**University of St. Gallen**
Master in Banking and Finance

Referee: Prof. Dr. Karl Frauendorfer

**Master's thesis**

# Monte Carlo methods: with application to the pricing of interest rate derivatives

Roman Frey*

July 21, 2008

*roman.frey@student.unisg.ch / roman.frey@gmx.ch

**Abstract**

This paper provides an extensive treatment of the entire Monte Carlo simulation theory. Furthermore, the Monte Carlo technique is used for addressing the pricing of various interest rate derivatives in different term structure models by the simulation approach. With the rising complexity and diversity of upcoming derivative securities, analytically tractable or closed-form pricing methods are difficult to find or even inexistent. If the thoroughly popular lattice valuation approach additionally fails due to non-recombining characteristics, Monte Carlo simulation represents a powerful and flexible alternative pricing method.

The goal of this paper is to discuss and implement the fundamentals of Monte Carlo methods and to introduce the wide use of this approach in finance, especially in interest rate derivative valuation. The paper is roughly divided into three parts. The first part focuses on random number generation and on increasing efficiency methods for Monte Carlo, such as variance reduction techniques or low-discrepancy sequences. In the following part different term structure models are developed and the link to the simulation theory is eventually established. In the third and final part some ordinary and extended Monte Carlo algorithms are implemented and corresponding simulations are run in order to analyze Bermudan swaption prices in detail.

Even though Monte Carlo methods feature a relatively slow but given convergence rate, they remain a competitive tool in financial applications. They owe their rising popularity to a large extent to their flexibility and to recent progress in methods which improve their accuracy and precision in estimating quantities of interest. Moreover, some of the leading yield curve models are heavily relying on Monte Carlo techniques. Several extensions of the standard Monte Carlo approach, such as least-squares Monte Carlo, for instance, are able to overcome the early-exercise hurdle and thus capable of pricing American-style options by simulation. Corresponding results for Bermudan swaptions are found to be robust, unbiased and consistent.

## Contents

## List of Tables

## List of Figures

# 1 Fundamentals

This chapter's purpose is to give a first introduction into the very useful tool Monte Carlo and to develop the key ideas of the simulation technique, its properties as well as its limits. Monte Carlo methods are generally applied in a wide field of areas, including physics, mathematics, statistics, finance and so on.

However, this paper will mainly focus on its application in finance, especially the representation of rather complex derivative prices as expectations, as it is fundamentally done by Glasserman [46]. This representation then will provide a basis for the application of Monte Carlo to the pricing of derivatives.

## 1.1 Principles of Monte Carlo

### 1.1.1 Introduction

The Monte Carlo method was introduced by Count Buffon who, in 1777 [23], proposed a new, at that early period exotic as well as pioneering method of calculating $\pi$. This basically simple though subtle method is relying on the execution of repeated performances what we will see in Section 1.1.3 in detail. However, the actual beginning or breakthrough of the Monte Carlo method dates back to the introduction of the first computers and their application during the years 1940-45 with the purpose of computing neutron diffusion in atomic bombs. The Monte Carlo approach was primarily promoted and advanced by physics researchers Stanislaw Ulam, Nicholas Metropolis and John von Neumann. They were totally aware of the power and importance of statistical sampling techniques and already saw the unlimited potential that comes with them even though experiments including repeated samplings had fallen, due to the length and tediousness of the calculations, into impracticality. Together with the first electronic computer - the *ENIAC* - which was able to solve differential equations at a tremendous, so far unconceivable speed, the Monte Carlo method was eventually triggered [79, p.125ff]. Metropolis and Ulam published the first paper *The Monte Carlo Method* [80] in the year 1949. Their work generally studies methods of resolutions for differential equations that occur in several fields of the natural sciences. The name *Monte Carlo* refers to a famous casino in Monaco where Ulam's uncle used to indulge his gambling passion. The characteristics of randomness and the repetitive nature of many processes correspond with the games played in a casino [42, p.15ff]. Note that the famous roulette wheel is one of the simplest mechanical devices to generate random variates.

Simulation tools find their application mostly in cases where the inherent complexity of a problem makes the use of other techniques impossible or where no analytically tractable solution with a deterministic algorithm is existing. In short, the Monte Carlo method is a numerical approach which aims for solving mathematical problems by the simulation of so called *random variables*. Therefore, Monte Carlo offers a direct alternative to analytical mathematics. It makes it possible both to understand the sampling distribution of a quantity of interest and to evaluate a statistic's behavior in random samples, for instance. It started to be en vogue in financial mathematics in the 1980s, particularly when the theories of the random walk of

asset prices came up. Monte Carlo's key idea is to estimate a particular parameter $\theta$, with a corresponding estimator $\hat{\theta}$, computed from observed, empirical data. Thereby the Monte Carlo estimator $\hat{\theta}$ is expected to meet the important and intuitive criteria *unbiasdness*, *efficiency* and *consistency* [82, p.1ff].

Monte Carlo simulations basically consist of creating artificial history and help to understand a sequence of random historical events. It is an instrument to toy with uncertainty [100, p.44-45].

### 1.1.2 Monte Carlo integration

The equivalence between probability and volume establishes a basis for Monte Carlo methods. The intuitive notion of an event's probability is defined as its volume or measure relative to that of an universe of possible outcomes [46, p.1]. In probability theory [19] in general, a random variable is formalized as a certain function that is defined over the set of all possible outcomes, referred to as sample space $\Omega$. This sample space again takes values of another set $E$. Furthermore, $\omega$ is defined as a general element of $\Omega$ while $X$ simply denotes a random variable [67, p.1f]:

$$X : \Omega \to E.$$

In all cases that are taken into account here, $E$ will be equal to $\mathbb{R}^d$. Consider, now, the different realizations $\omega$ of $\Omega$. For this purpose, the notion of the *probability* $P$ is introduced. The probability represents the positive measure on $\Omega$ with a total mass of 1. When $X$ takes values in $\mathbb{R}$ (or more precisely in $\mathbb{R}^d$), $P$ enables us to compute the *expected value* of $X$ [67, p.2]:

$$E(X) = \int_{\Omega} X(\omega) dP(\omega). \tag{1.1}$$

Considering the distributional characteristics of $X$ in a further step, one recognizes that the random variable $X$ is basically an image measure of the probability $P$ under the mapping of $X$. Consequently, the distribution of $X$ under $P$, which we denote by $h_X$, is characterized by the following property [15, 67, p.221f, p.2f]:

$$E[f(X)] = \int_{E} f(x) dh_X(x). \tag{1.2}$$

Finally, $n$ random variables $X_1, \ldots, X_n$ are independent if the following property holds for all positive functions $f_1, \ldots, f_n$ [67, p.2f]:

$$E[f_1(X_1) \ldots f_n(X_n)] = E[f_1(X_1)] \ldots E[f_n(X_n)].$$

Monte Carlo, however, uses this general identity inversely [46, p.2ff]. It calculates the volume of a set by interpreting the volume of a probability. And from calculating volumes, only small steps take us to integrals. Integration generally puts the quantity of interest in the form of an expected value. Let us consider, for example, a high dimensional integral of the function $f(u_1, \ldots, u_d)$ that depends on $d$ variables $u_1, \ldots, u_d$ over the so called unit hypercube $[0, 1]^d$, denoted as $\mathcal{A}$. Furthermore, we assume that $f$ is integrable (or to be more precise, square-integrable). As a shorthand notation a point in the unit hypercube is denoted by

$x = (u_1, \ldots, u_d)$, where $u_1, \ldots, u_d$ are independent random variables distributed uniformly on the interval $[0, 1]$, and a function evaluated at the point $x$ is indicated by $f(x) = f(u_1, \ldots, u_d)$. In a next step, consider the problem of computing this multidimensional integral of the form [15, p.222f]

$$I = \int_{\mathcal{A}} f(x)dx = \int_{\mathcal{A}} f(u_1, \ldots, u_d)du_1 \ldots du_d. \tag{1.3}$$

We may estimate $I$ by randomly sampling a sequence of points $x_i \in \mathcal{A}, i = 1, \ldots, n$, and building the Monte Carlo estimator $\hat{I}_n$ of $I$

$$\hat{I}_n = \frac{\text{vol}(\mathcal{A})}{n} \sum_{i=1}^{n} f(x_i). \tag{1.4}$$

where $\text{vol}(\mathcal{A})$ denotes the volume of the region $\mathcal{A}$. Since we consider the unit hypercube $\mathcal{A} = [0, 1]_1 \times [0, 1]_2 \times \cdots \times [0, 1]_d$, $\text{vol}(\mathcal{A}) = 1$ [15, p.223].

If $f$ is indeed square integrable over $[0, 1]^d$ then the strong *law of large numbers* ensures that the Monte Carlo estimate converges to the true value of the integral:

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} f(x_i) = I. \tag{1.5}$$

This means that $\hat{I} \to I$ with the probability 1 as $n \to \infty$. $\hat{I}$ is therefore an *unbiased* estimator for $I$ [67, p.2ff].

Until now we know that we may obtain an estimate for an integral by Monte Carlo which is asymptotically correct. Nevertheless, we still do not know anything about the speed of convergence or about the accuracy and precision of our estimate. The following sections will shed some light on that topic.

### Limits and convergence

Equation (1.5) showed that the Monte Carlo estimate converges to the correct value as the number of draws $n$ is increased. Since the Monte Carlo estimate is only an approximation with randomness included, we have to allow for an approximation error, which has been playing a decisive role in the entire simulation theory and research. In oder to consider an error estimate for a finite $n$, the variance $\sigma_f^2$ of the function $f(x)$ is introduced [67, p.3]:

$$\sigma_f^2 = \int_{\mathcal{A}} (f(x) - I)^2 dx. \tag{1.6}$$

By some simple transformations it can be shown [67, p.5] that

$$\int dx_1 \ldots \int dx_n \left( \frac{1}{n} \sum_{i=1}^{n} f(x_i) - I \right)^2 = \frac{\sigma_f^2}{n}. \tag{1.7}$$

Due to the *central limit theorem* the standardized estimator $(\hat{I} - I)/(\sigma_f/\sqrt{n})$ converges in distribution to the standard normal as the number of replications $n$ increases. This fact is often

expressed as [46, p.10]:

$$\frac{\hat{I} - I}{\sigma_f/\sqrt{n}} \;\Rightarrow\; \phi(0,1). \tag{1.8}$$

The quality of this approximation is thus improving with an increasing $n$. Accordingly, the error $\hat{I} - I = \varepsilon$ in the Monte Carlo estimate is itself a random variable and approximately normally distributed with mean zero and standard deviation $\sigma_f/\sqrt{n}$ [46, p.10]:

$$\hat{I} - I \;\approx\; \phi(0, \sigma_f^2/n). \tag{1.9}$$

Our Monte Carlo estimate now lies between $I - a\sigma_f/\sqrt{n}$ and $I + b\sigma_f/\sqrt{n}$ with the probability of [67, p.5]

$$\lim_{n\to\infty} P\left(-a\frac{\sigma_f}{\sqrt{n}} \leq \frac{1}{n}\sum_{i=1}^{n} f(x_i) - I \leq b\frac{\sigma_f}{\sqrt{n}}\right) = \frac{1}{\sqrt{2\pi}}\int_{-a}^{b} dt\, \exp\left(-\frac{t^2}{2}\right). \tag{1.10}$$

The paramter $\sigma_f$ would typically be unknown in a setting where $I$ is unknown. Of course it would be preferable to find an estimator that does not require to repeat the entire simulation several times. This estimator is the Monte Carlo standard error that can be estimated by using the sample standard deviation [46, p.2]:

$$s_f = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(f(x_i) - \hat{I})^2}. \tag{1.11}$$

Thus, the function values $f(x_1), \ldots, f(x_n)$ do not only provide an estimate $\hat{I}$ of the integral $I$ but simultaneously a measure of the error in $\hat{I}$.

Up to this point we have seen that the standard error in a Monte Carlo integration scales like $1/\sqrt{n}$. This is a central feature of the Monte Carlo method because the main advantage of Monte Carlo is the fact that the error estimate is independent of the dimension $d$ of the integral. Of course, a change in dimensions will change $f$ and with $f$, $\sigma_f$ will change as well. Nevertheless, the standard error will always have the form $\sigma_f/\sqrt{n}$, based on $n$ draws from the unit hypercube $[0,1]^d$. Thus, the convergence rate $O(n^{-1/2})$ will apply for an arbitrary $d$ [12, 46, p.1270, p.3].

However, the price we pay is that the Monte Carlo estimate converges rather slowly to the true value. In order to halve this error it requires to increase the number of points by the factor of four. This represents the so called square-root convergence rate of the Monte Carlo estimate, which stems from the $\sqrt{n}$ in the denominator of the standard error. Thus, the main disadvantage of the Monte Carlo method is its relatively slow convergence rate. Therefore, the Monte Carlo method is generally not competitive and thus rather not advisable in one-dimensional integrals since there are other methods that converge much faster in low dimensions than the Monte Carlo method. Yet, Monte Carlo methods are very attractive in evaluating integrals in high dimensions [46, p.2f].

Much effort has been spent on research activities with the aim of increasing efficiency of Monte Carlo tools over the past years. Nevertheless, the Monte Carlo approach still exhibits a

| $n$ | 10 | 100 | 1 000 | 10 000 | 100 000 |
|---|---|---|---|---|---|
| $\hat{I}_n$ | 89.0911 | 95.3314 | 93.4988 | 94.0059 | 93.8314 |
| $\varepsilon_n$ | −4.7240 | 1.5164 | −0.3162 | 0.1909 | 0.0190 |

Table 1: Monte Carlo estimates and errors

rather slow convergence rate which is generally given and therefore it is often impossible, except for quasi-Monte Carlo techniques, to overcome that handicap. On this account, to achieve reasonably accurate simulation estimates, high – sometimes even too high – sample sizes are required. Yet, in order to enhance the Monte Carlo technique's efficiency and simultaneously keep it a competitive tool for expensive computations, some methods to improve the simulation characteristics have been proposed [38, p.255ff]. Two main topics, *variance reduction techniques* and *quasi-Monte Carlo*, will be elaborately discussed in the Sections 3 and 4.

### 1.1.3 Some first examples

#### Computing Monte Carlo integration

As we have seen in the previous section, computing integrals by Monte Carlo can be attractive especially in high dimensional cases. For the sake of completeness we look at a trivial example which demonstrates the application of the Monte Carlo tool in a stochastic framework [15, p.221f]. Consider the following integral:

$$I = \int_4^5 \exp(x)dx. \tag{1.12}$$

Of course, we can solve Equation (1.12) analytically, leading to the true value $I = \exp(5) - \exp(4) = 93.8150$. However, let us compare our Monte Carlo estimate $\hat{I}_n$ of $I$ and its error $\hat{I}_n - I = \varepsilon_n$ in Table 1. Apparently, our estimate $\hat{I}$ tends to converge to the correct value as the number of draws $n$ increases.

#### Buffon's needle problem

In the Buffon's needle problem the following procedure to estimate the number $\pi$ is considered: "Given a needle of length $l$ and an infinite grid of parallel lines with common distance $d$ between them, what is the probability $P(H)$ that a needle, tossed at the grid randomly, will cross one of the parallel lines?" [91, p.916]. Each time a needle lands on the grid in such a way that it touches or crosses a line it is considered as a hit. This problem was introduced by the French mathematician Buffon in 1733 [22] and reproduced with solution by Buffon in 1777 [23].

Considering Figure 1, we can recognize that if the distance between a needle's center and the next line, denoted by $b$, is higher than $q$, then the needle does not cross or hit a line. The centers of the needles are expected to be independently and uniformly distributed. In other words, a

center will land with equal probabilities somewhere within the space, here between the parallel lines. Since the angles $\theta$ can take values only from 0 up to 180°, the following two properties are assumed to hold [75, 91, p.11f, p.916]:

1. $0 \leq b \leq d/2$
2. $0 \leq \theta \leq 180°$ and $0 \leq \theta \leq \pi$, respectively.

Accordingly, $b$ and $\theta$ are the only two variables which have the characteristics of randomness. Thus, they will play a decisive role in order to compute a Monte Carlo estimate for the variable $\pi$.

Following trigonometric theory the relation $\sin(\theta) = q/(1/2\ l)$ must hold. While solving for $q$ this expression allows us now to conclude that a needle, tossed randomly, will touch or even cross a line if [75, p.11]

$$b \leq \frac{1}{2}\ l \sin(\theta) = q.$$

Based on this condition Ramaley proposed the following way to calculate the probability of a hit, expressed as $P(b \leq 1/2\ l \sin(\theta)) = P(H)$ [91, p.916]:

$$P(H) = \int_0^\pi \frac{l \sin(\theta) d\theta}{\pi d} = \frac{l}{\pi d} \int_0^\pi \sin(\theta) d\theta = \frac{2l}{\pi d}. \tag{1.13}$$

Of course if $l$ is assumed to be equal to $d$, i.e. the needle is equally long as the distance between the parallel lines, then the calculations are straightforward and the probability of a hit is given by $P(H) = 2/\pi$.

According to the probability theory, the law of large numbers ensures that the observed frequency of an event eventually coincides with its probability as the sample size goes to infinity. It follows that while tossing a sufficiently large number of needles, the proportion of all hits is expected to match the probability of a hit $P(H)$ [75, p.12]. Formally, if the number of hits is denoted by $h$ and the number of tossed needles by $n$, the law of large numbers is responsible for the asymptotic validity of the following equation:

$$\lim_{n \to \infty} P(H) = \frac{h}{n}. \tag{1.14}$$

As from now there is only a small step to the estimate of $\pi$ calculated by Monte Carlo. Equations (1.13) and (1.14) show that $h/n$ is totally equal to $2l/\pi d$. By solving for $\pi$, the Monte Carlo estimate $\hat{\pi}$, again indicated by a hat notation, for $\pi$ is given by [75, 91, p.12, p.917]

$$\hat{\pi} = \frac{2ln}{dh}. \tag{1.15}$$

Recall that the strong law of large numbers ensures that $\hat{\pi} \to \pi$ with the probability 1 as $n \to \infty$ [46, p.2].

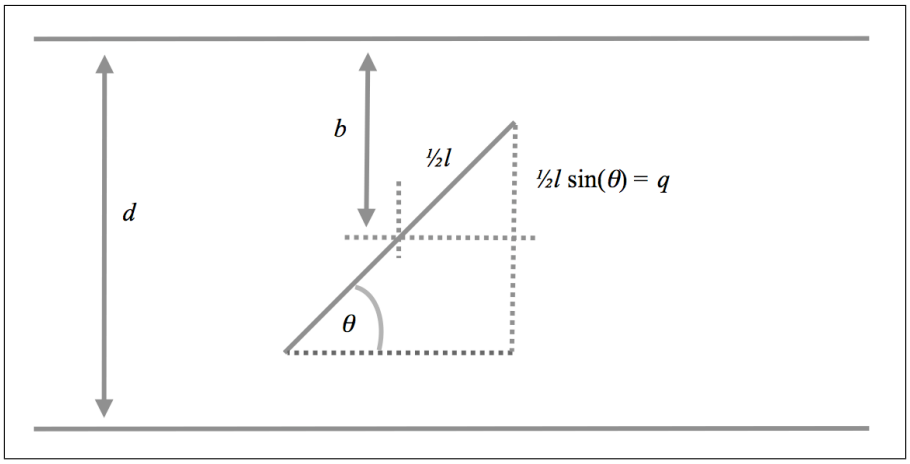



Figure 1: Buffon's needle problem [91, p.916]

## 1.2 General application in finance

As mentioned above, Monte Carlo methods find application in numerous areas, amongst others also in finance. In financial engineering simulation tools have been increasingly gaining importance due to both the continuous enhancement of computational efficiency and the development of very complex financial, especially derivative instruments.

### 1.2.1 Asset and derivative price dynamics

A model of the dynamics of asset prices must reflect the random nature of price movements. In this section only a simple and standard case of numerous random sample paths is considered in order to allow for computing derivative prices by simulation. It is referred to Sections 7, 8 and 9 for more sophisticated models.

Stock prices are usually assumed to follow a *Markov process*. Markov processes can be characterized as a particular type of stochastic processes without any history. Past values and the way the present has emerged from the past are therefore irrelevant [56, p.263f]. A particular type of a Markov stochastic process and presumably the simplest process is the so called *Wiener process* or elementary *Brownian motion*, $\Delta W = z\sqrt{\Delta t}$, where $z$ is assumed to be iid[1] and $z \sim \phi(0,1)$. The mean of $\Delta W$ is zero and its variance is $\Delta t$, which means the standard deviation grows with the square root of time. It follows that $W(t) \sim \phi(0,t)$ [46, p.80]. Since it is common to proceed from small changes to the limit, $\Delta W$ goes to $dW$ as $\Delta t \to 0$.

If we allow for a *drift* and *variance rate*, the process $X(t)$ is called an *arithmetic Brownian motion* with drift $\mu$ and diffusion coefficient $\sigma^2$ if

$$\frac{X(t) - \mu t}{\sigma}$$

[1] independent and identically distributed

follows an elementary Brownian motion [46, p.80]. Thus $X(t) \sim \phi(\mu t, \sigma^2 t)$ and $X$ solves the stochastic differential equation (SDE)

$$dX(t) = \mu dt + \sigma dW(t). \tag{1.16}$$

Note that the parameters $\mu$ and $\sigma$ are deterministic and independent of both the time $t$ and the variable $X$. If we allow them to be time-varying and dependent on the level of $X$, the result will be an *Itô process* [56, p.269] $dX(t) = \mu(X,t)dt + \sigma(X,t)dW(t)$. Because Brownian motion increments are independently normally distributed, simulating $W(t)$ or $X(t)$ is straightforward.

Since an arithmetic Brownian motion can also take negative values, this type of process is apparently unsuitable for simulating asset prices. These are rather assumed to follow a *geometric Brownian motion* (GBM). A stochastic process $S(t)$ is regarded as a GBM, abbreviated $S(t) \sim GBM(\mu, \sigma^2)$, if $\ln S(t)$ is a Brownian motion with initial value $\ln S(0)$ [46, p.93]. In this case we would write a GBM as

$$S(t) = S(0) + \int_0^t \mu S_u du + \int_0^t \sigma S_u dW_u \tag{1.17}$$

or in shorthand and divided by $S(t)$,

$$\frac{dS(t)}{S(t)} = \mu dt + \sigma dW(t). \tag{1.18}$$

A GBM is one of the most fundamental models to describe the evolution of the value of financial assets. The SDE defined by Equation (1.17) is well-defined. The first integral on the right-hand side is a regular Riemann integral and the second integral is a *stochastic*, or also called *Itô integral*. Note that the left-hand side of Equation (1.18) represents the proportional change in the asset price in the interval $(t, t+dt)$ and additionally shows that $dS/S \sim \phi(\mu dt, \sigma^2 dt)$ [46, p.93].

Applying *Itô's lemma* (see Section 1.2.2) one can find the following solution

$$S(t) = S(0) \exp((\mu - \frac{1}{2}\sigma^2)t + \sigma W(t)) \tag{1.19}$$

for the SDEs (1.17) and (1.18), respectively. Equation (1.19) allows us now to simulate an arbitrary number of possible asset price paths by Monte Carlo. Particularly, a random path for an asset price can be simulated by sampling repeatedly for $z$ from $\phi(0,1)$ and substituting in Equation (1.19). It follows that the logarithm of the asset price is thus normally distributed, and the price itself has a lognormal distribution [56, p.275]. The source of this normality lies in the Wiener process, which is a reasonable assumption due to the central limit theorem.

Figure 2 shows possible asset price paths that are assumed to follow a geometric Brownian motion with the parameters $S(0) = 100$, $\mu = 0.05$, $\sigma = 0.4$, $t = 1$ and $dt = 1/365$. The simulated asset prices in one year are $S(t)_1 = 134.72$, $S(t)_2 = 72.12$ and $S(t)_3 = 62.20$.

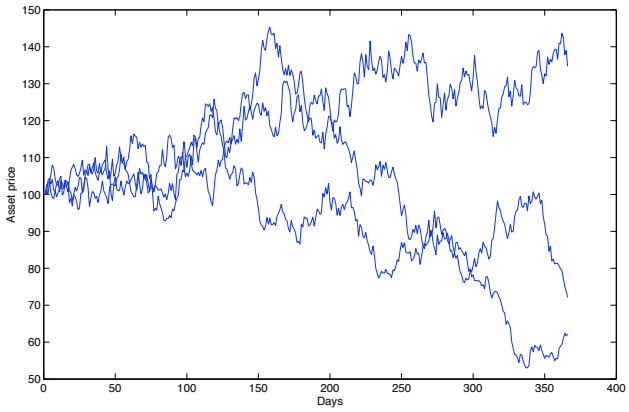



Figure 2: Three independent sample paths following a geometric Brownian motion

### 1.2.2 Pricing vanilla European options by Monte Carlo

Having generated the sample paths of asset prices under the assumption of a certain diffusion process, only small steps bring us to the pricing of derivative instruments using Monte Carlo. The doubtless most important one was firstly taken by Kiyoshi Itô in 1951 when he published Itô's lemma

$$dG = \left( \frac{\partial G}{\partial x}\mu + \frac{\partial G}{\partial t} + \frac{1}{2}\frac{\partial^2 G}{\partial x^2}\sigma^2 \right) dt + \frac{\partial G}{\partial x}\sigma dW \tag{1.20}$$

in the article *On Stochastic Differential Equations* [58], where $G$ is a continuous and differentiable function of the variable $x$. He basically showed by a simple Taylor series expansion that, firstly, the term involving $\Delta x^2$ in Equation (1.20) includes a component of order $\Delta t$ and therefore must not be ignored as $\Delta t \to 0$. Secondly, the variance of $z^2\Delta t$ is of order $\Delta t^2$ and therefore $\Delta x^2$ becomes nonstochastic and equal to $\sigma^2 dt$ as $\Delta t \to 0$ [56, p.279-280].

A price of derivatives is eventually a function of the underlying asset's price and time. Applying Itô's lemma, we can calculate the stochastic process followed by a function whose inputs come from the stochastic process followed by the underlying itself [56, p.273ff]. Thus with Itô's lemma in our toolbox we are able to understand the dynamics of derivatives. A key point to keep in mind is that the Wiener process underlying the stochastic process for the asset price is exactly the same as the Wiener process underlying the stochastic process for the derivative. Therefore, both are subject to the same source of uncertainty, which is of prime importance not only for the application of Monte Carlo but also for the entire option pricing theory.

Let us now consider the application of Monte Carlo to a vanilla European call option. This may seem a little pointless, as there is with Black-Scholes [9] an analytical method that delivers the true option price much easier. Nevertheless, it builds a good and easy introduction to identify the exact Monte Carlo procedure. The payoff of a call option with strike $K$ and duration $T$ is

$$(S(T) - K)^+ = \max\{0, S(T) - K\}. \tag{1.21}$$

By multiplying this payoff by a discount factor $e^{-rT}$ we get the present value of the payoff. This is commonly done with a continuously compounded risk-free interest rate, denoted as $r$. The expected present value and thus the call option price $C$ is defined by [46, p.4]

$$C = e^{-rT} E[(S(T) - K)^+].$$

However, without the knowledge of any distributional characteristics of the random variable $S(T)$, this expectation is totally meaningless. It is impossible to price an option without any information about its distribution of the underlying price or its return. Therefore, we necessarily need to specify the particular distribution of the final asset price $S(T)$. In the previous section a model for the dynamics of stock prices, namely the geometric Brownian motion, was introduced. This model provides us with the lognormal distribution of $S(T)$, the required information for a successful option pricing via expected values. Since we are in a risk-neutral environment[2], the drift parameter $\mu$ in Equation (1.19) may be replaced by the risk-free rate $r$.

Hence it is straightforward to find the asymptotically correct value for the European call option by Monte Carlo simulation. In a first step, simulate one path for the asset price as it is done in Figure 2 and calculate, according to (1.21), the corresponding payoff. Secondly, repeat this first step numerous times which provides you with several payoffs. Third and last step, compute the mean payoff and multiply it with the risk-free discount factor. For any $n \geq 1$, the Monte Carlo estimator $\hat{C}_n$ is unbiased and strongly consistent, in the sense that its expectation

$$E[\hat{C}_n] = C \equiv e^{-rT} E[(S(T) - K)^+],$$

meaning that $\lim_{n\to\infty} \hat{C}_n = C$ [46, p.4f].

The fact that this Monte Carlo estimator is indeed consistent is illustrated in Figure 3. The figure shows the development of the relative pricing error, computed as $(\hat{C}_n - C)/C$, with a steadily growing sample $n$. Note that the pricing error is gradually decreasing and eventually converting to 0 as $n$ goes to infinity. This is a first proof that Monte Carlo provides correct results in the limit and is basically the prerequisite for its increasing importance in valuing derivatives.

### 1.2.3 Pricing exotic and path-dependent options by Monte Carlo

The payoff function $(S(T) - K)^+$ of a vanilla European option reflects that the payoff is completely determined by the terminal value of the asset price $S(T)$. Thus, it does not depend on the evolution of the price between time 0 and $T$, each price path consists of merely two relevant points $S(0)$ and $S(T)$. However, as soon as we allow for more sophisticated derivatives, it is often essential to simulate paths over multiple intermediate dates. Possible reasons are firstly, that a payoff depends on prices of underlying assets on several dates and secondly, that distribution

[2] Risk-neutrality is correctly assumed because it is possible to find a perfect replication (dynamic hedge) of a derivative instrument and thus, the entire stochastic as well as all preferences can be cancelled. For an extensive treatment of this topic it is referred to Hull [56, p.241ff].

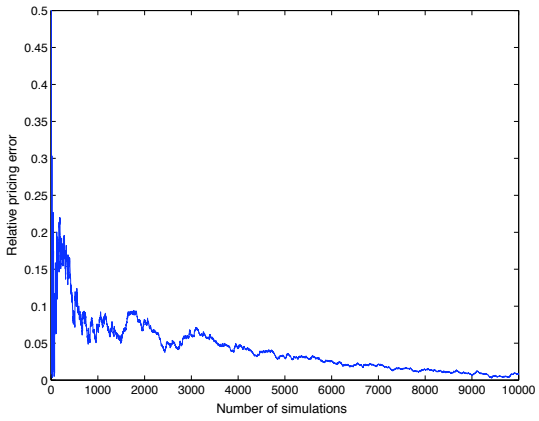



Figure 3: European call option – Pricing error with increasing $n$

parameters may not remain constant until expiration $T$ [14, p.330f]. In many cases one often do not find an explicit and closed formula such as Black-Scholes. Then, Monte Carlo simulations turn out to be of high importance even though they also gain complexity, as we will see when we price interest rate derivatives in Sections 7, 8 and 9 by Monte Carlo.

It is thoroughly possible to price exotics like barrier, Asian and lookback options by Monte Carlo. Here we are having a closer look at Asian options and how to value them by simulation. Their payoff depends on either the arithmetic or geometric average price of the underlying asset during a certain past time. Since the distribution of the arithmetic average of a set of lognormal distributions does not have analytically tractable properties, there is no explicit pricing formula available [56, p.539]. Asian options are arguably the simplest path-dependent options for which Monte Carlo is a competitive computational tool. The payoff of an Asian option with arithmetic averaging is defined as

$$\left(\frac{1}{n}\sum_{i=1}^{n} S(t_i) - K\right)^+$$

where the maturity is in $T$ years, $t_i = i\delta t$, and $\delta t = T/n$. In a standard Monte Carlo approach, we simply generate asset price paths $S(t_1), \ldots, S(t_n)$ as we have done in Figure 2, compute the average along the path and then estimate the discounted payoff, as it is done in detail by Brandimarte [14, p.340f]. A corresponding algorithm to price an Asian call option in MATLAB is given in Appendix A on Page II.

## 2 Random numbers and variables

A good deal of statistical methods rely heavily on the source of random samples, and so does Monte Carlo. Random sampling from a desired or given distribution is at the heart of the Monte Carlo method, as we have already partially perceived in the previous section. The applicability and success of a Monte Carlo calculation generally rest on two pillars. Firstly, of course, the performance of the underlying model is a crucial factor. But secondly, and this to a considerable extent, the quality of the random samples that are used is decisive, where, by quality, it is meant how well the random samples are able to reflect true randomness. Or, to put it into other words, how even they fill or cover the unit hypercube. Hence, Monte Carlo often stands or falls with the characteristics of the random samples. Therefore, the whole subject of random numbers and their generation has become a discrete field of research besides the entire Monte Carlo theory [85, p.1-4 & p.161-164].

