Computational Finance

Christian Bayer

July 15, 2010

Contents

Chapter 1

Introduction

One of the goals in mathematical finance is the pricing of derivatives such as options. While there are certainly also many other mathematically and computationally challenging areas of mathematical finance (such as portfolio optimization or risk measures), we will concentrate on the problems arising from option pricing. The techniques presented in this course are also often used in computational finance in general, as well as in many other areas of applied mathematics, science and engineering.

The most fundamental model of a financial market consists of a probability space $(\Omega, \mathcal{F}, P)$, on which a random variable S is defined. In the simplest case, S is $\mathbb{R}$ (or $[0, \infty[$) valued and simply means the value of a stock at some time T. However, S might also represent the collection of all stock prices S_t for $t \in [0, T]$. Then S is a random variable taking values in the (infinite-dimensional) path space, i.e., either the space of continuous functions $C([0, T]; \mathbb{R}^d)$ or the space of càdlàg functions $D([0, T]; \mathbb{R}^d)$ taking values in $\mathbb{R}^d$. Then the payoff function of almost any *European option* can be represented as $f(S)$ for some functional f.

Example 1.1. The European call option (on the asset S^1) is given by

$$f(S) = \left(S_T^1 - K\right)^+.$$

Example 1.2. An example of a look-back option, consider the contract with payoff function

$$f(S) = \left(S_T^1 - \min_{t\in[0,T]} S_t^1\right)^+.$$

Example 1.3. A simple barrier option (down-and-out) could look like this (for the barrier $B > 0$):

$$f(S) = \left(S_T^1 - K\right)^+ \mathbf{1}_{\min_{t\in[0,T]} S_t^1 > B}.$$

In all these cases, the problem of pricing the option can therefore be reduced to the problem of computing

$$E\left[f(S)\right]. \tag{1.1}$$

Indeed, here we have assumed that we already started with the (or a) risk neutral measure P. Moreover, if the interest rate is deterministic, then discounting is trivial. For stochastic interest rates, we may assume that the stochastic interest rate is a part of S (depending on the interest rate model, this might imply that the state space

of the stochastic process S_t is infinite-dimensional, if we use the Heath-Jarrow-Morton model, see [16]). Therefore, the option pricing problem can still be written in the form (1.1) in the case of stochastic interest rates by incorporating the discount factor in the "payoff function" f.

Of course, we have to assume that $X := f(S) \in L^1(\Omega, \mathcal{F}, P)$. Then the most general form of the option pricing problem is to compute $E[X]$ for an integrable random variable X. Corresponding to this extremely general modeling situation is an extremely general numerical method called *Monte-Carlo simulation*. Assume that we can generate a sequence $(X_i)_{i\in\mathbb{N}}$ of independent copies of X.[1] Then, the strong law of large numbers implies that

$$\frac{1}{M}\sum_{i=1}^{M} X_i \xrightarrow[M\to\infty]{} E[X] \tag{1.2}$$

almost surely. Since the assumptions of the Monte-Carlo simulations are extremely weak, we should not be surprised that the rate of convergence is rather slow: Indeed, we shall see in Section 2.2 that the error of the Monte-Carlo simulation decreases only like $\frac{1}{\sqrt{M}}$ for $M \to \infty$ in a certain sense – note that the error will be random. Nevertheless, Monte-Carlo simulation as a very powerful numerical method, and we are going to discuss it together with several modifications in Chapter 2.

While the assumption that we can generate samples from the distribution of S might seem innocent, it poses problems in many typical modeling situations, namely when S is defined as the solution of a *stochastic differential equation* (SDE). Let $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in[0,T]}, P)$ be a filtered probability space satisfying the usual conditions. In many models, the stock price S_t is given as solution of an SDE of the form

$$dS_t = V(S_t)dt + \sum_{i=1}^{d} V_i(S_t)dB_t^i, \tag{1.3}$$

where $V, V_1, \ldots, V_d : \mathbb{R}^n \to \mathbb{R}^n$ are vector fields and B denotes a d-dimensional Brownian motion. (If we replace the Brownian motion by a Lévy process, we can also obtain jump-processes in this way.) In general, it is not possible to solve the equation (1.3) explicitly, thus we do not know the distribution of the random variable $X = f(S)$ and cannot sample from it. In Chapter 3 we are going to discuss how to solve SDEs in a numerical way, in analogy to numerical solvers for ODEs (ordinary differential equations). Then, the option price (1.1) can be computed by a combination of the numerical SDE-solver (producing samples from an approximation of $f(S)$) and the Monte-Carlo method (1.2) (applied to those approximate samples).

If the option under consideration is "Markovian" in the sense that the payoff function only depends on the value of the underlying at time T, i.e., the payoff is given by $f(S_T)$, then the option price satisfies a partial differential equation (PDE).[2] Indeed, let

$$u(s,t) = E\left[f(S_T)|S_t = s\right],$$

and define the partial differential operator L by

$$Lg(s) = V_0 g(s) + \frac{1}{2}\sum_{i=1}^{d} V_i^2 g(s),$$

[1] By this statement we mean that we have a random number generator producing (potentially infinitely many) random numbers according to the distribution of X, which are independent of each other.

[2] In fact, we can find such PDEs in much more general situations!

$s \in \mathbb{R}^n$, where the vector field V is applied to a function $g : \mathbb{R}^n \to \mathbb{R}$ giving another function $Vg(s) := \nabla g(s) \cdot V(s)$ from $\mathbb{R}^n$ to $\mathbb{R}$ and $V_i^2 g(s)$ is defined by applying the vector field V_i to the function $V_i g$. Moreover, we have

$$V_0(x) := V(x) - \frac{1}{2} \sum_{i=1}^{d} DV_i(x) \cdot V_i(x),$$

with DV denoting the Jacobian matrix of the vector field V. Then we have (under some rather mild regularity conditions)

$$\begin{cases} \dfrac{\partial}{\partial t} u(t,s) + Lu(t,s) = 0, \\ u(T,s) = f(s). \end{cases} \tag{1.4}$$

Therefore, another approach to solve our option pricing problem in a numerical way is to use the well-known techniques from numerics of PDEs, such as the finite difference or finite element methods. We will present the finite difference method in Section 4.1. We note that a similar partial differential equation also holds when the SDE is driven by a Lévy process. Then the partial differential operator L is non-local, i.e., there is an integral term. Note that there are also finite difference and finite element schemes for the resulting partial integro-differential equations, see [8] and [29], respectively.

There is a very fast, specialized method for pricing European call options (and certain similar options) on stocks S_T, such that the characteristic function of $\log(S_T)$ is known (we take S_T to be one-dimensional). This condition is actually satisfied in quite a large class of important financial models. Let ϕ_T denote the characteristic function of $\log(S_T)$ and let $C_T = C_T(K)$ denote the price of the European call option with strike price K. Moreover, we denote its Fourier transform by $\hat{C}_T$. Then

$$\hat{C}_T(\mu) = \frac{\phi_T(\mu - i)}{i\mu - \mu^2},$$

i.e., we have an explicit formula for the Fourier transform of the option price.[3] Now we only need to compute the inverse Fourier transform, which is numerically feasible because of the FFT-algorithm.

