up clearly by Crispoldi in his book "Sabr and Sabr-Libor Market Models in Practice" [8]:

- As we outlined before, a self financing strategy which allows to hedge the stochastic volatility risk and build a replicate portfolio of the derivative's payoff is feasible for the authors in the SABR model framework; in fact they came up with an analytical formula for the price of vanilla options (such as caplets, floorets, swaptions) under the SABR model, in terms of a "Black-implied volatility" (whose equation is given as well). For our purpose we will not go deeper in trying to prove this fact (deriving the differential equation for the derivative $f(t, S_t)$, set to zero the drift...), but we will simply assume that the options can be modelled as a \mathbb{Q} -mg such that we can implement our Monte Carlo simulation to get a risk neutral price for the derivatives.
- The SABR parameters have a direct and clear influence on the calibrated implied volatility smile.
- The model is generally able to provide a very good fit to the implied volatility smile quoted in the market. The computational simplicity allows the model to be naturally used as a good interpolation scheme along the strike axis.

The SABR Parameters

Let's see how the parameters impact the volatility surface under the SABR model (we consider the explanation given originally by Hagan [17] and the excellent summary on this topic given by Crispoldi [8]):

- The α_0 parameter is the value of the stochastic volatility process α_t in $t = 0$. The value of α_0 shifts the volatility smile up or down with almost no effect on the shape of the smile.
- The β parameter is the "constant elasticity of variance (CEV)" parameter and it must be taken between -1 and 1 since the SABR model is a martingale only if $0 \leq \beta < 1$ or as long as $\rho \leq 0$ for $\beta = 1$ (see Henry-Labordere 2008 [18]). The main effect of β is a change in the smile slope. The slope increases as β moves from 1 to 0. The reason behind this behaviour is the fact that the model switches from a lognormal-like to normal-like behaviour when β is lowered. Since β controls also which distribution the forward rate F_t has, Hagan suggests that this can be chosen a priori to reflect a market view:

If we take $\beta = 0$ the equation for the forward price reduces to $dF_t = \alpha dW_t^1$. If α is taken constant too, the SABR model actually stops being a stochastic volatility model and corresponds to the normal model. This choice allows to assign a probability mass to negative outcomes of F_t (important for interest rate modeling).

If we take $0 < \beta < 1$ the SABR model becomes a CEV type one. If α is taken constant and $\beta = 0.5$, the equation for the forward price reduces to a squared root CEV model: $dF_t = \alpha\sqrt{F_t}dW_t^1$ (see Cox and Ross 1976 [7]).

If we take $\beta = 1$ the equation for the forward price reduces to $dF_t = \alpha F_t dW_t^1$. If α is taken constant, the SABR model corresponds to the lognormal Black & Scholes model (in this case the distribution of the forward price is truly lognormal). When α_t is a stochastic quantity, the actual distribution of F_t is not exactly lognormal. However in this case F_t cannot change sign .

- The ν parameter is the "volatility" of α_t ; for this reason it is also known as "volatility of volatility". The effect of ν on the smile is to change its curvature. A higher value of ν makes the implied volatility increase for the options with more extreme strikes (out of the money (OTM) and in the money (ITM) options).
- The ρ parameter is the correlation between the Brownian motion in the forward rate process (B^1) and the Brownian motion in the volatility process(B^2). It can take any value between -1 and 1. In many markets the sign of ρ is negative, which means that the volatility α_t (t) increases when the forward interest rate F_t decreases and vice versa. The effect of ρ on the smile is similar to the one produced by *beta*: the smile gets more steep as we move from 1 to -1.

3.3.2 Calibration of the Implied volatility Surface

The calibration procedure has been performed following the work of Hagan [17]. First of all we collect from the market the observed implied volatilities in a certain option market with different strikes and maturities, such that we obtain a cloud of different implied volatilities to use in our calibration procedure.

The implied volatility σ^{MKT} is the value of the parameter to be inserted in the BS formula in order to obtain a certain price observed of a European derivative $C^{MKT}(K, T)$. The data we used for our calibration were the ones of a European call option on the FTSE Mib Index observed the "19-May-2021" from the Bloomberg Platform. The observed implied volatilities are synthesized in the following table.

K	18Jun21	17Sep21	17Dec21	18Mar22	17Jun22	16Dec22	16Jun23	15Dec23
21000	35,06%	27,80%	26,39%	24,81%	24,00%	23,57%	22,63%	22,49%
22000	31,12%	25,98%	24,45%	23,25%	22,65%	22,19%	21,56%	21,55%
23000	27,23%	23,63%	22,65%	21,73%	21,43%	21,28%	20,51%	20,66%
24000	22,95%	21,32%	20,95%	20,47%	20,32%	20,14%	19,59%	19,91%
25000	18,81%	19,26%	19,44%	19,21%	19,32%	19,39%	18,82%	19,31%
26000	16,64%	17,57%	18,15%	18,13%	18,46%	18,69%	17,80%	18,24%
27000	17,14%	16,41%	17,21%	17,35%	17,81%	17,50%	17,70%	18,48%
28000	20,03%	16,03%	16,64%	16,79%	17,31%	18,03%	17,39%	18,16%

In order to use the original formulas of Hagan we need to work with forward values: we collect the actual value of the index $S_0 = 24486,45$ and we compute the forward values at the different maturities by capitalizing the value of the index at a certain rate r : $f_t = e^{r(T-t)}S_t$; an estimate of the implied fwd value of the index is calculated directly by Bloomberg, which is what we took as fwd values.

Now we need to calibrate all the parameters except for the β which, as Hagan [17] suggested, must be fixed before the calibration: "Selecting β from "aesthetic" or other a priori considerations usually results in $\beta = 1$ (stochastic lognormal), $\beta = 0$ (stochastic normal), or $\beta = \frac{1}{2}$ (stochastic CIR) models".

As expected $\beta = 1$ results also in our empirical tests in the best result in calibration since the equities are better described by a log-normal behaviour.