This section deals with algorithms for generating uniformly distributed random numbers from the unit interval $[0, 1]$ and methods for transforming those numbers to other desired distributions. Since normal variates are doubtless of primary importance in financial applications, this case is treated in more detail.

### 2.1 Pseudorandom numbers

As discussed, the success of Monte Carlo models is highly dependent on how well the random numbers fit the *known* distribution they are pretended to be drawn from. Since a computer is a completely deterministic machine, truly random numbers are fictitious from a practical point of view [85, p.161]. Nevertheless, this is not a disadvantage if there is some source of *pseudorandom* numbers that is sufficiently good at mimicking genuine randomness [46, p.39]. Sufficiently good in a sense that each proposed pseudorandom number generator must pass a series of statistical tests [85, p.161]. Henceforth, to simplify matters we nonetheless use the term random numbers conscious of its pseudorandom character.

An utterly useful type of generator of random processes *updates* a current sequence of numbers in a manner that seems to be random, we may also call these updates random innovations. A deterministic generator, denoted by $f$, yields the required numbers recursively. Recursively in the sense that the previous $k$ numbers entirely determine the following number

$$x_i = f(x_{i-1}, \ldots, x_{i-k}). \tag{2.1}$$

The number of previous variates used, $k$, is called the *order* of the generator and the set of values at the start of the recursion is denoted as *seed* [45, p.3].

By today's standard an efficient and precise pseudorandom generator should be satisfying the following four criteria [46, p.42]:

- *Long period.* Any random generator of the form (2.1) will, after a certain period, begin to generate the same sequence again, i.e. it will begin to repeat itself. Producing the

same sequence for several times will obviously result in undesired correlations among the generated numbers. Therefore one should never come anywhere near exhausting the length of the period.

- *Efficiency.* Random number generators may be used repeatedly during a single simulation. Therefore, speed is absolutely crucial. The linear congruential generator (see Section 2.2.1) is one of the fastest and simplest algorithm existing. Nonetheless, many generators can be designed and applied efficiently, as we will see later on.

- *Randomness.* A generator should firstly feature good theoretical properties and secondly pass a series of statistical tests. Fortunately, this area is sufficiently advanced so that one can use several generators which are supposed to provide uniformly, independently distributed variates.

- *Reproducibility.* It sometimes may be necessary, e.g. for testing or research purposes, to repeat the same random sequence of a previous run. Apparently, this is only possible with pseudorandom generators. One considerable drawback of a genuinely random sequence is that it can hardly be reproduced.

## 2.2 Generation of uniform variates

### 2.2.1 Linear congruential generators

The simple linear congruential generator was introduced by Lehmer [68] as a general source of random numbers. Although this generator may be limited especially in its ability to produce very long sequences of numbers that appear to be uniformly independently distributed, it remains a basic element in other, more adequate or sophisticated generators.

The form of the linear congruential generator is

$$x_i \equiv (ax_{i-1} + c) \bmod m, \quad \text{with } 0 \leq x_i < m, \tag{2.2}$$

where $a$ is the so called *multiplier*, $c$ the *increment* and $m$ the *modulus* [15, p.226f]. From this sequence we derive the random numbers $u_i$

$$u_i = \frac{x_i}{m}. \tag{2.3}$$

In the linear congruential generator $f$ from Equation (2.1) is a simple linear function (i.e., $k = 1$) combined with a modular reduction.

A common distinction has been made between a homogeneous case where $c$ is set equal to zero, also called the *multiplicative congruential method*, and the inhomogeneous case where $c \neq 0$, also called the *mixed congruential method* [85, p.168ff]. Following conditions ensure that the generator has full period. Full period means that the number of distinct values generated from any seed $x_0$ is at least as large as $m - 1$. If $c \neq 0$, then Knuth [65, p.17] names the following conditions [46, p.43]:

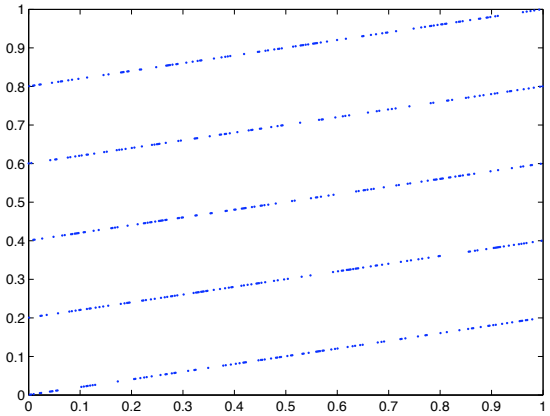



Figure 4: Lattice structure of linear congruential generators

(i) $c$ and $m$ are relatively prime (i.e. their only common divisor is 1),

(ii) every prime number that divides $m$ simultaneously divides $a-1$ and

(iii) $a-1$ must be divisible by 4 if $m$ is.

In the second case, if $c=0$ and $m$ is prime, we achieve full period from any $x_0 \neq 0$ if [46, p.43]

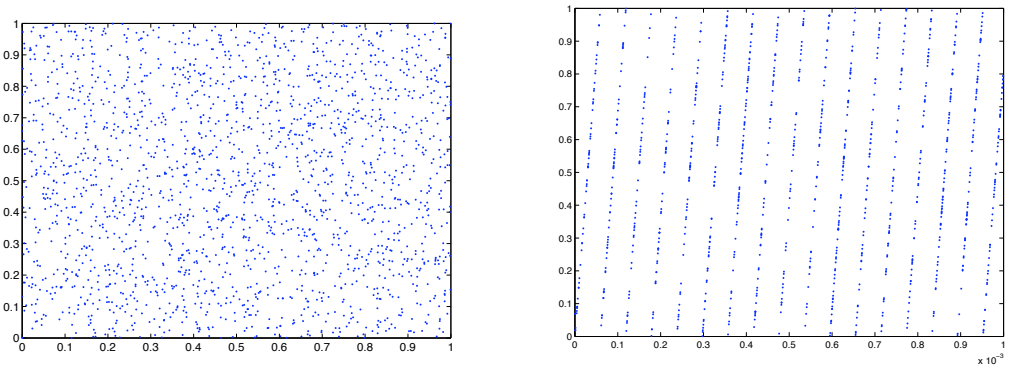
(i) $a^{m-1}-1$ is a multiple of $m$,

(ii) $a^j - 1$ is not a multiple $m$ for $j = 1, \ldots, m-2$.

If a number $a$ indeed fulfills all these conditions above it is called a *primitive root* of $m$. If $c=0$, the sequence first returns to $x_0$ at the smallest $k$ for which $a^k x_0 \bmod m = x_0$. Therefore, one can define the requirement of a primitive root as follows: a sequence must not return to $x_0$ until $a^{m-1}x_0$ [46, p.43]. Furthermore, Marsaglia [73] showed in his paper that a generator with a nonzero $c$ is providing negligible additional generality but makes the simulation considerably slower. Since speed is crucial in solving problems by simulation, it is customary to take $c=0$ [46, p.44].

As already touched before the parameter's choice is crucial for linear congruential generators. Hence, there are extensive empirical studies reported in literature. Fishman and Moore [39], for example, indicate that different values of multipliers, all of which perform well under different statistical tests (e.g. *spectral test* (see below)), may yield pseudorandom samples statistically different compared to genuinely random samples from a uniform distribution [46, p.44].

Let us compare some distributional properties and provide a bit more detail on how these comparisons are made. Apparently, if the random variables $u_1, u_2, \ldots$ are independent and uniformly distributed over $[0,1]$, then $(u_1,u_2),(u_3,u_4),\ldots$ are uniformly distributed over the unit square $[0,1]^2$, and so on. If we look at Figure 4 we see consecutive overlapping random



(a) Apparent randomness

(b) Evident lattice structure

Figure 5: IBM linear congruential generator with $a = 7^5, m = 2^{31} - 1$ and $x_0 = 8\,835$

pairs $(u_1, u_2), (u_2, u_3), \ldots, (u_{2047}, u_{2048})$ produced by a simple linear congruential generator with the parameters $a = 1\,229$, $m = 2\,048$, $c = 1$ and the seed $x_0 = 1$ [15, p.228f]. This graph immediately reveals some regularity, i.e. a clearly definite pattern since the generated points all lie on several parallel lines. Thus, they hardly deserve to be denoted as random. This phenomenon is characteristic of all linear congruential generators and is often called *lattice structure*.

If we secondly take the famous IBM linear congruential generator with parameters $a = 7^5$, $c = 0$, $m = 2^{31} - 1$ and seed $x_0 = 8\,835$ as proposed by Lewis, Goldman and Miller [69] as well as by Park and Miller [86] [46, p.48], the generated deviates truly appear to be random at first sight as the left panel (a) of Figure 5 shows. Nevertheless, appearances are deceiving. If we increase the number of draws to one million but keep the parameters and the seed unchanged and zoom the x-coordinate to the interval $[0, 0.001]$, as it is illustrated in the right panel (b) of Figure 5, the lattice structure and thus the deterministic character, which lies behind it, becomes evident. In spite of this obvious flaw the linear congruential method is a widely used method [15, p.229].

The lattice structure of linear congruential generators is often used to assess different generators and select appropriate parameters. A widely used test in the analysis of random number generators is the *spectral test*, originally proposed by Coveyou and MacPherson [30]. The spectral test determines the maximum distance between adjacent parallel hyperplanes defined by the lattice [45, p.65]. The maximum distance between adjacent parallel hyperplanes in a $d$-dimensional lattice can be represented by $\delta_d(m, a)$. For any $d$-dimensional lattice with $m$ points, there is a lower bound, $\delta_d^*(m)$, on $\delta_d(m, a)$. Thus, the closer the ratio

$$S_d(m, a) = \frac{\delta_d^*(m)}{\delta_d(m, a)} \tag{2.4}$$

is to one, the better the generator is with respect to this measure [65, 45, p.65]. However, even

though the spectral test is an informative measure of uniformity, it does not provide a strict ranking criteria for the quality of generators because its computed values are not independent of the dimension $d$. Fishman and Moore [39] recommend in their empirical study the application of the spectral test up to dimension $d = 6$, in higher dimension the computational effort becomes too high. Glasserman therefore suggests, due to these inevitable shortcomings, to use only a considerably small fraction of the generator's period [46, p.49]. This highlights the importance of including long periods in a generator, i.e. much longer than the $2^{31}$ of the above presented linear congruential generator.

### 2.2.2 Multiple recursive generators

We now turn to methods that combine elements of different linear congruential generators. These methods seem to be pretty attractive because they preserve favorable computational features while simultaneously extending their maximum period. Furthermore, lattice structure problems appear to be diluted even though complexity tends to increase.

The multiple recursive generator naturally extends the linear congruential generator. Here we do not directly compute the next random variate from the one just computed. We rather calculate the linear combination of the past $k$ generated random numbers as follows [34, p.146]:

$$x_i \equiv (a_1 x_{i-1} + a_2 x_{i-2} + \cdots + a_k x_{i-k}) \bmod m \tag{2.5}$$

followed by $u_i = x_i/m$. Obviously, when $k = 1$, the multiple recursive generator is reduced to the linear one [34, p.146].

Knuth [65, p.29ff] has shown that in Equation (2.5) each of the (lagged) values $x_{i-j}$ can adopt up to $m$ whereas the vector $(x_{i-1}, \ldots, x_{i-k})$, obviously, take up to $m^k$ distinct values. Thus, the period of such a multiple recursive generator can be much longer than that of a simple linear congruential generator. Particularly, taking $m$ as a prime, the maximum period is $m^k - 1$, which is significantly larger than just $m - 1$ [46, p.51].

However, one of the multiple recursive generator's drawback is its computing time, which is about $k$ times longer than that of the linear congruential generator. To compensate for that shortfall, Deng and Lin [34, p.146ff] propose to set as many terms of $a_i$ in (2.5) equal to 0 or $\pm 1$ as possible. In particular, there are multiple recursive generators with the parameters chosen to be $a_1 = \pm 1$, $a_i = 0$ for $2 \leq i \leq k-1$ and $a_k = b$. The choice of $b$ is crucial insofar, that it should be carried out properly in order to obtain the maximum period of $m^k - 1$. So they propose the *fast* multiple recursive generator

$$x_i = (b x_{i-k} - x_{i-1}) \bmod m. \tag{2.6}$$

The main difference between the fast multiple recursive and the classical linear congruential generators may be explained as follows. In the former we add and subtract a *variable* increment $x_{i-1}$, respectively. In the latter, however, there is a *constant* increment $c$ added only, where $c \geq 0$. Obviously, this minor but nonetheless far reaching difference between these two methods does

not influence their computation time in any way [34, p.146f]. In summary, the fast multiple recursive generator features the advantages of either the ordinary multiple recursive and the linear congruential generator and therefore seems to be clearly superior to its alternatives.

### 2.2.3 Nonlinear congruential generators

The *inversive congruential generator* is one of various nonlinear congruential generators that have been proposed so far. This generator, introduced by Eichenauer and Lehn [36], generates the next variate of a sequence by using the modular multiplicative inverse. Inversive congruential generators include recursions as [45, p.36]

$$x_i \equiv (ax_i^- + c) \bmod m \tag{2.7}$$

where $x^-$ stands for the multiplicative inverse of $x$ modulo $m$, if it exists. If it does not exist, it simply denotes 0.

The main advantage of this generator is that it is not affected by the lattice structure characteristic. On the other hand, the price we pay is a substantial efficiency loss. Although there are some ways of speeding up the calculations, the overall difficulties of the inversive congruential generators, which are almost impossible to overcome, have eventually prevented their widespread use [45, p.36f].

### 2.2.4 Testing random number generators

Random number generators are often tested empirically. In particular, a certain segment of the output stream is firstly taken and afterwards subjected to statistical tests in order to determine whether there are unusual departures from randomness or significant defects [33, p.29]. Therefore, empirical test approaches additionally assess the program that has actually been used to generate the random numbers. This is a great advantage compared to other kinds of tests. The specifications of the tests are that the null hypothesis ($H_0$) is that the elements of a given sequence of real numbers are iid and from $U[0,1]$. The alternative hypothesis ($H_1$) is that the sequence does not have the assumed distribution. Recall at this point that not rejecting $H_0$ does not automatically mean that $H_0$ it true indeed. The only conclusion that may be drawn is that there is not enough evidence to successfully reject the null hypothesis.

#### Chi-squared Goodness-of-Fit test

The $H_0$ of the Chi-squared Goodness-of-Fit test, sometimes also simply called *uniformity test* or *frequency test* [85, p.166], is that a random variable follows a uniform $[0,1]$ distribution. At the beginning, the whole interval from 0 to 1 is divided into $q$ subintervals of the lengths $v$. Thus, the individual subintervals scale like $[0,v],[v,2v],\ldots,[(q-1)v,1]$, while $qv = 1$. In a next step we count how many of the $n$ taken consecutive numbers are in each of the $q$ intervals. Then we compare the total numbers of observed variates, $o_i$, in the intervals with the (in advance) ex-

pected numbers of variates, $e_i$, in the corresponding intervals. Of course, the expected frequency is defined as $e_i = n/q$.

Under the $H_0$ the test statistic

$$\chi^2 = \sum_{i=1}^{q} \frac{(o_i - e_i)^2}{e_i} \tag{2.8}$$

follows a chi-squared distribution with $q - 1$ degrees of freedom. Therefore, one may reject $H_0$ at a significance level of $\alpha$ if $\chi^2 > \chi^2_{q-1,\alpha}$ [33, 45, p.29, p.74].

#### Serial tests

The serial test starts from a sequence of random numbers $u_1, u_2, \ldots$. Again, independence and uniformity of the random variates are supposed. If these conditions are met once again, nonoverlapping pairs $(u_1, u_2), (u_3, u_4), \ldots$, for instance, should be iid and uniforms from $[0, 1]^2$. The serial test is a more severe test for statistical independence and basically a multidimensional version, i.e. $d \geq 2$, of the Chi-squared Goodness-of-Fit test [85, p.167].

The null hypothesis of the serial test with $d = 2$ is that pairs $(u_{2i-1}, u_{2i})$, $i = 1, 2, \ldots$, are uniformly distributed over $[0, 1]^2$. Thus the possibility of serial dependence is explored because if consecutive numbers are indeed serially dependent, then the required uniformity will be lost. As in the Chi-squared Goodness-of-Fit test, we run a standard chi-squard test with $q^2$ subsquares, which contain an area of $1/q^2$ each. This time, the observed frequency in the $i$th subsquare must be indicated by a function $f_i$, where $i = 1, 2, \ldots, q^2$ and $\sum_{i=1}^{q^2} f_i = n$. Under $H_0$, the expected frequency is given by $e_i = n/q^2$ and the null distribution is $\chi^2_{q^2-1}$ [33, 45, p.30, p.78].

Here we assumed that $n$ points in the sample are independent. In the Chi-squared Goodness-of-Fit test, however, the assumption was that the random numbers are independent. Since both assumptions are not unproblematic, an alternative would be to draw non-consecutive random points, i.e. to allow for some distance between them. This clearly diminishes the probability of dependency among the generated variates [33, p.30].

### 2.3 Transformations of uniform variates

We have seen so far how to generate a sequence of random numbers from $U[0, 1]$. However, most of the simulations include the sampling of random variables, or even random vectors in a multidimensional case, from specified, not uniform distributions. Sampling of nonuniform random variates is done indirectly, i.e. one transforms samples from the uniform to a desired nonuniform distribution. Accordingly, each realization of a nonuniform random variable is directly obtained from a single uniform variate or from a sequence of uniforms [46, p.54].

There are numerous algorithms that allow you to generate random numbers from a certain desired distribution. However, two main criteria in order to choose between them are accuracy and speed. The former is usually assessed by conducting a simple goodness-of-fit test as we have already seen in Section 2.2.4. In this section we want to have a closer look at the most

widely used general techniques, the *inverse transformation method* and the *acceptance-rejection method*.

### 2.3.1 Inverse transform method

Suppose a cumulative distribution function (CDF) of the form $F(x) = P(X \leq x)$ is given and that we would like to draw samples according to $F$. Applying the inverse transform method [15, p.230]

$$X = F^{-1}(u), \quad u \sim U[0,1] \tag{2.9}$$

where $F^{-1}$ is the inverse of $F$, provides us with the random variable $X$ according to the distribution of $F$. The inverse of $F$ is well-defined if $F$ is strictly increasing.

It is straightforward to verify that the random variates $X$ generated by this method are characterized by $F$ [15, 46, p.230, p.55]:

$$\begin{aligned} P(X \leq x) &= P(F^{-1}(u) \leq x) \\ &= F(u \leq F(x)) \\ &= F(x). \end{aligned}$$

Even though the inverse transform method may not be the fastest method for sampling from a distribution, it possesses some important features that make it attractive nonetheless [46, p.58]:

- It is often used in order to sample from conditional distributions.
- It maps the input $u_i$ monotonically and continuously to the output $X$, which can be of great benefit in the implementation of variance reduction techniques as we will see in Section 3.
- The inverse transform method features a one-to-one relationship, i.e. it requires one $u$ for each sample $X$ generated only. This will turn out to be of particular importance in using quasi-Monte Carlo methods as illustrated in Section 4.

Note, however, that the inverse transform method does not directly apply to a multivariate distribution. Nevertheless, via marginal and conditional univariate distributions one can crab-wise generate multivariate random deviates in an inverse transform method [45, p.109]. If a decomposition of the CDF of the multivariate random variables $(X_1, X_2, \ldots, X_d)$ like

$$F_{X_1 X_2 \ldots X_d}(x_1, x_2, \ldots, x_d) = F_{X_1}(x_1) F_{X_2|X_1}(x_2|x_1) \cdots F_{X_d|X_1 X_2 \ldots X_{d-1}}(x_d|x_1, x_2, \ldots, x_{d-1})$$

is possible and the functions are also invertible, the inverse transform method can be applied sequentially by including iid random variables of a $U[0,1], u_1, u_2, \ldots, u_d$ [45, p.109]:

$$\begin{aligned} x_1 &= F_{X_1}^{-1}(u_1), \\ x_2 &= F_{X_2|X_1}^{-1}(u_2), \\ \ldots & \quad \ldots \\ x_d &= F_{X_d|X_1 X_2 \ldots X_{d-1}}^{-1}(u_d). \end{aligned}$$

| iid from $g$ | $y_i$ | $y_{i+1}$ | $y_{i+2}$ | $y_{i+3}$ | $\cdots$ | $y_{i+k}$ | $\cdots$ |
|---|---|---|---|---|---|---|---|
| accept? | no | yes | no | yes | $\cdots$ | yes | $\cdots$ |
| iid from $f$ | | $x_j$ | | $x_{j+1}$ | $\cdots$ | $x_{j+l}$ | $\cdots$ |

Table 2: Acceptance-rejection method [45, p.115]

### 2.3.2 Acceptance-rejection method

The acceptance-rejection method was proposed by von Neumann in 1951 [104] and is used in order to generate random variates that follow a probability density function denoted by $f(x)$. Since we are occasionally struggling in inverting the CDF that corresponds to our $f(x)$, the just discussed inverse transform method loses ground and the acceptance-rejection method is applied [15, p.233f].

Given a so called target distribution $f(x)$, the acceptance-rejection method generates samples according to $f(x)$ by first of all generating variates from a more suitable distribution $g(y)$. Afterwards a random subset of these generated variates is rejected according to certain rejection rules. Thereby the choice of this rejection rule, or let us call it rejection approach, will be decisive if the accepted samples will eventually be distributed according to $f(x)$. The property $f(x) \leq cg(x)$, for some constant $c$, tells us now how to generate samples from $g(y)$. Therefore, we conclude that in the acceptance-rejection method a sample $Y$ is generated from $g$ and at the same time accepted with probability $f(Y)/cg(Y)$ [46, p.58].

The generic implementation, i.e. the algorithm of the acceptance-rejection method that we can use to sample from density $f$ using candidates from $g$, can be stated as follows [15, 45, 46, p.243, p.114, p.58f]:

1. generate the sample $Y$ from the distribution with density $g$
2. generate $u$ from $U[0,1]$, which must be independent of $Y$
3. if $u \leq f(Y)/cg(Y)$, then
   3a. deliver $X = Y$,
   otherwise
   3b. return to step 1.

Accordingly, the acceptance-rejection method can be illustrated as taking a subsequence out of a sequence of iid realizations from the more convenient distribution $g$ in exactly such a way that the subsequence indeed has the density $f$, as this approach is shown in Table 2 [45, p.115].

Unlike the inverse transform method, this method applies immediately to multivariate random variables. Yet, it may not be very efficient in higher dimensions. The discussion about the efficiency of the acceptance-rejection method is not pushed any further in this paper and it is referred to Devroye [35], Fishman [38] or Gentle [45] for a more profound treatment of that topic [45, 46, p.113ff, p.60].

In Section 2.3.1 the one-to-one relationship was found to be an attractive feature of the inverse transform method, meaning that it uses exactly one uniform in order to generate one nonuniform random variable. While applying the acceptance-rejection method this relationship is, as it is generally known, not existing. There is actually no upper limit on the number of uniforms needed to generate even one single nonuniform variable. Hence, simulations based on this method therefore end up in infinite-dimensional integration problems. As we will see in Section 4, acceptance-rejection methods are generally inapplicable with quasi-Monte Carlo methods [46, p.62f].

## 2.4 Univariate normal random variables

As we have partially noticed in Section 1.2 normal random variables build a form of a building block of many simulation models, especially in finance. Here we examine carefully different algorithms for generating samples from univariate normal distributions. If $Z \sim \phi(0,1)$ and $X = \sigma Z + \mu$, then $X \sim \phi(\mu, \sigma^2)$. Due do this simple relationship, it is totally sufficient to sample from a standard normal distribution, $\phi(0,1)$ [46, p.63]. In order to keep notations short, we always use the expression normals knowingly that we actually generate standard normal random variables.

Basically, there are two different approaches to generate normal variables, by either inversion of the CDF or transforming uniform variables [49, p.108f]. Since there is neither a closed form of the normal CDF, denoted as $\Phi$, nor an analytical form for its inverse $\Phi^{-1}$ existing, the former would have been done by complex numerical and often inefficient inversions and is therefore rather inapplicable to generate normals. The latter, however, performs very well and has therefore been extended to several different methods. Some old-fashioned possibilities are to exploit the power of the central limit theorem. Such an utterly simple – possibly too simple – approach is to generate iid uniforms $u_i$, for $i = 1, 2, \ldots, 12$, and afterwards generate normals by $x_i = \sum u_i - 6$. Note that in this method the central limit theorem finds its application in a sample size of no larger than 12, which is far from sufficient [49, p.109]. This method is not only a poor approximation but also considerably slow and therefore rather not to use.

### Box-Muller method

An alternative and more promising approach is the so called Box-Muller algorithm, named after its proposers Box and Muller [10]. The Box-Muller approach contains two independent standard normal random variables, $Z_1$ and $Z_2$, which team up to a bivariate standard normal distribution, where the sample is now drawn from. If $u_1$ and $u_2$ are independent and from $U[0,1]$, a polar transformation is applied, as described by Box and Muller [10, p.610]:

$$\begin{aligned} Z_1 &= \sqrt{-2\log(u_1)}\cos(2\pi u_2), \\ Z_2 &= \sqrt{-2\log(u_1)}\sin(2\pi u_2), \end{aligned} \tag{2.10}$$

where $Z_1$ and $Z_2$ are now supposed to be independently normally distributed as $\phi(0,1)$ [10, 15, 46, p.236, p.610, p.65].

#### Marsaglia-Bray approach

According to (2.10) the Box-Muller transformation requires the evaluation of one square root and two trigonometric functions for every two deviates generated. Consequently, it is a comparatively slow and inefficient algorithm. Therefore, Marsaglia and Bray [74] proposed a slight modification of the Box-Muller method that reduces computing time by simply avoiding the evaluation of the sine and cosine function. It instead uses the acceptance-rejection method to generate uniforms from $[0,1]$ and afterwards transforms these candidates to $\phi(0,1)$ [46, p.66].

However, since the acceptance-rejection method is involved, there is no upper bound on the number of uniforms this algorithm may require to generate only one single normal. Accordingly, the Marsaglia-Bray method can not be used for quasi-Monte Carlo simulation.

### 2.5 Multivariate normals

A $d$-variate normal distribution $\phi(\mu, \Sigma)$ it generally specified by its mean vector $\mu$ and variance-covariance matrix $\Sigma$. Random vectors from a multivariate normal distribution can be generated directly by generating a $d$-vector of iid standard normal deviates $z' = (z_1, z_2, \ldots, z_d)$ [45, p.197].

Next we know from the linear transformation property that if the vector $z \sim \phi_d(0, I)$ and $x = \mu + Az$, then $x \sim \phi_d(\mu, AA')$. Therefore, we can generate a sequence of univariate normal random variables $z_i$, as we have seen in Section 2.4, and put them into a vector $z$. Thus the challenge is merely made up of finding the matrix $A$ for which the property $AA' = \Sigma$ holds. The representation of $\Sigma$ as $AA'$ with $A$ as a lower triangular matrix is denoted as a *Cholesky factorization* of the variance-covariance matrix $\Sigma$ [46, p.72]. If $\Sigma$ is symmetric positive definite, i.e. $\det \Sigma \neq 0$, there is a Cholesky factorization and the sampling of multivariate normals is easily done by basically sampling from a univariate $\phi(0,1)$ [45, p.197f].

### 2.6 Correlated normal variables and vectors

In some applications in finance random variables or vectors are required to depend on each other in a prescribed way, i.e. they must not be totally uncorrelated. Correlated random variables are calculated according to the following algorithm [45, 49, p.199, p.113]:

1. calculate the Cholesky factorization $AA' = \Sigma$,
2. calculate $z \sim \phi(0, I)$ (by components) by $z_i \sim \phi(0,1)$, for $i = 1, 2, \ldots, d$,
3. $x = \mu + Az$, then $x \sim \phi(\mu, \Sigma)$.

Suppose we have a two-dimensional case, $d = 2$. Next we are looking for a vector $x' = (x_1, x_2) \sim \phi(0, \Sigma)$. The correlation between $x_1$ and $x_2$ is denoted by $\rho$ and thus the Cholesky factorization provides $\Sigma = AA'$:

$$\begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} = \begin{pmatrix} \sigma_1 & 0 \\ \rho\sigma_2 & \sigma_2\sqrt{1-\rho^2} \end{pmatrix} \begin{pmatrix} \sigma_1 & \rho\sigma_2 \\ 0 & \sigma_2\sqrt{1-\rho^2} \end{pmatrix}.$$

If $z_1$ and $z_2$ are independent and from $\phi(0,1)$, then

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = A \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} \sigma_1 z_1 \\ \sigma_2(\rho z_1 + \sqrt{1-\rho^2} z_2) \end{pmatrix} \tag{2.11}$$

are distributed according to $\phi(0,\Sigma)$ [49, p.114].

## 3 Variance reduction techniques

Monte Carlo methods have proved to be very valuable tools in modern financial applications. As we have seen in Section 1.2, pricing derivative securities by Monte Carlo often requires to compute complex multidimensional integrals. Therefore, Monte Carlo becomes increasingly attractive compared to other methods of numerical integration due to its dominant advantage that its convergence rate is independent of the dimension of the problem. On the other hand, one drawback is that for sophisticated calculations a substantially large number of replications $n$ may be required to obtain reasonably precise results [12, 46, p.1269, p.3].

At exactly this point variance reduction techniques take center stage. The goal of these methods is to increase efficiency and precision of Monte Carlo applications by simply reducing the variance of simulation estimates. Given a certain precision, if we decrease the variance we can simultaneously reduce $n$ or with a given $n$, the precision increases with a lower variance. However, smaller variance only is not the whole story. By preferring one estimator over another, computational efficiency must not be overlooked [12, p.1269].

### Selection criteria

Suppose we want to estimate a parameter $\theta$ by Monte Carlo. An appropriate estimator of $\theta$, as it has been discussed in Section 1.1, based on $n$ replications is then the sample mean $1/n \sum_{i=1}^{n} \hat{\theta}_i$, which is approximately normally distributed, $\phi(\theta, \sigma^2/n)$. Suppose, now, that we have to choose between two Monte Carlo estimators $\hat{\theta}_{i,1}$ and $\hat{\theta}_{i,2}$ and that both are unbiased, i.e. more formally $E[\hat{\theta}_{i,1}] = E[\hat{\theta}_{i,2}] = \theta$. Even though both estimators have the same sample mean, they do not automatically have equal variances, say $\mathrm{Var}[\hat{\theta}_1] < \mathrm{Var}[\hat{\theta}_2]$. Obviously, a sample mean based on $n$ replications of $\hat{\theta}_1$ definitely provides a more precise and therefore more accurate estimate of $\theta$ than does a sample mean based on $n$ replications of $\hat{\theta}_2$. However, Boyle et al. [12, p.1270f] showed in their paper that this is not the whole story. Generating $n$ replications of $\hat{\theta}_1$ may be far more time-consuming than generating $n$ replications of $\hat{\theta}_2$. Furthermore, to simplify matters we assume that the work required to generate a single replication of $\hat{\theta}_j$ is a constant $b_j$, $j = 1, 2$. Of course in certain problems the time or work needed per replication is stochastic. Nevertheless, this does not influence the initial position significantly. However, two possible estimators, allowing for a given computing time $t$, are [12, p.1271]

$$\frac{b_1}{t} \sum_{i=1}^{t/b_1} \hat{\theta}_{i,1} \qquad \text{and} \qquad \frac{b_2}{t} \sum_{i=1}^{t/b_2} \hat{\theta}_{i,2}. \tag{3.1}$$

These estimators in (3.1) are approximately normally distributed, $\phi(\theta, \sigma_j \sqrt{b_j/t})$, $j = 1, 2$, provided that $t$ is sufficiently large. Thus, we can derive the following decision rule: the first estimator is preferred over the second if

$$\sigma_1^2 b_1 < \sigma_2^2 b_2. \tag{3.2}$$

Out of this discussion one may conclude that it is reasonable to take the product of variance and work or time per run as a measure of efficiency, and not using the variance as a sole decision criterion [12, 47, p.1271, p.506f]. Thus we should allow for cases where a higher variance estimator may actually be preferable if it takes much less time to generate.