Unfortunately, most options encountered in practise are American options, and the above treated methods do not directly apply for American options. Indeed, the pricing problem for an American option is to find

$$\sup_{\tau \leq T} E\left[f(S_\tau)\right], \tag{1.5}$$

where τ ranges through all stopping times in the filtered probability space. So, it is not obvious how to apply any of the methods presented above. We will discuss one numerical method for American options in detail and hint at some modifications of the standard methods suitable for computing prices of American options, see Section 2.4

The book of Glasserman [16] is a wonderful text book on Monte Carlos based methods in computational finance, i.e., it covers Chapter 2 and Chapter 3 in great detail. On the other hand, Seydel [36] does also treat Monte Carlo methods, but concentrates more on finite difference and element methods. Wilmott [41] is a very popular, easily

[3] For integrability reasons, the above formula is not true. Indeed, we have to dampen the option price, introducing a damping parameter. For the precise formulation, see Section 4.2.

accessible book on quantitative finance. It covers many of the topics of the course, but the level of mathematics is rather low. For the prerequisites in stochastic analysis, the reader is referred to Øksendal [32] for an introduction of SDEs driven by Brownian motion. Cont and Tankov [7] is the text book of choice for Lévy processes, and Protter [34] treats stochastic integration and SDEs in full generality.

Chapter 2

Monte Carlo simulation

2.1 Random number generation

The key ingredient of the Monte Carlo simulation is sampling of independent realizations of a given distribution. This poses the question of how we can obtain such samples (on a computer). We will break the problem into two parts: First we try to find a method to get independent samples from a *uniform distribution* (on the interval $]0, 1[$, then we discuss how to get samples from general distributions provided we know how to sample the uniform distribution.

Uniform pseudorandom numbers

Computers do not know about randomness, so it is rather obvious that we cannot get truly random numbers if we trust a computer to provide them for us. Therefore, the numbers produced by a random number generator on a computer are often referred to as *pseudorandom numbers*. If the random numbers, say, $u_1, u_2, \ldots$ produced by a random number generator, are not random (but deterministic), they cannot really be realizations of a sequence $U_1, U_2, \ldots$ of independent, uniformly distributed random variables. So what do we actually mean by a random number generator? More precisely, what do we mean by a *good* random number generator?

Remark 2.1. Even though the questions raised here are somehow vague, they are really important for the success of the simulation. Bad random number generators can lead to huge errors in your simulation, and therefore must be avoided. Unfortunately, there are still many bad random number generators around. So you should rely on "standard" random number generators which have been extensively tested. In particular, you should not use a random number generator of your own. Therefore, the goal of this section is not to enable you to construct and implement a random number generator, but rather to make you aware of a few issues around random number generation.

Before coming back to these questions, let us first note that a computer usually works with finite arithmetic. Therefore, there is only a finite number of floating point numbers which can be taken by the stream random numbers $u_1, u_2, \ldots$. Therefore, we can equivalently consider a random string of integers $i_1, i_2, \ldots$ taking values in a set $\{0, \ldots, m\}$ with $u_l = i_l/m$.[1] Then the uniform random number generator producing

[1] Integer is here used in its mathematical meaning not in the sense of a data type.

$u_1, u_2, \ldots$ is good, if and only if the the random number generator producing $i_1, i_2, \ldots$ is a good random number generator for the uniform distribution on $\{0, 1, \ldots, m-1\}$. Of course, this trick has not solved our problems. For the remainder of the session, we study the problem of generating random numbers $i_1, i_2, \ldots$ on a finite set $\{0, 1, \ldots, m-1\}$.

Formally, a random number generator can be defined like this (see L'Ecuyer [25]):

Definition 2.2. A random number generator consist of a finite set X (the *state space*), an element $x_0 \in X$, (the *seed*), a *transition function* $T : X \to X$, and a function $G : X \to \{0, \ldots, m-1\}$. Given a random number generator and a seed x_0, the pseudorandom numbers are computed via the recursion $x_l = T(x_{l-1})$, $l = 1, 2, \ldots$, and $i_l := G(x_l)$.

There is an immediate (unfortunate) consequence of the definition: since X is finite, the sequence of random numbers (i_l) must be periodic. Indeed, there must be an index ℓ such that $x_\ell = x_l$ for some $l < \ell$. This implies that $x_{\ell+1} = x_{l+1}$ and so forth. Note that this index ℓ can occur much later than the first occurrence of $i_k = i_{k'}$ for some $k' < k$! Nonetheless, it arguably contains all possible candidates for good random number generators.

The following criteria for goodness have evolved in the literature on random number generators ([25],[16]):

Statistical uniformity: the sequence of random numbers $i_1, i_2, \ldots$ produced by the generator for a given seed should be hard to distinguish from truly random samples (from the uniform distribution on $\{0, \ldots, m-1\}$). This basically means that no *computationally feasible* statistical test for uniformity should be able to distinguish $(i_l)_{l \in \mathbb{N}}$ from a truly random sample. The restraint to computationally feasible tests is important: since we know that the sequence is actually deterministic (even periodic), it is easy to construct tests which can make the distinction. (The trivial test would be to wait for the period: then we see that the pseudorandom sequence repeats itself.)[2]

Speed: In modern applications, a lot of random numbers are needed. In molecular dynamics simulations, up to 10^{18} random numbers might be used (during several months of computer time). Often, the generation of random numbers is the bottleneck during a simulation. Therefore, it is very important that the RNG is fast.

Period length: If we need 10^{18} random numbers, then the period length of the RNG must be at least as high. In fact, usually the quality of randomness deteriorates well below the actual period length. As a rule of thumb it has been suggested that the period length should be an order of magnitude larger than the square of the number of values used ([35]).

Reproducibility: For instance for debugging code it is very convenient to have a way of exactly reproducing a sequence of random number generated before. (By setting the seed this is, of course, possible for any RNG satisfying Definition 2.2.)

[2]This condition basically means that we cannot guess the next number i_{l+1} given only the previously realized numbers $i_1, \ldots, i_l$, at least not better than by choosing at random among $\{0, \ldots, m-1\}$, if we assume that *we do not know the algorithm*. There is a stronger notion of *cryptographic security* which requires that we cannot guess i_{l+1} even if we are intelligent in the sense that we do know and use the RNG. In essence this means that we cannot compute the state x_l from $i_1, \ldots, i_l$. While this property is essential in cryptography, it is not important for Monte Carlo simulations.

Portability and jumping ahead: The RNG should be portable to different computers. By "jumping ahead" we mean the possibility to quickly get the state x_{l+n} given the state x_l for n large (i.e., without having to generate all the states inbetween). This is important for parallelization.

How do RNGs implemented on the computer actually look like? The prototypical class of RNGs are *linear congruential generators*. There, we have $X = \{0, \dots, m-1\}$ and $x_l = i_l$ and the transition map is given by

$$x_{l+1} = (ax_l + c) \mod m. \tag{2.1}$$

Remark 2.3. Linear congruential generators are very well analyzed from a theoretical point of view, see Knuth [22]. For instance, we know that the RNG (2.1) has full period (i.e., the period length is m) if $c \neq 0$ and the following conditions are satisfied:

- c and a are relatively prime,
- every prime number dividing m also divides $a - 1$,
- if m is divisible by four then so is $a - 1$.

Source	m	a	c
Numerical Recipes	2^{32}	1664525	1013904223
glibc (GCC)	2^{32}	1103515245	12345
Microsoft C/C++	2^{32}	214013	2531011
Apple Carbonlib	$2^{31} - 1$	16807	0

Table 2.1: List of linear congruential RNGs as reported in [40]

Table 2.1 has a list of linear congruential RNGs used in prominent libraries. Note that $m = 2^{32}$ is popular, since computing the remainder of a power of 2 in base-2 only means truncating the representation.