Now the remaining parameters of the model α_t , ρ_t , ν_t are calibrated 8 times for each maturity (T_1, T_2, \dots, T_8) through a minimization exercise which has been performed by employing the Optimization Toolbox function "lsqnonlin" in MATLAB: given a certain maturity T_i , it generates the parameter values that minimize the sum of the squared error for each strike between the market volatilities and the volatilities computed by "blackvolbysabr":

$$\sum_{K=K_1}^{K_8} \sigma_K^{MKT} - \sigma_K^B \quad (3.7)$$

Where "blackvolbysabr" is the MATLAB function which computes σ^B , the Black (log-normal) implied volatility calculated for a specific strike K and time to expiry which is one of the main results of the work of the work of Hagan:

$$\begin{aligned} \sigma^B(K, f; \alpha_0, \beta, \rho, \nu) = & \frac{\alpha_0}{(fK)^{\frac{1-\beta}{2}} [1 + \frac{(1-\beta)^2}{24} \log^2(\frac{f}{K}) + \frac{(1-\beta)^4}{1920} \log^4(\frac{f}{K}) + \dots]} * \\ & (\frac{z}{x(z)}) * \{1 + [\frac{(1-\beta)^2}{24} \frac{\alpha_0^2}{(fK)^{1-\beta}} + \frac{1}{4} \frac{\rho\beta\nu\alpha_0}{(fK)^{\frac{1-\beta}{2}}} + \frac{(2-3\rho^2)\nu^2}{24}]T + \dots\} \end{aligned}$$

where:

- $z = \frac{\nu}{\alpha_0} (fK)^{\frac{1-\beta}{2}} \log(\frac{f}{K})$;
- $x(z) = \log(\frac{\sqrt{1-2z+z^2}+z-\rho}{1-\rho})$;
- f is the current forward value of the underlying (in $t=0$);
- α_0 is the current SABR volatility (in $t=0$);
- K is the strike value
- T is the option's maturity.

Here we have all the calibrated parameters for the different maturities:

	18Jun21	17Sep21	17Dec21	18Mar22	17Jun22	16Dec22	16Jun23	15Dec23
α	0,2076	0,2100	0,2125	0,2088	0,2129	0,2139	0,2150	0,2177
β	1	1	1	1	1	1	1	1
ρ	-0,6567	-0,7279	-0,7019	-0,6845	-0,6703	-0,6328	-0,6543	-0,6140
ν	2,8120	1,3778	1,1268	0,9312	0,8073	0,8019	0,7516	0,7055

Once we have calibrated our model, we can easily calculate the implied volatility through the Hagan Formula and plot it is easily for each strike and maturity as a surface in MATLAB, alongside the observed implied volatility (blue dots in fig. 3.4).

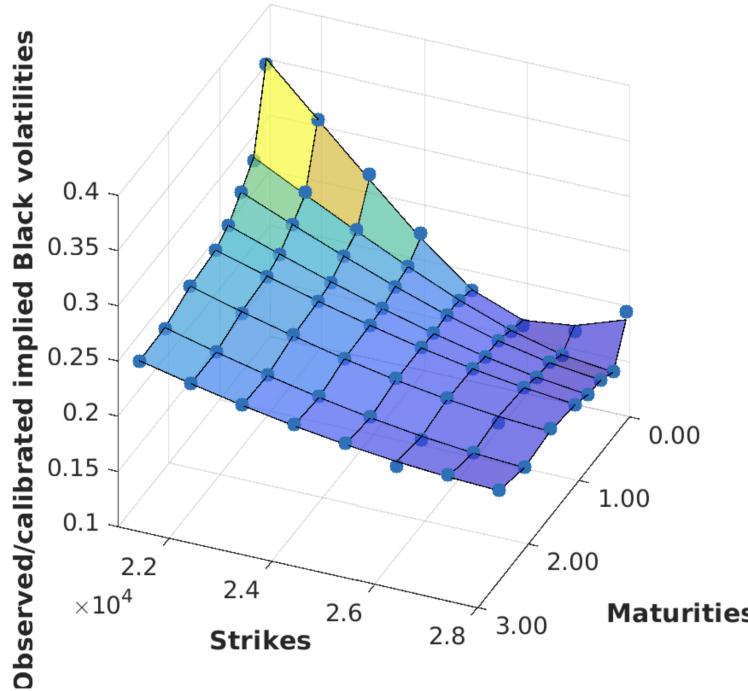


Figure 3.3: Implied volatilities and calibrated volatility surface

Here we present the algorithm employed to get these results:

Algorithm 11: Calibration algorithm (MATLAB)

```
% Import market files
S0 = 24486.45; NumMaturities = 8 ; Beta = 1
for k = 1 : NumMaturities
    objFun = @(X) MarketVolatilities(:,k) - blackvolbysabr(X(1), Betas(k),
        X(2), X(3), Settle, ExerciseDate(k), CurrentFwdValues(k),
        MarketStrike(:,k));
    X = lsqnonlin(objFun, [0.5 0 0.5], [0 -1 0], [Inf 1 Inf]);
    Alphas(k) = X(1);
    Rhos(k) = X(2);
    Nus(k) = X(3);
end
CalibratedParameters = [Alphas Betas Rhos Nus]
for k = 1 : NumMaturities
    ComputedVols(:,k) = blackvolbysabr(CalibratedParameters(k,1), ...
        CalibratedParameters(k,4), Settle, ExerciseDate(k),
        CurrentFwdValues(k), PlottingStrikes);
end
```

3.3.3 Monte Carlo and SABR: Pricing of a Barrier Call Option down & out

Clearly, the SABR model was developed to price derivatives on the interest rates like swaptions, caplets, florets... By changing the reference variable from the forward price f to S , we can introduce a SABR model which works on spot prices instead of

forward values, in order to price easily also equity derivatives. This approach can be found in several academic papers like [6], [5].

In the real measure of probability \mathbb{P} we have the following dynamic for the SABR:

$$\begin{cases} d\alpha_t = \nu \alpha_t dB_t^2 \\ dS_t = \mu S_t dt + \alpha_t S_t^\beta dB_t^1 \end{cases}$$

The SABR is now similar to the stochastic differential equation studied in the chapter 2; what changes is that now we have a volatility parameter alpha which is described itself by another differential equation, and other parameters β and ν which will change the slope and smile of the volatility surface. The two equations are connected by B^i , which are correlated Brownian motions, with correlation $\langle dB^1, dB^2 \rangle = \rho dt$. The SABR is a stochastic volatility model which reproduces the CEV if $\alpha = 0$, e $\rho = 0$.