To come back to the actual topic of this section, variance reduction, we should first recall the variance of our Monte Carlo simulation:

$$\mathrm{Var}[\hat{\theta}] = \frac{\sigma^2}{n}. \tag{3.3}$$

Apparently, one simple way to improve the precision and accuracy of a Monte Carlo estimate is to increase the number of replications $n$. However, this brute force approach requires an excessive computational effort and is therefore not a reasonable solution [15, p.243f]. The second approach is to abandon the denominator and rather try to work on the nominator, $\sigma^2$, of this fraction. This can be accomplished in numerous different ways, which are more or less complicated and more or less rewarding as well, as it is shown in the following subsections [11, 12, 15, 45, 46, p.324ff, p.1271ff, p.244ff, p.239ff, p.185ff].

## 3.1 Covariates

### 3.1.1 Antithetic variates

Antithetic variates is one of the simplest and most widely used variance reduction technique in financial applications. Originally, it was introduced to option pricing by Boyle [11, p.327] in his paper *Options: a Monte Carlo approach*. For applying antithetic variates we normally do not require a deep knowledge of the problem's specific features, therefore complexity as well as potential efficiency gains are rather limited [46, p.205]. The key idea behind this approach is straightforward: if a random variable $U$ is uniformly distributed over $[0, 1]$, then $1 - U$ must be too. Henceforth, generating two paths using $U_1, U_2, \ldots, U_n$ and $1 - U_1, 1 - U_2, \ldots, 1 - U_n$ as input, respectively, will automatically result in a variance reduction because an unusually large or small value computed from the first path will be balanced by the output calculated from the second path, the so called antithetic path. The same property also holds for several other distributions through the inverse transformation method: $F^{-1}(U)$ and $F^{-1}(1-U)$ both are distributed according to $F$, nevertheless they are antithetic to each other due to the monotonicity of the function $F^{-1}$ [46, p.205]. Suppose we have a symmetric distribution about the origin, then $F^{-1}(1 - h)$ and $F^{-1}(h)$ will have the same magnitudes but opposite signs: if $Z \sim \phi(0, 1)$ then $-Z \sim \phi(0, 1)$, for instance.

Consider, now, the problem of computing the price of a vanilla European put option by Monte Carlo. In Section 1.2.2 we have seen that $E[\hat{P}] = \frac{1}{n}\sum_{i=1}^{n} P_i \equiv e^{-rT}E[(K - S(T))^+]$ is an unbiased estimator of the option price. The underlying asset price $S(T)$ obtained from Equation (1.19) with $Z_i$ replaced by $-Z_i$, the antithetic variate, is thus a valid sample from the distribution of the terminal stock price [12, p.1272]. Similarly, each

$$\tilde{P}_i = e^{-rT}(K - \tilde{S}(T))^+$$

| $\boldsymbol{E[Z^2]}$ | mean | standard error |
|---|---|---|
| Ordinary Monte Carlo | 1.0062 | 0.1442 |
| Monte Carlo based on antithetic variates | 0.9950 | 0.1947 |
| $\boldsymbol{E[\exp(Z/2)]}$ | mean | standard error |
| Ordinary Monte Carlo | 1.1351 | 0.0596 |
| Monte Carlo based on antithetic variates | 1.1337 | 0.0198 |

Table 3: Monte Carlo and antithetic variates

is an unbiased and consistent antithetic variates estimator of the option price. Therefore the following equation holds [12, p.1272]:

$$E[\hat{P}_{AV}] = \frac{1}{2n}\left(\sum_{i=1}^{n} P_i + \sum_{i=1}^{n} \tilde{P}_i\right) = \left(\frac{1}{n}\sum_{i=1}^{n} \frac{P_i + \tilde{P}_i}{2}\right). \tag{3.4}$$

This gives rise to the question under what specific conditions an antithetic variates estimator is preferable to a standard Monte Carlo estimator. Considering that $P_i$ and $\tilde{P}_i$ have the same variance and that $\hat{P}_{AV}$ uses twice as many replications as $\hat{P}$, antithetics eventually increase efficiency if [12, 46, p.1273, p.206f]

$$\mathrm{Var}[P_i + \tilde{P}_i] < 2\,\mathrm{Var}[P_i],$$

which can be rearragend

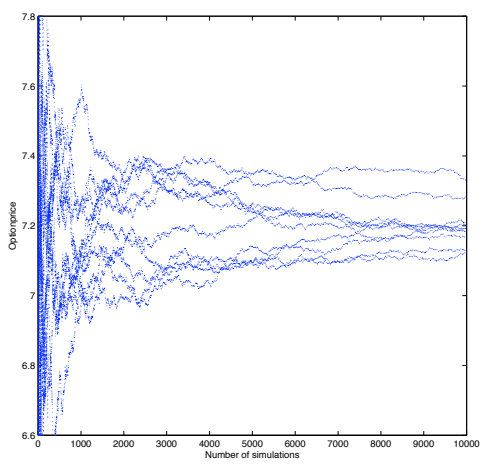
$$\mathrm{Var}[P_i] + \mathrm{Var}[\tilde{P}_i] + 2\,\mathrm{Cov}[P_i, \tilde{P}_i] < 2\,\mathrm{Var}[P_i].$$

Thus, antithetic sampling does indeed reduce variance if

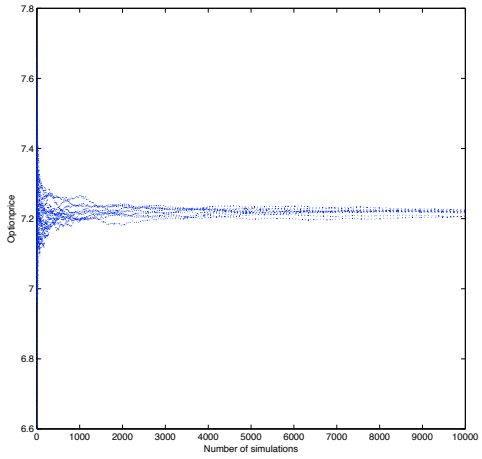
$$\mathrm{Cov}[P_i, \tilde{P}_i] < 0. \tag{3.5}$$

Notice that Equation (3.5) requires that a negative dependence between the inputs also produces a negative correlation between the outputs. Monotonicity of the mapping from inputs to outputs, which is eventually defined by the simulation algorithm, is a sufficient condition ensuring this relationship [46, p.207]. If we use a non-monotone function the correlation in the antithetic output sample will not be negative. As a consequence, the variance of our simulation will rather increase than decline as originally intended [12, p.1273f].

To illustrate the importance of monotonicity, compare the ordinary and the antithetic variates Monte Carlo estimates and their standard errors of the two functions $Z^2$ and $e^{(Z/2)}$, both with $Z \sim \phi(0, 1)$. Table 3 shows that the standard error of the former increases, due to the non-monotonicity of the function $Z^2$, as we run the simulation including antithetics with a sample size of $v = 100$ and a number of replications $n = 1\,000$. This example shows impressively that relying on antithetic variates may backfire if one does not make a wise choice of the corresponding function.




(a) Ordinary Monte Carlo. Ten price paths of a vanilla European put option generated by a standard Monte Carlo method.




(b) Monte Carlo based on antithetic variates. Ten price paths of a vanilla European put option generated by Monte Carlo using antithetic variates.

Figure 6: Figure (a) and (b) demonstrate the effect of antithetic variates. They show ten price paths each of a vanilla European put option with the parameters $S(0) = 12$, $K = 20$, $r = 0.06$, $\sigma = 0.5$ and $T = 1$. The corresponding Black-Scholes price lies at 7.23.

Consider now the European put option, which we have discussed above, with the parameters $S(0) = 12$, $K = 20$, $r = 0.06$, $\sigma = 0.4$ and $T = 1$. The correct Black-Scholes price therefore is 7.23. Figure 6 demonstrates the effect of antithetic variates. The left panel (a) shows ten different option price paths computed by Monte Carlo simulations with $n = 10\,000$ replications each. The right panel (b) does basically the same except that it uses antithetic variates as inputs. The variance reduction brought by antithetics compared to ordinary Monte Carlo is obvious.

### 3.1.2 Control variates

The control variates approach basically exploits available information about the errors in estimates of quantities which are known in advance to reduce the error in an estimate of a still unknown quantity. Boyle [11] firstly applied the control variates technique to option pricing in 1977. The previous section revealed that the antithetic variates approach does not require a profound knowledge about the system we are simulating. Control variates, however, is relying heavily on specific and detailed features of the simulation and may obtain better results by systematically exploiting them [46, p.185f]. The price for a higher potential in variance reduction we pay, reluctantly, by an increased complexity of the model.

Let us describe this method by the analysis of Asian options as described by Boyle [11, p.326ff]

as well as by Kemna and Vorst [63, p.121ff]. The basic idea underlying this method is to replace the original problem under consideration by a similar but simpler problem for which an exact analytically tractable solution can be found. As we have already seen in Section 1.2.3, the price of an Asian call option whose payoff depends on the arithmetic average of the underlying asset, denoted as $C_a$, is analytically intractable. Through the control variates method the knowledge of the price of the same option with the price $C_g$, which depends on the geometric average and can be therefore evaluated in closed form, may be leveraged to compute $C_a$ [12, p.1276]. Suppose, now, that the pairs $(C_{i,a}, C_{i,g}), i = 1, \ldots, n$, are iid and that the expectation of $C_{i,g}$, $E[C_g]$, is a known quantity. Then for any fixed $\beta$ an unbiased estimator of $E[C_a]$ is thus provided by

$$C_{i,a}^{\beta} = C_{i,a} - \beta(C_{i,g} - E[C_g]). \tag{3.6}$$

Calculating the sample means reveals

$$\bar{C}_a^{\beta} = \bar{C}_a - \beta(\bar{C}_g - E[C_g]), \tag{3.7}$$

where the observed error $\bar{C}_g - E[C_g]$ is used as a *control* in estimating $E[C_a]$ [12, 46, 59, p.1275, p.186, p.113]. This estimator is unbiased and consistent.

The variance of each $C_{i,a}^{\beta}$ is given by [12, p.1275]

$$\begin{aligned} \operatorname{Var}[C_{i,a}^{\beta}] &= \operatorname{Var}[C_{i,a} - \beta(C_{i,g} - E[C_g])] \\ &= \operatorname{Var}[C_a] + \beta^2 \operatorname{Var}[C_g] - 2\beta \operatorname{Cov}[C_a, C_g]. \end{aligned} \tag{3.8}$$

Now it is straightforward to see that the control variate estimator exhibits smaller variance than the ordinary Monte Carlo estimator if $\beta^2 \operatorname{Var}[C_g] < 2\beta \operatorname{Cov}[C_a, C_g]$ [59, p.113ff].

The optimal coefficient $\beta^*$ minimizes the control variates estimator's (3.8) variance and is therefore given by [12, p.1275]

$$\beta^* = \frac{\operatorname{Cov}[C_a, C_g]}{\operatorname{Var}[C_a]}. \tag{3.9}$$

As some concluding observations one can say that the effectiveness of a control variate heavily depends on the strength of the correlation between the quantity of interest $C_a$ and the control $C_g$, whereas the sign of the correlation is absolutely irrelevant. Thus, an estimator that is based on $\beta^*$ is guaranteed not to increase variance. It will rather result in a strict decrease in variance as long as $\rho_{C_a,C_g} \neq 0$. Furthermore, the variance reduction factor $1/(1 - \rho_{C_a,C_g}^2)$ and $|\rho_{C_a,C_g}|$ moves in parallel. The former rises sharply as the latter approaches 1. Accordingly, the variance reduction factor declines quickly as $|\rho_{C_a,C_g}|$ decreases away from 1 [46, p.187]. In summary, a high degree of correlation is required for control variates to yield substantial benefits.

In practice, of course, we rarely know $\beta^*$ because we seldom happen to know $\operatorname{Cov}[C_a, C_g]$. However, given $n$ independent replications $(C_{i,a}, C_{i,g})$ we can estimate the true parameter $\beta^*$ by a least-squares regression approach [12, 103, p.1275f, p.11]:

$$b = \frac{\sum_{i=1}^{n}(C_{i,g} - \bar{C}_g)(C_{i,a} - \bar{C}_a)}{\sum_{i=1}^{n}(C_{i,g} - \bar{C}_g)^2}. \tag{3.10}$$

This equation represents the slope of the regression line through the points $(C_{i,g}, C_{i,a})$, therefore the link between control variates and least-squares regression is obvious. Yet, replacing $\beta^*$ with $b$ introduces some bias because $b$ and $\bar{C}_g$ are not independent. Even though we introduced a bias the estimator is still consistent, i.e. if $n \to \infty$ then $b \to \beta^*$ [12, p.1276]. However, increasing $n$ is actually not the purpose of variance reduction techniques since we basically want to decrease the variance with a limited $n$. Nevertheless, there is a relatively simple way to eliminate bias. Use, in a first step, $n_1$ replications to calculate a reasonable estimate $b$ and do then, in a second step, apply this $b$ with the remaining $n - n_1$ replications of $(C_{i,g}, C_{i,a})$ [46, p.200]. Thus, the resulting estimator $C_a^b$ will be unbiased even though we use $b$ as an estimate for the true coefficient $\beta^*$ [12, p.1276].

## 3.2 Stratified sampling

The goal of the stratified sampling approach is to make the random variates, which serve as inputs for the actual simulation, more regular [12, p.1279f]. This is basically what many other variance reduction techniques do as well. In particular, it constrains a certain fraction of observations drawn from specific subsets, also called strata, of the sample space. Suppose that our goal is to estimate $E[X]$ and that $X$ is in some way dependent on the value of another random variable $Y$, the so called *stratification variable*, with known probability. Let $A_1, \ldots, A_k$ be disjoint strata. Thus, the stratification variable $Y$ has a discrete probability function $P(Y \in A_j) = p_j, \forall\ j = 1, \ldots, k$. Using conditioning, then [76, p.175ff]

$$E[X] = \sum_{j=1}^{k} E[X|Y \in A_j] p_j. \tag{3.11}$$

Consider, for example, we want to use Monte Carlo to compute $\theta = \int_0^1 h(x)dx = E[h(U)]$. In a standard Monte Carlo simulation one would simply draw $n$ uniforms $U_i$ and compute the Monte Carlo estimator, i.e. here a simple sample mean

$$\frac{1}{n} \sum_{i=1}^{n} h(U_i).$$

However, one may obtain a superior estimator by dividing the integration interval $(0, 1)$ into $k$ subintervals $((j-1)/k, j/k)$. Secondly, for each subset $j = 1, \ldots, k$, $n_j$ random uniforms $U_m$ are generated in order to estimate

$$\hat{\theta}_j = \frac{1}{n_j} \sum_{m=1}^{n_j} h\left(\frac{U_m + j - 1}{k}\right).$$

Afterwards, using these single estimates as inputs we can build the final estimator [15, p.260f]

$$\hat{\theta} = \sum_{j=1}^{k} \hat{\theta}_j p_j. \tag{3.12}$$

Suppose that we use stratified sampling to force 100 normal random variates $Z_1, \ldots, Z_{100}$ to lie between the $j-1$ and $j$ percentile and thus produce a much better match to the normal distribution although of the limited sample size. Since $\tilde{Z}_1, \ldots, \tilde{Z}_{100}$ will be highly dependent, it is more complicated to estimate a standard error and it typically requires batching the runs. Assume, for instance, that your computational budget allows a maximum of 100 000 replications. Then we may run 100 independent stratified samples each of 1 000 size, rather than one single stratified sample of the maximal budget size only [12, p.1280]. Thus, it is inevitable to sacrifice some variance reduction potential in order to estimate standard errors. However, this is usually the case in general variance reduction techniques.

Let us go one step further and have a look at the variance of the estimator $\hat{\theta}$. Suppose that all existing strata are independently sampled. Then the variance rate is given by [15, p.261]

$$\mathrm{Var}[\hat{\theta}] = \sum_{j=1}^{k} p_j^2 \,\mathrm{Var}[\hat{\theta}_j] = \sum_{j=1}^{k} \frac{p_j^2}{n_j} \,\mathrm{Var}[X_j], \tag{3.13}$$

where the random variables sampled in each subset are denoted by $X_j$. Hence, one can read out of (3.13) that the variance minimization strategy is to simply allocate more samples to the strata where $\mathrm{Var}[X_j]$ is larger.

Different stratified sampling approaches vary in both complexity and in effectiveness. Stratifying a uniform distribution, for instance, is rather trivial but unfortunately also of limited benefit. On the other hand, stratified sampling where the stratification variable is exactly tailored to a certain model is at a much higher level of complexity. In contrast to the control variates approach of Section 3.1.2, which only required knowing its mean, using a variable for stratification requires the knowledge of its entire distribution and is thus more powerful than just accounting for its mean [46, p.278].

In principle, the stratified sampling approach applies in arbitrary dimensions. Stratified sampling extended to multiple dimensions $d$ is called *latin hypercube* sampling. The difficulty in especially high dimensions is that generating even a single stratified sample of size $n^d$ may be unattainable unless $n$ is exceptionally small, in which case stratification provides rather little benefit. If we partition each coordinate into $k$ strata, we will end up with already $k^d$ strata for the hypercube [12, 46, p.1281, p.236]. Therefore latin hypercube sampling may be possible in principle but often simply infeasible in practice. Nevertheless, the interested reader is referred to the paper of Stein [99, p.144ff] or to the Chapter 4.4 of Glasserman [46, p.236ff], both provide a detailed discussion about this topic.

## 3.3 Importance sampling

Importance sampling is in many respects by far the most delicate of the variance reduction techniques discussed in this chapter. In comparison with other approaches it has the capacity to exploit detailed knowledge of the model to achieve a higher level of variance reduction [46, p.278].

Importance sampling is based on the observation that a simple expectation under one probability measure can be expressed as an expectation under another. Thus, it basically attempts to reduce variance by changing probability measures [12, p.1283]. Changing measures is a standard tool in finance. Consider, for instance, option pricing principles where we switch from an objective probability measure to risk-neutrality. Importance sampling tries to give more weight to important outcomes and may therefore be particularly useful if you intend to simulate rare events or from the tails of a distribution. Then, in contrast to a crude Monte Carlo approach, not needlessly many samples will be wasted, as the required events will rarely occur, and thereby efficiency of the simulation procedure may be increased substantially [59, p.115f]. This is what importance sampling may let to become very useful in certain situations.

To flesh out the key idea of this approach, consider the problem of estimating

$$\theta = E[h(X)] = \int h(x)f(x)dx,$$

where $X$ is a random vector with joint density $f(x)$. The crude Monte Carlo estimator is therefore given by $\hat{\theta} = 1/n \sum_{i=1}^{n} h(X_i)$. Suppose we know another density $g$ satisfying $f(x) \geq 0$ whenever $g(x) \geq 0$. Then we may alternatively represent $\theta$ as an expected value with respect to the density $g$ [15, 46, p.262f, p.255f]:

$$\theta = \int h(x)\frac{f(x)}{g(x)}g(x)dx = E_g\left[h(X)\frac{f(X)}{g(X)}\right]. \tag{3.14}$$

According to (3.14) the estimator associated with $g$ for the importance sampling approach therefore is [46, p.255]

$$\hat{\theta}_g = \frac{1}{n}\sum_{i=1}^{n} h(X_i)\frac{f(X_i)}{g(X_i)}, \tag{3.15}$$

where the weight $f(X_i)/g(X_i)$ is the so called *likelihood ratio* evaluated at $X_i$, which is used to correct the change in probability measure. Note that if we use random sampling, this ratio will be a random variable, too. From Equation (3.14) follows that $E_g[\hat{\theta}_g] = \theta$, what means that $\hat{\theta}_g$ must be an unbiased estimator of $\theta$. Thus, in order to compare variances it is sufficient to simply compare second moments. Therefore importance sampling successfully achieves a variance reduction if [46, p.255ff]

$$E_g\left[\left(h(X)\frac{f(X)}{g(X)}\right)^2\right] = E\left[h(X)^2\frac{f(X)}{g(X)}\right] < E[h(X)^2].$$

Notice that importance sampling does not always bring a reduction in variance. Indeed, the method may backfire if $g$ is not chosen with care and the resulting variance might even be infinitely larger than applying a crude Monte Carlo. Obviously, successfully applying importance sampling lies in the art of choosing an effective importance sampling density function $g$.

To gain more insight into the key idea of choosing $g$ well [15, p.262f], let us take a closer look at the example above, where we are interested in estimating

$$\theta = E[h(X)]$$

with $h(x) \geq 0$. As we have already seen above, $\theta$ may be estimated in two different procedures:

$$\begin{aligned} E[h(X)] &= \int h(x)f(x)dx \\ &= \int h(x)\frac{f(x)}{g(x)}g(x)dx = E_g\left[h(X)\frac{f(X)}{g(X)}\right]. \end{aligned}$$

The two estimators have the same expected values, but the variances may be different:

$$\mathrm{Var}[h(X)] = \int h^2(x)f(x)dx - \theta^2 \tag{3.16}$$

$$\mathrm{Var}_g\left[h(X)\frac{f(X)}{g(X)}\right] = \int h^2(x)\frac{f(x)}{g(x)}f(x)dx - \theta^2. \tag{3.17}$$

Equation (3.17) discloses that the following choice of $g$,

$$g(x) = h(x)\frac{f(x)}{\theta},$$

leads to the ideal condition of a *zero variance*, $\mathrm{Var}_g[h(X)f(X)/g(X)] \equiv 0$. Thus, however we sample, the result will always be correct. Of course, there is a rub in it, using this density requires that we already know the answer we are looking for in advance, i.e. the value of $\theta$. Nevertheless, this example does provide some useful guidance regarding the optimal choice of the importance sampling function. Consider the difference between the two variances

$$\Delta\,\mathrm{Var} = \mathrm{Var}[h(X)] - \mathrm{Var}_g\left[h(X)\frac{f(X)}{g(X)}\right] = \int h^2(x)\left[1 - \frac{f(x)}{g(x)}\right]f(x)dx.$$

We can read out of this equation that ensuring a variance reduction by importance sampling requires to choose the new density $g$ such that [15, p.263]

$$\begin{cases} g(x) > f(x) & \text{when the term } h^2(x)f(x) \text{ is large,} \\ g(x) < f(x) & \text{when the term } h^2(x)f(x) \text{ is small.} \end{cases}$$

Thus, we should try to sample in proportion to the product of $h$ and $f$. The name importance sampling derives from this doubtless crucial observation.

### 3.4 Conclusion

To top this section about variance reduction techniques off, we will provide a short comparison of the different approaches discussed in the previous subsections. Firstly, consider Figure 7 [46, p.277], which gives an overview of all four techniques in terms of complexity and potential efficiency gains. As already anticipated and also partially mentioned, complexity and effectiveness are related positively, meaning that the former is gradually increasing with the latter.

Secondly, looking at Table 4, which shows the implementation of antithetic variates, control variates as well as importance sampling to price a vanilla European call option, we can read out

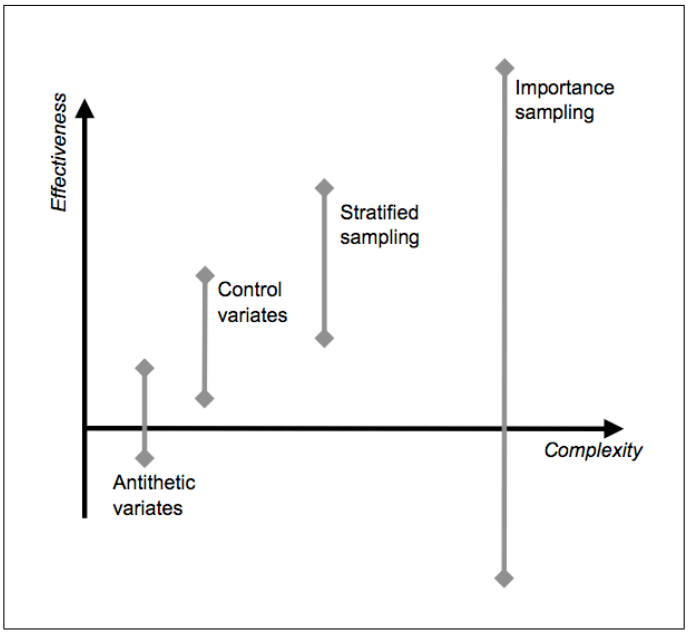



Figure 7: Variance reduction: overview [46, p.277]

the effectiveness in terms of convergence rate or speed. Antithetic as well as control variates both provide a substantial variance reduction compared to the crude Monte Carlo estimates. The corresponding standard errors of these two approaches, which are indicated in parentheses, drop off much faster than those of the standard Monte Carlo. Moreover, importance sampling did not bring any reduction in variance at all. This phenomenon is already discussed in Section 3.3 and illustrated in Figure 7. The observed increase instead of an intended decline in variance may be ascribed to an unwise choice of the importance sampling density $g$. For more detailed explanations regarding the single calculations and formula it is referred to Appendix A where the commented MATLAB codes are individually listed on Pages IV to VII.

| $n$ | Standard MC | Antithetic variates | Control variates | Importance sampling |
|---|---|---|---|---|
| 10 000 | 10.8473<br>(0.1706) | 11.3479<br>(0.0934) | 11.1833<br>(0.0680) | 11.4153<br>(0.2262) |
| 100 000 | 11.2682<br>(0.0784) | 11.2060<br>(0.0295) | 11.2230<br>(0.0205) | 11.2718<br>(0.0720) |
| 500 000 | 11.2205<br>(0.0245) | 11.2544<br>(0.0133) | 11.2376<br>(0.0090) | 11.2309<br>(0.0319) |
| 1 000 000 | 11.2489<br>(0.0174) | 11.2453<br>(0.0093) | 11.2379<br>(0.0065) | 11.2491<br>(0.0226) |

Table 4: Using antithetic variates, control variates and importance sampling to price a vanilla European call option, with the parameters $S(0) = 75$, $K = 72$, $r = 0.03$, $T = 0.75$ and $\sigma = 0.35$, by Monte Carlo simulation. The standard errors are indicated in parentheses.

## 4 Quasi-Monte Carlo

A *quasi-Monte Carlo*, also called *low-discrepancy* method can be described in simple terms as the *deterministic* version of an ordinary Monte Carlo method. Deterministic in the sense that the random, or rather pseudorandom samples in the Monte Carlo method are replaced by wisely-chosen deterministic points. Quasi-Monte Carlo therefore makes no attempt to mimic randomness. It rather generates sample points that are literally too evenly distributed to be random and thus selectively tries to increase accuracy [46, p.281]. In other words, it covers the $d$-dimensional space with fewer unintentional gaps compared with independent random variables. Unlike random numbers, low-discrepancy numbers aim not to be serially uncorrelated, since perfect decorrelation and thus randomness is "merely an artefact of the custom to construct multidimensional vector variates from one and the same underlying one-dimensional number generator" [59, p.77]. They rather analyze the domain to be sampled and take these points into account, which have already been probed. Therefore, quasi-Monte Carlo sequences are autocorrelated by construction and thus would fail almost all of the tests (see Section 2.2.4) applied to pseudorandom number generators. The only exception are those testing the uniformity of the marginal distribution. Detailed analysis revealed [85, p.3] that, in Monte Carlo integration, for instance, not the (true) randomness of the samples is primarily relevant, but rather that the samples should be spread in a uniform manner over the integration domain.

We have already seen that the main deficiencies of Monte Carlo methods are that to achieve a reasonable level of accuracy may be computationally burdensome and that the Monte Carlo error term is probabilistic. Therefore, the main goal is to choose those deterministic points in such a way that the resulting deterministic – and thus guaranteed – error bound is smaller than the probabilistic Monte Carlo error bound. In principle, there is always the possibility to determine in advance an integration rule that yields a prescribed level of accuracy. Moreover, with the same computational effort, the quasi-Monte Carlo method achieves a significantly higher accuracy. Thus, in terms of *determinism* and *precision*, quasi-Monte Carlo is clearly superior to the ordinary Monte Carlo method [85, p.10].

Niederreiter showed in his pathbreaking book *Random Number Generation and Quasi-Monte Carlo Methods* [85, p.14ff] that low-discrepancy methods are utterly powerful techniques that have the potential to accelerate convergence from the $O(1/\sqrt{n})$ rate associated with Monte Carlo to a deterministic error bound of nearly $O(1/n)$. Variance reduction techniques, as we have seen in Section 3, only affect the implicit constant in $O(1/\sqrt{n})$ and are therefore not a quarter as ambitious [46, p.282].

The only problem of quasi-Monte Carlo methods is that their convergence is not independent of dimensionality anymore. Thus, especially for high-dimensional computations we can not know in advance if, with respect to the accuracy required for the specific computation, the use of low-discrepancy numbers will be superior to crude Monte Carlo. Pure theoretical considerations are no longer sufficient to decide whether any particular low-discrepcancy sequence will lead to a speed-up in convergence. Accordingly, we have to fall back on empirical results [59, p.77f].

### 4.1 Discrepancy

The focus of quasi-Monte Carlo methods lies on the actual objective, namely approximating an integral, rather than simply trying to imitate the behavior of independent uniforms. In particular, suppose this objective is to compute

$$E[f(U_1, \ldots, U_d)] = \int_{[0,1]^d} f(x)dx. \tag{4.1}$$

Quasi-Monte Carlo will approximate Integral (4.1) using an average of the function at $n$ points like

$$\int_{[0,1]^d} f(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i), \tag{4.2}$$

for wisely and deterministically chosen points $x_1, \ldots, x_n \in [0,1]^d$ [46, 59, 76, p.282, p.78, p.249f]. The goal of low-discrepancy sequences is to construct points $x_1, \ldots, x_n$ in such a way that makes the error in Approximation (4.2) as small as possible for a wide class of integrands. It is intuitively clear and

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} f(x_i) = \int_{[0,1]^d} f(x)dx \tag{4.3}$$

also refers that this is equivalent to choosing the points $x_i$ to fill the hypercube uniformly, thus to achieve a maximal degree of uniformity and a low degree of discrepancy, respectively. Various notions of discrepancy have been considered to measure the deviation from the uniform distribution quantitatively, or, to put it into other words, how inhomogeneously a set of $d$-dimensional vectors is distributed in the unit hypercube [76, p.249ff].

Let $B \subseteq [0,1]^d$ be an arbitrary axially parallel $d$-dimensional rectangle in the unit cube $[0,1]^d$. The key idea behind discrepancy is that for an evenly distributed point set the fraction of the points lying within $B$ should correspond to the rectangle's volume. Then, a general notion of discrepancy of the point set $P = \{x_1, \ldots, x_n\}$ is given by [46, 76, 85, p.283, p.249f, p.14]

$$D_n(P) = \sup_B \left| \frac{\#\{x_i \in B\}}{n} - \mathrm{vol}(B) \right|. \tag{4.4}$$

Note that $0 \leq D_n(P) \leq 1$, always. Thus, the ordinary discrepancy $D_n$, also called *extreme* discrepancy, is the supremum over errors in integrating the indicator function of $B$ using the point set $P$. Restricting $B$ to rectangles for which one corner is the origin,

$$B^* = \prod_{j=1}^{d} [0, u_j)$$

defines the *star* discrepancy, denoted as $D_n^*(P)$. The more evenly the points of a sequence are distributed, the closer the discrepancy $D_n$ is to zero [46, p.283f]. Furthermore, Niederreiter showed [85, p.15 (Proposition 2.4.)] that

$$D_n^*(P) \leq D_n(P) \leq 2^d D_n^*(P),$$

and therefore if we fix $d$, $D_n$ and $D_n^*$ have an equal order of magnitude.

Niederreiter [85, p.31ff] explained that the term *low-discrepancy* is customary used do describe methods that achieve a star discrepancy of $O((\log n)^d/n)$, which expresses the dependence of quasi-Monte Carlo methods on the dimension $d$. Since $\log n$ grows only modestly and becomes negligible relative to $n$, low-discrepancy essentially means that $D_n^* \approx O(1/n)$ as long as $d$ is not too large. Accordingly, the application of quasi-Monte Carlo methods has traditionally been limited to problems of rather low dimensions. However, recent research in finance has low-discrepancy methods been found to be effective in much higher dimensions, too [46, p.285].