We end the discussion by pointing out a common weakness of all linear congruential RNGs. Set $d \geq 1$ and consider the sequence of vectors $(i_l, i_{l+1}, \dots, i_{l+d-1})$ indexed by $l \in \mathbb{N}$. Note that for every l the truly random vector $(I_l, \dots, I_{l+d-1})$ is uniformly generated on the set $\{0, \dots, m-1\}^d$. On the other hand, the pseudorandom vectors generated by linear congruential RNGs fail in that regard: they tend to lie on a (possibly) small number of hyperplanes in the hypercube $\{0, \dots, m-1\}^d$, see Figure 2.1. It has been proved that they can at most lie on $(d!m)^{1/d}$ hyperplanes, but often the actual figure is much smaller.

Finally, we would like to mention one of the most popular modern random number generators: the Mersenne Twister (available on http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html). This RNG produces 32-bit integers, the state space is $\mathbb{F}_2^{32\times 624}$, where $\mathbb{F}_2$ denotes the finite field of size two, the period is $2^{19937} - 1$. It is not a linear congruential RNG, but the basis of the transformation map T is a linear map in X – with additional transformations, though.

Non-uniform random numbers

In many applications, we do not need uniform random numbers, but random numbers from a certain distribution. For instance, the Black-Scholes model represents the stock

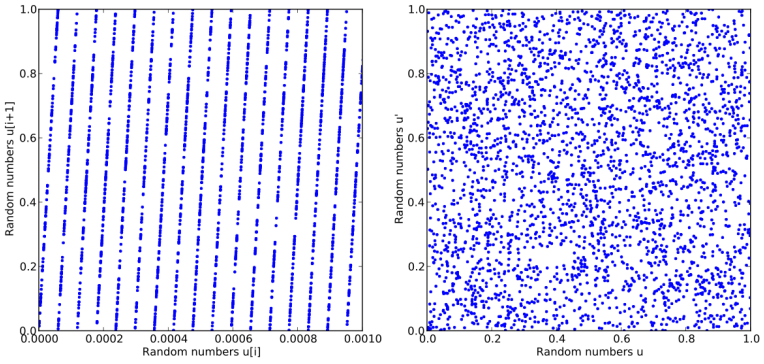

Figure 2.1: Hyperplane property for the linear congruential generator with $a = 16807$, $c = 0$, $m = 2^{31} - 1$. On the left, we have plotted 2 000 000 points (u_i, u_{i+1}), on the right 3000 pairs (i.e., 6000 random numbers plotted as pairs).

prize as

$$S_T = S_0 \exp\left(\sigma B_T + \left(\mu - \frac{1}{2}\sigma^2\right)T\right).$$

Therefore, the stock prize S_T has a log-normal distribution. On the other hand, B_T has a normal distribution. Thus, there are two ways to sample the stock prize: we can either sample from the log-normal or from the normal distribution.

For the rest of this section, and indeed, the whole text, we assume that we are given a perfect (i.e., truly random) RNG producing a sequence $U_1, U_2, \ldots$ of independent uniform random numbers. We will present several general techniques to produce samples from other distributions, and then some specialized methods for generating normal (Gaussian) random numbers. An exhaustive treatment of random number generation can be found in the classical book of Devroye [10].

We start with a well-known theorem from probability theory, which directly implies the first general method for random number generation.

Proposition 2.4. *Let F be a cumulative distribution function and define*

$$F^{-1}(u) := \inf\{\, x \mid F(x) \geq u \,\}.$$

Given a uniform random variable U, the random variable $X := F^{-1}(U)$ has the distribution function F.

Proof. Since by definition of F^{-1} we have $F^{-1}(u) \leq x \iff F(x) \geq u$,

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

Example 2.5. The exponential distribution with parameter $\lambda > 0$ has the distribution function $F(x) = 1 - e^{-\lambda x}$, which is explicitly invertible with $F^{-1}(u) = -\frac{1}{\lambda}\log(1-u)$. Thus, using the fact that $1 - U$ is uniformly distributed if U is, we can generate samples from the exponential distribution by

$$X = -\frac{1}{\lambda}\log(U).$$

Remark 2.6. If an explicit formula for the distribution function F but not for its inverse F^{-1} is available, we can try to use numerical inversion. Of course, this results in random numbers, which are samples from an approximation of the distribution F only. Nevertheless, if the error is small and/or the inversion can be done efficiently, this method might be competitive even if more direct, "exact" methods are available.[3] For instance, approximations of the inverse of the distribution function Φ of the standard normal distribution have been suggested for the simulation of normal random variables, see [16].

Remark 2.7. The transparent relation between the uniform random numbers $U_1, \dots, U_l$ and the transformed random numbers $X_1, \dots, X_l$ (with distribution F) underlying the inversion method allows to translate many structural properties on the level of the uniform random numbers to corresponding properties for the transformed random numbers. For instance, if we want the random numbers $X_1, \dots, X_l$ to be correlated, we can choose the uniforms to be correlated. Another example is the generation of the maximum $X^* := \max(X_1, \dots, X_l)$. Apart from the obvious solution (generating $X_1, \dots, X_l$ and finding their maximum), there are also two other possible methods for generating X^* based on the inversion method:

- Since X^* has the distribution function F^l, we can compute a sample from X^* by $(F^l)^{-1}(U_1)$. Efficiency of this method depends on the tractability of F^l.
- Let $U^* = \max(U_1, \dots, U_l)$. Then $X^* = F^{-1}(U^*)$. Since we only have to do one inversion instead of l, this method is usually much more efficient than the obvious method.

Next we present a general purpose method, which is based on the densities of the distributions involved instead of their distribution functions. More precisely, let $g : \mathbb{R}^d \to [0, \infty[$ be the density of a d-dimensional distribution, from which we can sample efficiently (by whatever method). We want to sample from another d-dimensional distribution with density f. The *acceptance-rejection method* works if we can find a bound $c \geq 1$ such that

$$f(x) \leq cg(x), \quad x \in \mathbb{R}^d.$$

Algorithm 2.8 (Acceptence-rejection method)**.** *Given an RNG producing independent samples X from the distribution with density g and an RNG producing independent samples U of the uniform distribution, independent of the samples X.*

1. *Generate one instance of X and one instance of U.*
2. *If $U \leq f(X)/(cg(X))$ return X;*[4] *else go back to 1.*

Proposition 2.9. *Let Y be the outcome of Algorithm 2.8. Then Y has the distribution given by the density f. Moreover, the loop in the algorithm has to be traversed c times on average.*

[3] We should note that many elementary functions like exp and log cannot be evaluated exactly on a computer. Therefore, one might argue that even the simple inversion situation of Example 2.5 suffers from this defect.

[4] Note that $P(g(X) = 0) = 0$.

Proof. By construction, Y has the distribution of X conditioned on $U \leq \frac{f(X)}{cg(X)}$. Thus, for any measurable set $A \subset \mathbb{R}^d$, we have

$$\begin{aligned} P(Y \in A) &= P\left(X \in A \,\middle|\, U \leq \frac{f(X)}{cg(X)}\right) \\ &= \frac{P\left(X \in A,\ U \leq \frac{f(X)}{cg(X)}\right)}{P\left(U \leq \frac{f(X)}{cg(X)}\right)}. \end{aligned}$$

We compute the nominator by conditioning on X, i.e.,

$$\begin{aligned} P\left(X \in A,\ U \leq \frac{f(X)}{cg(X)}\right) &= \int_{\mathbb{R}^d} P\left(X \in A,\ U \leq \frac{f(X)}{cg(X)} \,\middle|\, X = x\right) g(x)dx \\ &= \int_A P\left(U \leq \frac{f(x)}{cg(x)}\right) g(x)dx \\ &= \int_A \frac{f(x)}{cg(x)} g(x)dx \\ &= \frac{1}{c}\int_A f(x)dx \end{aligned}$$

On the other hand, a similar computation shows that $P\left(U \leq \frac{f(X)}{cg(X)}\right) = \frac{1}{c}$, and together we get

$$P(Y \in A) = \int_A f(x)dx.$$

Moreover, we have seen that the probability that the sample X is accepted is given by $1/c$. Since the different runs of the loop in the algorithm are independent, this implies that the expected "waiting time" is c. □

Exercise 2.10. Why can c only be larger or equal to 1? What does $c = 1$ imply?