Apply Girsanov

Let's riformulate the SABR model in terms of two uncorrelated Brownian Motions dW_t^1 and dW_t^2 with $\langle dW_t^1, dW_t^2 \rangle = 0$, as we did in section (3.2.3).

$$\begin{cases} d\alpha_t = \nu \alpha_t dW_t^2 \\ dS_t = \mu S_t dt + \alpha_t S_t^\beta (\sqrt{1 - \rho^2} dW_t^1 + \rho dW_t^2) \end{cases}$$

We impose through the Girsanov theorem (2.2.4) that the drift of the risky asset S is r :

$$\begin{cases} d\alpha_t = \nu \alpha_t dW_t^2 \\ dS_t = r S_t dt + \alpha_t S_t^\beta (\sqrt{1 - \rho^2} dW_t^1 + \rho dW_t^2 + \frac{(\mu - r) S_t^{1-\beta}}{\sigma_t} dt) \end{cases}$$

We can define in this way: $\lambda \equiv \frac{(\mu - r) S_t^{1-\beta}}{\sigma_t}$. So we obtain the following equations between the Brownian processes respectively in the measure \mathbb{P} and \mathbb{Q} .

$$\sqrt{1 - \rho^2} dW_t^1 + \rho dW_t^2 + \lambda dt = \sqrt{1 - \rho^2} d\tilde{W}_t^1 + \rho d\tilde{W}_t^2$$

In this case we have infinite ways to redefine the two BM in the measure \mathbb{Q} as Girsanov's Theorem allow us to do. The model is not complete.

One of the option , used also in practice is to impose a null drift of the volatility, mantaining dW_2 as before and defining only dW_1 as:

$$\begin{cases} \sqrt{1 - \rho^2} dW_t^1 + \lambda dt = \sqrt{1 - \rho^2} d\tilde{W}_t^1 \\ dW_t^2 = d\tilde{W}_t^2 \end{cases}$$

Where \tilde{W} is a Brownian motion in the martingale measure \mathbb{Q} .

The model under the risk neutral probability \mathbb{Q} is:

$$\begin{cases} d\alpha_t = \nu \alpha_t d\tilde{W}_t^2 \\ dS_t = r S_t dt + \alpha_t S_t^\beta (\sqrt{1 - \rho^2} d\tilde{W}_t^1 + \rho d\tilde{W}_t^2) \end{cases}$$

The parameters calibrated in the previous section will give us realistic values for our simulation even if not extremely precise since we calibrated them in the original forward model framework. Anyway, the parameters calibrated and the brownian

motions in the measure \mathbb{Q} define now our martingale measure properly (see lesson 6.1). It is possible to calculate finally the price of a call option as:

$$C_0 = e^{-rT} \mathbb{E}^{\mathbb{Q}}[(S_T - K)^+] \quad (3.8)$$

Now in order to implement properly a Monte Carlo evaluation for the derivative, we calculate the evolution of the stock price and of the volatility parameter *alpha* with a Euler discretization method (a similar discretization can be found in [8]):

$$\begin{cases} \Delta \alpha_{t_k} = \nu \alpha_{t_k} \Delta \tilde{W}_{t_k}^2 \\ \Delta S_{t_k} = r S_{t_k} \Delta t_k + \alpha_{t_k} S_{t_k}^{\beta} (\sqrt{1 - \rho^2} \Delta \tilde{W}_{t_k}^1 + \rho \Delta \tilde{W}_{t_k}^2) \end{cases}$$

So we can calculate iteratively:

$$\begin{cases} \alpha_{t_0} = \alpha_0 \\ \alpha_{t_{k+1}} = \alpha_{t_k} + \nu \alpha_{t_k} \Delta \tilde{W}_{t_k}^2 \end{cases}$$

and:

$$\begin{cases} S_{t_0} = S_0 \\ S_{t_{k+1}} = S_{t_k} + r S_{t_k} \Delta t_k + \alpha_{t_k} S_{t_k}^{\beta} (\sqrt{1 - \rho^2} \Delta \tilde{W}_{t_k}^1 + \rho \Delta \tilde{W}_{t_k}^2) \end{cases}$$

Once we have discretized our trajectories we can calculate the price for a call/put option for every strike and maturity presented in the previous section on the FTSE Mib index, with settle date "19-May-2021". Here we present a plot of the price of a call option for maturity "17-Sep-2021":

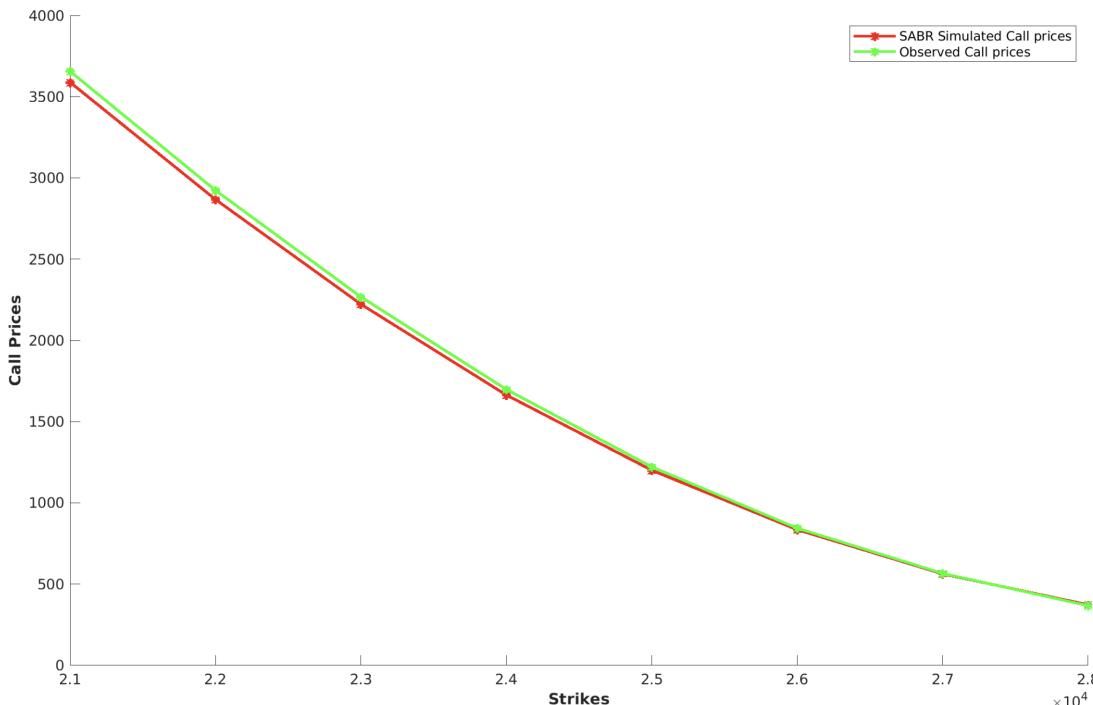


Figure 3.4: SABR simulated vs Observed EU call prices on FTSE Mib (Settle:"19-May-2021"; Expiry: "17-Sep-2022")