## 4.2 Low-discrepancy sequences

### 4.2.1 Halton

Primarily, a specific class of one-dimensional low-discrepancy sequences is introduced. In a one-dimensional case the point set

$$x_i = \frac{2i-1}{2n}, \quad i = 1, \ldots, n \tag{4.5}$$

has the value $D_n^* = 1/(2n)$, which can not be improved [85, p.23]. This monotonous Sequence (4.5) can be applied only when a reasonable $n$ is fixed. However, $n$ is often considerably large or even $n \to \infty$. Thus, $x_i$ will be newly placed and therefore it is absolutely essential that with a growing $n$ previously calculated results can be used again. This means that the points $x_1, x_2, \ldots$ must be replaced dynamically. And this is exactly achieved by the *Van der Corput* sequence, which is based on a simple recipe [15, p.270]:

1. Represent an integer number $k$ in a certain base $b$, where $b$ is also an integer and $b \geq 2$:

$$k = (\ldots d_4 d_3 d_2 d_1 d_0)_b, \quad \text{with digits } d_j \in \{0, 1, \ldots, b-1\}.$$

2. Firstly, reverse the binary digits and secondly add a radix point to obtain a number within the unit interval:

$$\psi = (.d_0 d_1 d_2 d_3 d_4 \ldots)_b.$$

Let us have a look at a straightforward example, say, $x_6$. The index $k = 6$ is written as a binary number $6 = (0110)_2 \equiv (d_3 d_2 d_1 d_0)_2$. Thus, $\psi_2(6) = (.d_0 d_1 d_2 d_3) = (1/2^2) + (1/2^3) = 3/8$. If this is done for all indices $k = 1, 2, \ldots$, the Van der Corput sequence $x_1, x_2, \ldots$ results [76, p.252].

We can write this more formally as

$$k = \sum_{j=0}^{\infty} d_j(k) b^j, \tag{4.6}$$

then the $k$th number in the Van der Corput sequence with base $b$ is defined by the radical-inverse function [46, p.286]

$$\psi_b(k) = \sum_{j=0}^{\infty} d_j(k) b^{-(j+1)}. \tag{4.7}$$

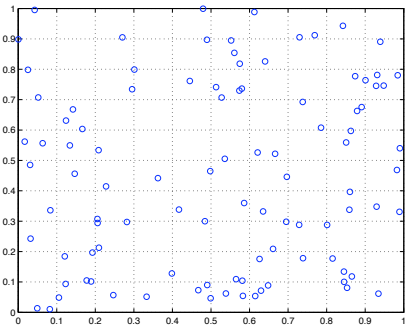
A Van der Corput sequence is therefore obtained by $x_k \equiv \psi_b(k)$ [85, p.24].

The *Halton* sequence, proposed by Halton [50] in 1960, is simply the multivariate extension of the Van der Corput sequence. Hence, the coordinates of a Halton sequence follow a crude Van der Corput sequence in distinct bases [46, p.293]. Thus to construct a simple Halton sequence let $b_1, \ldots, b_d$ be relatively prime integers and set

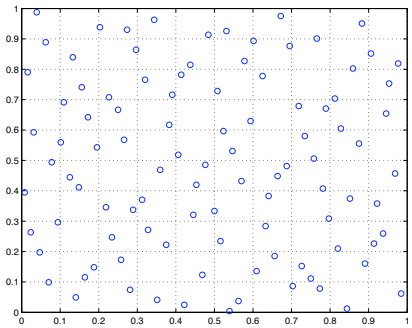
$$x_k = (\psi_{b_1}(k), \psi_{b_2}(k), \ldots, \psi_{b_d}(k)), \quad k = 0, 1, 2, \ldots. \tag{4.8}$$

Compared to Equation (4.7) one may quickly recognize that the Halton sequence is indeed obtained by associating a Van der Corput sequence to each of the multiple dimensions. Thus, it is an extended Van der Corput sequence only [76, p.252f].

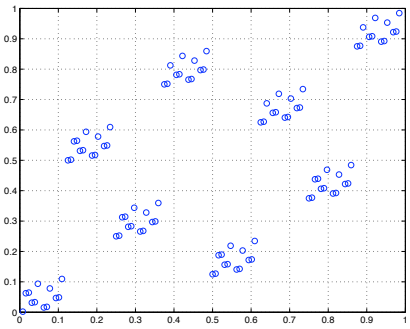
Let us now apply Equation (4.8) and generate some Halton sequences. Consider the two panels (a) and (b) on the top of Figure 8 which show that, in a two-dimensional case, Halton points, generated with bases $b_1 = 2$ and $b_2 = 3$, are obviously more evenly distributed than ordinary random points. Therefore, one usually takes $b_1, \ldots, b_d$ to be the first $d$ prime numbers [15, 46, p.279f, p.293]. The requirement that the $b_i$ are relatively prime is necessary to prevent any asymptotic pairwise periodicity and thus to fill the hypercube as evenly as possible. If we choose the bases to be 2 and 8, respectively, the resulting Halton sequence will be rather unsatisfactory in terms of uniformity as the bottom left panel (c) illustrates. The second requirement, beside that the $b_i$ must be primes, is that the bases are relatively small. This stems from the deterioration of the Halton sequence in high dimensions, which directly follows from the behavior of the Van der Corput sequence with a large base. A Van der Corput sequence in base $b$ consists of numerous consecutive monotone segments of length $b$. If $b$ is large, the sequence consequently produces undesirably long monotone segments. Furthermore, if we project a Halton sequence onto coordinates and use too large bases, the result will be long diagonal segments in the projected hypercube [46, p.294f]. This phenomenon becomes clear in panel (d) of Figure 8. Accordingly, applying Halton sequences in especially high-dimensional problems is of little benefit since the required uniformity degrades in higher dimensions because the base of the $d$th coordinate is at least as large as the $d$th prime, and this grows superexponentially with $d$ [85, p.47]. Therefore other methods as *Faure* or *Sobol* sequences have been proposed.



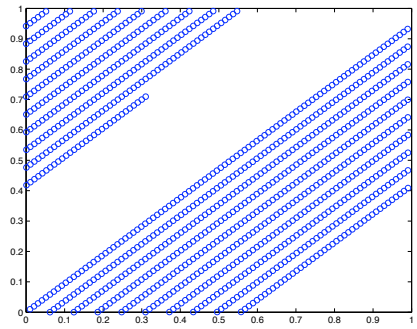
(a) Bidimensional pseudorandom numbers



(b) Bidimensional Halton sequences with $b_1 = 2$ and $b_2 = 3$



(c) Bidimensional Halton sequences with $b_1 = 2$ and $b_2 = 8$.



(d) Bidimensional Halton sequences with $b_1 = 97$ and $b_2 = 103$.

Figure 8: Random numbers and Halton sequences. The corresponding code to generate these Halton sequences in MATLAB is listed in Appendix A on Page VIII.

### 4.2.2 Faure

Faure [37] developed a sequence that is similar to the Halton sequence in that each dimension is a permutation of a Van der Corput sequence, however, all coordinates use a common base $b$. The base of a Faure sequence is the smallest prime that is larger than or equal to the number of dimensions, $b \geq d$ [15, p.280].

The Faure sequence possesses better regularity properties than does the Halton sequence especially for moderately high dimensions. Consider, for instance, that for the dimensionality $d = 20$, the Halton sequence takes the 20th prime number, which is 71, as the last component, whereas the Faure sequence uses the first prime number after 20, which is 23. Therefore the gaps in high dimensionality are filled substantially faster with the Faure than with the Halton sequence. Consequently, Faure sequences are expected to be superior to Halton sequences in large $d$ [101, p.317f].

### 4.2.3 Sobol

Niederreiter [84] introduced the notion of a $(t, m, d)$-*net* and a $(t, d)$-*sequence* in order to construct and describe point sets and sequences with low discrepancy. Briefly, a $(t, m, d)$-net is a finite set composed of points stemming from $[0, 1]^d$. The degree of uniformity that this net contains is quantified by the variable $t$. Furthermore, $(t, d)$-sequences eventually form $(t, m, d)$-nets. They consist of points from carefully chosen segments. Sobol [98] extended this idea and introduced the first construction of a specified $(t, d)$-sequence with base 2.

By way of comparison, Faure points are $(0, d)$-sequences in a base equal or larger than $d$ whereas Sobol points are $(t, d)$-sequences in a fixed base 2 valid for all $d$. The values of the parameter $t$ depend upon the dimensionality $d$. Thus, while Faure points achieve the best value of the uniformity parameter $t$, Sobol points take advantage of a considerably smaller base [46, p.303].

Like all other methods we have already discussed above, Sobol points also always start from a Van der Corput sequence. Yet it does not include different bases anymore, but exclusively base 2. The Sobol sequence is generated using a set of so called *direction numbers* $\{v_i\}$,

$$v_i = \frac{m_i}{2^i},$$

where $m_i < 2^i$ is an odd positive integer. These direction numbers are generated by exploiting primitive polynomials of modulo 2, i.e. polynomials with binary coefficients only [15, 43, p.281f, p.71f]:

$$f(z) = z^p + c_1 z^{p-1} + \cdots + c_{p-1} z + 1, \qquad c_k \in \{0, 1\}.$$

A primitive polynomial is irreducible (i.e. can not be factorized into polynomials of smaller degree) and does not divide the polynomial $x^q + 1$ for $q < 2^p - 1$ [15, p.282]. The procedure for generating the required direction numbers with a given primitive polynomial of degree $p$ is based on the recurrence formula

$$v_i = c_1 v_{i-1} \oplus c_2 v_{i-2} \oplus \cdots \oplus c_p v_{i-p} \oplus [v_{i-p}/2^p], \qquad i > p,$$

where $\oplus$ denotes a bit-by-bit binary exclusive-or operator. This may be easier implemented in simple integer arithmetic as [15, 43, p.282, p.72]

$$m_i = 2c_1 m_{i-1} \oplus 2^2 c_2 m_{i-2} \oplus \cdots \oplus 2^{p-1} c_{p-1} m_{i-p+1} \oplus 2^p c_p m_{i-p} \oplus m_{i-p}. \tag{4.9}$$

Because Equation (4.9) generates only $m_i$ for $i > d$, the first $d$ odd integers (i.e., $m_1, m_2, \ldots, m_p$), which are needed to initialize the recursion, must be supplied. Fortunately, they may be chosen arbitrarily, provided that each $m_i$ is odd and $m_i < 2^i$ [43, p.72].

The $i$th number of the Sobol sequence may now be formed as

$$x_j = b_1 v_1 \oplus b_2 v_2 \oplus b_3 v_3 \oplus \cdots , \tag{4.10}$$

where $\ldots b_3 b_2 b_1$ is the binary representation of the integer $j$, $j = (\ldots b_3 b_2 b_1)_2$ [45, p.96].

However, Antonov and Saleev [5] proposed an improved method, which is approximately 20% faster than the Sobol's original [43, p.71], and proved that the asymptotic discrepancy remains unchanged by using the *Gray code* representation of $j$. A Gray code is basically a function that maps a nonnegative integer $j$ to a corresponding binary representation $G(j)$. A Gray code representation for $j$ is therefore obtained by computing

$$\ldots g_3 g_2 g_1 = (\ldots b_3 b_2 b_1)_2 \oplus (\ldots b_4 b_3 b_2)_2.$$

The main feature of a Gray code is that the binary representation of $G(j)$ and $G(j+1)$ differ in exactly one bit [15, 45, 59, p.283f, p.96, p.83ff]. This makes it possible to construct a Sobol sequence as follows:

$$x_j = g_1 v_1 \oplus g_2 v_2 \oplus g_3 v_3 \oplus \cdots . \tag{4.11}$$

Applying the advantages of such Gray code representations, we may streamline a Sobol sequence generation and arrange the sequence of Equation (4.11) recursively with a given $x_j$ by

$$x_{j+1} = x_j \oplus v_r, \tag{4.12}$$

where $r$ is determined so that $b_r$ is the rightmost zero bit in the binary representation of $j$ [15, p.284]. Note that, especially in higher dimensions, the uniformity of a Sobol sequence can be considerably sensitive to the starting values [45, p.97].

A Sobol sequence of total $d$ dimensions fills in the unit hypercube in cycles of length 2. Thus, the cycles of a Sobol sequence are shorter than those used by either Halton or Faure sequences in high dimensions. Consequently, a Sobol sequence is able to fill the gaps in the existing sequence at a faster rate with the same budget of points than either of the two other sequences. However, there is the problem that at high dimensions the Sobol points tend to repeat themselves across dimensions, resulting in high-dimensional clustering [43, p.74]. The traditional solution is again to discard the first $n$ points, as it is also done using control variates (see Section 3.1.2). Boyle et al. [12] propose to discard the first 64 points, but this choice appears to be rather arbitrary.

### 4.3 Efficiency of low-discrepancy sequences

There have been published numerous empirical comparisons of quasi- and pseudorandom sequences in the literature that are somewhat inconclusive if not even contradictory [43, 45, p.75, p.97]. On the one hand, Boyle et al. [12] and Joy et al. [62] generally illustrated by computing some examples in their empirical studies that low-discrepancy sequences are more efficient than pseudorandom sequences. Paskov and Traub [87, p.114] compared low-discrepancy methods based on Halton and Sobol sequences with the crude Monte Carlo method and also with the Monte Carlo method combined with antithetic variates. They conclude that the Sobol method consistently outperforms the classical Monte Carlo method because the former terminates two to five times faster than the latter and therefore converges significantly faster. They additionally state that the Sobol sequence beats the Halton sequence. Moreover, the authors deduce that the Sobol method consistently defeats the antithetic variates, which in turn consistently outperform the standard Monte Carlo approach.

On the other hand, Bratley et al. [18] concluded that pseudorandom sequences are generally more efficient than low-discrepancy sequences for dimensions higher than twelve. Thus, the authors do not agree on which of the quasi-Monte Carlo sequences may be more efficient.

Fox [41] compared the efficiency of low-discrepancy sequences and found the performance of Faure sequences to be superior to that of Halton sequences. This finding is shared by the study of Bratley and Fox [17], who also concluded that the performance of the Sobol sequence is roughly the same as that of the Faure sequence and therefore superior to the Halton sequence.

In summary, we can draw the following conclusions from these results of research. Firstly, low-discrepancy sequences tend to outperform pseudorandom sequences for low-dimensional simulations. Secondly, it is still an open issue whether they are more efficient for higher dimensions. But is has been generally agreed that this particular point depends upon the specific case and can not be answered finally. Thirdly, some low-dicrepancy sequences may be more efficient than others, but this conclusion highly depends on the specific problem's dimensionality and is by no means finally drawn.

## 5 Simulating stochastic differential equations

Section 1.2 showed that we need to simulate SDEs in order to estimate quantities of interest if we are unable to compute them analytically. Thus far, we have only introduced methods that may simulate, at a discrete set of dates, continuous-time processes *exactly*. These methods are exact in a sense that the joint distribution of the simulated quantities coincides with the joint distribution of the continuous-time process on the simulation grid [46, p.339]. Exact simulation methods usually rely on some special features of a model and are generally available only for models that offer some kind of tractability. However, such models are exceptional since more complex models for pricing derivatives, especially interest rate derivatives, as we will see in the following sections, must be simulated through, for instance, discretization of SDEs. This means the simulation is no longer exact, but rather an *approximation*.

In general, we can draw the conclusion that there are two potential sources of error in generating sample paths [15, p.430]:

- sampling error,
- discretization error or bias.

Sampling error simply goes back to the random nature of Monte Carlo methods and can therefore be mitigated using either ordinary variance reduction techniques or quasi-Monte Carlo methods, as it is shown in Sections 3 and 4. Note that the former or the latter strategy may be able to improve precision in estimating an expectation at the fixed level of discretization, but it can not do anything to reduce the discretization bias or error [46, p.339]. To establish a basic understanding of this bias, we will look at some discretization methods that provide us with an approximate simulation of an SDE.

### 5.1 Discretization methods

There are several kinds of numerical schemes to solve different SDEs. However, we restrict our discussion to the two probably most fundamental methods, the *Euler scheme* and the *Milstein scheme*. The book by Kloeden and Platen, especially Chapter 9, [64, p.305ff] provides an excellent and comprehensive treatment of that topic and is therefore recommended to everybody who wants to delve into the solutions of SDEs.

As a starting point for both methods, we firstly consider an ordinary SDE of the form

$$dX(t) = \mu(t, X(t))dt + \sigma(t, X(t))\, dW(t). \tag{5.1}$$

Note that the drift $\mu$ and the diffusion $\sigma$ are functions of the process variable $X$ and the time $t$. Suppose that we are interested in simulating values of $X(T)$ without knowing its distribution. Possible reasons for this are either that (5.1) is simply not solvable and therefore one is not able to find an explicit solution for $X(T)$ or that even though one can solve (5.1) the distribution of $X(T)$ can still not be determined [46, 64, p.339f, p.305f].

Keep in mind that when we are simulating an SDE, we are actually simulating a discretized version of an SDE. Therefore, a general hat notation $\hat{X}$ is introduced to indicate that $\hat{X}$ is a time-discretized approximation of the true value $X$.

### 5.1.1 Euler scheme

The Euler scheme is the simplest and probably most common discretization scheme available. As already mentioned we generally simulate a discretized process, $\{\hat{X}_b, \hat{X}_{2b}, \ldots, \hat{X}_{mb}\} = \{\hat{X}(t_i), \hat{X}(t_{i+1}), \ldots, \hat{X}(t_{i+n})\}$, where $\hat{X}(0) = X(0)$ and the time steps $\Delta t$ are constant and denoted by $b$. Furthermore, the variable $m$ indicates the total number of simulated time steps, while $mb = T$.

Taking these preliminary considerations into account, the Euler scheme is given by

$$\hat{X}(t_{i+1}) = \hat{X}(t_i) + \mu\left(t_i, \hat{X}(t_i)\right) b + \sigma\left(t_i, \hat{X}(t_i)\right) \sqrt{b} Z_{i+1} \tag{5.2}$$

where $Z_i$ are independent standard normal random vectors, $Z_i \sim \phi(0, I)$. The implementation of this method is straightforward, at least if the drift, $\mu$, and the diffusion, $\sigma$, are easy to evaluate [33, 46, 59, 64, p.111, p.340f, p.32f, p.305].

### 5.1.2 Milstein scheme

The accuracy of numerical solutions of ordinary differential equations may often be improved by simple Taylor expansions. A very similar strategy also applies to SDEs. Yet it is crucial to account for the rules of Itô calculus in order to maintain consistency [46, p.340].

Inspecting the Euler scheme (5.2) from the perspective of Taylor expansions leads to a possible inconsistency: running a Taylor approximation expands the drift to the order $O(b)$ but the diffusion term only to $O(\sqrt{b})$. This suggests that we focus on the diffusion term in order to advance the Euler scheme [46, p.340]. And exactly there the *Milstein scheme* is applied.

The Milstein scheme is defined as an ordinary Euler scheme where the next order terms of the Itô-Taylor expansion of Equation (5.1) are additionally included. This eventually leads to

$$\begin{aligned}\hat{X}(t_{i+1}) = \hat{X}(t_i) + \mu\left(t_i, \hat{X}(t_i)\right) b + \sigma\left(t_i, \hat{X}(t_i)\right) \sqrt{b} Z_{i+1} + \\ \frac{1}{2}\sigma'\left(t_i, \hat{X}(t_i)\right) \sigma\left(t_i, \hat{X}(t_i)\right) b(Z_{i+1}^2 - 1).\end{aligned} \tag{5.3}$$

The Milstein approximation thus simply involves the addition of a term to the Euler scheme so that both the drift and diffusion terms are expanded to $O(b)$ in order to eliminate the inconsistency. The Milstein scheme therefore provides a refinement of the Euler scheme based on expanding the diffusion term to $O(b)$ rather than just $O(\sqrt{b})$ [46, 59, p.343, p.33]. Note that the additional term

$$\frac{1}{2}\sigma'\left(t_i, \hat{X}(t_i)\right) \sigma\left(t_i, \hat{X}(t_i)\right) b(Z_{i+1}^2 - 1)$$

has a zero mean and is uncorrelated with the Euler terms because the correlation between $Z_{i+1}^2 - 1$ and $Z_{i+1}$ is zero as well [46, p.343].

However, the question remains whether it is of great difference if one applies Euler's or Milstein's approximation for a particular diffusion equation. In conclusion, the approximations turn out to be very close in many simple cases. In more sophisticated models, the Milstein scheme appears to be a little better, which may be the additional benefit of the increased complexity. Of course, the two approximations do not differ at all if the diffusion coefficient $\sigma(t, X(t))$ does not depend on $X(t)$ [76, p.153].

## 5.2 Discretization error

In general, the discretization error can be defined as the error that occurs by approximating a continuous-time function in a discrete-time grid. Formally, the discretization error is given by $e \equiv |E[g(Y(t))] - E[g(\hat{Y}(T))]|$. Recall that we are eventually interested in an unbiased estimate of $E[g(Y(T))]$. Therefore, while discretizing a continuous-time function, the ultimate goal is to keep $e$ as small as possible. In any case where $e$ is large, our simulation will provide us with unbiased estimates for $E[g(\hat{Y}(T))]$ but not for $E[g(Y(T))]$, our actual quantity of interest. Therefore, an appropriate discretization method is crucial to achieve precise and correct results by Monte Carlo [51, 64, 85, p.4f, p.278, p.42].

Moreover, taking discretization errors into account, there is a tradeoff between the sample size, $n$, and the number of discretization points, $m$. Since both parameters are firstly positively related to the accuracy and precision of the Monte Carlo estimate and secondly eating up computational performance at the same time, it is of primary importance to allocate the existing computational budget to $n$ and $m$ in an optimal manner [51, p.5].

## 6 Term structure modeling

Thus far, we have only dealt with the Monte Carlo method in general and its application to the pricing of equity derivatives, since they seem to build a reasonable starting point to introduce and motivate simulation techniques. However, the second purpose of this paper is, together with a profound treatment of Monte Carlo methods by itself, to implement the discussed Monte Carlo techniques and provide at the same time some useful guidance on how different interest rate derivatives may be priced by simulation.

The volume of traded interest rate derivatives in both the OTC and exchange-traded markets have been increasing sharply over the past decades. As the number of new and simultaneously exotic interest rate products virtually exploded, it has been a key challenge to find good and primarily robust procedures for pricing and also hedging these products. Interest rate derivatives are more difficult to value than equity derivatives, which are often priced by an ordinary or some slight variation of the Black-Scholes model. The following points give some reasons for this [56, p.611]:

- Interest rates do typically not follow a GBM as stock prices normally do.
- In most cases, we have to describe the development and general behavior of the entire term structure in order to get reasonable prices.
- Volatilities can no longer assumed to be constant since they highly depend on particular points of the yield curve. This makes intuitively sense if we consider the volatility of a bond price, which becomes smaller and smaller as the duration decreases and its maturity is approaching.
- Interest rates are no longer used for discounting purposes only but they additionally define the payoffs from the derivatives.

One could actually say that any bond is an interest rate derivative because its value depends on interest rates. Therefore, if one tries to model interest rates as stochastic processes, as we have done with stock prices in Section 1.2.1 in Figure 2, one might apply the option pricing theory in order to price simple zero coupon bonds [15, p.124]. Doubtlessly, the most simple approach to value interest rate derivatives is the so called *Black model*. This model was proposed by Fischer Black [7] and originally constructed for the pricing of options on commodities. However, the great success of this straightforward model for vanilla interest rate options, such as bond options, interest rate caps and floors as well as swaptions, has been both the blessing and the bane of more sophisticated approaches. It is, in fact, vital to emphasize the strict limitations of the Black model [92, Preface, p.xv]. It assumes, what is admittedly rather unrealistic, that changes in interest rates are lognormally distributed and that volatilities are constant. Yet, it does not require a GBM as this the ordinary Black-Scholes model does [56, p.613].

The need to go beyond Black's closed-form formula in order to be able to value more sophisticated products has led to the development of several methods for simulating stochastic

interest rate models. More precisely, these models describe the entire term structure instead of a single final value. The ultimate purpose of interest rate modeling is to acquire a profound understanding of interest rate behavior. We would like to explore the dynamics of interest rates by fitting a model to available interest rate data. Furthermore, we are interested in the way that interest rates and corresponding derivative prices eventually relate together, both to understand them and last but not least to be able to price interest rate products correctly [60, p.16].

## 6.1 Types of interest rate models

Numerous different models have been introduced over the past years, all with the ultimate purpose to capture the uncertain dynamics of interest rates. In general, we can classify these models by the following two different criteria.

### Number of stochastic factors

In the simplest models, the so called *one factor* models, we describe the dynamics of the *short rate* $r(t)$ (Section 6.2) or the *forward rate* $f(t,T)$ (Section 6.3) by just one stochastic factor, thus there is one source of uncertainty only. As a matter of fact, bond prices, for instance, depend not only on one single but rather on the entire term structure of interest rates. Therefore, one factor models are essentially assuming that the dynamics of the whole term structure is captured by the short or forward rate, and its future evolution [15, p.126]. Nevertheless, one factor models are not as restrictive as it might appear. They imply that all rates move in the same direction over any short time interval, but they do not assume that they move by the same amount [56, p.650].

Yet, it is sometimes difficult or even infeasible to find a more or less realistic model based on one factor only. A principal component analysis, conducted by Rebonato [92, p.263], indicated that a large proportion of the variance of interest rates of different maturities may be satisfactorily explained by including two or three orthogonal factors. The first relevant principal component explained the average level of the yield curve, the second the curve's slope, and the third its curvature. These three factors eventually accounted together for approximately 95 to 99% of the observed term structure variability.

On the other hand, a *multifactor* model means that we have to deal with an increased complexity of the model as we will learn in the course of the following sections. Generally, in a $q$-factor model, where $q$ denotes an arbitrary, usually small number of factors, the price changes of these $q$ bonds completely explain the changes in the prices of all other existing bonds. Moreover, with additional factors the required number of sample paths to achieve a particular level of accuracy in the simulation grows with the power of the number of factors [15, 60, p.126, p.351]. Therefore, the choice of the number of factors in a model should be made with care.

### Equilibrium or no-arbitrage models

The starting point for *equilibrium* models is usually given by assumptions about how the economy works and about economic variables themselves. Then they attempt to derive a process for the short rate $r(t)$ and explore what the defined process for $r$ may imply for bond and option prices [56, p.650]. Thus, the key idea behind this approach is to build an economically sound model, whose output are interest rates evolved as a consequence of market equilibrium [15, p.126]. It may happen that the initial theoretical set of bond prices is not equal to the prices observed on the market. This might result in possible arbitrage opportunities. This is exactly the point where equilibrium models have often been criticized. They do not automatically fit today's term structure of interest rates. The current term structure is included as an output rather than an input. This eventually gives rise to inconsistencies with the actual term structure. Therefore, in practice there is little confidence in the bond option prices when the applied model is not even able to price the underlying bond correctly [24, 56, 77, p.24, p.654, p.93].

The *no-arbitrage* model is an alternative idea which is trying to build models that match the currently observed term structure exactly, that means the current term structure is used as an input. Thus, these models also provide us with underlying bond prices that are correct. However, in order to fit today's term structure, the model parameters have to be calibrated to prevailing market data. The calibration process is, amongst others, responsible that a model eventually yields consistent instrument prices. Nonetheless, calibrating a model correctly is somehow burdensome and takes significant additional computing effort [77, p.93].

## 6.2 Short rate models

*Short rate models* have been introduced in order to be able to explore the development of the dynamics of an instantaneous continuously compounded short rate $r(t)$. In this section the focus lies on simulating some of these simple yet important models. Assume that an investment of 1 in a money market account at time $t$, earning an interest rate of $r(u)$ at time $u$, grows to a value of

$$V(T) = \exp\left(\int_t^T r(u)du\right) \tag{6.1}$$

at time $T$ [46, 60, p.108, p.176f]. Therefore, the arbitrage-free price of a derivative instrument at time $t$ that yields a payoff of $Y(T)$ at time $T$ is the expected value of $Y(T)/V(T)$ [20, p.52]:

$$E_t\left[\exp\left(-\int_t^T r(u)du\right)Y(T)\right], \tag{6.2}$$

where $E_t$ stands for the time $t$-conditional expectation. Even though (6.1) is now a stochastic quantity, it remains the numeraire for risk-neutral pricing. Particularly, the price of a zero coupon bond at time $t$ that pays 1 at time $T$ is defined, given that $V(T) = 1$, as

$$P(t,T) = E_t\left[\exp\left(-\int_t^T r(u)du\right)\right]. \tag{6.3}$$

Equation (6.3) guarantees that we are always able to calculate bond prices $P$ whenever we can define the distribution of $\exp\left(-\int_t^T r(u)du\right)$ in terms of a certain dynamic structure for $r(t)$ [20, p.51].

Short rate models generally owe their popularity to both their high degree of tractability and their flexibility [60, p.177]. For many models one can find explicit solutions for bond prices and for bond option prices. They may be applied to price simple instruments in closed form or sometimes by deterministic numerical methods. However, there are still some extensions of the basic models which require Monte Carlo simulation for the computation of expressions of the form (6.2) [46, p.108].

In particular, the stochastic evolution of short rate models is identified by the following generalized SDE [20, 25, 77, 92, p.52f, p.51f, p.91, p.181]:

$$dr(t) = (\theta(t) - \alpha(t)r(t))\,dt + \sigma(t)r(t)^{\gamma}dW(t). \tag{6.4}$$

This equation denotes a general Gaussian Markov process with $\theta$, $\alpha$ and $\sigma$ all deterministic functions of time. Most of the particular short rate models can be expressed by (6.4). Notice that this process does not follow a GBM anymore. Interest rates, unlike stock prices, appear to be pulled back so some long-run average interest rate level over time. This phenomenon is well-known as *mean reversion*. The process in (6.4) does exhibit mean reversion where the parameter $\alpha$ measures the mean reversion speed towards a long-term interest rate level $\theta/\alpha = b$. Additionally, $\gamma$ is a measure of the degree to which the volatility $\sigma$ of the short rate depends on the current level of $r(t)$. Obviously, a higher $\gamma$ simultaneously means that the volatility reacts sensitively [77, p.91]. Note that in short rate models, in contrast to other models we will look at in Sections 6.3, 7 and 8, the dynamics of all interest rates are totally determined by the dynamics of the overnight rate.

### Vasicek

One of the earliest one factor stochastic equilibrium models of the term structure was developed and proposed by Vasicek [102] in 1977. This classical model describes the short rate through a so called *Ornstein-Uhlenbeck* process. An unspecified short-term interest rate serves as a basis for the model. Furthermore, the Vasicek model corresponds to the choices $\alpha(t) \equiv \alpha$, $\theta(t)/\alpha(t) \equiv b$ and $\sigma(t) \equiv \sigma$ constant while $\gamma = 0$ [25, p.52]. Thus, the risk adjusted dynamics in the Vasicek model of $r(t)$ are given by the following SDE [20, 25, 102, p.53, p.52, p.185]:

$$dr(t) = \alpha(b - r(t))dt + \sigma dW(t). \tag{6.5}$$

In order to be able to simulate this process, we require the solution for the SDE (6.5). The solution for any $0 < u < t$ [25, 46, p.53, p.109]

$$r(t) = r(u)e^{-\alpha(t-u)} + \alpha\int_u^t e^{-\alpha(t-s)}b(s)ds + \sigma\int_u^t e^{-\alpha(t-s)}dW(s) \tag{6.6}$$

can be found through an application of Itô's formula. Consequently, for a given $r(u)$ the value $r(t)$ is normally distributed with mean [46, 102, p.109f, p.186]

$$r(u)e^{-\alpha(t-u)} + \mu(u,t), \quad \text{with} \quad \mu(u,t) \equiv \alpha \int_u^t e^{-\alpha(t-s)} b(s) ds,$$

and variance

$$\sigma^2(u,t) \equiv \sigma^2 \int_u^t e^{-2\alpha(t-s)} ds = \frac{\sigma^2}{2\alpha} \left(1 - e^{-2\alpha(t-u)}\right).$$

In a next step we are interested in how we may eventually simulate a short rate process by Vasicek's model. The simulation of $r$ at the discrete-time steps $0 = t_0 < t_1 < \cdots < t_n$ can be carried out by [46, p.110]

$$r(t_{i+1}) = r(t_i)e^{-\alpha(t_{i+1}-t_i)} + \mu(t_i, t_{i+1}) + \sigma_r(t_i, t_{i+1}) Z_{i+1}, \tag{6.7}$$

where we repeatedly draw $Z_i, \ldots, Z_n$ independent variates from $\phi(0,1)$. Note that Equation (6.7) allows an exact simulation. Exact in the sense that the distribution of the simulated short rates $r(t_1), \ldots, r(t_n)$ is precisely that of the original Vasicek process at times $t_1, \ldots, t_n$ for the identical value of $r(0)$ [46, p.110].