Naturally, we want c to be as small as possible. That is, in fact, the tricky part of the endeavour. As an example, we give another method to sample normal random variables, starting from the exponential distribution, which we can sample by Example 2.5.

Example 2.11. The *double exponential distribution* (with parameter $\lambda = 1$) has the density $g(x) = \frac{1}{2}\exp(-|x|)$ for $x \in \mathbb{R}$. Let $f = \varphi$ denote the density of the standard normal distribution. Then

$$\frac{\varphi(x)}{g(x)} = \sqrt{\frac{2}{\pi}} e^{-\frac{x^2}{2}+|x|} \leq \sqrt{\frac{2e}{\pi}} \approx 1.315 =: c.$$

Exercise 2.12. Give a method for generating doubly exponential random variables – using only one uniform random number per output. Moreover, justify our bound c above.

Solution. The distribution function F of the double exponential distribution satisfies

$$F(x) = \begin{cases} \frac{1}{2}e^x, & x \leq 0, \\ 1 - \frac{1}{2}e^{-x}, & x > 0. \end{cases}$$

Thus, we can explicitly compute the inverse and get that

$$X := \begin{cases} \log(2U), & U \leq \frac{1}{2}, \\ -\log(2(1-U)), & U > \frac{1}{2}, \end{cases}$$

has the double exponential distribution.

For the bound, note that $e^{-x^2/2+|x|} \leq e^{1/2}$, since $-\frac{x^2}{2} + |x| \leq \frac{1}{2}$. □

We end the section by presenting a specific method for generating, again, standard normal random numbers. The *Box Muller method* is probably the simplest such method, although not the most efficient one. For a comprehensive list of random number generators specifically available for Gaussian random numbers, see the survey article [38].

Algorithm 2.13. *1. Generate two independent uniform randoms numbers U_1, U_2;*

2. Set $\theta = 2\pi U_2$, $\rho = \sqrt{-2\log(U_1)}$;

3. Return two independent standard normals $X_1 = \rho\cos(\theta)$, $X_2 = \rho\sin(\theta)$.

Exercise 2.14. Show that (X_1, X_2) indeed have the two-dimensional standard normal distribution.
Hint: Show that the density of the two-dimensional uniform variate (U_1, U_2) is transformed to the density of the two-dimensional standard normal distribution.

Solution: We use the transformation $(X_1, X_2) = h(U_1, U_2)$ with $h : [0,1]^2 \to \mathbb{R}^2$ defined by

$$h(u) = \begin{pmatrix} \sqrt{-2\log(u_1)}\cos(2\pi u_2) \\ \sqrt{-2\log(u_1)}\sin(2\pi u_2) \end{pmatrix}.$$

h is invertible with inverse

$$h^{-1}(x) = \begin{pmatrix} \exp\left(-\frac{1}{2}(x_1^2 + x_2^2)\right) \\ \frac{1}{2\pi}\arctan\left(\frac{x_2}{x_1}\right) \end{pmatrix}.$$

From probability theory we know that the density of (X_1, X_2) is given by the absolute value of the determinant of the Jacobian of h^{-1}, namely

$$\begin{aligned} \frac{\partial(u_1, u_2)}{\partial(x_1, x_2)} &= \begin{vmatrix} -x_1 \exp\left(-\frac{1}{2}(x_1^2 + x_2^2)\right) & -x_2 \exp\left(-\frac{1}{2}(x_1^2 + x_2^2)\right) \\ -\frac{1}{2\pi}\frac{1}{1+x_2^2/x_1^2}\frac{x_2}{x_1^2} & -\frac{1}{2\pi}\frac{1}{1+x_2^2/x_1^2}\frac{1}{x_1} \end{vmatrix} \\ &= -\frac{1}{2\pi}\exp\left(-\frac{1}{2}(x_1^2 + x_2^2)\right), \end{aligned}$$

whose absolute value is the two-dimensional Gaussian density. □

Remark 2.15. For generation of samples from the general, d-dimensional normal distribution $\mathcal{N}(\mu, \Sigma)$, we first generate a d-dimensional vector of independent standard normal variates $X = (X_1, \ldots, X_d)$ using, for instance, the Box-Muller method. Then we obtain the sample from the general normal distribution by

$$\mu + AX,$$

where A satisfies $\Sigma = AA^T$. Note that A can be obtained from Σ by Cholesky factorization.

Exercise 2.16. Implement the different methods for generating Gaussian random numbers and compare the efficiency.

2.2 Monte Carlo method

The Monte Carlo method belongs to the most important numerical methods. It was developed by giants of mathematics and physics like J. von Neumann, E. Teller and S. Ulam and N. Metropolis during the development of the H-bomb. (For a short account of the beginnings of Monte Carlo simulation see [30].) Today, it is widely used in fields like statistical mechanics, particle physics, computational chemistry, molecular dynamics, computational biology and, of course, computational finance! For a survey of the mathematics behind the Monte Carlo method see, for instance, the survey paper of Caflisch [4] or, as usual, Glasserman [16].

Error control in the Monte Carlo method

As we have already discussed in the introduction, we want to compute the quantity

$$I[f;X] \coloneqq E\left[f(X)\right], \tag{2.2}$$

assuming only that $f(X)$ is integrable, i.e., $I[|f|\,;X] < \infty$, and that we can actually sample from the distribution of X. Taking a sequence $X_1, X_2, \ldots$ of independent realizations of X, the law of large numbers implies that

$$I[f;X] = \lim_{M\to\infty} \frac{1}{M}\sum_{i=1}^{M} f(X_i), \quad P-\text{a.s.} \tag{2.3}$$

However, in numerics we are usually not quite satisfied with a mere convergence statement like in (2.3). Indeed, we would like to be able to control the error, i.e., we would like to have an error estimate or bound and we would like to know how fast the error goes to 0 if we increase M. Before continuing the discussion, let us formally introduce the Monte Carlo integration error by

$$\epsilon_M = \epsilon_M(f;X) \coloneqq I[f;X] - I_M[f;X], \qquad \text{where } I_M[f;X] \coloneqq \frac{1}{M}\sum_{i=1}^{M} f(X_i) \tag{2.4}$$

is the estimate based on the first M samples. Note that $I_M[f;X]$ is an *unbiased* estimate for $I[f;X]$ in the statistical sense, i.e., $E\left[I_M[f;X]\right] = I[f;X]$, implying $E\left[\epsilon_M(f)\right] = 0$. We also introduce the *mean square error* $E\left[\epsilon_M(f;X)^2\right]$ and its square root, the error in L^2. The *central limit theorem* immediately implies both error bounds and convergence rate provided that $f(X)$ is square integrable.

Proposition 2.17. *Let $\sigma = \sigma(f;X) < \infty$ denote the standard deviation of the random variable $f(X)$. Then the root mean square error satisfies*

$$E\left[\epsilon_M(f;X)^2\right]^{1/2} = \frac{\sigma}{\sqrt{M}}.$$

Moreover, $\sqrt{M}\epsilon_M(f;X)$ is asymptotically normal (with standard deviation $\sigma(f;X)$). i.e., for any constants $a < b \in \mathbb{R}$ we have

$$\lim_{M\to\infty} P\left(\frac{\sigma a}{\sqrt{M}} < \epsilon_M < \frac{\sigma b}{\sqrt{M}}\right) = \Phi(b) - \Phi(a),$$

where Φ denotes the distribution function of a standard normal random variable.