Using the Monte Carlo method, it is possible to calculate the price of other

exotic options, like a call option down out with barrier 21000, which has a slightly different payoff function from the simple EU call option, in which if the price of the underlying goes under a certain threshold (barrier), the payoff becomes zero:

$$C_0 = e^{-rT} \mathbb{E}^{\mathbb{Q}}[(S_T - K)^+] * \mathbb{I}_{\{\min S_t > B, \forall t > 0\}} \quad (3.9)$$

Here (fig. 3.5) we present an example of the trajectories of the underlying S which go to zero till maturity if they go below the Barrier:

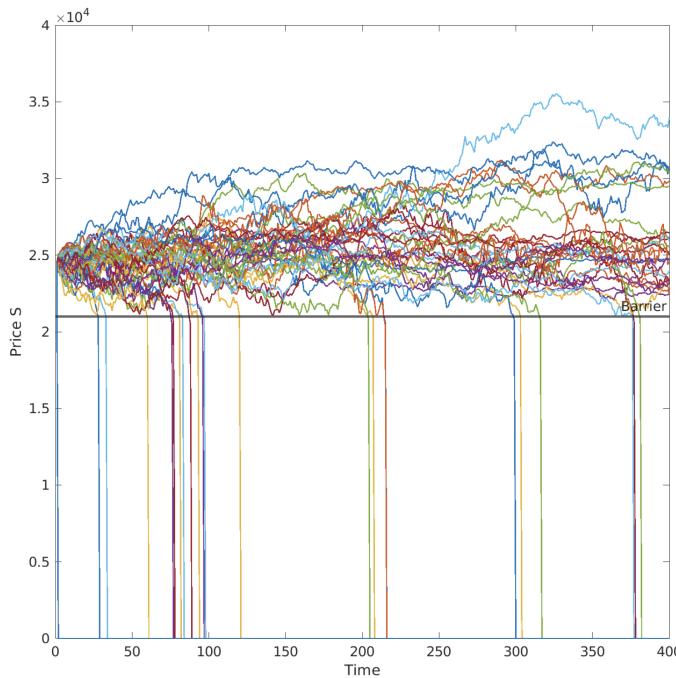


Figure 3.5: 50 Simulated trajectories with barrier (21000), FTSE Mib (Settle: "19-May-2021", expiry: "17-Sep-22"), 400 time intervals.

Here (fig. 3.6) we present a plot of the price of the barrier call option prices with maturity "17-Sep-2021":

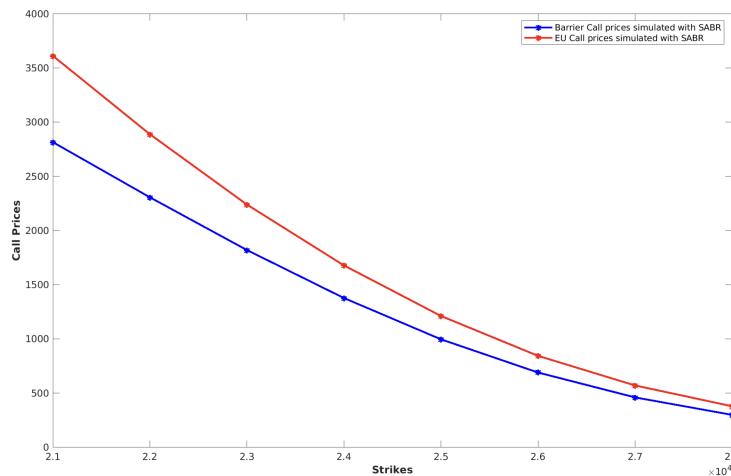


Figure 3.6: SABR simulated call prices with barrier (21000) vs SABR simulated EU call prices on FTSE Mib (Settle: "19-May-2021"; Expiry: "17-Sep-2022").

Here we present the algorithm used for the Monte-Carlo simulation and the Euler discretization used to price the derivatives under the SABR model:

Algorithm 12: Monte Carlo pricing of a Barrier option with SABR model
(MATLAB)

```

 $S_0 = 24486.45; r = -0.05; m = 10^5; nsim = 100;$ 
 $T = 1.0778; t = 400; dt = T/t; barrier = 21000;$ 
 $K = [21000, 22000, 23000, 24000, 25000, 26000, 27000, 28000];$ 
 $S = zeros(m,t); S_B = zeros(m,t); S(:,1) = S_0; S_B(:,1) = S_0;$ 
 $\alpha = zeros(m,t); \alpha(:,1) = 0.2129; \beta = 1; \rho = -0.6703; \nu = 0.8073$ 
for j = 1 : nsim
  for i = 1 : t-1
    dW1 = sqrt(dt) * randn(1,m);
    dW2 = sqrt(dt) * randn(1,m);
     $\alpha(:,i+1) = \alpha(:,i) + \nu * \alpha(:,i). * dW_2';$ 
    S(:,i+1) = S(:,i) + r*S(:,i)*dt +
     $(\alpha(:,i). * (S(:,i)^\beta). * (\sqrt{1 - \rho^2} * dW_1 + \rho * dW_2)');$ 
    for p = 1 : m-1
      if  $S_B(p,i) \leq barrier$ 
         $S_B(p,i+1) = 0;$ 
      else
         $S_B(p,i+1) = S_B(p,i) + r * S_B(p,i) * dt +$ 
         $(\alpha(p,i). * (S_B(p,i)^\beta). * (\sqrt{1 - rho^2} * dW_1(p) + \rho * dW_2(p))');$ 
      end
    end
  end
  for k = 1 : numel(K)
    Q_T(:,j) = SS2(:,end);
    Q_TB(:,j) = SS(:,end);
    payoffC = Q_T - K(k);
    payoffC_B = Q_TB - K(k);
    C = exp(-r*T)*mean(max(payoffC,0));
    C_B = exp(-r * T) * mean(max(payoffC_B,0));
    Cmean(k) = mean(C);
    Cmean_B(k) = mean(C_B)
  end
end

```

Conclusion

Having analyzed the probability background and the pricing theory behind the option pricing applications allowed us a more accurate and aware use of the Monte Carlo Method and of the risk neutral pricing. For example, even if for simplicity we used the generators already implemented in MATLAB, the sampling techniques and the generators exposed in the first chapter allowed us to understand why these kind of tools are available today and what is the interesting theory behind them.