The main advantage of the Vasicek model is its analytical tractability what makes the model relatively easy to understand, implement and eventually apply. However, there are also some shortcomings compared to rival models. Firstly, under the risk-neutral measure the short rate $r$ is normally distributed. Thus there is a positive probability that $r$ becomes negative. Secondly, the Vasicek model does not leave complete flexibility in simulating any specific shapes of the term structure. In particular, possible shapes are upward-sloping, downward-sloping or slightly humped only. Furthermore, since the parameter $\gamma$ is assumed to be equal to zero, the volatility of changes is forced to be both constant and independent of $r$. This assumption is restrictive with respect to the incorporation of different volatility structures [16, 20, 56, 60, p.168f, p.58ff, p.651f, p.181].

### Cox, Ingersoll and Ross

Cox, Ingersoll and Ross [31] proposed a new class of equilibrium processes, generally referred to as CIR models, which include a square-root diffusion term. The risk adjusted dynamics of the short term rate in a CIR model are defined by the SDE [31, p.391]

$$dr(t) = \alpha(b - r(t))dt + \sigma\sqrt{r(t)}dW(t) \tag{6.8}$$

and correspond to the choices $\alpha(t) \equiv \alpha$, $\theta(t)/\alpha(t) \equiv b$ and $\sigma(t) \equiv \sigma$ constant equal to the Vasicek model, whereas now $\gamma = 1/2$ [25, p.53]. In this general equilibrium framework holds if the change in production opportunities is assumed to follow a process of the form (6.8), then the short rate does as well [46, p.120]. The CIR model has been very successful and often applied for many years and thus established some kind of a benchmark. The reason for this lies in its analytical tractability and the fact that the instantaneous short rates remain always positive.

Contrary to the Vasicek model, the square-root diffusion term $\sigma\sqrt{r(t)}$ drops off to zero while $r(t)$ approaches the origin and this prevents $r(t)$ from becoming negative [20, 25, p.64, p.52]. As in the Vasicek model, the term structure in a CIR model may take the forms upward- and downward-sloping as well as slightly humped. The level of the entire term structure, yet not the general shape, at time $t$ is fully determined by the value of $r(t)$ [56, p.653].

Since there are still explicit derivative pricing formulae available, though harder to find than in the Vasicek model, numerical approaches as simulation techniques are not the center stage and therefore the CIR model is not amplified in this paper.

#### Ho-Lee

Ho and Lee [55] did pioneer work by proposing the first no-arbitrage model consistent with the initial term structure. Allowing for time-dependent drift parameters typically makes short rate models consistent with a prevailing set of bond prices. In the continuous-time Ho-Lee model the stochastic process followed by the short rate is [55, 60, p.1020, p.184]

$$dr(t) = \theta(t)dt + \sigma dW(t). \tag{6.9}$$

Equation (6.9) corresponds to an Ornstein-Uhlenbeck process with a time dependent drift parameter. This enables the model to be consistent with current market data. $\theta(t)$ directly defines the average direction that $r$ moves at time $t$. The fact that the SDE (6.9) does not incorporate mean reversion is clearly a shortcoming of this approach [56, p.654f]. Similar to the Vasicek model, for every $t$ there is a positive probability that the interest rates become negative. On the other hand, the main advantage of the Ho-Lee model is that the market price of risk is irrelevant and does not have to be estimated in order to determine the risk-adjusted process required to price interest rate derivatives. The risk-neutral process is automatically defined by fitting the model to the current term structure [54, 56, 77, p.11ff, p.654f, p.93].

Applying Itô's lemma, the SDE (6.9) can be solved explicitly for bond prices and short rates. There is an explicit formula available in the Ho-Lee model and thus the term structures are defined as [46, 60, p.111, p.184]

$$r(t) = r(0) + \int_0^t \theta(s)ds + \sigma dW(t). \tag{6.10}$$

Given this equation, only a small step brings us to the simulation of the Ho-Lee model. Sampling $Z_1, Z_2, \ldots$ from a standard normal distribution and inserting into (6.10) gives us a possible term structure generated by Ho-Lee.

#### Hull-White

The unsatisfactory fitting of the observed term structure of interest rates implied by the Vasicek model brought Hull and White in their paper [57] to propose a new model that basically combines the advantages of the Vasicek and the Ho-Lee model. Therefore, in literature the Hull-White

| **Model** | **Dynamics** | # factors | MR | $r < 0$ | $r \sim$ |
|---|---|---|---|---|---|
| *Equilibrium* | | | | | |
| Vasicek | $dr(t) = \alpha(b - r(t))dt + \sigma dW(t)$ | 1 | ✓ | Yes | $\mathcal{N}$ |
| CIR | $dr(t) = \alpha(b - r(t))dt + \sigma\sqrt{r(t)}dW(t)$ | 1 | ✓ | No | nc$\chi^2$ |
| LS [71] | $dX = (a - hX)dt + c\sqrt{X}dW(t)$<br>$dY = (d - eY)dt + f\sqrt{Y}dW(t)$ | 2 | ✓ | No | $\mathcal{N}$ |
| *No-arbitrage* | | | | | |
| Ho-Lee | $dr(t) = \theta(t)dt + \sigma dW(t)$ | 1 | – | Yes | $\mathcal{N}$ |
| Hull-White | $dr(t) = \alpha\left(b(t) - r(t)\right)dt + \sigma dW(t)$ | 1 | ✓ | Yes | $\mathcal{N}$ |
| BDT [8, 29] | $d\ln r = \left(\theta(t) + \frac{\sigma'(t)}{\sigma(t)}\ln r\right)dt + \sigma(t)dW(t)$ | 1 | (✓) | No | L$\mathcal{N}$ |

Table 5: Term structure instantaneous short rate models. MR=Mean reversion, LS=Longstaff and Schwartz model, BDT=Black, Derman and Toy model (see also Section 9.3). $r \sim$ indicates the distribution of the short rates, while $\mathcal{N}$, L$\mathcal{N}$ and nc$\chi^2$ stand for normal, lognormal and non-central $\chi^2$.

model is also referred to as *Extended Vasicek* model. The dynamics of the short rates in the Hull-White model are given by the SDE [20, 57, p.72, p.577]

$$dr(t) = \alpha\left(b(t) - r(t)\right)dt + \sigma dW(t). \tag{6.11}$$

Firstly, the Hull-White model makes the parameter $b$ time-dependent. Thereby enough degrees of freedom are available to fit the model to the initial term structure and the term structure of spot or forward rate volatilities. Thus the Hull-White model falls into the category of no-arbitrage models and is therefore consistent with the current market data [16, 57, p.173, p.577f]. Secondly, it can be characterized as the Ho-Lee model with mean reversion at rate $\alpha$. The model is to the same extent analytically tractable as the Ho-Lee model. The mean reversion level, i.e. the function $b(t)$, can be directly calculated from the initial term structure [56, p.656].

For an extensive discussion of this model the interested reader is referred either to the initial paper of Hull and White [57] or to the Chapter 3.3 of the book written by Brigo and Mercurio [20, p.71-80].

Table 5, according to Brigo and Mercurio [20, p.57], gives a review of some established instantaneous short rate models. Even though this list is not exhaustive, it includes most fundamental approaches in order to model a term structure of short rates. Note that some of these models, Hull-White, for instance, have been further developed and extended to multifactor models.

### 6.3 Forward rate models

*Forward rate models* describe the arbitrage-free dynamics of the term structure of interest rates through the evolution of *forward rates* $f(t,T)$. The distinguishing feature of these models is that they explicitly describe the evolution of the *full* term structure, contrary to the previously discussed short rate models, which do only provide a description of the dynamics of the short rate $r(t)$ [46, p.149]. Instantaneous short rate models, as we have seen before, are mostly easy to implement and are usually able to price many standard and also nonstandard interest rate derivatives consistently [56, p.679].

However, to legitimize the application of forward rate models, there are the two major limitations of short rate models that are generally overcome by using different types of forward rate models:

- In short rate models the current value of all term structure quantities is solely determined by the current value of the short rate. Thus, the term structure is entirely summarized by today's short rate. In multifactor models the complexity of the yield curve dynamics is subsumed into the current values of a finite, but usually small number of underlying factors [46, 56, p.149, p.679f]. The new generation of correlation-dependent instruments, however, requires models that describe the state of the world by the full term structure and not necessarily by a finite number of factors, which are often not able to deal with this new dimension of complexity [92, p.311].

- Short rate models do not provide us with a complete freedom in the choice of the corresponding volatility structure. In a forward rate framework the forward rate dynamics are completely specified through their instantaneous volatility structure. In contrast, in short rate models the volatility of the short rate alone is not sufficient to fully characterize the relevant interest rate model [20, 56, 92, p.183, p.679, p.311f].

Forward rate models can basically be divided into two classes, namely models based on *continuous* and *simple* rates, respectively. The by far most popular continuous forward rate model is the *Heath, Jarrow* and *Morton* framework. It was one of the first forward rate models proposed and simultaneously establishes a basis for models based on simple rates. These models are therefore closely related to the Heath, Jarrow and Morton approach and called *LIBOR market models*. The seemingly minor shift from continuous to simple rates has surprisingly far-reaching practical and theoretical implications [46, p.165f], as we will see later on. As in the application of both models, Heath–Jarrow–Morton as well as LIBOR market model, Monte Carlo techniques play a decisive role, we will give these two approaches a fundamental treatment and will implement them in Section 7 and 8, respectively.

## 7 Heath, Jarrow and Morton

In 1992 Heath, Jarrow and Morton (hereafter, HJM) published the important paper *Bond pricing and the term structure of interest rates* [53] describing an alternative framework for modeling the term structure of interest rates. Under the risk-neutral measure, they derived a generalized formula for the drifts of instantaneous forward rates in terms of the volatilities of the forward rates. One of the key insights of this framework is that the HJM model is completely defined by specifying the *volatilities* of forward rates. Consequently, by using the observed term structure of forward rates as an input one can easily match a HJM model with the current market discount bond prices [6, p.310]

Basically, there are two common formulations of the model. Firstly, the *price based* and secondly the *forward based* approach. The former technique is based on the dynamics of discount bonds, i.e. it takes them as the fundamental building block. The latter directly obtains the no-arbitrage SDE which is obeyed by the forward rates [92, p.314]. For a rigorous proof of the equivalence of these two approaches it is referred to the paper of Carverhill [27]. Note, though, that, historically, the pioneering results of the HJM were first achieved in the forward rates context.

The virtue of the HJM theory lies in the fact that within such a framework virtually any (exogenous term structure) interest rate model can be derived. Even market models (see Section 8), for instance, have evolved starting from the instantaneous forward rate dynamics of the HJM approach. However, the disadvantage is that there are only a few volatility structures which lead to a corresponding short rate process which is indeed Markovian. This means that models of this form will, except for some wise choices of volatility specifications, which we will look at in Section 7.2, generally be path-dependent and thus non-Markovian processes [6, 20, p.310, p.184]. A short rate process can be characterized as being Markovian if its future evolution is in no way affected, in a stochastic sense, by its past realizations. If a process is indeed not influenced by its past, then the short rate process is, to use Carverhill's [26] apt oxymoron, *randomly determined* [92, p.347f].

In general, prices of fixed-income instruments, e.g. bonds, do not depend on the values of a few individual factors but rather on the entire history of the forward rate process. This fact makes computation exceptionally difficult and often only possible by simulation since the nodes of non-recombining, or bushy trees grow exponentially and therefore the approximating lattice will literally explode already after a small number of steps [6, p.310]. As a result, Monte Carlo simulation is the computational tool of choice in the HJM setting.

### 7.1 Framework

As mentioned before, the HJM characterizes the dynamics of the entire forward rate curve $f(t,T)$, whereas $0 \leq t \leq T \leq \tau$. The origin of any consistent implementation of the HJM approach is the current yield curve, which is obviously based on market data. This yield curve can be described either by the collection of discount bonds $P(0,T)$ or by the instantaneous

forward rates $f(t,T)$. The forward rate at time $t$ for date $T > t$ is defined by [46, 53, 92, p.150, p.79, p.314]

$$f(t,T) = -\frac{\partial \ln P(t,T)}{\partial T}. \tag{7.1}$$

By solving the differential equation of Expression (7.1) the relationship between forward rates, spot rates and bond prices become obvious. In particular, spot rates, $r(t,T)$, and bond prices, $P(t,T)$, are written in terms of forward rates, $f(t,T)$, as

$$r(t,T) = \frac{1}{T-t} \int_t^T f(t,s)ds \tag{7.2}$$

and

$$P(t,T) = \exp\left(-\int_t^T f(t,s)ds\right), \tag{7.3}$$

respectively, for all $T \in [0,\tau]$, $t \in [0,T]$ [6, 53, 92, p.310, p.80, p.314]. Note that the spot rate at time $t$, $r(t)$, is exactly the same as the instantaneous forward rate at time $t$ for date $t$, i.e., $r(t) = f(t,t)$ [53, p.80].

In the HJM setting, the dynamics of the instantaneous forward rate curve are specified, under a given measure, through the SDE [46, 53, 60, p.150, p.80f, p.200]

$$df(t,T) = \alpha(t,T,\omega)dt + \sigma(t,T,\omega)dW(t), \tag{7.4}$$

where $\omega$ is a vector containing the past and present values of interest rates as well as bond prices at time $t$ in the sample space $\Omega$. The process $W$ is a standard $d$-dimensional Brownian motion, whereas $d$ is the number of factors or volatility curves which describe the forward rates. Thus, the HJM approach can be constructed as a one factor as well as a multifactor model.

Due to the absence of arbitrage, requested by the no-arbitrage condition, asset prices must be martingales when divided by the numeraire, which in this case is $\exp\left(\int_0^t r(s)ds\right)$. However, forward rates and asset prices are by no means the same. Therefore, the restrictions imposed on the dynamics in (7.4) with the purpose to avoid any arbitrage opportunities are different. Nevertheless, asset prices, particularly bonds, serve as a starting point in order to find these restrictions. To ensure that the discounted bond prices $P(t,T)\exp\left(-\int_0^t r(s)ds\right)$ are positive martingales, Heath et al. [53, p.81f] derived the dynamics of the form

$$\frac{dP(t,T)}{P(t,T)} = r(t)dt + v(t,T,\omega)dW(t), \tag{7.5}$$

where $0 \le t \le T \le \tau$ [46, 60, p.151, p.200]. The volatilities of the bonds $v(t,T,\omega)$ may either be functions of observed bond prices or, equivalently, of current forward rates since (7.1) indicates that they are one-to-one related to each other. Applying Itô's formula yet again, Heath et al. showed [53, p.80ff] that forward rate volatilities may be derived from bond price volatilities and we therefore must have

$$\sigma(t,T,\omega) = -\frac{\partial}{\partial T} v(t,T,\omega)$$

and

$$v(t,T,\omega) = -\int_t^T \sigma(t,s)ds + \text{constant}, \tag{7.6}$$

simultaneously [46, p.152]. Notice, however, that $P(t,T)$ approaches 1 as $t \to T$ and therefore we must have $v(T,T,\omega) = 0$ because bond's price volatility declines to zero at maturity [92, p.318]. Thus the constant in Equation (7.6) is zero which allows us to rewrite the expression for $\alpha$ as [20, 46, 60, p.186, p.152f, p.200f]

$$\alpha(t,T,\omega) = \sigma(t,T,\omega)\int_t^T \sigma(t,s)ds = \sum_{i=1}^{N} \sigma_i(t,T,\omega)\int_t^T \sigma_i(t,s)ds. \tag{7.7}$$

This expression represents the risk-neutral drift imposed by the no-arbitage condition. Additionally, (7.7) reveals a key result of the HJM research activities. The assumption of arbitrage-free dynamics has led to a specific relationship between the drift and the volatility. Namely that the drift of the dynamics in (7.4) is completely determined by the volatility structure of the instantaneous forward rates. Thus we know that the HJM drifts themselves are simply a function of the volatilities, either of forward rates, or of discount bonds. If we substitute (7.7) into (7.4) then the new HJM dynamics are [46, 53, 92, p.152, p.89f, p.318ff]

$$df(t,T) = \left(\sigma(t,T,\omega)\int_t^T \sigma(t,s)ds\right)dt + \sigma(t,T,\omega)dW(t). \tag{7.8}$$

This SDE defines the no-arbitrage dynamics of the forward rate curve under the risk-neutral measure and is simultaneously the central insight of the HJM approach. Recall why the model eventually fulfills the required no-arbitrage condition. While deriving the forward rate restriction (7.7) and the corresponding dynamics (7.8), a proper specification of the drift was chosen in order to guarantee the absence of arbitrage, or more precisely, to make the discounted bond prices martingales. However, integrating (7.8) leads to the dynamics of $f(t,T)$

$$f(t,T) = f(0,T) + \int_0^t \sigma(u,T,\omega)\int_u^T \sigma(u,s)ds\,du + \int_0^t \sigma(s,T,\omega)dW(s), \tag{7.9}$$

which are completely determined once the vector volatility function $\sigma$ is specified [20, 53, p.186, p.89f]. Only a small step takes us now from the dynamics of the forward rates to the dynamics of the zero coupon bond price $P(t,T)$:

$$dP(t,T) = P(t,T)\left[r(t)dt - \left(\int_t^T \sigma(t,s)ds\right)dW(t)\right], \tag{7.10}$$

where $r(t)$ denotes the instantaneous short term interest rate at time $t$. The dynamics of $r(t)$ are given by [20, 53, p.186, p.90]

$$r(t) = f(t,t) = f(0,t) + \int_0^t \sigma(u,t,\omega)\int_u^t \sigma(u,s)ds\,du + \int_0^t \sigma(s,t,\omega)dW(s). \tag{7.11}$$

The decisive discrepancy between HJM and previously discussed instantaneous short rate models has become apparent. We know that the HJM drift is determined once the volatility is finally specified. In contrast, the derivation of the short rate models disclosed that their dynamics and thus their drift parameters could be completely specified independent of any diffusion coefficients. Even though no volatility structures were taken into account, there was still no arbitrage introduced. A wise choice of the drift parameters is indeed crucial for calibrating short rate models to the observed bond prices. Provided that the initial forward rate curve is chosen in a way that the consistency, implied by (7.1), is guaranteed, the HJM model, however, is automatically calibrated to prevailing bond prices. To conclude, while calibrating a HJM model the initial forward curve conditions rather than individual parameters are essential. With regard to the calibrating process of a HJM framework itself, the main effort lies in choosing $\sigma$ properly to firstly match bond prices and eventually the market prices of interest rate derivatives [46, 60, 92, p.153, p.200, p.312f].

## 7.2 Volatility functions

The proceeding section has already foreshadowed that the consequences of the choice of volatility specification for the implementation and simulation of a HJM model are enormous. The short rate process (7.11) is, as mentioned before, not a Markov process in general. However, there are suitable specifications of the volatility function $\sigma(t, T, \omega)$ for which $r(t)$ is yet Markovian [20, 60, p.186f, p.203f]. For an extensive treatment of Markovian short rate processes it is referred to either the paper of Carverhill [26] or the Chapter 16 of Rebonato [92, p.347-359]. In the general case when $\sigma(t, T, \omega)$ is not Markovian, we may encounter major computational problems when discretizing the Dynamics (7.11) for the pricing of even simple derivatives. However, several standard Markovian functional forms for $\sigma(t, T, \omega)$ have been explored. Two of them, both single factor models ($d = 1$), are listed and shortly explained below [60, p.203]:

- *Constant* $\sigma$: $\sigma(t, T, \omega) \equiv \sigma$.
  In this one factor model with constant volatility each increment $dW(t)$ moves the entire forward curve, i.e. all existing points on the curve, by an equal amount of $\sigma dW(t)$. Thus, the forward curve is constrained to move only in parallel [46, p.153]. For this particularly simple specification, bond prices are assumed to be lognormal and all forward rates are normal and exhibit exactly the same volatility. If these distributional assumptions are fulfilled, the according models are qualified as Gaussian [92, p.317f]. They are analytically well-tractable but also unrealistic to some extent, what we will see later on. If we insert $\sigma$ into Equation (7.7) the resulting drift of a HJM model with constant $\sigma$ has the form [46, p.153]
  $$\alpha(t, T, \omega) = \sigma \int_t^T \sigma \, ds = \sigma^2 (T - t).$$
  Solving (7.8) with constant $\sigma$ leads to the dynamics of the forward rates
  $$f(t, T) = f(0, T) + \frac{1}{2}\sigma^2 [T^2 - (T - t)^2] + \sigma W(t),$$

which is simultaneously the continuous-time equivalent of the Ho-Lee model [60, 92, p.202, p.318].

Flesaker [40] conducted an empirical test about the constant volatility version of the HJM model for Eurodollar futures and futures options. He found that this general version of the HJM model is unable to explain cross-sectional pricing pattern of futures options and that it exhibits systematic biases, why the model is eventually soundly rejected. Moreover, he observed that the model tends to overvalue short-term options relative to long-term options. Accordingly, he clearly motivated further empirical studies of HJM models where the volatility is allowed to vary over time rather than artificially kept constant.

- *Exponential* $\sigma$: $\sigma(t,T,\omega) = \sigma \exp(-\lambda(T-t))$, for some constants $\sigma$, $\lambda > 0$.
  This diffusion term $\sigma(t,T,\omega)$ has a greater impact on forward rates for short maturities than on forward rates for long maturities [46, p.154]. Due to this fact the HJM model including an exponential $\sigma$ is eventually completely equal to the general short rate dynamics proposed by Hull and White [57], see (6.11). This makes clear that one can establish a one-to-one equivalence between the HJM one factor model and the general formulation of the Gaussian one factor instantaneous short rate model of Hull and White [20, p.187].

  This volatility specification keeps its analytical tractability but is able to perform slightly better than the HJM model with constant $\sigma$. Yet, the results are still unsatisfactory and therefore the HJM model including exponential volatility is also rejected [60, p.203].

Asides from many standard approaches with the goal of finding suitable specifications of $\sigma(t,T,\omega)$, which make the short rate process indeed Markovian, there have been developed some more sophisticated techniques, too. They basically exploit that, even though a short rate process might not be Markovian, there may yet exist a higher-dimensional Markov process that possesses the instantaneous short rate as one of its components [20, p.188]. Ritchken and Sankarasubramanian [95], for instance, proposed a model for simply capturing the path dependence of the short rate $r(t)$ through a single statistic. In particular, they found a few necessary and sufficient conditions which are imposed on the volatility structure of the forward rates. These conditions enable them to control the short rate's path dependence and make them analytically tractable [20, 60, p.189, p.205].

A second model, which was proposed by Mercurio and Moraleda [78], derives an interest rate model within the HJM framework which explicitly assumes a humped volatility structure in the instantaneous forward rate dynamics and thus is also analytically tractable. They motivate their assumption of a humped volatility structure by the fact that forward rates themselves commonly exhibit humped volatility functions that are generally implied by simple market quotes. They have shown that their model consequently outperforms similar existing models, which do not explicitly contain such a humped shape in volatilities [20, 78, p.191f, p.213].

### 7.3 Simulation

In regard to simulation, the HJM model enters unknown territory since an exact simulation of (7.8) is, except under very special choices of $\sigma(t, T, \omega)$, generally infeasible. Therefore, in the HJM model we depart from the previously discussed instantaneous short rate models. Simulating the HJM forward rate dynamics requires introducing a discrete approximation and thus some discretization error, which was treated in Section 5.2, is usually inevitable [46, p.150].

To be able to simulate a forward rate like $f(t, T)$, for $T > t$, each of the two arguments of $f$, time $t$ and maturity $T$, requires to be put into a discrete grid. However, it would both take too much effort and be computationally burdensome to simulate an entire Euler scheme, for instance, especially for small time steps $\Delta t$, which are denoted by $b$, where $b \equiv t_{q+1} - t_q$ again. Of course one could simply choose large values of $b$ to reduce computational effort instead. However, this brute force approach will apparently deteriorate the approximation and is therefore rather not the technique to be applied. A possible remedy might be to use an arbitrage-free HJM model in discrete- instead of continuous-time [46, p.155f]. Since this approach has gained wide acceptance in practice, it is briefly explored in this section according to Glasserman [46, p.149ff], who provides a well-structured and profound analysis of this topic. Therefore the interested reader is referred to the Chapter 3.6 of Glasserman [46, p.149-165] for a detailed derivation.

In order to keep discretization errors on a preferably small level, we try to achieve that the prices of exact bonds $P(0, T)$ and of discretized bonds $\hat{P}(0, T)$ finally match. Taking (7.1) into account, it follows that this is indeed the case if

$$\hat{f}(0, t_q) = \frac{1}{b_q} \int_{t_q}^{t_{q+1}} f(0, s) ds, \tag{7.12}$$

for all $q = 0, 1, \ldots, N-1$. Note that hat notation is used to denote discretized variables. Equation (7.12) reveals that each discretized forward rate is the average of the underlying forward curve over the discretization interval $b$. To keep notation as simple as possible, we will assume that $d = 1$. However, taking $d \geq 2$ does not change anything except that exposition increases in complexity. A generic simulation of the forward rates in a discrete-time HJM setting evolves as follows:

$$\hat{f}(t_i, t_j) = \hat{f}(t_{i-1}, t_j) + \hat{\alpha}(t_{i-1}, t_j, \omega) b_{i-1} + \hat{\sigma}(t_{i-1}, t_j, \omega) \sqrt{b_{i-1}} Z_i, \quad j = i, \ldots, N, \tag{7.13}$$

where $Z_1, \ldots, Z_N$ are iid random variables drawn from $\phi(0, 1)$ and $\hat{\alpha}$ and $\hat{\sigma}$ denote the discrete drift and diffusion according to their continuous-time counterparts in (7.8). Recall that again, the entire forward rate dynamics are driven by the same source of randomness, i.e. $Z_i$.

The ultimate goal is to achieve consistency between the simulated model prices and the prevailing market prices of traded derivatives. Therefore, a specific calibration procedure is commonly applied in order to specify the discrete diffusion $\hat{\sigma}$ to fulfill these requirements. The conditions for the discrete drift coefficient $\hat{\alpha}$, however, are different. Remember that previously the drift was specified to meet the requested no-arbitrage condition, i.e. discounted bond prices must be

martingales. Now we want to approximate the continuous-time drift by a properly chosen discrete drift $\hat{\alpha}$. One possible approach pursues the strategy that always maintains the martingale property of the discretized discounted bond prices. If we choose this way we get as a result

$$\hat{\alpha}(t_{i-1}, t_j, \omega) b_j = \frac{1}{2} \left( \sum_{q=i}^{j} \hat{\sigma}(t_{i-1}, t_q) b_q \right)^2 - \frac{1}{2} \left( \sum_{q=i}^{j-1} \hat{\sigma}(t_{i-1}, t_q) b_q \right)^2 . \tag{7.14}$$

This expression denotes the discrete-time version of the HJM drift. Some simple calculations eventually show [46, p.158] that the discrete drift in (7.14) is indeed consistent with the continuous-time limit in (7.7).

### Pricing a cap by HJM

With Equations (7.13) and (7.14) there is now a basis for pricing derivative securities in discrete-time HJM models using Monte Carlo simulation [46, 52, p.163f, p.9]. Basically, the price of a derivative in a risk less setting at time $t = 0$, $V_0$, with a payoff $V_T$ at time $T = t_N$ is defined as

$$V_0 = E \left[ \exp \left( - \sum_{q=0}^{N-1} \hat{f}(t_q, t_q) b_q \right) V_T \right].$$

This is similar to what has already been developed in Section 1. However, let us become more concrete and have a closer look at how to price a standard caplet[3] with maturity $T$ and cap rate $R_{cap}$ by the HJM approach. The underlying of a cap is a particular spot rate, in this case given by the spot rate $R_k$. Note that the underlying spot rate differs from the instantaneous short rate since the former applies over a certain interval, suppose $[t_i, t_{i+1}]$, and is based on discrete compounding.

The payoff of the caplet normally occurs at $T + l$ and is thus given by $(R_k - R_{cap})^+$. For the sake of comparison, in a continuous-time HJM model $R_k$ is given by

$$R_k = \frac{\int_T^{T+l} f(T, s) ds}{l}. \tag{7.15}$$

Now we change to our discrete-time HJM framework. Suppose that in a discrete-time grid $T = t_i$ and $T + l = t_{i+1}$. Hence, the discrete-time payoff occurring at time $t_{i+1}$ is $(R_k(t_i) - R_{cap})^+$ and $R_k$ changes to

$$R_k(t_i) = \frac{\exp\left(\hat{f}(t_i, t_i) b_i\right) - 1}{b_i}$$

whereas $b$ is the time step $\Delta t$, $b_i = t_{i+1} - t_i$. More generally, similar ideas apply if the caplet covers more than a single simulation interval. Suppose that the certain caplet covers an interval

[3] A caplet is an interest rate derivative that protects against an increase in an interest rate for a single period and therefore provides a payoff when a specified interest rate is above a certain level, the so called cap rate. A cap, however, is a portfolio of caplets covering multiple periods and its value is simply the sum of the values of its component caplets [56, p.619].

$[t_i, t_{i+n}]$ then the relevant spot rate is redefined and given by

$$R_k(t_i) = \frac{1}{t_{i+n} - t_i} \left( \exp \left( \sum_{q=0}^{n-1} \hat{f}(t_i, t_{i+q})[t_{i+q+1} - t_{i+q}] \right) - 1 \right).$$

In summary, the procedure for assessing the price of an ordinary caplet in the HJM framework by Monte Carlo simulation goes as follows. In a first step, simulate the forward rates generically in a discrete-time HJM setting by the forward rate dynamics (7.13). As usual, draw as many iid random variables from $\phi(0, 1)$ as you need and run the simulation by inserting these increments. Secondly, compute an individual payoff for each forward rate path simulated in the first step. Thirdly, discount all calculated payoffs to the time you need them. In a fourth and final step, average all the discounted payoffs. This proceeding provides you with unbiased and consistent interest rate derivative prices according to the HJM model.

## 7.4 Conclusion

Given the potential complexity of the HJM models, as discussed in the previous sections, what is the comparative advantage, in context of simulation and from a practical point of view, brought about by the HJM approach? Firstly, a general feature that contrasts with short rate models is the required number of time steps to simulate. Suppose we are interested in valuing a 3-year option on a 10-year swap. If we priced this instrument in a simple short rate model we would have to simulate yield curve paths over a 13-year horizon. In a HJM framework, however, we merely need to simulate these paths over a period of 3 years, provided that the initial forward rate curve extends for at least 13 years. When the option eventually expires, all the required information to be able to value the underlying swap consistently is embodied in the remaining forward rates. Even though the HJM setting is eventually more complex and requires the updating of more variables at each time step than short rate models, we need to simulate by far not as many time steps and therefore, HJM models may outperform similar models based on short rates [46, p.165].

Secondly, there are derivative securities that indeed require that the imperfect correlation among different rates should be accounted for in order to price them primarily correctly but also consistently. With respect to such instruments the HJM model has a very strong appeal. Single as well as multifactor HJM settings value exotic interest rate derivatives, which highly depend in an important way on the correlation structure among different quantities, correctly [92, p.332].

In summary, it is undeniable that the HJM approach enjoys a great importance with regard to the pricing of interest rate derivatives by Monte Carlo simulation.

## 8 LIBOR market model

The LIBOR market model (LMM) was firstly introduced by Miltersen, Sandmann and Sondermann [81] in 1997 and then further developed to different forms applicable in practice by Brace, Gatarek and Musiela [13] and Jamshidian [61]. Therefore, the LMM is sometimes also called Brace-Gatarek-Musiela (BGM) model.

There is a close relationship between the LMM and the previously discussed HJM framework. They both rest upon the evolution of the forward rates in order to describe the no-arbitrage dynamics of the term structure of interest rates. The most important difference between these two models is that the LMM is based on *simple* rather than on continuous compounded forward rates. At a first and probably superficial glance, this minor shift in focus seems to have essential consequences. However, in spite of this difference, the LMM and HJM approaches have one decisive feature in common. They share the goal to express the arbitrage-free dynamics of the forward rates, for instance, purely as a function of both the volatilities of and the correlations among themselves [94, p.3].

Over the past years, the LMM has become increasingly popular in the entire financial industry and thus has drawn a growing stream of research into further developments of this sophisticated technique. It is now one of the most popular models for pricing interest rate derivatives. This ascending popularity is, among other things, a result of consistency with practice. The valuation of vanilla products in ordinary LMM can be eased to applying simple standard market formulae [44, 48, 94, p.xiii, p.13, p.3f], i.e. Black's model [7]. This was also the aim of the Brace, Gatarek and Musiela, namely to show that the existing market practice can be made consistent with an arbitrage-free term structure model [13, p.127]. An essential feature of a successful pricing model is that standard pricing formulae for certain standard derivative instruments, such as caps, floors and swaptions, can be more or less easily reproduced [48, p.13f]. Market models were indeed the pioneering interest rate dynamics that are completely compatible with either Black's formula for caps or swaptions. They firstly derived the Black's formula based on rigorous interest rate dynamics [20, p.195]. Provided that both yield and volatility are flat, swaptions, for instance, can be correctly valued in a LMM with closed formulae that eventually reduce to exactly the Black swaption model [13, p.127].