Proof. Using independence of the X_i and the fact that $I_M[f;X]$ is unbiased,

$$E\left[\epsilon_M^2\right] = \operatorname{var}\left(\frac{1}{M}\sum_{i=1}^{M} f(X_i)\right) = \frac{1}{M^2}\sum_{i=1}^{M}\operatorname{var}(f(X_i)) = \frac{M\operatorname{var}(f(X_1))}{M^2} = \frac{\sigma^2}{M}.$$

Asymptotic normality is an immediate consequence of the central limit theorem. □

Proposition 2.17 has two important implications.

1. The error is probabilistic: there is no deterministic error bound. For a particular simulation, and a given sample size M, the error of the simulation can be as large as you want. However, large errors only occur with probabilities decreasing in M.
2. The "typical" error (e.g., the root mean square error $\sqrt{E\left[\epsilon_M^2\right]}$) decreases to zero like $1/\sqrt{M}$. In other words, if we want to increase the accuracy of the result tenfold (i.e., if we want to obtain one more significant digit), then we have to increase the sample size M by a factor $10^2 = 100$. We say that the Monte Carlo method *converges with rate* $1/2$.

Before continuing the discussion of the convergence rate, let us explain how to control the error of the Monte Carlo method taking its random nature into account. The question here is, how do we have to choose M (the only parameter available) such that the probability of an error larger than a given tolerance level $\varepsilon > 0$ is smaller than a given $\delta > 0$, symbolically

$$P\left(|\epsilon_M(f;X)| > \varepsilon\right) < \delta.$$

Fortunately, this question is already almost answered in Proposition 2.17. Indeed, it implies that

$$P\left(|\epsilon_M| > \varepsilon\right) = 1 - P\left(-\frac{\sigma\tilde{\varepsilon}}{\sqrt{M}} < \epsilon_M < \frac{\sigma\tilde{\varepsilon}}{\sqrt{M}}\right) \sim 1 - \Phi(\tilde{\varepsilon}) + \Phi(-\tilde{\varepsilon}) = 2 - 2\Phi(\tilde{\varepsilon}),$$

where $\tilde{\varepsilon} = \sqrt{M}\varepsilon/\sigma$. Of course, the normalized Monte Carlo error is only asymptotically normal, which means the equality between the left and the right hand side of the above equation only holds for $M \to \infty$, which is signified by the "$\sim$"-symbol. Equating the right hand side with δ and solving for M yields

$$M = \left(\Phi^{-1}\left(\frac{2-\delta}{2}\right)\right)^2 \sigma^2\varepsilon^{-2}. \tag{2.5}$$

Thus, as we have already observed before, the number of samples depends on the tolerance like $1/\varepsilon^2$.

Remark 2.18. This analysis tacitly assumed that we know $\sigma = \sigma(f;X)$. Since we started the whole endeavour in order to compute $I[f;X]$, it is, however, very unlikely that we already know the variance of $f(X)$. Therefore, in practice we will have to replace $\sigma(f;X)$ by a sample estimate. (This is not unproblematic: what about the Monte Carlo error for the approximation of $\sigma(f;X)$?)

Exercise 2.19. Compute the price of a European call option in the Black-Scholes model using Monte Carlo simulation. Study the convergence of the error and also the asymptotic normality of the error. Then, use (2.5) for a more systematic approach.

Exercise 2.20. If we want to compute the expected value of an integrable random variable, which is not square integrable, the above analysis does not apply. Compute the expected value of $E[1/\sqrt{U}]$ for a uniform random variable U using Monte Carlo simulation. Study the speed of convergence and whether the errors are still asymptotically normal.

Remark 2.21. Let us come back to the merits of Monte Carlo simulation. For simplicity, let us assume that X is a d-dimensional uniform random variable, i.e.,

$$I[f] \coloneqq I[f;U] = \int_{[0,1]^d} f(x)dx.$$

Note that the dimension of the space did not enter into our discussion of the convergence rate and of error bounds at all. This is remarkable if we compare the Monte Carlo method to traditional methods for numerical integration. Those methods are usually based on a grid $0 \leq x_1 < x_2 < \cdots < x_N \leq 1$ of arbitrary length N. The corresponding d-dimensional grid is simply given by $\{x_1, \ldots, x_N\}^d$, a set of size N^d. The function f is evaluated on the grid points and an approximation of the integral is computed based on interpolation of the function between grid-points by suitable functions (e.g., piecewise polynomials), whose integral can be explicitly computed. Given a numerical integration method of order k, the error is the proportional to $\left(\frac{1}{N}\right)^k$. However, the we had to evaluate the function on N^d points. Therefore, the accuracy in terms of points merely is like $n^{-k/d}$, where n denotes the total number of points involved, which is proportional to the computational cost. This is known as the *curse of dimensionality*: even methods, which are very well suited in low dimensions, deteriorate very fast in higher dimensions.

The curse of dimensionality is the main reason for the popularity of the Monte Carlo method. As we will see later, in financial applications the dimension of the state space can easily be in the order of 100 (or much higher), which already makes traditional numerical integration methods completely unfeasible. In other applications, like molecular dynamics, the dimension of the state space might be in the magnitude of 10^{12}!

Variance reduction

While there are no obvious handles of how to increase the convergence rate in Proposition 2.17, we might be able to improve the constant factor, i.e., the variance $\sigma(f;X)^2 = \mathrm{var}(f(X))$. Therefore, the idea is to obtain (in a systematic way) random variables Y and functions g such that $E[g(Y)] = E[f(X)]$, but with smaller variance $\mathrm{var}(g(Y)) < \mathrm{var}(f(X))$. Inserting $\sigma(g;Y) = \sqrt{\mathrm{var}(g(Y))}$ into (2.5) shows that such an approach will decrease the computational work – proportional to the number of trajectories, provided that generation of samples $g(Y)$ is not prohibitively more expensive than generation of samples from $f(X)$.

Antithetic variates

If U has the uniform distribution, then the same is true for $1 - U$. Similarly, if $B \sim \mathcal{N}(0, I_d)$ (the d-dimensional normal distribution), then so is $-B$. Therefore, these transformations do not change the expected value $E[f(X)]$ if $X = U$ or $X = B$.[5] In gen-

[5] Since many random number generators for non-uniform distributions are based on uniform ones, we can often view our integration problem as being of this type.

eral, let us assume that we know a (simple) transformation $\widetilde{X}$ having the same law as X, such that a realization of $\widetilde{X}$ can be computed from a realization of X by a deterministic transformation. Define the *antithetic* Monte Carlo estimate by

$$I_M^A[f;X] = \frac{1}{M}\sum_{i=1}^{M}\frac{f(X_i)+f\left(\widetilde{X}_i\right)}{2}. \tag{2.6}$$

Since $E\left[(f(X_i)+f(\widetilde{X}_i))/2\right] = E[f(X)]$, this can be seen as a special case of the Monte Carlo estimate (2.3). If we assume that the actual simulation of $(f(X_i)+f(\widetilde{X}_i))/2$ takes at most two times the computer time as the simulation of $f(X_i)$, then the computing time necessary for the computation of the estimate $I_M^A[f;X]$ does not exceed the computing time for the computation of $I_{2M}[f;X]$.[6] Then the use of antithetic variates makes sense if the means square error of $I_M^A[f;X]$ is smaller than the one for $I_{2M}[f;X]$, i.e., if

$$\frac{\operatorname{var}\left(\frac{f(X_i)+f(\widetilde{X}_i)}{2}\right)}{M} < \frac{\operatorname{var}(f(X_i))}{2M}.$$