We also saw that most of the results valid for a discrete time model are valid also for a continuous time one, which better approximates the situation in the reality. Options arise very interesting problems for researchers and practitioners: a correct evaluation of these contracts is crucial for the latter ones and it cannot be done properly without the theoretical notions of hedging, arbitrage and calibration. The binomial model remains a relatively easy and quick tool to price basic options. It is also an excellent educational starting point for understanding more advanced theories and models.

Monte Carlo methods proved to be a useful tool to price in continuous time European options with an Asian payoff and barrier options, which often do not have an explicit solution for their calculation. The Monte Carlo convergence to the mean has been reached with a number of simulation which can be quickly computed without too many simulations needed (10^5 - 10^6). The variance reduction techniques used in the thesis proved to be another very useful and easy-to-implement tool and helped us to get a good idea of the value of the options.

If we want to apply more advanced models like the SABR in practice, we need to implement a method for the calibration of the parameters.

The volatility surface observed on the European call options on the FTSE Mib has been satisfactorily well approximated by the SABR model, even though this turns out to be originally designed to price interest rate derivatives. We decided to use SABR for equity option pricing without introducing new theory on the interest rate derivatives; we chose to give the priority to the task of simulating with Monte Carlo, using anyway a well known model that provided us with a simple dynamic for the stochastic volatility and an easy calibration procedure thanks to the closed formula for the implied volatility. As we expected, the volatility surface is not empirically flat but is generally higher for more extreme strikes (deeply at or out of the money options) and tends to decrease for more distant maturities; the choice of going beyond B & S allowed us to get closer to a real market structure.

Appendix A

Appendices

A.1 Definitions

A.1.1 Expected value

If X is a r.v. with finite discrete distribution, the expected value of X over a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is defined by:

$$E[X] := \sum_{k=1}^m x_k P(X = x_k) \quad (\text{A.1})$$

If X takes infinite non-countable values, then the expected value will be defined as the integral of X (continuous r.v.) with respect to the measure of probability P :

$$\int_{\Omega} X dP(\omega) \quad (\text{A.2})$$

Properties of the integral of Random Variables in \mathbb{R}^d :

i) Linearity: $\forall X, Y \in L^1(\Omega, \mathbb{P})$ and $\alpha, \beta \in \mathbb{R}$ we have that:

$$\int_{\omega} (\alpha X + \beta Y) dP = \alpha \int_{\omega} X dP + \beta \int_{\omega} Y dP \quad (\text{A.3})$$

ii) Monotonicity: $\forall X, Y \in L^1(\Omega, \mathbb{P})$ such that $X \leq Y$ P -a.s. ($P(X > Y) = 0$) we have that:

$$\int_{\omega} X dP \leq \int_{\omega} Y dP \quad (\text{A.4})$$

iii) σ -additivity: Let $A = \bigcup_{n \in \mathbb{N}} A_n$ where $(A_n)_{n \in \mathbb{N}}$ disjoint sequence in \mathcal{F} . If $X \in m\mathcal{F}^+$ or $X \in L^1(\Omega, \mathbb{P})$ then we have that:

$$\int_{\omega} X dP = \sum_{n \in \mathbb{N}} \int_{A_n} X dP \quad (\text{A.5})$$

A.1.2 L^P Spaces and norm

In general, we say that the L^p -spaces are defined as the set of measurable functions f such that the integral $\int |f(x)|^p dx$ is convergent ($\sum |f(x)|^p dx < +\infty$ for discrete variables).

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $p \geq 1$. The p-norm of a R.V. X is defined as:

$$\|X\|_p := (\mathbb{E}[|X|^p])^{\frac{1}{p}} \quad (\text{A.6})$$

we say that $L^p(\Omega, \mathbb{P})$ is the vectorial space of summable r.v. of order p:

$$L^p(\Omega, \mathbb{P}) = \{X \in m\mathcal{F} \mid \|X\|_p < +\infty\}$$

$\|X\|_p$ is defined as a semi-norm on the $L^p(\Omega, \mathbb{P})$ space, if it's true that:

- i) $\|X\|_p = 0$ iff $X \stackrel{a.s.}{=} 0$
- ii) $\|\lambda X\|_p = |\lambda| \|X\|_p \forall \lambda \in \mathbb{R}$ and $X \in L^P \in (\Omega, \mathbb{P})$;
- iii) It is valid the Minkowski inequality (for proof see A.3.5):

$$\|X + Y\|_p \leq \|X\|_p + \|Y\|_p$$

$\forall \lambda \in \mathbb{R}$ and $X, Y \in L^P \in (\Omega, \mathbb{P})$.

A.2 Convergences

A.2.1 Almost Sure Convergence: $X_n \xrightarrow{a.s.} X$

Let's consider a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ over which are defined a sequence of R.V. $(X_n)_{n \in \mathbb{N}}$ and a random variable $X \in \mathbb{R}^d$.

$(X_n)_{n \in \mathbb{N}}$ converges "almost sure" to X if:

$$\mathbb{P}\left(\lim_{n \rightarrow +\infty} X_n = X\right) = 1 \quad (\text{A.7})$$

where: $(\lim_{n \rightarrow +\infty} X_n = X) \equiv \{\omega \in \Omega \mid \lim_{n \rightarrow +\infty} X_n(\omega) = X(\omega)\}$ is an event.

A.2.2 L^p Convergence: $X_n \xrightarrow{L^p} X$

Let's consider a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ over which are defined a sequence of R.V. $(X_n)_{n \in \mathbb{N}}$ and a random variable $X \in \mathbb{R}^d$.