The description of the evolution of the forward rates in the LMM is given by a lognormal SDE. This lognormality, however, results in completely state-dependent drift terms for individual forward rates and thus makes the application of recombining tree lattices simply impossible. Therefore, in the LMM framework derivatives are evaluated, due to its high dimensionality, by Monte Carlo methods, i.e. simulating the relevant forward LIBOR rates, since Monte Carlo simulations are not affected by the high dimensionality of the problem [44, 59, 90, p.xiii-xvii, p.159f, p.97f].

## 8.1 Model dynamics

The basic consideration in Section 7 was that the HJM framework models the dynamics of the instantaneous, continuously compounded forward rates $f(t,T)$. But these forward rates are not directly observable in the market and therefore must be estimated and extracted from actively traded discount bonds, respectively. Even the instantaneous short rates $r(t)$, treated in the models of Section 6.2, are somehow a mathematical idealization because they are not one-to-one observable either [46, 48, 94, p.166, p.16, p.57f]. In contrast, the LMM is expressed in terms of *discrete* forward rates, which are indeed observable in the market. On this account the term *market model* is used to describe these approaches in general. The probably most popular benchmark interest rate is the London InterBank Offered Rates, or in short LIBOR. The LIBOR is a daily computed reference rate and always quoted for numerous different maturities, such as one, three or six months, and also for many different currencies [46, 48, p.166, p.16].

The spot LIBOR rate $L_{t,\tau}(t)$ set at time $t$ for an accrual period of length $\tau - t = \delta$ is based on simple rates and defined as [16, p.150]

$$L_{t,\tau}(t) = \frac{1}{\delta}\left(\frac{1}{P(t,\tau)} - 1\right).$$

A *forward* LIBOR rate works similarly. Fix $\delta = \tau - s$ and consider a maturity $\tau$. Thus, the forward LIBOR rate for the period from $s$ to $\tau$ set at time $t$, where $t \leq s \leq \tau$, is given by [13, 16, p.128f, p.150]

$$L_{s,\tau}(t) = \frac{1}{\delta}\left(\frac{P(t,s) - P(t,\tau)}{P(t,\tau)}\right). \tag{8.1}$$

Suppose that the discount bond $P(t,\tau)$ is taken as a numeraire. Under this measure the price of the trading strategy $P(t,s) - P(t,\tau)$ must be a martingale. But if we divide this trading strategy by the numeraire

$$\frac{P(t,s) - P(t,\tau)}{P(t,\tau)}, \tag{8.2}$$

the resulting price must be $\delta L_{s,\tau}(t)$ again. Now it directly follows from Equation (8.1) that since Expression (8.2) is a martingale, $\delta L_{s,\tau}(t)$ must also be a martingale. Thus, under the risk adjusted probability measure $Q^\tau$ for the maturity $\tau$, the far-reaching conclusion can be drawn that the process $L_{s,\tau}(t)$ indeed possesses the martingale property because $\delta$ is constant [48, p.26f]. It follows that if the forward LIBOR rate, $L_{s,\tau}(t)$, is defined according to a particular diffusion process, it needs to be driftless under the probability measure $Q^\tau$ [20, p.208]. To specify the model, we assume the following driftless dynamics of $L_{s,\tau}(t)$ under $Q^\tau$:

$$dL_{s,\tau}(t) = L_{s,\tau}(t)\,\gamma_{s,\tau}(t)\,dW(t), \tag{8.3}$$

where $\gamma_{s,\tau}(t)$ denotes the instantaneous volatility of $L_{s,\tau}(t)$, which is a deterministic function of time at time $t$. In general, under $Q^\tau$, $L_{s,\tau}(t)$ is a martingale and always lognormal if $\gamma_{s,\tau}(t)$ is deterministic. This leads to the fact, what will be shown in the next Section 8.2 in detail, that the pricing of interest rate caps is consistent with Black's formula. According to this the LMM

with a deterministic $\gamma_{s,\tau}(t)$ is also referred to as the lognormal forward-LIBOR market model [13, 16, 48, 60, p.129, p.150, p.27, p.214].

Equation (8.1) expresses the forward LIBOR rate in continuous-time from $s$ to $\tau$. But the simulation of a LMM is actually based on a discrete-time grid. Therefore, it is already done some spadework, supporting the following insights by allowing for a finite set of maturities, also referred to as *tenor dates*, $0 = T_0, T_1, \ldots, T_M, T_{M+1}$. These tenor dates are all fixed in advance. Again, $\delta_i = T_{i+1} - T_i$, for $i = 0, \ldots, M$, indicates the time intervals between the individual tenor dates. Furthermore, a finite set of bond prices is introduced. Assume that $P_n(t)$ denotes a bond price at time $t$ maturing at $T_n$, where $0 \leq t \leq T_n$. Thus far, bond prices have been written as $P(t, T_n)$. For the sake of a throughout consistent notation, they are now denoted by $P_n(t)$, while $P_n(t) \equiv P(t, T_n)$ and for $n = 1, 2, \ldots, M+1$. Comparably, the forward LIBOR rate at time $t$ for the accrual period $[T_n, T_{n+1}]$ is based on an equal notation and therfore given by $L_n(t)$ [16, 46, p.151f, p.167].

#### Dynamics of forward LIBOR rates under the forward measure

The dynamics of the forward LIBOR rate $L_{t_{i-1},t_i}(t)$ in Equation (8.3) were indicated under the probability measure $Q^{t_i}$. Accordingly, each forward rate was specified under an individual probability measure and thus it was a martingale only under this particular measure [16, p.151]. This approach might work well while pricing standard interest rate derivatives, such as caps and floors. To price more sophisticated derivatives, however, we need to describe the dynamics under one and the same probability measure. The choice of a particular measure is restricted to the *spot* and the *forward measure*. The derivation of the forward rate dynamics under one of these alternative measures is based on the *change-of-numeraire* technique [48, 94, p.27f, p.57ff]. In this paper only the derivation's results of the dynamics of the forward LIBOR rates under the forward measure will be considered. For a general and rigorous treatment of the entire derivation process of the arbitrage-free dynamics of forward LIBOR rates it is referred either to Jamshidian [61] or to Chapter 3 of Rebonato [94, p.57-84].

However, using Itô's formula and applying Girsanov's theorem[4] one can find that the no-arbitrage dynamics of forward LIBOR rates, $L_n(t)$, under the forward measure are given by [16, 46, 48, p152f, p.172, p.31f]

$$dL_n(t) = \sum_{j=g(t)}^{n} \frac{\delta_j L_j(t) L_n(t) \gamma_n(t) \gamma_j(t)}{1 + \delta_j L_j(t)} \, dt + L_n(t) \gamma_n(t) \, dW(t), \tag{8.4}$$

where $g(t)$ represents the index for the next reset date at time $t$, which means that $g(t)$ is the smallest integer such that $t \leq t_{g(t)}$ holds [56, p.683]. Note that (8.4) has a *linear* characteristic because the numerator in the drift is quadratic but is stabilized by the term $1 + \delta_j L_j(t)$ in the

[4] Girsanov's theorem is used to transform an underlying probability measure $P$ to an equivalent martingale measure $Q$. Discounted prices of underlying assets, for instance, exhibit martingale properties under the martingale measure. Therefore, this theorem is of high relevance especially in financial mathematics. For a proof of Girsanov's theorem the reader should consult Lamberton and Lapeyre [66, p.66].

denominator. If we approached the limit and decreased the compounding period $\delta_j$ to the limit, this attractive feature would be lost. This is the reason why we have to distinguish between continuous and simple forward rates in the LMM [46, p.172].

## 8.2 Valuing ordinary interest rate derivatives

In summary, we have encountered two important characteristics of the LMM so far. Firstly, they are based on directly observable and thus simple market rates. Secondly, the model allows for deterministic volatilities $\gamma_j$. However, if we go a step further and explore the pricing of interest rate caps or swaptions in detail, a third essential feature appears [46, p.172].

At the beginning we take a standard interest rate cap that provides payoffs at the times $T_{n-v}, \ldots, T_{n+p}$ into account. Consider, now, a single caplet for the period $[T_n, T_{n+1}]$ with the underlying $L_n(T_n)$ and strike rate $K$, both fixed at $T_n$. Thus, the caplet's payoff is $\delta_n \left(L_n(T_n) - K\right)^+$ and paid at $T_{n+1}$. Under the forward measure $Q^{T_{n+1}}$, which takes a discount bond maturing at $T_{n+1}$ as numeraire, the price of the caplet $C_n(t)$ at time $t$ must be [16, 46, 60, p.154, p.172f, p.209]

$$C_n(t) = P_{n+1}(t) E^{Q^{T_{n+1}}} \left[\delta_n \left(L_n(T_n) - K\right)^+\right],$$

with a martingale property that applies to $C_n(t)/P_{n+1}(t)$. If we take the volatility of the LIBOR rate $\gamma_n(t)$ to be a deterministic function of $t, T_n$ and $T_{n+1}$, then the forward LIBOR rate $L_n(T_n)$ is both lognormally distributed and a martingale under the measure $Q^{T_{n+1}}$. The price of this caplet is therefore defined as [16, 20, 60, p.154, p.200-202, p.209]

$$C_n(t) = \delta_n P_{n+1}(t) \left(L_n \Phi(d) - K\Phi(d - \sigma)\right) \tag{8.5}$$

with

$$d = \frac{1}{\sigma}\left(\ln \frac{L_n}{K} + \frac{1}{2}\sigma^2\right),$$

the variance

$$\sigma^2 = \mathrm{Var}^{Q^{T_{n+1}}} \left[\ln L_n(T_n)\right] = \int_t^{T_n} \gamma_n(u)^2 du$$

and $\Phi$ the cumulative normal distribution. This reveals the third pathbreaking characteristic of the LMM. Brace et al. succeeded in proofing that if volatilities are taken to be deterministic, standard interest rate caplets can be priced in closed form by the Black formula [13, 16, 20, 60, p.141, p.154, p.202, p.209]. You can check that relationship by comparing a caplet's price computed in a LMM by simulation, thus based on (8.4) and (8.6), with a price of the same caplet calculated by Black's formula. As a result you will see that these two prices do not differ significantly.

Additionally, Formula (8.5) may also be used in reverse direction. If we start there where it actually ended before, we take caplet prices which are given now by the market. You can assume $\sigma$ to be the unknown parameter and therefore solve (8.5) for the so called *implied volatility*. The implied volatility makes model prices to exactly match the market prices. This approach will prove to be useful in calibrating a model to current market data [46, p.173].

## 8.3 Simulation

Due to the complexity of the drift terms of the forward LIBOR rate, the high dimensionality of market models as well as the large number of state variables, efficient pricing of interest rate derivatives in LMM usually requires Monte Carlo simulation [48, 90, p.75, p.16]. In Section 7.3 we have seen that in general forward rate models an exact simulation is not possible and therefore always a discretization error exists. However, the LMM is originally based on a finite set of maturities that is already defined in advance. Therefore, only the time argument included in LMM requires discretization in order to make simulation even possible [46, p.174].

As a first step of the successful implementation of a LMM the numeraire of the model must be determined. In the previous section, for instance, we included a standard discount bond $P_{n+1}(t)$ as numeraire. Furthermore, it has been shown that the derivative's price is defined by the discounted expected payoffs over all possible price paths of forward LIBOR rates. This expectation of a random variable, however, can only be built if the random variable itself is simulated. In a next step, to enable the simulation of forward rates and the generation of random paths, respectively, the SDE of the according forward rate has to be discretized using, for instance, numerical schemes like Euler [48, p.75f]. There are countless possibilities for time-discrete approximation schemes. Gatarek et al. [44, p.200] compared the following methods: Euler method, Predictor-Corrector, Brownian bridge, combined Brownian bridge-predictor-corrector, Milstein method and the lognormal approach. As a result they found that the Brownian bridge and the combined Brownian bridge-predictor-corrector methods tend to outperform the other approaches regarding their density of approximation errors. Nevertheless, here we restrict ourselves to the Euler scheme.

Now we proceed to the actual simulation and apply a simple Euler scheme in oder to discretize the SDE (8.4), which leads to [46, p.175f]

$$\hat{L}_n(t_{i+1}) = \hat{L}_n(t_i) + \mu_n(\hat{L}(t_i), t_i)\hat{L}_n(t_i)[t_{i+1} - t_i] + \hat{L}_n(t_i)\sqrt{t_{i+1} - t_i}\sigma_n(t_i)Z_{i+1} \tag{8.6}$$

with

$$\mu_n(\hat{L}(t_i), t_i) = \sum_{j=g(t)}^{n} \frac{\delta_j \hat{L}_j(t_i)\sigma_n(t_i)\sigma_j(t_i)}{1 + \delta_j \hat{L}_j(t_i)}$$

and $Z_1, Z_2, \ldots$ are iid random vectors drawn from $\phi(0, I)$. Note that, as in Section 7.3, discretized variables are identified by a hat notation.

After this discretization the forward LIBOR rates can be simulated in one simulation trial using (8.6). Figure 9 shows the possible development of eight different paths of the 3 months US-Dollar LIBOR rate simulated for the next 3.5 years in a lognormal one factor LMM. Note that the starting value $\hat{L}_{0,T_1}(t_0)$, which was at 2.64% in May 2008, must not be simulated because it is already known in $t = 0$. To achieve the best possible approximation of discrete-time simulation runs to a continuous sample path of the diffusion process, many possible random paths have to be simulated [48, p.80]. Now the same procedure as presented in Section 1.2.2 applies. If the number of Monte Carlo runs is large enough, the strong law of large numbers ensures that the resulting Monte Carlo estimator is unbiased and strongly consistent.

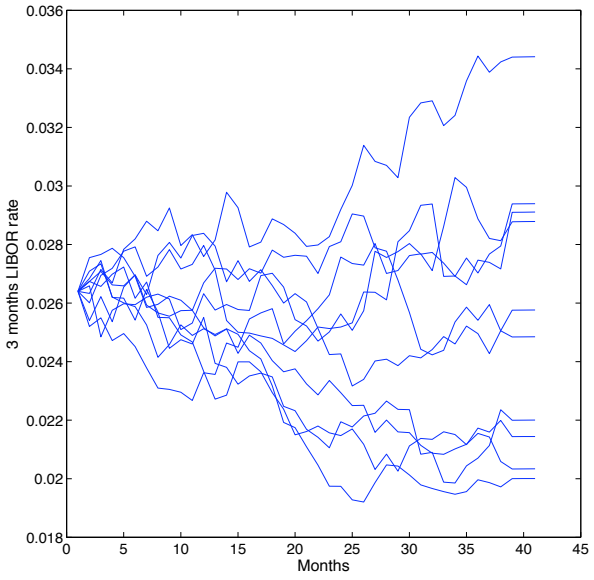



Figure 9: Simulated 3 months US-Dollar LIBOR rate paths

Note that the LMM can be easily extended to incorporate several independent factors. This means we can use the LMM as a single and a multifactor model. The optimal number of factors is typically determined by a principal component analysis of either the covariance matrix or the correlation matrix of changes in forward rates as it has already been explained in Section 6.1 [46, p.183f]. For an extensive discussion of this topic Chapter 16 of James and Webber [60, p.455-485] should be consulted.

## 8.4 Calibration

Thus far we have taken the volatilities $\gamma_n(t)$ in the LMM as given and therefore somehow neglected how they may be specified in practice. Yet, volatility factors play a decisive role in calibrating the LMM to current market prices of traded derivatives. Furthermore, they primarily determine both the level of volatility in forward rates and the correlations among them [46, p.181]. As we have seen in Section 8.2, some of the otherwise burdensome calibration process of the LMM can be done through closed form expressions. Algorithms to calibrate caps to the prevailing market prices are the simplest algorithms used in practice and do not even require the use of optimization techniques. This feature makes calibration fast what is one of the main advantages of market models [46, 94, p.184, p.211-214]. However, assumed that no formulae or effective approximations are available, the calibration process requires repeated simulation until the model price eventually matches the market. Thus, calibrating a LMM may become an interactive procedure [46, p.184]. Since the derivation of the exact calibrating procedure of the

LMM is an utterly technical affair and is not one of the primary goals of this work, it lies beyond the scope of this paper. However, the interested reader is referred to an excellent paper [93] or to Part III of the book [94, p.211-329], both written by Rebonato, which provide an extensive discussion of the entire calibration process of market models.

## 9 Pricing Bermudan swaptions in single and multifactor LIBOR market models

The purpose of this section is to show a detailed implementation of the previously discussed LMM in order to price Bermudan swaptions by Monte Carlo simulation. Bermudan-style swaptions give the holder the right, but not the obligation, to enter into a swap on a set of clearly prespecified exercise dates [59, p.159]. Thus, it is an option on a swap that includes an early exercise opportunity, i.e. the option may be exercised on some or even all payment dates, depending on the particular specification of the option, of the underlying swap [56, p.691]. Therefore, Bermudan swaptions belong to the category of basic American claims, which are typically endowed with an early exercise opportunity.

Derivatives of American type and thus also Bermudan-style interest rate derivatives in general can not be priced correctly by straightforward Monte Carlo simulation. It was believed for a long time that Monte Carlo would not even be able to tackle the problem of early exercises for the following two reasons. Firstly, the general Monte Carlo technique typically works *forward*, as the preceding sections have shown, while American options, for instance, have been usually priced by *backward* induction of lattice trees. Secondly, they require the simultaneous solution of an optimal *stopping time* problem, as we will see later in this section [21, 59, 72, p.1323ff, p.159f, p.113ff]. Until recently, the Monte Carlo approach was unable to deal with this additional tasks.

Nevertheless, in this paper the implementation of the LMM by Monte Carlo is carried out by using the example of Bermudan swaptions due to the fact that Bermudan swaptions are one of the, if not *the*, most traded interest rate derivatives on the market [59, 90, p.160, p.1]. Yet, until a relatively short time ago they have commonly been priced using only one factor short rate models such as the Black-Derman-Toy (BDT) model, for instance [88, p.1]. This was the case due to the problem of pricing American derivatives, particularly when more than one factor have an influence on the value of the option. And this is exactly the second reason for choosing Bermudan swaptions. The valuation of American-style derivatives has been one of the most challenging problems in derivative finance. Therefore, the existence of early exercise opportunities has lately attracted a great stream of research activities where the LMM has gained increasingly in importance and now already plays some kind of a benchmark role.

In recent years, some extended and thus more sophisticated Monte Carlo simulation techniques have been proposed with the goal to overcome the above-named hurdle and eventually to be able to price American derivatives even in multifactor models by simulation. In the meantime, Monte Carlo simulation has become an important tool for pricing early-exercise derivative securities. In this paper we restrict ourselves to one possible approach, referred to as *least-squares Monte Carlo*, which was introduced by Longstaff and Schwartz [72] in 2001. This regression-based algorithm and its implementation will be presented in Section 9.2, after exploring the general theory, including the optimal stopping time problem, of how to price American options by Monte Carlo simulation in Section 9.1. Section 9.3 represents an excursion about the short rate model Black-Derman-Toy. Subsequently, Bermudan swaption prices, computed in a LMM by least-squares Monte Carlo and in a BDT model, will be compared in Section 9.4. Furthermore, as

discussed before the volatility structure plays a distinctive role in the LMM theory. Therefore different volatility scenarios and general extensions to the LMM, such as volatility skews, will be examined in Section 9.5. Finally, since the LMM may incorporate several independent factors driving the yield curve, the factor impact on Bermudan swaption prices is considered in the last Chapter 9.6.

## 9.1 Monte Carlo pricing of American options

As briefly discussed above, the embedded right of an American-style derivative can be exercised at *any* time prior to the expiration date $T$. In the strict sense, the early exercise opportunity of a Bermudan derivative is restricted to a *finite* number of exercise dates. However, if we assume that the number of allowed exercise dates goes to infinity, the value of a Bermudan option eventually coincides with that of an American option. Therefore, it is basically not distinguished between American and Bermudan claims in this paper.

The value of an American option is maximized by following an optimal exercise strategy. Accordingly, for a correct valuation of the corresponding derivatives, an optimization problem must be simultaneously solved. This additional challenge has made it utterly difficult to price early-exercise derivatives by Monte Carlo simulation [12, 21, 96, p.1309f, p.1327f, p.271].

Suppose that the stopping time is a random variable itself, denoted by $\tau$, and that $\mathcal{T}$ indicates all possible exercise times within the duration of the option $[t, T]$. It directly follows that $\tau \in \mathcal{T}$. Thus, the value of a classical American put option at time $t$ is given by [21, 96, p.1327f, p.272]

$$P = \sup_{t \leq \tau \leq T} e^{-r\tau} E\left[K - (S(\tau))^+\right], \tag{9.1}$$

where the supremum is taken over the set of all stopping times $\mathcal{T}$. Note that the expectation in (9.1) is taken under the risk-neutral probability measure.

Of course, if we knew the optimal stopping time in advance, the correct value would be $e^{-r\tau}(K - S(\tau))^+$. Unfortunately, the optimal stopping strategy is never known in advance and therefore we also have to determine it by a simulation procedure. Consider, now, the application of ordinary Monte Carlo to price this American put option, as discussed in the paper of Broadie and Glasserman [21, p.1327f]. Then the optimal stopping time would be calculated for each simulated path. Taking a single path only, this procedure leads to the path estimate

$$\max_{i=0,\ldots,d} e^{-rt_i}(K - S(t_i))^+. \tag{9.2}$$

However, this approach does not only require a prefect foresight, which is admittedly unrealistic, but is also biased. From the inequality

$$\max_{i=0,\ldots,d} e^{-rt_i}(K - S(t_i))^+ \geq e^{-r\tau}(K - S(\tau))^+$$

follows that Equation (9.2) consequently overestimates the option price. To understand why, assume that the price path of the underlying has never reached a level where exercising the

option would have been superior to keeping it, i.e. it has never reached the optimal exercise region. Thus, according to the optimal but unknown exercise policy, the option will be kept until maturity $T$. On the other hand, under perfect foresight, as Expression (9.2) assumes, the option would be exercised at the trough of the underlying price path, regardless whether the optimal exercise region has been entered. This fact indicates the bias problem and shows that a standard Monte Carlo approach does not solve the pricing problem of an American option correctly. Therefore, we are in need of another simulation methodology that produces unbiased estimators for early-exercise derivatives. With the least-squares Monte Carlo approach such an simulation technique is explained and also implemented in Section 9.2.

### 9.1.1 Optimal stopping problem

Pricing American and Bermudan derivative securities requires the solution of the optimal stopping time problem. Therefore, the purpose of this subsection is to briefly establish the relationship between early-exercise derivative prices and the optimal stopping time. In general, the notion of an optimal stopping point is a stopping time, usually denoted by $\tau_*$, at which the supremum in Expression (9.1) is eventually attained [66, 89, p.17, p.35]. In a first step, the structure of the optimal stopping time of (9.1) is described. The entire set space can be divided into the *continuation set*, $C$, and the *stopping set*, $D$. Consequently, one can deduce that the optimal stopping time is obtained and the option should be exercised as soon as the observed value, here $S(t)$, enters $D$. In order to make this reasoning really meaningful, the sets $C$ and $D$ must be determined. For this purpose, Peskir and Shiryaev introduced a simple gain function $G(x) = (K - x)^+$ [89, p.379f]. Taking this function into account, the continuation and stopping sets may be defined as

$$C = \{(t, x) \in [0, T) \times (0, \infty) : P(t, x) > G(x)\} \tag{9.3}$$

and

$$D = \{(t, x) \in [0, T) \times (0, \infty) : P(t, x) = G(x)\}, \tag{9.4}$$

respectively, where $P(t, x)$ indicates the price of the American put option from (9.1) [89, p.379]. From these equations it follows that the stopping time $\tau_D$ is defined by

$$\tau_D = \inf\{0 \leq s \leq T - t : S(t + s) \in D\}. \tag{9.5}$$

Thus, the general stopping rule can be characterized as the first time the observed value, $S(t)$ for instance, enters a specified exercise region. Notice that this is a brief introduction to the optimal stopping time theory. The mathematically interested reader is referred to Lamberton and Lapeyre [66, p.17-28] or Peskir and Shiryaev [89, p.375-395]. Both provide a profound treatment of the optimal stopping problem for American options.

## 9.2 Least-squares Monte Carlo

Least-squares Monte Carlo (LS-MC) is a very powerful new approach that is able to price American-style derivatives consistently and correctly by Monte Carlo simulation. The decision

problem of a Bermudan or American option lies in comparing the payoff from immediate exercise with the expected payoff from continuation. If the immediate payoff is higher than the so called continuation value then the option will be exercised. The key insight of the LS-MC method is that the continuation value is an expectation conditional on continuation and can be approximated under the assumption that the continuation value is eventually a simple function of those variables that are observable at the exercise date in question. The estimation of these required functions is carried out by extracting cross-sectional information in the simulation using standard least-squares regressions. Particularly, ex-post realized payoffs from continuation are regressed on the specified functions of the ex-ante observed values of the state variables. The fitted values from this regression then provide an efficient and unbiased estimate of the conditional expectation of the continuation value, which simultaneously specifies the optimal exercise strategy along each path. This means the optimal stopping time can also be estimated from this expectation [72, 88, p.114f, p.8].

We now explore the implementation of the LS-MC method for pricing Bermudan swaptions. The goal of LS-MC is to provide a pathwise approximation to the optimal stopping rule. As mentioned above, the characteristics of the optimal stopping rule are that it maximizes the value of the early-exercise option. Suppose that the option can be exercised at $K$ discrete times $0 < t_1 \leq t_2 \leq \cdots \leq t_K = T$. Furthermore, the optimal stopping policy is taken into account at each exercise date. At each time $t_k$, the immediate-exercise value ($EV$) of the option is equal to its payoffs and therefore completely known. The continuation value ($CV$), however, is not known at time $t_k$ and therefore must be calculated, under no-arbitrage condition and risk-neutral pricing measure, by taking the expectation of the remaining discounted cash flows [72, 83, p.121f, p.110]:

$$CV(t_k) = E\left[\sum_{j=k+1}^{K} \exp\left(-\int_{t_k}^{t_j} r(s)ds\right) Y(\omega, t_j; t_k, T) \,|\, \mathcal{F}_{t_k}\right]. \tag{9.6}$$

The notation $Y(\omega, s; t, T)$, where $\omega \in \Omega, s \in (t, T]$, indicates the path of cash flows generated by the option, conditional on (a) the exercise of the option after $t$ and (b) the optimal stopping strategy is followed by the optionholder for all $s, t < s \leq T$ [83, p.110]. Furthermore, $\mathcal{F}_{t_k}$ denotes the information set at time $t_k$ under which the above expectation is taken. Hence, the problem of finding the optimal exercise time is actually reduced to the comparison of $EV$ and $CV$. The option will be exercised if the $EV$ at time $t_k$ is positive, i.e. if $EV(t_k) \geq CV(t_k)$.

### 9.2.1 Least-squares Monte Carlo algorithm

The LS-MC method runs least-squares regressions to approximate the conditional expectation function, i.e. the $CV$s, at the times $t_{K-1}, t_{K-2}, \ldots, t_1$. Since the path of cash flows, $Y(\omega, s; t, T)$, of the option is defined recursively, the implementation algorithm accordingly works backward. Remember that $Y(\omega, s; t_k, T)$ and $Y(\omega, s; t_{k+1}, T)$ may not be the same since it might be optimal to stop and thus exercise the option at time $t_{k+1}$ [72, p.122].

In order to acquire a detailed understanding of Longstaff and Schwartz's sophisticated simulation technique and also to get an idea of how to implement it in MATLAB, a step-by-step implementation of LS-MC for pricing Bermudan swaptions is described. Recall that a Bermudan swaption is generally characterized by three different dates: the lockout date $t_s$, the last possible exercise date $t_x$ and the maturity of the underlying swap $t_e$. The general pattern of this implementation procedure is according to Amin [1, p.7-9], Brigo [20, p.586-590] and Longstaff and Schwartz [72, p.122f].

1. *Number of paths.* At the beginning, choose a sufficiently large number of paths, $n$, to ensure that the calculated values will be smooth and precise. The total number of simulation steps is in case of Bermudan swaptions usually chosen to be equal to the number of possible exercise dates [1, p.7].

   Consider, for instance, that the option is exercised at $\tau = t_i$. Then the value of the remaining swaption is given by

   $$S_{i,e}(t_i) = \sum_{k=i}^{e-1} P(t_i, t_{k+1}) \delta_k \left( L_k(t_i) - K \right), \tag{9.7}$$

   where $L_k(t_i)$ indicates the forward LIBOR rate observed at the exercise time $t_i$ for the period $[t_k, t_{k+1}]$ and, of course, $t_x = t_{e-1}$.

2. *Basis functions.* In order to approximate the $CV$ as a function of the past and present values of the underlying, $i_k$ basis functions $p_1(t_k, L), \ldots, p_{i_k}(t_k, L)$ are chosen. To price Bermudan swaptions, these functions are commonly determined to be quadratic. Nevertheless, there are basically five different choices of possible basis functions which are specified in Amin's paper [1, p.10f].

3. *Simulation of the underlying variables.* In case of a Bermudan swaption, totally $n$ paths of the the LIBOR forward rate are simulated according to Equation (8.6) in a LMM. It was discussed in Section 8 that simulating forward rates requires a numerical discretization such as Euler, for instance.

4. *Exercise value and stopping time.* As mentioned above, the algorithm to value Bermudan swaptions by LS-MC works backward and thus begins at time $t_x$. At first, the value of the swaption at time $t_x$, denoted by $V(t_x)$, is simply equated with the value of the option in case of exercise or zero otherwise. Since $t_x$ is the last possible exercise date, $V(t_x)$ automatically corresponds to the exercise value, $EV(t_x)$. Thus, the following equation holds:

   $$V(t_x) = \max\left(S_{x,e}(t_x), 0\right) = EV(t_x).$$

   Furthermore, the variable $\nu$ is introduced which indicates the optimal stopping time along the LIBOR rate process. Initially, the stopping time is set equal to the final time $T$. It follows that expression

   $$\nu = t_x$$

holds at the beginning.

5. *Least-squares regression.* At this point of the implementation the iterative part of the entire scheme begins and we work backwards from $t_x$. The goal is to approximate the discounted $CV$ at time $t_{x-1}$ by a linear combination of the basis functions $p_1(t_k, L), \ldots, p_{i_k}(t_k, L)$ from Step 2, i.e. the discounted $CV$ is regressed on the basis functions. The corresponding regression coefficients of the individual basis functions at time $t_i$ are estimated by ordinary least-squares regression and expressed by the vector $\beta(t_i)$.

6. *Comparison of continuation and immediate-exercise value.* The $CV$ is estimated by the fitted values from $\sum \beta(t_i)\, p(t_i)$, the least-squares regression. Recall that the $CV$ corresponds to the conditional expected payoff under the assumption that the option is kept alive. Now, the $CV(t_i)$ is compared with the $EV(t_i)$. If the latter is greater, i.e. $CV(t_i) < EV(t_i)$, the option is immediately exercised. From this it follows that $t_i$, the present time, is indeed the optimal stopping time. Therefore, $V$ is automatically equal to $S_{i,e}(t_i)$ and the optimal stopping time is changed from $T$ to $t$ at exercise. Formally, these two expressions must hold:

$$\begin{aligned} V(t_i) &= S_{i,e}(t_i) \\ \nu &= t. \end{aligned} \tag{9.8}$$

7. *Restart.* If Step 6 is successfully accomplished, we restart at Step 4 and repeat the Steps 4 to 6 for all $t_{x-1} \leq t_i \leq t_s$ until the first exercise time of the Bermudan swaption is eventually reached.

This is a first, yet rudimentary description of a possible implementation of the LS-MC in a LMM. To get more into technical details it is referred to Appendix A on Page IX, where a MATLAB code, including detailed descriptions of the single steps, for pricing Bermudan swaptions is listed.