This is equivalent to $\operatorname{var}(f(X_i)+f(\widetilde{X}_i)) < 2\operatorname{var}(f(X_i))$. Since $\operatorname{var}(f(X_i)+f(\widetilde{X}_i)) = 2\operatorname{var}(f(X_i)+2\operatorname{cov}(f(X_i),f(\widetilde{X}_i))$, antithetic variates can speed up a Monte Carlo simulation iff

$$\operatorname{cov}\left(f(X),f(\widetilde{X})\right) < 0. \tag{2.7}$$

Control variates

Assume that we are given a random variable Y and a functional g such that we know the exact value of $I[g;Y] = E[g(Y)]$. (Note that we allow $Y = X$.) Then obviously

$$I[f;X] = E\left[f(X) - \lambda(g(Y) - I[g;Y])\right],$$

for any deterministic parameter λ. Thus, a Monte Carlo estimate for $I[f;X]$ is given by

$$I_M^{C,\lambda}[f;X] := \frac{1}{M}\sum_{i=1}^{M}(f(X_i)-\lambda g(Y_i)) + \lambda I[g;Y], \tag{2.8}$$

where (X_i,Y_i) are independent realizations of (X,Y). Similar to the situation with antithetic variates, we may assume that simulation of $I_M^{C,\lambda}[f]$ takes at most twice the time of simulation of $I_M[f]$, but often it does take less time than that, especially if $X = Y$. We are going to choose the parameter λ such that $\operatorname{var}(f(X)-\lambda g(Y))$ is minimized. A simple calculation gives that

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

is minimized by choosing λ to be equal to

$$\lambda^* = \frac{\operatorname{cov}(f(X),g(Y))}{\operatorname{var}(g(Y))}. \tag{2.9}$$

[6] Since we only need to sample one random number X_i and obtain $\widetilde{X}_i$ by a simple deterministic transformation, in many situations it is much faster to compute $(f(X_i)+f(\widetilde{X}_i))/2$ then to compute two realizations of $f(X_i)$.

Assuming that the computational work per realization is two times higher using control variates, (2.5) implies that the control variates technique is $1/(2(1-\rho^2))$-times faster than normal Monte Carlo, where ρ denotes the correlation coefficient of $f(X)$ and $g(Y)$. For instance, for $\rho = 0.95$, the use of the control variate improves the speed of the Monte-Carlo simulation by a factor five. In particular, the speed-up is high if $f(X)$ and $g(Y)$ are highly correlated.

Remark 2.22. We can only determine the optimal factor λ^* if we know $\mathrm{cov}(f(X), g(Y))$ and $\mathrm{var}(g(Y))$. If we are not in this highly unusual situation, we can use sample estimates instead (obtained by normal Monte Carlo simulation with a smaller sample size).

Exercise 2.23. In the setting of a Black-Scholes model consider the Asian option maturity T and payoff function

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

Moreover, consider the (artificial) geometrical-average Asian option with payoff function

$$\left(\left[\prod_{i=1}^{n} S_{t_i}\right]^{1/n} - K\right)^+.$$

a) Find an explicit formula for the geometrical Asian option (see also Glasserman [16, page 99 f.]).

b) Simulate the option price of an Asian option using normal Monte Carlo, Monte Carlo with antithetic variates and Monte Carlo with the geometrical-average Asian option as control variate. Compare the results in terms of accuracy and run-time. Finally, try to combine both variance reduction techniques. Is there a further effect?

Importance sampling

Importance sampling is somehow related to the acceptance-rejection method. The idea is to sample more often in regions, where the variance is higher. Assume that the underlying random variable X has a density p (on $\mathbb{R}^d$). Moreover, let q be another probability density. Then we can obviously write

$$I[f;X] = \int_{\mathbb{R}^d} f(x)p(x)dx = \int_{\mathbb{R}^d} f(x)\frac{p(x)}{q(x)}q(x)dx = E\left[f(Y)\frac{p(Y)}{q(Y)}\right] = I\left[f\frac{p}{q};Y\right],$$

where Y is a d-dimensional random variable with density q. Thus, a Monte Carlo estimate for $I[f]$ is given by

$$\tilde{I}_M[f;X] = \frac{1}{M}\sum_{i=1}^{M} f(Y_i)\frac{p(Y_i)}{q(Y_i)} = I_M\left[f\frac{p}{q};Y\right]. \tag{2.10}$$

As usual, a possible speed up is governed by the variance of $f(Y)\frac{p(Y)}{q(Y)}$, which is determined by

$$\mathrm{var}\left(f(Y)\frac{p(Y)}{q(Y)}\right) + I[f;X]^2 = E\left[\left(f(Y)\frac{p(Y)}{q(Y)}\right)^2\right] = E\left[f(X)^2\frac{p(X)}{q(X)}\right].$$

So how do we have to choose q? Assume for a moment that $f \geq 0$ itself. Take q proportional to $f \cdot p$. Then, the new estimator is based on the random variable

$$f(Y)\frac{p(Y)}{q(Y)} \equiv 1,$$

thus, the variance is zero! Of course, there is a catch: q needs to be normalized to one, therefore in order to actually construct q, we need to know the integral of $f \cdot p$, i.e., we would need to know our quantity of interest $I[f]$. However, we can gain some intuition on how to construct a good importance sample estimate: we should choose q in such a way that $f \cdot p/q$ is almost flat.

Conclusions

Comparing the three methods of variance reduction presented here, we see that antithetic variates are easiest to implement, but can only give a limited speed-up. On the other hand, both control variates and importance sampling can allow us to use very specific properties of the problem at hand. Therefore, the potential gain can be large (in theory, the variance can be reduced almost to zero). On the other hand, this also means that there is no general way to implement control variates or importance sampling.

2.3 Quasi Monte Carlo simulation

As we have seen, Monte Carlo simulation is a method to compute

$$I[f] \coloneqq \int_{[0,1]^d} f(x)dx \tag{2.11}$$

– in fact, by composition with the inverse of the distribution function, all the integration problems in this section were of the form (2.11). This means that we use the approximation

$$J_M[f] \coloneqq \frac{1}{M}\sum_{i=1}^{M} f(x_i), \tag{2.12}$$

where the $x_i \in [0,1]^d$ are chosen in such a way as to mimic the properties of a sequence of independent uniform random variates – but they are, in fact, still deterministic. The idea of Quasi Monte Carlo simulation is to instead choose a (deterministic) sequence $x_i \in [0,1]^d$ which are especially even distributed in $[0,1]^d$. Figure 2.2 shows samples in $[0,1]^2$ as generated from a uniform (pseudo) RNG. We can see a lot of clumping of the drawn points. This is not a sign of a bad RNG: indeed, for truly random realizations of the uniform distribution on $[0,1]^2$ we would expect a similar kind of clumping. However, it is easy to see that it should be possible to construct sequences (x_i) with much less clumping, see again Figure 2.2. So, in some sense the idea is the replace pseudo random number by "more evenly distributed" but deterministic sequences.

For more information on Quasi Monte Carlo methods, we refer to Glasserman [16] and the survey articles Caflisch [4] and L'Ecuyer [26].