Let $(X_n)_{n \in \mathbb{N}}$ and X be in $L^p(\Omega, \mathbb{P})$ with $p \geq 1$. We say that $(X_n)_{n \in \mathbb{N}}$ converges to X in L^P norm if:

$$\lim_{n \rightarrow +\infty} \mathbb{E}[|X_n - X|^p] = 0 \quad (\text{A.8})$$

In this case we write: $X_n \xrightarrow{L^p} X$.

This is also called "convergence in p-th mean" and tells us that the expectation of the p-th power of the difference between X_n and X converges to zero.

The most important cases of convergence in p-th mean are:

- When X_n converges in p-th mean to X for $p = 1$, we say that X_n converges in mean to X .

- When X_n converges in p-th mean to X for $p = 2$, we say that X_n converges in mean square (or in quadratic mean) to X .

It is also worth noticing that if:

$$X_n \xrightarrow{L^p} X \Rightarrow \lim_{n \rightarrow \infty} E[|X_n|^p] = E[|X|^p] < +\infty \text{ (is bounded - see definition of } L^P \text{ space A.1)}$$

A.2.3 Convergence in Probability: $X_n \xrightarrow{P} X$

Let's consider a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ over which are defined a sequence of R.V. $(X_n)_{n \in \mathbb{N}}$ and a random variable $X \in \mathbb{R}^d$.

X_n converges in probability to X if, $\forall \epsilon > 0$, it is true that:

$$\lim_{n \rightarrow \infty} \mathbb{P}(|X_n - X| \geq \epsilon) = 0 \quad (\text{A.9})$$

In this case we write: $X_n \xrightarrow{P} X$.

A.2.4 Convergence in Distribution: $X_n \xrightarrow{d} X$

Let's consider a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ over which are defined a sequence of R.V. $(X_n)_{n \in \mathbb{N}}$ and a random variable $X \in \mathbb{R}^d$.

We say that $(X_n)_{n \in \mathbb{N}}$ converges to X in law or in distribution if:

$$\lim_{n \rightarrow +\infty} \mathbb{E}[f(X_n)] = \mathbb{E}[f(X)], \quad (\forall f \in bC) \quad (\text{A.10})$$

where $bC = bC(\mathbb{R}^d)$ is the family of continuous and bounded functions from $\mathbb{R}^d \rightarrow \mathbb{R}$.

In this case we write: $X_n \xrightarrow{d} X$.

A.2.5 Relations between Convergences

Let's consider a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ over which are defined a sequence of R.V. $(X_n)_{n \in \mathbb{N}}$ and a random variable $X \in \mathbb{R}^d$.

The following implications are true:

- i) If $X_n \rightarrow X$ (pointwise) $\Rightarrow X_n \xrightarrow{a.s.} X$;
 - ii) If $X_n \xrightarrow{a.s.} X \Rightarrow X_n \xrightarrow{P} X$ (direct consequence of A.3.6);
 - iii) If $X_n \xrightarrow{L^p} X$ (for some $p \geq 1$) $\Rightarrow X_n \xrightarrow{P} X$;
 - iv) If $X_n \xrightarrow{P} X \Rightarrow \exists$ a sub-sequence $(X_{n_k})_{k \in \mathbb{N}} \mid X_{n_k} \xrightarrow{a.s.} X$;
 - v) If $X_n \xrightarrow{P} X \Rightarrow X_n \xrightarrow{d} X$;
 - vi) If $X_n \xrightarrow{P} X$ and $\exists Y \in L^p(\Omega, \mathbb{P}) \mid |X_n| \leq Y(a.s. \forall n \in \mathbb{N}) \Rightarrow X_n \xrightarrow{L^p} X$
- with X_n sequence of r.v. $X \in L^p(\Omega, \mathbb{P})$;

vii) If $X_n \xrightarrow{d} X$ (with $X \sim \delta_c, c \in \mathbb{R}^d$) $\Rightarrow X_n \xrightarrow{a.s.} X$.

A.3 Inequalities

A.3.1 Jensen's Inequality

Extension of the triangular inequality. Let:

- $-\infty \leq a < b \leq +\infty$;
- $X : \Omega \rightarrow]a, b[$ a r.v. over the space $(\Omega, \mathcal{F}, \mathbb{P})$;
- $f :]a, b[\rightarrow \mathbb{R}$ a convex function;

If X and $f(X) \in L^1(\Omega, \mathbb{P}) \rightarrow \|X\| < +\infty$, then we have that:

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]. \quad (\text{A.11})$$

For proof see [30]

A.3.2 p-Norms' Inequalities

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $p \geq 1$. If $1 \leq p_1 \leq p_2$ then:

$$\|X\|_{p_1} \leq \|X\|_{p_2}$$

So:

$$L^{p_2}(\Omega, \mathbb{P}) \subseteq L^{p_1}(\Omega, \mathbb{P})$$

Proof The proposition is a direct consequence of the Jensen's Inequality with $f(x) = x^q$, $x \in [0, \infty[, q = \frac{p_2}{p_1} \geq 1$; so we have that:

$$\mathbb{E}[|X|^{p_1}]^q \leq \mathbb{E}[|X|^{p_1 q}] \Rightarrow \mathbb{E}[|X|^{p_1}]^{\frac{p_2}{p_1}} \leq \mathbb{E}[|X|^{p_2}] \Rightarrow \mathbb{E}[|X|^{p_1}]^{\frac{1}{p_1}} \leq \mathbb{E}[|X|]^{\frac{1}{p_2}}$$

□

A.3.3 Holder's Inequality

Let $p, q > 1$ be exponents conjugates $\Rightarrow \frac{1}{p} + \frac{1}{q} = 1$.

If $X \in L^p(\Omega, \mathbb{P})$ and $Y \in L^q(\Omega, \mathbb{P})$, it is true that:

- i) $XY \in L^p(\Omega, \mathbb{P})$
- ii)

$$\|XY\|_1 \leq \|X\|_p \|Y\|_q \quad (\text{A.12})$$

For proof see [30]

A.3.4 Cauchy Schwarz Inequality (Corollary of Holder)

Let $p, q > 1$ be exponents conjugates $\Rightarrow \frac{1}{p} + \frac{1}{q} = 1$.