### 9.2.2 Convergence and robustness of least-squares Monte Carlo

Thus far, the LS-MC method itself and its implementation algorithm have been discussed. An additional topic is the general quality of this new simulation approach applied to price American derivatives. By quality we mean, amongst others, the convergence properties and the robustness of the LS-MC technique. Longstaff and Schwartz did already investigate the convergence characteristics of their introduced simulation model. Their findings about the convergence properties, however, remained rather shallow [72, p.124f]. A far more detailed analysis of the LS-MC method is provided by Clément et al. [28]. At the beginning, they distinguished between two different approximations in the LS-MC algorithm [28, 83, p.450, p.108]. The first approximation is that the algorithm replaces the conditional expectations in the dynamic pricing principle by

orthogonal projections on a finite set of functions. The approximation two is that Monte Carlo simulations and least-squares regressions are used in order to estimate the conditional expectation function. They basically prove that the entire LS-MC algorithm eventually converges as the number of basis functions goes to infinity. In other words, the estimated conditional expectation coincides with the true conditional expectation. They additionally determined the exact convergence rates of the LS-MC method [28, p.449f].

Moreno and Navas [83] took a slightly different approach and analyzed the overall robustness of the LS-MC method relative to the type and number of basis functions for pricing early-exercise options numerically. They found that this approach is considerably robust for pricing American put options. In this case the basis functions are reduced to the degree of a polynomial on the asset price. On the other hand, if the choice of the basis functions is not given, e.g. for complex derivatives, the robustness is no longer guaranteed since the number and especially the type of the basis functions may influence option prices considerably [83, p.126].

### 9.3 Exkursion: Black-Derman-Toy model

The Black-Derman-Toy model (BDT) is a one factor no-arbitrage short rate model and was proposed by Black, Derman and Toy in 1990 [8]. All interest rates or also security prices depend on a single factor, the short rate. The BDT model may therefore be viewed more properly as a topic for Section 6.2, as it is also listed in Table 5. Nevertheless, it is included here for two reasons. Firstly, the BDT model has been playing an important role in pricing Bermudan swaptions for a considerable time. Yet, it has been partially superseded by the LMM over the last couple of years. Secondly, the purpose of the subsequent section is to draw a comparison between the LS-MC approach in a LMM and the BDT model on the basis of computed Bermudan swaption prices.

The BDT model assumes the short rates to follow a lognormal process, which is given by the SDE [29, p.221]

$$d\ln r = \left(\theta(t) + \frac{\sigma'(t)}{\sigma(t)}\ln r\right)dt + \sigma(t)dW(t). \tag{9.9}$$

It basically corresponds to the lognormal version of the Ho-Lee model. The lognormal feature, however, prevents the interest rates in the BDT model from taking negative values and is thus a decisive advantage. The BDT approach has been in good odor among practitioners due to both the simplicity of its calibration process and its straightforward analytical results [92, p.203]. SDE (9.9) shows that in the BDT approach two independent functions of time, $\theta(t)$ and $\sigma(t)$, are included. These two functions make it possible that the model matches not only the current term structure of interest rates but also the prevailing structure of the interest rate volatilities [29, 92, p.221f, p.204f]. This provides the model with the necessary flexibility to be able to price sophisticated interest rate derivatives such as Bermudan swaptions, for instance.

Given these advantages, the BDT model also suffers from some significant shortcomings. As it is in the nature of one factor models, the yields in the BDT model tend to move too much in parallel. This is admittedly unrealistic. The Ho-Lee model, for instance, assumes constant

absolute volatility and thus yields can only move exactly in parallel. Hence, the BDT model allows for some more complex changes in the term structure shape than comparable one factor short rate models do. Yet, under certain circumstances the volatility function $\sigma(t)$ may affect the instantaneous short rates that they will even be mean-fleeing rather than mean-reverting. Another drawback is that it is not able to specify the volatility of different maturity yields independently of the future volatility of the short rate. This fact may be of relevance for long-maturity swaps with an underlying that is determined by a short-maturity index [92, p.209-213].

In a BDT model interest rate derivatives are usually priced using recombining binomial trees. For each time period, $\Delta t$, the spot rate $r(i)$ from 0 until $i\Delta t$ and its corresponding volatility must be known. Then the spot rate curve is modeled in such a way that a zero bond maturing at $2\Delta t$ is correctly priced [29, 92, p.234ff, p.205ff]. In the next section this approach is followed and the BDT model will be applied in order to price interest rate Bermudan swaptions in a binomial tree.

### 9.4 Least-squares Monte Carlo versus Black-Derman-Toy

The BDT model has been very popular in practice to price Bermudan swaptions and also other types of interest rate derivatives. Since BDT prices derivatives by backward induction in a binomial tree, it is undoubtedly a suitable approach to value derivatives with early exercise opportunities such as Bermudan swaptions. Nevertheless, it has been continuously replaced by the LMM model over the past years. Possible reasons for this fact lie in the different shortcomings of the BDT model, which has been briefly discussed in Section 9.3. In general, one decisive drawback compared to the LMM is that it is based on merely one factor. This requires that all changes in bond yields are perfectly correlated [29, 92, p.221f, p.206ff]. Therefore, the BDT model is not suited to value interest rate derivative securities whose prices strongly depend on the effective correlation structure between interest rates. With the introduction of more complex derivatives valuation trees have become frequently non-recombining. The obvious alternative tool has been Monte Carlo simulation. Since several techniques, which enable Monte Carlo to deal with American-type derivatives, have been proposed, the LMM has continuously gained importance. Today, it is a very popular tool among traders in practice [94, p.12f].

The main purpose of this section is actually to contrast the BDT with the LMM, simulated in a LS-MC model, by comparing computed Bermudan swaptions prices. This is exactly done in Table 6. This table exhibits such a comparison between the two models LS-MC in a LMM and BDT, which are often applied in Bermudan swaption valuation in practice. To achieve comparability, the LMM dynamics in the LS-MC approach include one factor. Thus, both the BDT and the LS-MC prices are computed in a one factor model. Furthermore, the volatility in the LMM is chosen to remain constant. The variable $\alpha$ denotes the constant elasticity of variance (CEV) parameter (see Subsection 9.5.1) and is used for reproducing a slightly curved implied volatility skew. Since $\alpha = 1$, the LMM corresponds to a lognormal LMM. The LS-MC estimates as well as the related regressions are based on the simulation of totally 50 000 paths whereas the BDT prices have been computed using ordinary BDT binomial tree lattices, as

| $T_s$ | $T_e$ | $\delta$ | Volatility ($\gamma(t,T)$) | $\alpha$ | P/R | LS-MC | BDT |
|---|---|---|---|---|---|---|---|
| 1 | 4 | 0.5 | 0.20 | 1 | +1 | 156.9 (0.0217) | 158.0 |
| 1 | 4 | 0.5 | 0.20 | 1 | −1 | 156.4 (0.0172) | 156.7 |
| 2 | 4 | 0.5 | 0.20 | 1 | +1 | 186.6 (0.0305) | 187.8 |
| 2 | 5 | 0.5 | 0.20 | 1 | −1 | 186.3 (0.0227) | 186.6 |
| 5 | 10 | 0.5 | 0.15 | 1 | +1 | 283.1 (0.0554) | 283.8 |
| 5 | 10 | 0.5 | 0.15 | 1 | −1 | 279.0 (0.0415) | 279.3 |

Table 6: LS-MC (one factor LMM with constant volatility) versus BDT prices (in basis points) on Bermudan swaptions with the strike rate $K = 6\%$. The initial forward LIBOR rate curve is assumed to be flat at 6%. $T_s$ indicates the lockout date of the swaption and $T_e$ shows the maturity of the underlying swap in years. Moreover, P/R stands for payer or receiver swaption, whereas the former is indicated by +1 and the latter by −1. The standard errors of the LS-MC are indicated in parentheses. Similar studies have been conducted by Andersen [2] and Pedersen [88].

they have been explained above. Note that Pedersen [88, p.23f] drew a similar comparison and calculated almost equal BDT prices.

From the results in Table 6 one can draw the following conclusions. The calculations show that the LS-MC prices tend to be slightly lower than the BDT Bermudan swaption values. The fact that the different prices are still close to each other suggests that the underlying dynamics of the two models are similar. Recall that due to the choice of $\alpha$ being equal to one both models, LMM and BDT, are based on lognormality. Furthermore, they indicate that the price differences between these two approaches are more pronounced for payer than receiver Bermudan swaptions. However, the discrepancies are still on a very low level, considering that the prices are indicated in basis points. It is of considerable difficulty to judge which model provides more accurate prices. Andersen [2, p.11], for instance, also drew a comparison of Bermudan swaptions prices between these two models. In general, he came to very similar results, i.e. BDT prices which are slightly higher compared to the LMM values. He eventually concluded that the LMM tends to be biased low relative to the BDT model. It would require a thorough scrutiny of these slight discrepancies in order to be able to reach a final conclusion. This, however, is beyond the scope of this paper.

### 9.5 Extensions of the LMM

The volatility structure plays a decisive role in the LMM theory. In this section we go beyond the standard approach, as it has been discussed so far, and extend the ordinary LMM by allowing for volatility smiles and skews. These extended models will be capable of producing volatility

smiles and skews which are completely consistent with current market data. Firstly, a brief outline of possible extensions of the LMM is given. In the second part, the effect of introducing different volatility scenarios on Bermudan swaption prices, computed by the LS-MC approach in a one factor LMM, is analyzed. In particular, it will be investigated whether allowing for volatility skews does directly influence Bermudan payer and receiver swaption prices.

### 9.5.1 Accounting for smiles and skews: CEV

The basic assumption of the LMM, lognormaility of the forward LIBOR rates, is often observed to be violated on cap or swaption markets. In particular, the distribution of the empirical forward LIBOR rates often exhibits a fat left tail, referred to as *volatility skew*. The fact that the usually assumed lognormality may be violated asks for alternative models. Models where the particular diffusion coefficients of the forward LIBOR rates are non-linear functions of the rates themselves. One specific possibility is that a certain power-function can describe the foward-dependence of the diffusion term. Such an approach is called *constant elasticity of variance (CEV)* model [3, p.2].

The CEV model was originally introduced by Cox and Ross [32] in 1976. But Andersen and Andreasen [3] developed it further and firstly applied the CEV dynamics as a model for the development of forward LIBOR rates. The general forward LIBOR rate dynamics, which served as starting point in the elaborations of Andersen and Andreasen, is given by [3, 20, 94, p.5, p.456, p.349f]

$$dL_n(t) = \varphi\left(L_n(t)\right)\sigma_n(t)dW(t) \tag{9.10}$$

and

$$\sigma_n(t) = \varphi(L_n(t))\lambda_n(t),$$

where $\varphi$ is a possibly non-linear function and $\lambda$ is a bounded $d$-dimensional deterministic function. While the crude market model characteristics are usually not rich enough to capture the frequently observed volatility skews, the $\lambda_n(t)$ function makes it possible to calibrate the model to the exact term structure of implied caplet volatilities [3, p.5]. Andersen and Andreasen then proposed a particularly tractable specification of (9.10) as the CEV model, where

$$\varphi(L_n(t)) = [L_n(t)]^{\alpha}$$

and with $\alpha$ as a positive constant [3, 20, p.6, p.456]. This is eventually the CEV model proposed by Cox and Ross [32] and elaborated by Andersen and Andreasen [3]. Consider, now, three special cases of the CEV model. Firstly, if $\alpha$ is set equal to zero, $\alpha = 0$, then we end up with normal dynamics. The second case where $\alpha = 1$, directly leads to lognormal dynamics, i.e. to the lognormal LMM proposed by Brace et al. [13]. Thirdly, setting $\alpha$ equal to a half, $\alpha = 0.5$, gives rise to a square-root process [3, 20, 94, p.6f, p.457, p.350].

These explanations show that the standard LMM can be extended to non-linear forward volatility functions. Forward rate models based on the CEV framework are able to consistently reproduce the on the market for caps or swaptions observed volatility smiles and skews. CEV

models are shown to maintain an equal level of tractability as the lognormal LMM does. Yet, they can generate upward- as well as downward-sloping volatility skews completely consistent with prevailing market data. Therefore, they may achieve a better fit to observed prices for caplets, for instance [3, p.2 & p.26f].

### 9.5.2 Volatility scenarios: effect of a skew

We now go a step further and illustrate the effect of introducing a volatility skew on Bermudan swaption prices. Recall that the variable $\alpha$ denotes the constant elasticity of variance (CEV) parameter. In particular, in order to incorporate a volatility skew, we simply change the value of $\alpha$. Setting $\alpha$ equal to one gives rise to the ordinary lognormal LMM. According to Amin [1, p.11] as well as Andersen and Andreasen [3, p.20], in order to keep at-the-money prices constant and independent of a volatility skew, the parameters $\lambda_n^{CEV}$ for the volatility in the CEV model and $\lambda_n^{LN}$ for the lognormal model are related as follows: $\lambda_n^{CEV} = \lambda_n^{LN} \times L_n^{1-\alpha}$, where $L_n$ indicates the forward LIBOR rate.

First of all, the effect of a volatility skew on in-the-money (ITM) payer swaptions and out-of-the-money (OTM) receiver swaptions are illustrated in Table 7. Notice that choosing $\alpha = 0.5$ leads to a distribution for the forward LIBOR rates that is skewed left of the lognormal distribution. Therefore, ITM payer swaptions and OTM receiver swaptions are valued considerably higher than similar Bermudan swaptions in a lognormal LMM. Consistently, while raising $\alpha$ to 1.5, ITM payer and OTM receiver Bermudan swaption prices fall below the values stemming from the lognormal model. This is a logical consequence if the left volatility skew is taken into account.

Contrary, prices of OTM payer and ITM receiver Bermudan swaptions are lower in a left-skewed CEV, produced by $\alpha = 0.5$, than in a lognormal model. This fact is illustrated in Table 8. Consequently, the rise in the OTM payer and ITM receiver swaption values continues while increasing $\alpha$ from 1 to 1.5. Thus, it seems to exist a positive relationship between the CEV parameter $\alpha$ and Bermudan swaption prices of OTM payer and ITM receiver swaptions. However, these calculations showed that changing $\alpha$ and thus introducing a volatility skew does directly effect the prices of Bermudan swaptions. Remember, however, that changing $\alpha$ simultaneously changes the effective volatility considered in the model. Similar studies about a skew and smile effect have been conducted by Amin [1] and Andersen [2]. Their findings go into the same direction, yet they are more pronounced and dispersed since they are based on different parameter characteristics and other volatility scenarios.

## 9.6 Factor dependence of Bermudan swaption pricing in a lognormal LMM

The dynamics of a LMM, which were derived in Section 8.1 and explicitly stated in Equation (8.4) for a one factor model, can be easily extended to incorporate more than only one independent factor driving the forward LIBOR rate curve. Assume, for instance, that a particular LMM includes $d$ factors in total. Of course, each factor supposed to drive the yield curve plays a

| $T_s$ | $T_e$ | $\delta$ | P/R | LS-MC $\alpha = 0.5$ | LS-MC $\alpha = 1.0$ | LS-MC $\alpha = 1.5$ |
|---|---|---|---|---|---|---|
| 2 | 5 | 0.25 | +1 | 173.5 (0.0175) | 169.5 (0.0185) | 165.0 (0.0193) |
| 2 | 5 | 0.25 | −1 | 40.0 (0.0071) | 37.6 (0.0063) | 35.5 (0.0057) |
| 4 | 10 | 0.25 | +1 | 371.6 (0.0444) | 361.7 (0.0486) | 367.3 (0.0422) |
| 4 | 10 | 0.25 | −1 | 133.7 (0.0218) | 123.1 (0.0186) | 118.2 (0.0167) |
| 10 | 20 | 0.25 | +1 | 619.9 (0.1040) | 604.3 (0.1109) | 598.3 (0.0904) |
| 10 | 20 | 0.25 | −1 | 312.6 (0.0563) | 298.9 (0.0473) | 294.8 (0.0428) |

Table 7: Introduction of a volatility skew in a in-the-money payer swaption. The initial forward LIBOR rate curve is assumed to be flat at 3%, whereas the strike rate is $K = 2.5\%$. The volatility scenario is chosen to be $\lambda_n(t) = 0.18 \times 0.03^{1-\alpha}$ and we used 30 000 paths for valuation. Note that all prices are in basis points, calculated by a LS-MC approach, and that the standard errors of the LS-MC are indicated in parentheses.

| $T_s$ | $T_e$ | $\delta$ | P/R | LS-MC $\alpha = 0.5$ | LS-MC $\alpha = 1.0$ | LS-MC $\alpha = 1.5$ |
|---|---|---|---|---|---|---|
| 2 | 5 | 0.25 | +1 | 23.2 (0.0064) | 26.7 (0.0077) | 31.6 (0.0097) |
| 2 | 5 | 0.25 | −1 | 287.5 (0.0189) | 290.8 (0.0182) | 294.9 (0.0176) |
| 4 | 10 | 0.25 | +1 | 96.3 (0.0233) | 112.2 (0.0299) | 123.5 (0.0213) |
| 4 | 10 | 0.25 | −1 | 567.4 (0.0469) | 579.4 (0.0441) | 598.3 (0.010) |
| 10 | 20 | 0.25 | +1 | 265.9 (0.0717) | 300.5 (0.0966) | 311.6 (0.0687) |
| 10 | 20 | 0.25 | −1 | 885.6 (0.1042) | 920.4 (0.0938) | 942.8 (0.0828) |

Table 8: Introduction of a volatility skew in a out-of-the-money payer swaption. The initial forward LIBOR rate curve is assumed to be flat at 3%, whereas the strike rate is $K = 4\%$. The volatility scenario is chosen to be $\lambda_n(t) = 0.18 \times 0.03^{1-\alpha}$ and we used 30 000 paths for valuation. Note that all prices are in basis points, calculated by a LS-MC approach, and that the standard errors of the LS-MC are indicated in parentheses.

part in contributing to the model's total volatility. The component of the volatility of $L_n(t)$, contributed by the $q$th factor, is denoted by $\gamma_{n,q}(t)$. Thus, the dynamics of a multifactor LMM may be defined as [56, p.685]

$$dL_n(t) = \sum_{j=g(t)}^{n} \frac{\delta_j L_j(t) L_n(t) \sum_{q=1}^{d} \gamma_{n,q}(t)\gamma_{j,q}(t)}{1+\delta_j L_j(t)}\, dt + L_n(t) \sum_{q=1}^{d} \gamma_{n,q}(t)\, dW_q(t), \tag{9.11}$$

where the Brownian increments, $dW_q$, are chosen to be orthogonal to each other [94, p.186].

The goal of this section is to study the impact of the number of factors in a lognormal LMM on Bermudan swaption prices. For this purpose there are several different volatility scenarios that could be incorporated in a multifactor LMM in order to investigate the factor impact on swaption prices. However, this work rests on the following specification, here exemplarily illustrated for a two factor model, proposed by Andersen and Andreasen [3, p.20] and also used in a study conducted by Andersen [2, p.20]:

$$\lambda_n(t) = \left(0.13, 0, 13 - \sqrt{0.009 \times n \times \delta}\right).$$

The factor impact in interest rate models has been the topic of several studies so far. Some of these studies, however, do not agree upon the factor dependence and are even contradictory in parts. On the one hand, Longstaff et al. [70], for instance, found a significantly strong positive relationship between Bermudan Swaption prices and the number of factors. Similarly, Sidenius [97] also indicated that there may exist a relatively strong dependence of Swaption prices on the number of factors in market models. On the other hand, Andersen and Andreasen [4] state that standard Bermudan swaption prices computed in a one and two factor LMM are almost identical. They thus conclude that swaptions show little sensitivity to the number of factors, but with the one factor prices being slightly higher than the two factor prices.

The findings of this work, with regard to different durations of the swaptions, slightly diverge from some of the studies mentioned above but are partially in line with the results of Pedersen's paper [88, p.25f], even though he allowed for different volatility and correlation structures.

In order to investigate the concrete factor impact Bermudan receiver swaptions are priced in a lognormal LMM by the LS-MC approach. Table 9 exhibits these Bermudan receiver swaption prices for different strike rates, durations as well as number of factors in a lognormal LMM. A first result is that the factor dependence must be explored with regard to the duration of the swaptions. There is a negative factor impact on prices of swaptions with a rather short duration, although only a weak one. In particular, most of short-dated swaption premia moderately decrease with additional factors in the model. Hence, there seems to be a tendency, though not a distinct one, that prices of short duration Bermudan swaptions decline from one factor to multifactor market models. The tide has turned if we consider durable Bermudan swaptions. For swaptions with 10-year swaps as underlying, for instance, the conclusion that their premia show a positive sensitivity to the number of factors appears to hold. A deep in-the-money and long-living Bermudan swaption price increases steadily if we raise the number of factors as we can see in the last rows of Table 9. Yet, the factor dependence does not seem to be pronounced

| | | | Number of factors | | | |
|---|---|---|---|---|---|---|
| $K$ | $T_s$ | $T_e$ | 1 | 2 | 3 | 5 |
| 4.0 | 1 | 3 | 16.7 (0.0032) | 16.3 (0.0032) | 16.2 (0.0032) | 15.9 (0.0036) |
| 5.0 | 1 | 3 | 110.4 (0.0085) | 108.81 (0.0085) | 109.2 (0.0084) | 109.0 (0.0085) |
| 6.0 | 1 | 3 | 276.0 (0.0110) | 274.0 (0.0109) | 273.1 (0.0111) | 274.3 (0.0112) |
| 4.0 | 2 | 6 | 63.4 (0.0102) | 63.1 (0.0103) | 64.8 (0.0110) | 68.4 (0.0120) |
| 5.0 | 2 | 6 | 232.9 (0.0208) | 234.3 (0.0214) | 241.1 (0.0220) | 242.4 (0.0220) |
| 6.0 | 2 | 6 | 511.4 (0.0280) | 512.1 (0.0281) | 512.3 (0.0281) | 517.3 (0.0297) |
| 4.0 | 5 | 10 | 142.7 (0.0224) | 144.6 (0.0244) | 149.4 (0.0254) | 156.1 (0.0261) |
| 5.0 | 5 | 10 | 328.3 (0.0351) | 334.8 (0.0364) | 336.0 (0.0365) | 341.2 (0.0380) |
| 6.0 | 5 | 10 | 599.4 (0.0477) | 606.2 (0.0498) | 613.1 (0.0506) | 621.6 (0.0519) |

Table 9: Factor dependence of Bermudan receiver swaption prices, given in basis points, in multifactor market models. The initial forward LIBOR rate curve is assumed to be flat at 4.5%. The volatility structure was chosen to be 0.13, $0.13 - \sqrt{0.009 \times n \times \delta}$, whereas $\delta = 0.5$ and $\alpha = 1$. The least-squares Monte Carlo estimates are based on 50 000 paths and the standard errors are indicated in parentheses.

either. This leads to the second conclusion that can be drawn. The number of factors does indeed influence Bermudan swaption premia in a lognormal LMM. Nevertheless, the impact remains on a relatively low level for short- as well as long-living Bermudan swaptions.

To conclude, these calculations highlight the importance of considering different maturities of swaptions while comparing their factor dependence. One can say that the factor impact turns from being slightly negative for short-dated to positive for durable Bermudan swaptions. However, unlike what was found in other studies for a range of exotic instruments, the factor impact on Bermudan swaption prices seems to be consistently moderate.

## 10 Conclusion

In this paper the foundations and several extensions of the Monte Carlo theory have been developed and implemented. After exploring the application of Monte Carlo methods in numerous important term structure models, the very powerful regression-based least-squares Monte Carlo approach has been introduced and eventually implemented to price Bermudan swaptions in a LIBOR market model.

The beginning of this paper has provided an introduction to the generic implementation of a standard Monte Carlo pricing tool and has shown that the crux hinges on determining the distributional characteristics of the underlying random variable. The distributional properties are required to generate and simulate, respectively, price or interest rate paths. Since Monte Carlo estimators approach the true value with probability one as the sample size goes to infinity, they are unbiased and consistent. A decisive advantage, especially for high dimensional calculations, is that its convergence rate $O(1/\sqrt{n})$ is independent of the problem's dimensionality. However, the convergence rate is considerably slow in low dimensions which is a clear shortcoming. As many statistical methods, Monte Carlo techniques are also relying on randomness. Random number generators are deterministic at heart, therefore they actually mimic genuine randomness by producing pseudorandom numbers. Since not the randomness itself but rather how well the numbers fit the given distribution they are pretended to be drawn from is crucial, a generator's pseudorandom character is basically irrelevant. Of primary importance are, however, the length of the period as well as the efficiency of the generator. With regard to these criteria, Deng and Lin stated that a fast multiple recursive generator tends to outperform its alternatives [34, p.146f].

Variance reduction techniques and low-discrepancy sequences accomplished a significant improvement in efficiency of Monte Carlo methods. The former achieve a reduction of the given variance rate $\sigma^2/n$ by using covariates or particular sampling techniques. Therefore, the precision of the estimate may be increased while keeping $n$ and thus the computational effort unchanged. Notice that variance reduction techniques must always be applied with care because some may even backfire and lead to an undesirable variance increase if the specifications are not chosen properly. In general, the variance reduction potential and the cost and complexity of implementation are positively related. Antithetic variates, for instance, are easily applied but also restricted in their effectiveness while importance sampling is utterly powerful but also expensive. It follows that a particular variance reduction technique must be coordinated with the specifications of an individual problem. The quasi-Monte Carlo approach represents a deterministic version of the crude Monte Carlo method. It chooses completely deterministic points which are supposed to cover a $d$-dimensional space with fewer gaps than random numbers. Wisely-chosen deterministic points are too evenly distributed to be random. Therefore, low-discrepancy sequences have the potential to accelerate the convergence rate to nearly $O(1/n)$ and are thus very effective in increasing speed and precision. Since the valuation of complex derivatives might be computationally burdensome, a great deal of research has been spent with the ultimate goal of

increasing efficiency over the past years.

Monte Carlo methods play a decisive role in the valuation of numerous interest rate derivatives, which usually require, unlike equity derivatives, the simulation of the entire yield curve. Short rate models, e.g. Vasicek, CIR, Ho-Lee or Hull-White, relying on one or several factors, explore the development of the dynamics of the instantaneous short rate. They generally press home the advantage of being easily implemented and well-tractable. Thus, short rate models have been of high importance for pricing standard and also some nonstandard interest rate derivatives. Forward rate models, however, draw on describing the evolution of the full yield curve with the advantage of incorporating exotic volatility and correlation structures. This is the reason why forward rate models like HJM or LMM have been increasingly gaining importance in recent years. The HJM framework is completely determined by the forward rate's volatility and correlation structure in a no-arbitrage world. Since upcoming interest rate derivative products are often influenced by the correlation structures among rates, the HJM approach has a very strong appeal. Furthermore, implementing market models in lattice valuation frameworks leads to non-recombining trees. Consequently, Monte Carlo is the tool of choice for market models.

We have seen that the advantage of the LMM is that it is based on simple rates, which are directly observable on the market. The LMM is one of the most popular models for pricing interest rate derivatives. It partially owes its popularity to the consistency with the existing market practice, i.e. with Black's formula. This feature again makes the calibration procedure comparably fast. In this paper, the LMM has been implemented in order to price Bermudan swaptions. Broadie and Glasserman [21, p.1327f] showed that applying standard Monte Carlo methods to value early-exercise derivatives leads to an upward bias. Therefore, Longstaff and Schwartz's [72] least-squares Monte Carlo method, which is based on a simple regression approach, is used to price Bermudan swaptions. The findings are that Bermudan swaption prices, computed by LS-MC in a LMM, are robust and consistent. Moreover, they are in line with Black-Derman-Toy prices. Introducing volatility skews, as they are observable on the market, does directly influence swaption prices. Additionally, some calculations highlighted that the factor dependence must be considered under different maturities. Short-dated swaptions tend to be negatively related to the number of factors whereas long-living Bermudan swaption prices increase while allowing for additional factors. However, the factor dependency in general does not seem to be dramatic.

In conclusion, Monte Carlo methods have been gaining increasingly importance and have become a main pillar in derivative valuation practice. Many derivative securities and structured products have reached a level of complexity where numerical methods are the preferably applied pricing techniques. However, with foresight there is definitely room for improvements with regard to efficiency, flexibility as well as precision. Even though processing power has advanced at an incredible pace, simulations may still be computationally expensive. Therefore, current and also future research activities are aiming at increasing accuracy and enhancing as well as speeding up Monte Carlo methods.