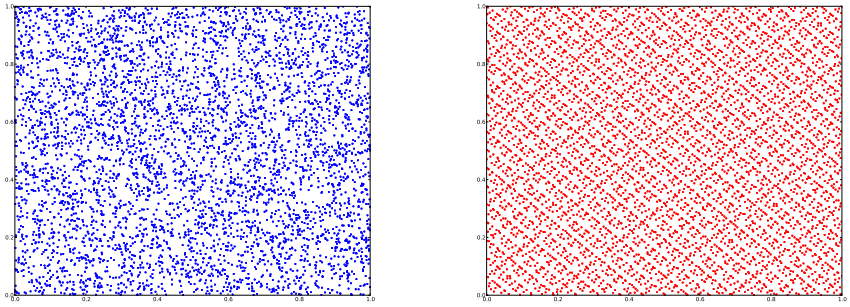

Figure 2.2: Pseudo random samples in $[0,1]^2$ (left picture) versus quasi random ones (right picture)

Discrepancy and variation

In order to proceed mathematically, we need a quantitative measure of "even distribution". This measure is provided by the notion of discrepancy. Let λ denote the restriction of the d-dimensional Lebesgue measure to the unit cube $[0,1]^d$, i.e., the law of the uniform distribution. Now consider a rectangular subset R of $[0,1]^d$, i.e., $R = [a_1, b_1[\times \cdots \times [a_d, b_d[$ for some $a_1 < b_1, \ldots, a_d < b_d$. Then for a given sequence $x_i \in [0,1]^d$ we can compare the Monte Carlo error for computing the volume of the set R using the first M elements of the sequence (x_i) and get

$$\frac{1}{M}\#\{1 \le i \le M : x_i \in R\} - \lambda(R).$$

This is the basis of the following two (supremum-norm type) definitions of discrepancy.

Definition 2.24. The *discrepancy* D_M of a sequence $(x_i)_{i\in\mathbb{N}}$ (or rather of its subsequence $(x_i)_{i=1}^M$) is defined by

$$D_M = \sup_R \left| \frac{1}{M}\#\{1 \le i \le M : x_i \in R\} - \lambda(R) \right|.$$

The *star-discrepancy* D_M^* is defined similar to D_M, but the supremum is taken over only those rectangles containing the origin $(0, \ldots, 0)$, i.e.,

$$D_M^* = \sup \left\{ \left| \frac{1}{M}\#\{1 \le i \le M : x_i \in R\} - \lambda(R) \right| \,\middle|\, R = \bigtimes_{j=1}^{d} [0, b_j[,\ b_1, \ldots, b_d \in [0,1] \right\}.$$

The quality of the quadrature rule (2.12) will depend both on the uniformity of the sequence (measured by some form of discrepancy) and the regularity of the function f. For Monte Carlo simulation, we only needed the function f to be square integrable, and the accuracy was determined by the variance $\mathrm{var}(f(X))$. Error bounds for Quasi Monte Carlo will generally require much more regularity. One typical measure of regularity is the following.

Definition 2.25. The *variation in the sense of Hardy-Krause* is recursively defined by

$$V[f] = \int_0^1 \left| \frac{df}{dx}(x) \right| dx$$

for a one-dimensional function $f : [0,1] \to \mathbb{R}$ and

$$V[f] = \int_{[0,1]^d} \left| \frac{\partial^d f}{\partial x^1 \cdots \partial x^d}(x) \right| dx + \sum_{j=1}^{d} V[f_1^{(j)}],$$

where $f_1^{(j)}$ denotes the restriction of f to the boundary $x^j = 1$, for a function $f : [0,1]^d \to \mathbb{R}$.[7]

Theorem 2.26. *For any integrable function $f : [0,1]^d \to \mathbb{R}$ the* Koksma-Hlawka inequality *holds:*

$$|I[f] - J_M[f]| \leq V[f] D_M^*.$$

Remark 2.27. The Koksma-Hlawka inequality is a deterministic upper bound for the integration error, a worst case bound. For the Monte-Carlo method, we only got probabilistic bounds (see Proposition 2.17), which could be seen as bounds for the average case. On the other hand, while the Monte-Carlo bounds are sharp, the error estimate given by the Koksma-Hlawka inequality usually is a gross over estimation of the true error. Indeed, even the basic assumption that $f \in C^d$ turns it useless for most financial applications. Fortunately, Quasi Monte Carlo works much better in practice!

In the literature, one can find other measures of variation and discrepancy, which together can give much better estimates than the Koksma-Hlawka inequality. The interested reader is referred to [26] and the references therein. Still, the good performance of Quasi Monte Carlo methods in practice seems to defy theoretical analysis.

We give the proof of the Koksma-Hlawka inequality in a special case only (the extension to the general case is left as an exercise).

Proof of Theorem 2.26 for $d = 1$. Assume that $f \in C^1([0,1])$. Then for any $0 \leq x \leq 1$ we have

$$f(x) = f(1) - \int_0^1 f'(t) \mathbf{1}_{]x,1]}(t) dt.$$

We insert this representation into the quadrature error

$$\begin{aligned}
|I[f] - J_M[f]| &= \left| \frac{1}{M} \sum_{i=1}^{M} \int_0^1 f'(t) \mathbf{1}_{]x_i,1]}(t) dt - \int_0^1 \int_0^1 f'(t) \mathbf{1}_{]x,1]}(t) dt dx \right| \\
&= \left| \int_0^1 f'(t) \left[\frac{1}{M} \sum_{i=1}^{M} \mathbf{1}_{]x_i,1]}(t) - \int_0^1 \mathbf{1}_{]x,1]}(t) dx \right] dt \right| \\
&\leq \int_0^1 \left| f'(t) \right| \underbrace{\left| \frac{1}{M} \sum_{i=1}^{M} \mathbf{1}_{[0,t[}(x_i) - \int_0^1 \mathbf{1}_{[0,t[}(x) dx \right|}_{\leq D_M^*} dt \\
&\leq V[f] D_M^*.
\end{aligned}$$

□

[7] If the integral is not defined, because the function f is not smooth enough, we set $V[f] = \infty$.

Sequences of low discrepancy

By Theorem 2.26, we need to find sequences of *low discrepancy*.

Definition 2.28. We say that a sequence $(x_i)_{i\in\mathbb{N}}$, $x_i \in [0,1]^d$, has low discrepancy, if $D_M^* \leq c\log(M)^d M^{-1}$.

We give a few examples of sequences of low discrepancy.

Example 2.29. Choose a prime number p (or more generally, an integer $p \geq 2$). Define the map $\psi_p : \mathbb{N}_0 \to [0,1[$ by

$$\psi_p(k) = \sum_{j=0}^{\infty} \frac{a_j(k)}{p^{j+1}}, \text{ where } k = \sum_{j=0}^{\infty} a_j(k)p^j.$$

The *Van der Corput sequence* is the one-dimensional sequence $x_i = \psi_p(i)$, $i \in \mathbb{N}_0$.

Example 2.30. The *Halton sequence* is a d-dimensional generalization of the Van der Corput sequence. Let $p_1, \dots, p_d$ be relatively prime integers. Define a d-dimensional sequence by $x_i = (x_i^1, \dots, x_i^d)$, $i \in \mathbb{N}_0$, with $x_i^j = \psi_{p_j}(i)$, $j = 1, \dots, d$.

Remark 2.31. When we work with RNGs, we do not have to define extra multi-dimensional RNGs. Indeed, if $(X_i)_{i\in\mathbb{N}}$ is a sequence of independent, uniform, one-dimensional random numbers, then the sequence

$$(X_{(i-1)d+1}, \dots, X_{id})_{i\in\mathbb{N}}$$

is a sequence of d-dimensional, independent, uniform random variables. On the other hand, if we take d-tuples of a one-dimensional sequence of low discrepancy, we cannot hope to obtain a d-dimensional sequence with with low discrepancy, see Figure 2.3.