If $X \in L^p(\Omega, \mathbb{P})$ and $Y \in L^q(\Omega, \mathbb{P})$ and iff $\exists a \in \mathbb{R} | X \stackrel{a.s.}{=} aY$ it is true that:

$$|\mathbb{E}[XY]| \geq \|X\|_2 \|Y\|_2 \quad (\text{A.13})$$

For proof see [30]

A.3.5 Minkowsky's Inequality

Let $\|X\|_p$ be a semi-norm on the $L^p(\Omega, \mathbb{P})$ space, then is true that:

$$\|X + Y\|_p \leq \|X\|_p + \|Y\|_p \quad (\text{A.14})$$

$\forall \lambda \in \mathbb{R}$ and $X, Y \in L^p(\Omega, \mathbb{P})$.

For proof [30]

A.3.6 Markov's Inequality

$\forall X$ r.v. $\in \mathbb{R}^d$, $\lambda > 0$ and $p \in [0, +\infty]$, it is true that:

$$P(|X| \geq \lambda) \leq \frac{\mathbb{E}[|X|^p]}{\lambda^p}$$

(A.15)

Proof If $\mathbb{E}[|X|^p] = +\infty$, (A.3.6) is always true.

If $\mathbb{E}[|X|^p] \neq +\infty$ then for the property of monotonicity ($|X| \geq \lambda \Rightarrow \int_{\omega} X dP \geq \int_{\omega} \lambda dP$), we have that:

$$\mathbb{E}[|X|^p] \geq \mathbb{E}[|X|^p \mathbb{I}_{|X| \geq \lambda}] \geq \lambda^p \mathbb{E}[\mathbb{I}_{|X| \geq \lambda}] = \lambda^p \mathbb{P}(|X| \geq \lambda)$$

□

Proof The statement follows by fixing $p=2$ and $X = Y - E[Y]$:

$$\mathbb{P}(|Y - E[Y]| \geq \lambda) \leq \frac{E[|Y - E[Y]|^2]}{\lambda^2} = \frac{\text{var}[Y]}{\lambda^2} \quad (\text{A.16})$$

□

A.4 Characteristic Function

Let X be a random variable, its characteristic function of a real variable η is a function:

$$\varphi_X(\eta) = E[e^{i\eta X}] = \int_{\mathbb{R}} e^{i\eta x} \phi_X(x) dx \quad \text{with } \phi \text{ probability density of } X \text{ r.v.}$$

$$\varphi_X : \mathbb{R} \Rightarrow \mathbb{C}$$

$$\eta \Rightarrow E[e^{i\eta X}]$$

Where the integral is a Fourier transformed.

A.4.1 Characteristic Function: Normal case

In general for a normal r.v $X \sim \mathcal{N}_{\mu, \sigma^2} \Leftrightarrow \varphi(\eta) = E[e^{i\eta X}] = e^{i\eta\mu - \frac{\sigma^2}{2}\eta^2}$

For a standard normal r. v. we have: $X \sim \mathcal{N}_{0,1} \Leftrightarrow \varphi(\eta) = E[e^{i\eta X}] = e^{-\frac{1}{2}\eta^2}$

The characteristic function allows for $\sigma = 0$ in which the function become the characteristic function of a delta di Dirac: $\varphi(\eta) = e^{i\eta\mu}$

A.4.2 Characteristic Function: Browninan motion case

In the Brownian motion case, the characteristic function where $X = W_t \sim \mathcal{N}(0, t)$ r.v. is:

$$\varphi_{W_t}(\eta) = E[e^{i\eta W_t}] = \int_{\mathbb{R}} e^{i\eta x} \phi_X(t, x) dx = \hat{\phi}(t, \eta) = e^{-\frac{t\eta^2}{2}}$$

A.5 Ito's Formula in the stochastic case

If $F = F(t, x) \in (\mathbb{R}^2)$ then $X_t = F((t, W_t))$ is an Ito Process and it's true that:

$$F(t, W_t) = F(0, W_0) + \int_0^t (\partial_t F)(s, W_s) ds + \int_0^t \partial_x F(s, W_s) dW_s + \frac{1}{2} \int_0^t (\partial_{xx} F)(s, W_s) ds$$

Idea of the demonstration is related to a second order development in x with Taylor (for the full proof see [31], [2]):

A.5.1 General Ito Formula

If X is an Ito process:

$$dX_t = u_t dt + v_t dW_t$$

and F is: $F(t, x) \in C^2(\mathbb{R}^2)$ (it's enough $C^{1,2}$) then:

$$\begin{aligned} dF(t, W_t) &= (\partial_t F)(t, W_t)dt + (\partial_x F)(t, W_t)dX_t + \frac{1}{2}(\partial_{xx} F)(t, W_t)d\langle X \rangle_t \\ &= \partial_t F dt + \partial_x F u_t dt + \partial_x F v_t dW_t + \frac{1}{2}\partial_{xx} F v^2 dt \\ &= (\partial_t F + \partial_x F u_t + \frac{1}{2}\partial_{xx} F v^2)dt + \partial_x F v_t dW_t \end{aligned}$$

Where $d\langle X \rangle_t$ is the quadratic variation process (for more details see [31]) of X_t , where $\langle X \rangle_t = \int_0^t v_s^2 ds = v_t^2 dt$

A.5.2 Geometric Brownian motion and Ito

Let's try to calculate the stochastic differential of the logarithm of S_t :

$$\begin{aligned} d\log(S_t) &= \frac{\partial \log(S_t)}{\partial t} + \frac{\partial \log((S_t))}{\partial (S_t)}dS_t + \frac{1}{2}\frac{\partial^2 \log(S_t)}{\partial S_t^2}d\langle S_t \rangle_t \\ &= 0 + \frac{1}{S_t}dS_t - \frac{1}{S_t^2}\sigma^2 S_t^2 dt \\ &= \frac{1}{S_t}(\mu S_t dt + \sigma S_t dW_t) - \frac{1}{S_t^2}(\sigma S_t)^2 dt \\ &= (\mu - \frac{\sigma^2}{2})dt + \sigma dW_t \end{aligned}$$

where:

$$dS_t = \mu S_t dt + \sigma S_t dW_t \quad \text{with: } \mu, \sigma \in \mathbb{R}, \sigma > 0$$

$$F(t, S_t) = \log(S_t)$$

$$\partial_t F(t, S_t) = 0$$

$$\partial_x F(t, S_t) = \frac{1}{S_t}$$

$$\partial_{xx} F(t, S_t) = -\frac{1}{S_t^2}$$

$$d\langle S_t \rangle_t = (\sigma S_t)^2 dt$$

Once we obtained this stochastic differential equation, we rewrite it in the integral form (for a time interval $[0, t]$) to get the final solution:

$$\begin{aligned} \log(S_t) - \log(S_0) &= \int_0^t (\mu - \frac{\sigma^2}{2})ds + \int_0^t \sigma dW_s \\ \log(\frac{S_t}{S_0}) &= (\mu - \frac{\sigma^2}{2})t + \sigma W_t \\ S_t &= S_0 e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t} \end{aligned}$$