## A Appendix: Matlab codes

### Section 1.1.3: Monte Carlo integration

```
%% Monte Carlo integration

% Author:   Roman Frey
% Date:     24.02.2008
% Contact:  roman.frey@student.unisg.ch

% Computation of the Monte Carlo estimate and its error for an integral.
% Here we have chosen the following example:
% I = Integral (from 4 to 5) exp(x)dx.

function I = MCIntegral(n)

    I = mean(exp(rand(1,n) + 4));   % Integral by Monte Carlo
    Std = std(exp(rand(1,n) + 4))   % Calculation of the standard deviation
    error = I - ((exp(5) - exp(4))) % Computes the error of the MC estimate

end
```

### Section 1.2.1: Asset price dynamics

```
%% Geometric Brownian motion

% Author:   Roman Frey
% Date:     04.03.2008
% Contact:  roman.frey@student.unisg.ch

% This code simulates three random asset price paths which are assumed to
% follow a geometric Brownian motion. Note that this is a fairly efficient
% code since it is a so called vectorized code. The general structure of
% this code is according to Paolo Brandimarte. Numerical Methods in
% Finance. John Wiley & Sons Inc. New York, 2002, p.318-319.

function paths = GBM(S0,mu,sigma,T,m,n)

% The variable m denotes the total number of steps whereas n stands for the
% number of simulated paths. Note that increasing m and n both increase
% computation time.
```

```
    dt = T/m;
    drift = (mu-0.5*sigma^2)*dt;    % Calculation of the drift term.
    diffusion = sigma*sqrt(dt);     % Calculation of the diffusion term.
    incr = drift + diffusion*randn(n,m);
    logpaths = cumsum([log(S0)*ones(n,1),incr],2);
                                    % In a first step we calculate the so
                                    % called logpaths.
    paths = exp(logpaths);

end

%%
% After the code is written, we simulate three different asset price paths
% with the following input parameters to generate the GBM paths of Figure 2:
% S(0) = 100
% mu = 5%
% sigma = 40%
% t = 1 year
% steps m = 365, thus dt = 1/365
% number of replications n = 3

paths=GBM(100,0.05,0.4,1,365,3);
plot(1:length(paths),paths(1,:))
hold on
plot(1:length(paths),paths(2,:))
hold on
plot(1:length(paths),paths(3,:))
```

**Section 1.2.3: Pricing an Asian option by Monte Carlo**

```
%% Pricing an Asian option by Monte Carlo simulation

% Author:   Roman Frey
% Date:     08.03.2008
% Contact:  roman.frey@student.unisg.ch

% This code shows the pricing of an Asian option by Monte Carlo simulation.
% The general structure of this code is according to Paolo Brandimarte.
% Numerical Methods in Finance. John Wiley & Sons Inc. New York, 2002, p.340.
```

```
function Price = Asian(S0,K,r,sigma,T,m,n)

% The variable m denotes the total number of steps whereas n stands for the
% number of simulated paths. Note that increasing m and n both increase
% computation time.

Payoff=zeros(n,1);
for i=1:n
    paths=GBM(S0,r,sigma,T,m,1);   % According to the function 'Geometric
                                   % Brownian Motion', which was explained
                                   % in a previous section, this line
                                   % generates asset price sample paths which
                                   % follow a geometric Brownian motion.

    Payoff(i)=max(0,mean(paths(2:(m+1)))-K);
                                   % Note that it is taken the current
                                   % arithmetic mean of the asset prices to
                                   % calculate the payoff.
end
Price=exp(-r*T)*Payoff;
P=mean(Price)
end

%%
% After the code is written, we calculate the price of an Asian call option:

% Price=Asian(S0,K,r,sigma,T,m,n);
```

**Section 3.1.1: Monte Carlo and antithetic variates**

```
%% Monte Carlo and antithetic variates

% Author:   Roman Frey
% Date:     08.04.2008
% Contact:  roman.frey@student.unisg.ch

% Computing of the mean and the corresponding standard error of the
% two functions X^2 and exp(x/2) by ordinary Monte Carlo as well as
% by the variance reduction technique antithetic variates.
```

```
function h=antithetics(v,n)
k1=zeros(n,1);
k2=k1;
for i=1:n
    x=randn(v,1);
    k1(i)=mean(x.^2);
    x=randn(v/2,1); x=[x;-x];
    k2(i)=mean(x.^2);
end
[mean(k1) std(k1); mean(k2) std(k2)]

%%

for i=1:n
    x=randn(v,1);
    k1(i)=mean(exp(x/2));
    y=randn(v/2,1); x=[x;-x];
    k2(i)=mean(exp(x/2));
end
[mean(k1) std(k1); mean(k2) std(k2)]
```

**Section 3.4: Monte Carlo and antithetic variates: convergence**

```
%% Monte Carlo and antithetic variates: convergence

% Author:   Roman Frey
% Date:     22.05.2008
% Contact:  roman.frey@student.unisg.ch
%
% This code computes the Monte Carlo estimator of a vanilla European call
% option including antithetic variates.
% The general structure of this code is according to Paolo Brandimarte.
% Numerical Methods in Finance. John Wiley & Sons Inc. New York, 2002,
% p.247. However, the code has been rewritten and properly described.


function [P] = MC_Antithetic_variates(S0,K,r,T,sigma,n)

A = (r - 0.5*sigma^2)*T;    % Computation of the parameters A for the drift
```

```
B = sigma * sqrt(T);          % and B denoting the volatility.


Z = randn(n,1);               % Generation of a standard normal random vector.


payoff = max( 0 , S0*exp(A+B*Z) - K);
                              % Calculation of the payoff including ordinary
                              % standard normal random increments.


payoff_antithetic = max( 0 , S0*exp(A+B*(-Z)) - K);
                              % Calculation of the antithetic payoff
                              % including the antithetic standard normal
                              % random increments -Z.


E_antithetic = exp(-r*T) * 0.5 * (payoff+payoff_antithetic);
                              % Computation of the unbiased antithetic
                              % estimator of the option price, as it is given
                              % in Equation (3.4).

[P] = normfit(E_antithetic);
end
```

**Section 3.4: Monte Carlo and control variates: convergence**

```
%% Monte Carlo and control variates: convergence

% Author:   Roman Frey
% Date:     22.05.2008
% Contact:  roman.frey@student.unisg.ch
%
% This code computes the Monte Carlo estimator of a vanilla European call
% option including control variates.
% The general structure of this code is according to Paolo Brandimarte.
% Numerical Methods in Finance. John Wiley & Sons Inc. New York, 2002,
% p.254. However, the code has been rewritten and properly described.
```

```
function [P] = MC_Control_variates(S0,K,r,T,sigma,n,n_pilot)

A = (r - 0.5*sigma^2)*T;    % Computation of the parameters A for the drift
B = sigma * sqrt(T);        % and B denoting the volatility.


V_stock = S0*exp(A+B*randn(n_pilot,1));
                            % Value of the stock with the pilot
                            % replications as input.


V_option = exp(-r*T) * max( 0 , V_stock - K);
                            % Computation of the discounted payoff, i.e.
                            % the option value according to the stock price.


Covariance = cov(V_stock, V_option);
                            % Calculation of the covariance between the
                            % stock and the option price.


Var_CV = S0^2 * exp(2*r*T) * (exp(T * sigma^2) - 1);


b = - Covariance(1,2) / Var_CV;
                            % Since the optimal coefficient beta_star is
                            % usually unknown, we estimate the true
                            % parameter beta by running a least-squares
                            % regression (OLS) as it is done in
                            % Equation (3.10).

E_CV = S0 * exp(r*T);       % Expected value of the control variate.


V_stock_new = S0*exp(A+B*randn(n,1));
                            % In a second step, we calculate now the new
                            % stock prices with the remaining n standard
                            % normal random variables.
```

```
V_option_new = exp(-r*T) * max( 0 , V_stock_new - K);
                            % Similarly, we compute the option price with
                            % the new stock price considered.


CV = V_option_new + b * (V_stock_new - E_CV);
                            % According to Equation (3.6) we calculate the
                            % control variates with the estimator b for the
                            % true parameter beta of Equation (3.9).
[P]=normfit(CV);
end
```

**Section 3.4: Monte Carlo and importance sampling: convergence**

```
%% Monte Carlo and importance sampling: convergence

% Author:   Roman Frey
% Date:     22.05.2008
% Contact:  roman.frey@student.unisg.ch
%
% This code computes the Monte Carlo estimator of a vanilla European call
% option including importance sampling.
% The general structure of this code is according to Paolo Brandimarte.
% Numerical Methods in Finance. John Wiley & Sons Inc. New York, 2002,
% p.254. However, the code has been rewritten and properly described.


function [P] = MC_Importance_sampling(S0,K,r,T,sigma,n)

A = (r - 0.5*sigma^2)*T;    % Computation of the parameters A for the drift
B = sigma * sqrt(T);        % and B denoting the volatility.


A_is = log(K/S0) - 0.5*sigma^2*T;   % Calculation of the importance sampling
                                    % drift.

Z = randn(n,1);             % Generation of a standard normal random vector.

VY = A_is + B*n;
```

```
is_ratio = exp( (2*(A - A_is)*VY - A^2 + A_is^2)/2/B^2);


E_is = exp(-r*T)*max(0, (S0*exp(VY)-K));
                            % Computation of the unbiased importance
                            % sampling estimator of the option price.

[P] = normfit(DiscPayoff.*ISRatios);
end
```

**Section 4.2.1: Generating Halton sequences**

```
%% Halton sequences

% Author:   Roman Frey
% Date:     02.05.2008
% Contact:  roman.frey@student.unisg.ch

% This code generates Halton sequences with different bases.
% This code is according to Paolo Brandimarte. Numerical Methods in Finance.
% John Wiley & Sons Inc. New York, 2002, p.271.


function S = Halton(m,b)

% The variable m denotes the total number of Halton points that will be
% generated in the sequence. b stands for the base of the sequences.

S = zeros(m,1);     % Generating a vector S (for sequence) containing a total
                    % of m zeros.

B = 1+ceil(log(m)/log(b));
V = b.^(-(1:B));
W = zeros(1,B);
for i=1:M
   % increment last bit and carry over if necessary
   j=1;
   ok = 0;
   while ok == 0
      W(j) = W(j)+1;
```

```
      if W(j) < b
         ok = 1;
      else
         W(j) = 0;
         j = j+1;
      end
   end
   S(i) = dot(W,V);
end
```

**Section 9: Pricing Bermudan swaptions in multifactor LIBOR market model by least-squares Monte Carlo**

```
%% Pricing Bermudan swaptions in a multifactor LIBOR market model

% Author:   Roman Frey
% Date:     May, 2008
% Contact:  roman.frey@student.unisg.ch
% Source:   Most of this code stem from

%           Justin London. Modeling Derivatives Applications in Matlab,
%           C++, and Excel. Pearson Education, Inc., 2007.
%           and
%           Ahsan Amin, Pericing Bermudan fixed income derivatives in
%           multi-factor extendend LIBOR market model. Working paper, 2002.

%           However, the code has been applied with different volatility
%           scenarios and correlation structures. Moreover, the code has
%           been described in detail.


% This code computes the value of a Bermudan swaption by least-squares
% Monte Carlo in a single and multifactor LIBOR market model,
% respectively.
% Note that this code actually works for an arbitrary number of factors.
% However, remember to adjust volatility structure for each factor
% accordingly. The volatility structure may be obtained from a standard
% principal component analysis of the correlation matrix and can be
% adjusted to calibrated volatilities as it is done by Rebonato in the
% paper Riccardo Rebonato. On the simultaneous calibration of multifactor
```

```
% lognormal interest rate models to Black volatilities and to the
% correlation matrix. The Journal of Computational Finance, 2:5-27, 1999.



function [Price, CI]=Bermudan_Swaption(M, delta, T_e1, T_x1, T_s1)

% M indicates the Number of simulated paths. Note that numbers higher than
% 10000 take a long time, especially when combined with several factors.

% The parameter delta denotes the tenor spacing, which is usually chosen to
% be 6 months (0.5) or 3 months (0.25).

% T_e1 = Maturity of underlying swap in years.
% T_x1 = Last possible exercise date of the swaption.
% T_s1 = Lockout date of the swaption.
%        Note that all three quantities must be exact multiples
%        of delta.

T_e=T_e1/delta+1;   % Number of periods until maturity of the swap.
T_x=T_x1/delta+1;   % Number of periods until last possible exercise.
T_s=T_s1/delta+1;   % Number of periods until lockout date.
N=T_e;


factor=2;       % Number of relevant factors in the applied LMM. Note that
                % if you change the number of factors F you have to adjust
                % the volatility structure (see below) accordingly.


alpha=1.0;      % Constant elasticity of Variance (CEV) parameter alpha
                % for introducing a slightly curved implied volatility
                % skew or smile. Remember that changing alpha simultaneously
                % changes the effective volatility.

                % If we choose alpha=1, it corresponds to a lognormal LMM.

K=0.05;         % Strike rate in percent, i.e. a fixed coupon of a bond.

theta=+1;       % Theta indicates whether it is a payer or receiver swaption.
```

```
                % Payer swaption = +1 / Receiver swaption = -1.


L_0=0.045;      % Observed forward LIBOR rate at time T_i for the period,
                % (T_k,T_k+delta) along a given path where T_i=tT_s,...,T_x.
n_spot=2;

L=repmat(L_0,[M,T_e+1]);    % Generates the matrix L (size: M rows and
                            % T_e+1 columns) with all elements equal to
                            % L_0.

lambda=repmat(0,[T_e,factor]);  % Generates the Variance-Covariance matrix.

for n=1:N,
   for F=1:factor,
      if(F==1)
         lambda(n,F)=0.13;  % Volatility of the first factor
      end
      if(F==2)
         lambda(n,F)=(0.13-(0.009*n*delta).^0.5);
                            % Volatility of the second factor
      end
   end
end
                % You can change the volatilities according to the desired
                % volatility scenario. Note, again, that if alpha=1 then the
                % volatility of all factors remain constant.
                % Remember to add more volatility factors if you increase
                % the number of factors.

moneymarket=repmat(1,[T_x,M]);  % Generates the matrix for the money market
                                % quantities, for now all elements are 1.

Swap=repmat(0,[T_x,M]);         % Generates the matrix for the Swap rates.

B=repmat(1,[M,T_e]);            % Generates the matrix B containing only 1.


moneymarket(2,:)=moneymarket(1,:).*(1+delta*L(:,1))';
                                % Multiplies the second row of the money
```

```
                                % market matrix by (1+delta+L_0).

incr=repmat(0,[M,1]);           % Generates the vector for the increments.


drift=repmat(0,[M,factor]);     % Generates the drift matrix.


for t= 2 : T_x,t;

   normals=randn([M,factor]);   % Generates a (M,factor) matrix of standard
                                % normal random variables.
   drift(:,:)=0;

   for n= t : T_e,
      incr(:,1)=0;

      for F=1:factor,

         drift(:,F)=drift(:,F)+ delta*lambda(n-n_spot+1,F).*((L(:,n)...
             .^alpha)./(1+delta.*L(:,n)));
                                % Determination of the drift term matrix.
                                % Note that the drift in the LMM is
                                % determined by the volatility structure.
                                % Compare also with Equation (8.6).

         incr(:,1)=incr(:,1)+lambda(n-n_spot+1,F).*(L(:,n)...
             .^alpha)./L(:,n).*(normals(:,F).*sqrt(delta)-.5.*lambda...
             (n-n_spot+1,F).*(L(:,n).^alpha)./L(:,n).*delta+drift(:,F)...
             .*delta);
                                % Determination of the increments, or also
                                % called innovations. Compare with Equation
                                % (8.6).

      end

      L(:,n)=L(:,n).*exp(incr(:,1));    % Computation of the LIBOR rate
                                        % paths.

      L(L(:,n)<.00001,n)=.00001;        % In order to avoid negative numbers.
```

```
   end

   B(:,t)=1.0;  % Generating the bond vector.

   for n=t+1:T_e,

      B(:,n)=B(:,n-1)./(1+delta.*L(:,n-1));
                                % Discounted bond prices.
   end

   moneymarket(t+1,:)=moneymarket(t,:).*(1+delta*L(:,n_spot))';
                                % Compounded money market.

   if((t>= T_s) & (t <=T_x))    % Check whether t lies between the lockout
                                % and the last possible exercise date.

      for n=t:(T_e-1),          % The swap leg is determined one date
                                % before the end.

                  Swap(t,:)=Swap(t,:)+(B(:,n+1).*(L(:,n)-K).*theta*delta)';
                                % Swap rate is updated by the difference
                                % between the current LIBOR and the strike
                                % rate, multiplied by delta and by +1 or -1
                                % for payer and receiver swaptions,
                                % respectively.
      end

   end

   n_spot=n_spot+1;

end

V=repmat(0,[M,1]);          % Generation of a value vector.

stop_rule=repmat(T_x,[M,1]);    % Introduction of a stop rule vector that
                                % denotes the optimal stopping time along a
                                % given path of the LIBOR rate process. All
                                % elements of the stop rule vector are,
```

```
                                % first, set equal to the final exercise
                                % time, i.e. the remaining periods, T_x,
                                % until the last exercise date.


V(Swap(T_x,:)>0,1) = (Swap(T_x,Swap(T_x,:)>0))';

C=repmat(0,[T_x,6]);            % C denotes the matrix for the necessary
                                % coefficients.

for t=(T_x-1):-1:T_s,
   i=0;
   a=0;
   y=0;
   for p=1:M,
      if (Swap(t,p)> 0.0)       % Determination of the coefficients.
         i=i+1;
         a(i,1)=1;
         a(i,2)=Swap(t,p);
         a(i,3)=Swap(t,p)*Swap(t,p);
         a(i,4)=moneymarket(t,p);
         a(i,5)=moneymarket(t,p)*moneymarket(t,p);
         a(i,6)=moneymarket(t,p)*Swap(t,p);

         y(i,1)= moneymarket(t,p)/moneymarket(stop_rule(p,1),p) * V(p,1);
                                % Now we work backward and run a regression
                                % of the basis functions of the state
                                % variables.
      end

   end

   beta=inv(a'*a)*(a'*y);       % Regression coefficients are calculated by
                                % an ordinary least-squares (OLS)
                                % regression.

   C(t,:)=beta';


   E_cont_V=repmat(0,[M,1]);    % Vector of expected continuation value is,
```

```
                              % first, set equal to zero.

  Exe_V=repmat(0,[M,1]);      % Vector of exercise value is, first, set
                              % equal to zero.

  E_cont_V(:,1)=(C(t,1)+C(t,2).*Swap(t,:)+C(t,3).*Swap(t,:)...
     .*Swap(t,:)+C(t,4).*moneymarket(t,:)+C(t,5).*moneymarket(t,:)...
     .*moneymarket(t,:)+C(t,6).*moneymarket(t,:).*Swap(t,:))';
                              % Calculation of the fitted values. Recall
                              % that these fitted values directly
                              % correspond to the conditional expected
                              % payoff under the assumption that the
                              % option is kept alive.Therefore, the
                              % resulting value is called the expected
                              % continuation value.

Exe_V(Swap(t,:)>0,1)=(Swap(t,Swap(t,:)>0))';
                              % Definition of the exercise value.


 V((Exe_V(:,1)>E_cont_V(:,1))&(Swap(t,:)>0)',1)...
    =Exe_V((Exe_V(:,1)> E_cont_V(:,1))&(Swap(t,:)>0)',1);
                              % Now the continuation value is compared
                              % with the immediate-exercise value. If the
                              % later is greater, the option will be
                              % immediately exercised. Thus, the present
                              % time represents the optimal stopping
                              % time. The value V is set equal to
                              % immediate-exercise value.


 stop_rule((Exe_V(:,1)>E_cont_V(:,1))&(Swap(t,:)>0)',1)=t;
                              % This stop rule checks firstly whether it
                              % is advisable to exercise the option, i.e.
                              % the exercise value is larger than the
                              % continuation value and secondly, whether
                              % the swap rate is indeed positive.
                              % If the immediate-exercise is greater than
                              % the continuation value, the present time
                              % is indeed the optimal stopping time.
```

```
                                % Therefore, the optimal stopping time is
                                % automatically changed from T to t.

end

   P=0;         % Price of our Bermudan swaption is, first, set equal to zero.

   for p=1:M,
      P=P + (V(p,1)/(moneymarket(stop_rule(p,1),p)))/M;
                % The former steps are repeated for all T_x-1<=t_i<T_s
                % until the first exercise time of the Bermudan swaption is
                % eventually reached.

   end

   [Price, CI]=normfit(P)
end
```

## References


[1] Ahsan Amin. Pricing Bermudan fixed income derivatives in multi-factor extended LIBOR market model. *Working Paper*, (`http://www.geocities.com/anan2999/`), 2002.

[2] Leif Andersen. A simple approach to the pricing of Bermudan swaptions in the multi-factor Libor market model. *Journal of Computational Finance*, 3:5–32, 2000.

[3] Leif Andersen and Jesper Andreasen. Volatility skews and extensions of the Libor market model. *Applied Mathematical Finance*, 7(1):1–32, March 2000.

[4] Leif Andersen and Jesper Andreasen. Factor dependence of Bermudan swaption prices: Fact or fiction? *Journal of Financial Economics*, 61(1):3–37, October 2001.

[5] I.A. Antonov and V.M. Saleev. An economic method of computing $LP_\tau$ sequences. *USSR Journal of Computational Mathematics and Mathematical Physics*, 19:252–256, 1979.

[6] Kerry Back. *A Course in Derivative Securities*. Springer-Verlag Inc., Berlin Heidelberg, 2005.

[7] Fischer Black. The pricing of commodity contracts. *Journal of Financial Economics*, 3(1-2):167–179, January-March 1976.

[8] Fischer Black, Emanuel Derman, and William Toy. A one-factor model of interest rates and its application to treasury bond options. *Financial Analysts Journal*, 46(1):33–39, January/February 1990.

[9] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, May/June 1973.

[10] George E. P. Box and Mervin E. Muller. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 29:610–611, January 1958.

[11] Phelim Boyle. Options: A Monte Carlo approach. *Journal of Financial Economics*, 4(3):323–338, May 1977.

[12] Phelim Boyle, Mark Broadie, and Paul Glassserman. Monte Carlo methods for security pricing. *The Journal of Economic Dynamics and Control*, 21:1267–1321, 1997.

[13] Alan Brace, Dariusz Gatarek, and Marek Musiela. The market model of interest rate dynamics. *Mathematical Finance*, 7(2):127–147, April 1997.

[14] Paolo Brandimarte. *Numerical Methods in Finance*. John Wiley & Sons Inc., New York, 2002.

[15] Paolo Brandimarte. *Numerical Methods in Finance and Economics*. John Wiley & Sons Inc., New Jersey, second edition, 2006.

[16] Nicole Branger and Christian Schlag. *Zinsderivate*. Springer-Verlag Inc., Berlin Heidelberg, 2004.

[17] Paul Bratley and Bennett L. Fox. Algorithm 659: Implementing Sobol's quasirandom sequence generator. *ACM Transactions on Modeling and Computer Simulation*, 2(3):195–213, 1988.

[18] Paul Bratley, Bennett L. Fox, and Harald Niederreiter. Implementation and tests of low-discrepancy sequences. *ACM Transactions on Mathematical Software*, 14(1):88–100, 1988.

[19] Leo Breiman. *Probability*. Addison-Wesley, Reading, Massachusetts, 1968.

[20] Damiano Brigo and Fabio Mercurio. *Interest Rate Models - Theory and Practice*. Springer-Verlag Inc., Berlin Heidelberg, second edition, 2006.

[21] Mark Broadie and Paul Glassserman. Pricing American-style securities using simulation. *The Journal of Economic Dynamics and Control*, 21(8-9):1323–1352, June 1997.

[22] Georges-Louis Leclerc Buffon. Editor's note concerning a lecture given 1733 by Mr. Leclerc de Buffon to the Royal Academy of Sciences in Paris. *Histoire de l'Académie royale des sciences avec les mémoires de mathématique et de physique tirés des registres de cette Académie*, pages 43–45, 1733.

[23] Georges-Louis Leclerc Buffon. Essai d'arithmétique morale. *Histoire naturelle, générale et particulière*, Supplément 4:46–123, 1777.

[24] Andrew J. G. Cairns. *Interest Rate Models - An Introduction*. Princeton University Press, New Jersey, 2004.

[25] René Carmona and Michael Tehranchi. *Interest Rate Models: An Infinite Dimensional Stochastic Analysis Perspective*. Springer-Verlag Inc., Berlin Heidelberg, 2006.

[26] Andrew Carverhill. When is the short rate Markovian? *Mathematical Finance*, 4:305–312, October 1994.

[27] Andrew Carverhill. A simplified exposition of the Heath, Jarrow, and Morton model. *Stochastics Reports*, 53:227–240, 1995.

[28] Emmanuelle Clément, Damien Lamberton, and Philip Protter. An analysis of a least squares regression method for American option pricing. *Finance and Stochastic*, 6(4):449–471, October 2002.

[29] Les Clewlow and Chris Strickland. *Implementing Derivative Models*. Wiley, Chichester, 1998.

[30] R. R. Coveyou and R. D. MacPherson. Fourier analysis of uniform random number generators. *Journal of the ACM*, 14:100–119, 1967.

[31] John C. Cox, Jonathan E. Ingersoll Jr., and Stephen A. Ross. A theory of the term stucture of interest rates. *Econometrica*, 53(2):385–407, March 1985.

[32] John C. Cox and Stephen A. Ross. The valuation of options for alternative stochastic processes. *Journal of Financial Economics*, 3:145–166, January-March 1976.

[33] J. S. Dagpunar. *Simulation and Monte Carlo*. John Wiley & Sons Ltd., Chichester West Sussex, 2007.

[34] Lih-Yuan Deng and Dennis K. J. Lin. Random number generation for the new century. *The American Statistician*, 54(2):145–150, May 2000.

[35] Luc Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag Inc., New York, 1986.

[36] Jürgen Eichenauer and Jürgen Lehn. A non-linear congruential pseudo random number generator. *Statistische Hefte*, 27:315–326, 1986.

[37] Henri Faure. Discrépance de suites associées à un système de numération (en dimension $s$). *Acta Arithmetica*, 41:337–351, 1983.

[38] George S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag Inc., New York, 1996.

[39] George S. Fishman and Louis R. Moore. An exhaustive analysis of multiplicative congruential random number generators with modulus $2^{31} - 1$. *SIAM Journal on Scientific and Statistical Computing*, 7:24–25, 1986.

[40] Bjorn Flesaker. Testing the Heath-Jarrow-Morton/Ho-Lee model of interest rate contingent claims pricing. *Journal of Financial and Quantitative Analysis*, 28(4):483–495, December 1993.

[41] Bennett L. Fox. Implementation and relative efficiency of quasirandom sequence generators. *ACM Transactions on Mathematical Software*, 12:362–376, 1986.

[42] Herbert C. Frey and Gero Niessen. *Monte Carlo Simulation: Quantitative Risikoanalyse für die Versicherungsindustrie*. Gerling Akademie Verlag, München, 2001.

[43] Silvio Galanti and Alan Jung. Low-discrepancy sequences: Monte Carlo simulation of option prices. *The Journal of Derivatives*, 5(1):63–83, Fall 1997.

[44] Dariusz Gatarek, Przemyslaw Bachert, and Robert Maksymiuk. *The LIBOR market model in practice*. John Wiley & Sons Ltd., Chichester West Sussex, 2006.

[45] James. E. Gentle. *Random Number Generation and Monte Carlo Methods*. Springer-Verlag Inc., New York, second edition, 2003.

[46] Paul Glassserman. *Monte Carlo Methods in Financial Engineering*. Springer-Verlag Inc., New York, 2004.

[47] Peter W. Glynn and Ward Whitt. The asymptotic efficiency of simulation estimators. *Operations Research*, 40(2):505–520, May/June 1992.

[48] Irina Götsch. *Libor Market Model: Theory and Implementation*. VDM Verlag Dr. Müller, 2006.

[49] Michael Günther and Ansgar Jüngel. *Finanzderivate mit* MATLAB. Vieweg & Sohn Verlag, Wiesbaden, 2003.

[50] John H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90, 1960.

[51] Martin Haugh. Simulating stochastic differential equations. *Columbia University*, (`http://www.columbia.edu/~mh2078/index.html`), 2004.

[52] Martin Haugh. The Heath-Jarrow-Morton framework. *Columbia University*, (`http://www.columbia.edu/~mh2078/TS05/HJM_models.pdf`), 2005.

[53] David Heath, Robert Jarrow, and Andrew Morton. Bond pricing and the term structure of interest rates: A new methodology for contingent claims valuation. *Econometrica*, 60(1):77–105, January 1992.

[54] Thomas S. Y. Ho. Evolution of interest rate models: A comparison. *The Journal of Derivatives*, 2(4):9–20, 1995.

[55] Thomas S. Y. Ho and Sang-Bin Lee. Term structure movements and pricing interest rate contingent claims. *The Journal of Finance*, 41(5):1011–1029, December 1986.

[56] John C. Hull. *Options, Futures, and other Derivatives*. Pearson Prentice Hull, New Jersey, sixth edition, 2005.

[57] John C. Hull and Alan White. Pricing of interest-rate-derivative securities. *The Review of Financial Studies*, 3(4):573–592, Winter 1990.

[58] Kiyoshi Itô. On stochastic differential equations. *Memoirs of the American Mathematical Society*, 4:1–51, 1951.

[59] Peter Jäckel. *Monte Carlo methods in finance*. John Wiley & Sons Ltd., Chichester West Sussex, 2002.

[60] Jessica James and Nick Webber. *Interest Rate Modelling*. John Wiley & Sons Ltd., Chichester West Sussex, 2000.

[61] Farshid Jamshidian. LIBOR and swap market models and measures. *Finance and Stochastic*, 1(4):293–330, September 1997.

[62] Corwin Joy, Phelim Boyle, and Ken Seng Tan. Quasi-Monte Carlo methods in numerical finance. *Management Science*, 42(6):926–938, June 1996.

[63] A. G. Z. Kemna and A. C. F. Vorst. A pricing method for options based on average asset values. *Journal of Banking and Finance*, 14(1):113–129, March 1990.

[64] Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag Inc., Berlin Heidelberg, third edition, 1999.

[65] D. E. Knuth. *The Art of Computer Programming, Volume II: Seminumerical Algorithms*. Addison-Wesley, Reading, Massachusetts, third edition, 1998.

[66] Damien Lamberton and Bernard Lapeyre. *Introduction to Stochastic Calculus Applied to Finance*. Chapman & Hall, London, first edition, 1996.

[67] Bernard Lapeyre, Etienne Pardoux, and Remi Sentis. *Introduction to Monte-Carlo Methods for Transport and Diffusion Equations*. Oxford University Press Inc., New York, 1998.

[68] D. H. Lehmer. Mathematical methods in large-scale computing units. *Proceedings of the Second Symposium on Large Scale Digital Computing Machinery (Harvard Univerity Press, Cambridge*, pages 141–146, 1951.

[69] P.A.W. Lewis, A.S. Goodman, and J.M. Miller. A pseudo-random number generator for the system/360. *IBM Systems Journal*, 8:136–146, 1969.

[70] Francis A. Longstaff, Pedro Santa-Clara, and Eduardo S. Schwartz. Throwing away a billion dollars: the cost of suboptimal exercise strategies in the swaptions market. *Journal of Financial Economics*, 62(1):39–66, October 2001.

[71] Francis A. Longstaff and Eduardo S. Schwartz. Interest rate volatility and the term structure: a two-factor general equilibrium model. *The Journal of Finance*, 47(4):1259–1282, September 1992.

[72] Francis A. Longstaff and Eduardo S. Schwartz. Valuing american options by simulation: a simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147, Spring 2001.

[73] George Marsaglia. The structure of linear congruential generators. In *Applications of Number Theory to Numerical Analysis*, pages 249–286. Academic Press, New York, 1972.

[74] George Marsaglia and T. A. Bray. A convenient method for generating normal variables. *SIAM review*, 6(3):260–264, July 1964.

[75] James McCarry and Firooz Khosraviyani. The Buffon needle problem extended. *Texas College Mathematics Journal*, 2(2):10–14, August 2005.

[76] Don L. McLeish. *Monte Carlo Simulation and Finance*. John Wiley & Sons Inc., New Jersey, 2005.

[77] Iwan Meier. *Estimating the Term Structure of Interest Rates and the Pricing of Interest Rate Derivatives*. PhD thesis, Universität Bern, 2000.

[78] Fabio Mercurio and Juan M. Moraleda. An analytically tractable interest rate model with humped volatility. *European Journal of Operational Research*, 120(1):205–214, January 2000.

[79] Nicholas Metropolis. The beginning of the Monte Carlo method. *Los Alamos Science*, Special Issue:125–130, 1987.

[80] Nicholas Metropolis and Stanislaw Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, September 1949.

[81] Kristian R. Miltersen, Klaus Sandmann, and Dieter Sondermann. Closed form solutions for term structure derivatives with lognormal interest rates. *The Journal of Finance*, 52(1):409–430, March 1997.

[82] Christopher Z. Mooney. *Monte Carlo simulation*. Quantitative Applications in the Social Sciences. Sage Publications Inc., California, 1997.

[83] Manuel Moreno and Javier F. Navas. On the robustness of least-squares Monte Carlo (LSM) for pricing American derivatives. *Review of Derivatives Research*, 6(2):107–128, 2003.

[84] Harald Niederreiter. Point sets and sequences with low discrepancy. *Monatshefte für Mathematik*, 104:273–337, 1987.

[85] Harald Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[86] S.K. Park and K.W. Miller. Random number generators: good ones are hard to find. *Communications of the ACM*, 31:1192–1201, 1988.

[87] Spassimir H. Paskov and Joseph F. Traub. Faster valuation of financial derivatives. *Journal of Portfolio Management*, 22(1):113–120, Fall 1995.

[88] Morten Bjerregaard Pedersen. Bermudan swaptions in the LIBOR market model. *Working Paper, Financial Research Department, SimCorp A/S*, July 1999.

[89] Goran Peskir and Albert Shiryaev. *Optimal Stopping and Free-Boundary Problems*. Birkenhäuser Verlag, Basel, 2006.

[90] Raoul Pietersz. *Pricing models for Bermudan-style interest rate derivatives*. PhD thesis, Erasmus University Rotterdam, December 2005.

[91] J. F. Ramaley. Buffon's noodle problem. *The American Mathematical Monthly*, 76(8):916–918, October 1969.

[92] Riccardo Rebonato. *Interest-Rate Option models*. John Wiley & Sons Ltd., Chichester West Sussex, 1996.

[93] Riccardo Rebonato. On the simultaneous calibration of multi-factor lognormal interest rate models to Black volatilities and to the correlation matrix. *The Journal of Computational Finance*, 2:5–27, 1999.

[94] Riccardo Rebonato. *Modern Pricing of Interest-Rate Derivatives*. Princeton University Press, New Jersey, 2002.

[95] Peter Ritchken and L. Sankarasubramanian. Volatility structures of forward rates and the dynamics of the term structure. *Mathematical Finance*, 5:55–72, 1995.

[96] L. C. G. Rogers. Monte Carlo valuation of American options. *Mathematical Finance*, 12(3):271–286, July 2002.

[97] Jakob Sidenius. LIBOR market model in practice. *The Journal of Computational Finance*, 3(3):5–26, Spring 2000.

[98] Ilya M. Sobol. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Journal of Computational Mathematics and Mathematical Physics*, 7(4):86–102, 1967.

[99] Michael Stein. Large sample properties of simulations using Latin hypercube sampling. *Technometrics*, 29(2):143–151, May 1987.

[100] Nassim Nicholas Taleb. *Fooled by Randomness - The Hidden Role of Chance in Life and in the Markets*. Penguin Books ltd., London, 2004.

[101] Shu Tezuka. Financial Applications of Monte Carlo and Quasi-Monte Carlo Methods. In Peter Hellekalek and Gerhard Larcher, editors, *Random and Quasi-Random Point Sets*, pages 303–332. Springer-Verlag Inc., New York, 1998.

[102] Oldrich Vasicek. An equilibrium characterization of the term structure. *Journal of Financial Economics*, 5:177–188, August 1977.

[103] Marno Verbeek. *A Guide to Modern Econometrics*. John Wiley & Sons Ltd., Chichester West Sussex, second edition, 2004.

[104] John von Neumann. Various techniques used in connection with random digits. *Applied Mathematics Series*, 12, 1951.

**Declaration of authorship**

I hereby declare

- that I have written this thesis without any help from others and without the use of documents and aids other than those stated above,
- that I have mentioned all used sources and that I have cited them correctly according to established academic citation rules,
- that I shall not pass on any copies of this thesis to any third parties without the President's consent, with the exception of fellow students or persons who have provided me with essential information for this thesis, to whom I may pass on copies of this thesis after the procedure has been concluded.

July 21, 2008 / Roman Frey