Bibliography

- [1] BACHELIER, L. J. B. Théorie de spèculation. *Annales scientifiques de l'É.N.S.* 3e série 17 (1900), 21–86.
- [2] BJORK, T. *Arbitrage Theory in Continuous Time*. Oxford Finance, Evry, France(2013).
- [3] BLACK-SCHOLES. The pricing of options and corporate liabilities. *Journal Of Political Economy* 81 (1973), 637–654.
- [4] BOYLE, P. P. Options: A monte carlo approach. *Journal Of financial Economics* 4, Canada (1976), 323–338.
- [5] CHEN-GRZELAK-OOSTERLEE. Calibration and monte carlo pricing of the sabr-hull-white model for long-maturity equity derivatives. *Journal of Computational Finance* 15 (2012), 79–113.
- [6] CHEN-OOSTERLEE-VANDERWEIDE. A low-bias simulation scheme for the sabr stochastic volatility model. *International Journal of Theoretical and Applied Finance* 15 (2012), No. 2.
- [7] COX-Ross. The valuation of options for alternative stochastic processes. *Journal of Financial Economics* 3 (1976), 145–166.
- [8] CRISPOLDI-WIGGER-LARKIN. *SABR and SABR LIBOR Market Models in Practice: With Examples Implemented in Python*. Applied Quantitative Finance, (2015).
- [9] CRÉPEY, S. *Financial Modeling: A Backward Stochastic Differential Equations Perspective*. Springer Finance, Stockholm (2019).
- [10] DURRETT, R. *Probability: theory and examples, second ed.* Duxbury Press, Belmont, CA (1996).
- [11] EICHENAUCHER-HERRMAN-HERMANN-WEGENKITTL. *A Survey of Quadratic and Inversive Congruential Pseudorandom Numbers, in Monte Carlo and quasi Monte Carlo Methods in Scientific Computing, Hellekalek, Larcher, Niederreiter, Zinterhof*. Springer-Verlag, Berlin (1998).

- [12] FISHMAN, G. S. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag, (1996).
- [13] FISHMAN-MOORE. An exhaustive analysis of multiplicative congruential random numbers generators with modulus $2^{31} - 1$. *SIAM Journal of Scientific and Statistical Computing* 7 (1986), 24–25.
- [14] GENTLE, J. E. *Random Number Generation and Monte Carlo Methods, Second Edition*. Springer, (2005).
- [15] GLASSERMAN, P. *Monte Carlo methods in financial engineering*. Springer-Verlag, New York (2004).
- [16] GOBET, E. *Monte Carlo methods and Stochastic Processes: From Linear to Non-Linear*. CRC Press, Ecole Polytechnique - University of Paris-Saclay, France (2016).
- [17] HAGAN-KUMAR-LESNIEWSKI-WOODWARD. Managing smile risk. *Wilmott September* (2002), 84–108.
- [18] HENRY-LABORDERE. *Analysis, Geometry and modeling in Finance: Advanced methods in Option Pricing*. Chapman and Hall, (2008).
- [19] HULL-WHITE. The use of the control variate technique in option pricing. *Journal of Financial and Quantitative Analysis* 23 (1988), 237–251.
- [20] KEMNA-VORST. A pricing method for options based on average asset values. *Journal of Banking and Finance* 14 (1990), 113–129.
- [21] KNUTH, D. *The Art of Programming, Volume II: seminumerical algorithms, third edition*. Addison Wesley Longman, Reading, Massachussets (1998).
- [22] KOLMOGOROV, A. Uber die summen durch den zufall bestimmter unabhängiger grossen. *Math. Ann.* 99 (1928), 309–319.
- [23] L'ECUYER, P. Efficient and portable combined random number generators. *Communications of the ACM* 31 (1988), 742–749.
- [24] L'ECUYER, P. Combined multiple recursive random number generators. *Journal of Modern Applied Statistical Methods Operations Research* (1996), 77–120.
- [25] LEHMER, D. H. Mathematical methods in large scale computing units. *Proceedings of the Second Symposium on Large Scale Digital Computing Machinery* (Cambridge (1951)), 141–146.
- [26] LEWIS-GOODMAN-MILLER. A pseudo random number generator for the system/360. *IBM Systems of Journal* 8 (1969), 136–146.

- [27] MARSAGLIA-BRAY. A convenient method for generating random variables. *SIAM 6* (1964), 260–264.
- [28] MATSUMOTO-NISHIMURA. Mersenne tister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation 8* (1998), 3–30.
- [29] MERTON, R. C. Theory of rational option pricing. *Bell J. Econom. and Management Sci. 4* (1973), 141–183.
- [30] PASCUCCI, A. *Teoria della Probabilità*. Springer, Bologna, Italy (2020).
- [31] PASCUCCI, A. *PDE and Martingale Methods in Option Pricing*. Springer - Verlag, Italy (2011).
- [32] PROTTER, J. J. *Probability Essentials*. Springer-Verlag, Berlin(2000).
- [33] SAWILOWSKY, S. S. You think you've got trivials? *Journal of Modern Applied Statistical Methods 2*, Wayne State University (2003), 218–225.
- [34] TAUSWORTHE, R. Random numbers generated by linear recurrence modulo two. *Mathematics of Computation 19* (1965), 201–209.
- [35] VAJARGAH-SALIMIPOUR-SALAHSHOUR. Variance analysis of control variate technique and applications in asian option pricing. *Int. J. Industrial Mathematics 8* (2015).
- [36] VONNEUMANN, J. Various techniques used in connection with random digits. *Applied Mathematics Series 12* (1951).