Remark 2.32. Clearly, a very evenly spaced (finite) sequence is given by taking all the $(n+1)^d$ points $\left\{0, \frac{1}{n}, \frac{2}{n}, \dots, 1\right\}^d$ for some fixed $n \in \mathbb{N}$. However, we would like to have a sequence of arbitrary length: we want to compute estimates $J_M[f]$ increasing M until some stopping criterion is satisfied – and, of course, this is only feasible if updating from $J_M[f]$ to $J_M[f]$ does not require to recompute $M+1$ terms. Using the tensorized sum above, we can only compute $J_{(n+1)^d}[f]$, since $J_M[f]$ would probably give a very bad estimate for $M < (n+1)^j$ and would require recomputing the whole sum for $M > (n+1)^d$, unless we refine the grid taking $n \to 2n$, which increases M by a factor 2^d. Thus, taking a regular tensorized grid is not feasible.

Additionally, there are several other prominent families of sequences with low discrepancy, like the *Sobol* or *Faure* sequences. For a sequence of low discrepancy, the Koksma-Hlawka inequality, when applicable, implies that the quadrature error satisfies

$$|I[f] - J_M[f]| \leq \frac{V[f]c\log(M)^d}{M}, \tag{2.13}$$

i.e., the rate of convergence is given by $1-\epsilon$, as compared to the meagre $1/2$ from classical Monte Carlo simulation. This is indeed the usually observed rate in practice, however, this statemented should be treated with care: apart from the regularity assumptions of the Koksma-Hlawka inequality, let us point out that $\log(M)^d/M \gg M^{-1/2}$ for all reasonably sized M even in fairly moderate dimensions d. For instance, in dimension $d = 8$, we only have

$$\log(M)^d/M \leq M^{-1/2}, \text{ for } M \geq 1.8 \times 10^{29}.$$

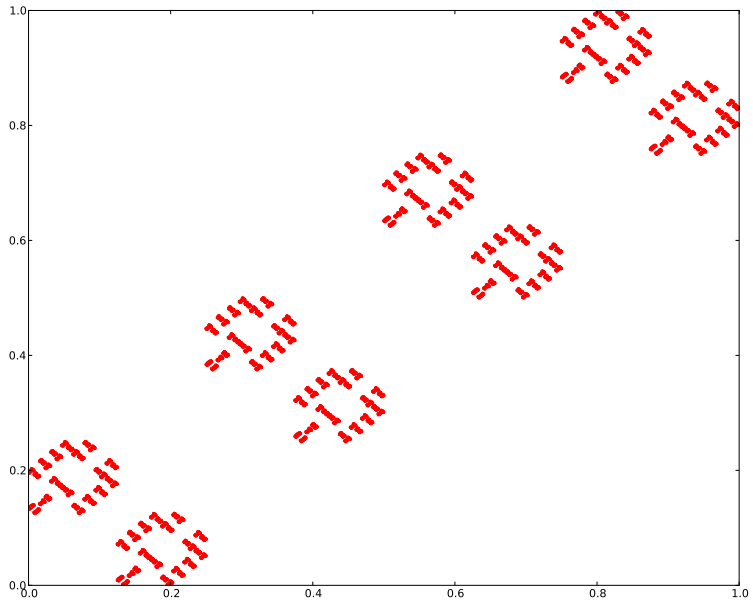

Figure 2.3: Pairs of one-dimensional Sobol numbers

Exercise 2.33. Solve the Exercises 2.19 and 2.23 using Quasi Monte Carlo. Report the results and compare the speed of convergence with the one obtained by Monte Carlo simulation.

Remarks on Quasi Monte Carlo

Low dimensionality

It is generally difficult to construct good sequences of low discrepancy in high dimensions $d \gg 1$. Indeed, even for the available sequences, it is usually true that the "level of even distribution" often deteriorates in the dimension in the sense that, e.g., the projection two the first two coordinates $(x_i^1, x_i^2)_{i\in\mathbb{N}}$ will often have better uniformity properties than the projections on the last two coordinates (x_i^{d-1}, x_i^d). Moreover, the theory suggests that functions need to be more and more regular in higher dimensions. So why does QMC work so well especially in higher dimensions?

One explanation is that many high-dimensional functionals f, especially those used in finance, often depend mostly on few dimensions, in the sense that in an ANOVA decomposition (of f into functions depending only on a few coordinates)

$$f(x^1, \dots, x^d) = \sum_{k=0}^{d} \sum_{(i_1<i_2<\cdots<i_k)\in\{1,\dots,d\}^k} f^{(i_1,\dots,i_k)}(x^{i_1}, \dots, x^{i_k})$$

the functions $f^{(i_1,\dots,i_k)}$ with big k only contribute little to the values of f. In many

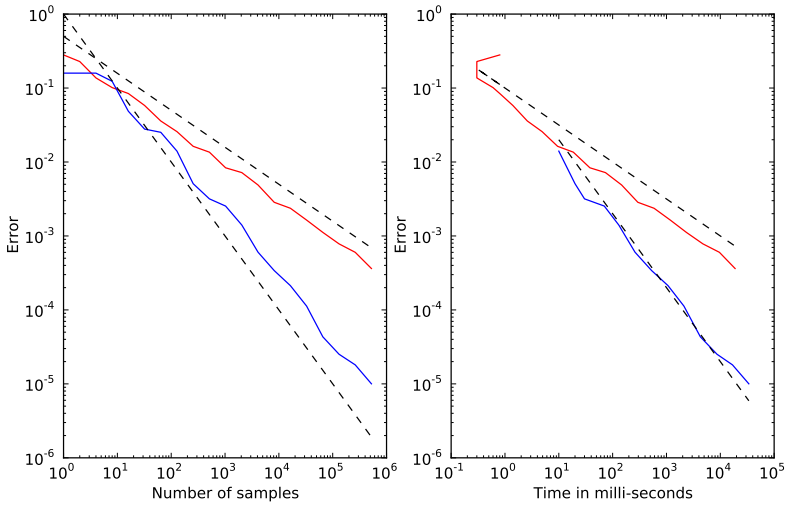

Figure 2.4: A call option in the Black-Scholes model using Monte Carlo and Quasi Monte Carlo simulation. Red: MC simulation, blue: QMC simulation, black: Reference lines proportional to $1/M$ and $1/\sqrt{M}$.

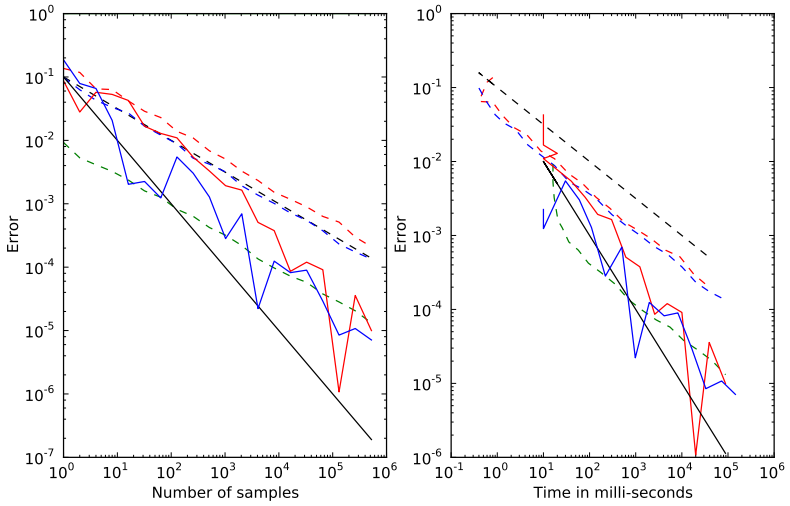

Figure 2.5: The Asian option from Exercise 2.23 using Monte Carlo and Quasi Monte Carlo simulation (Solid lines: QMC simulation, dashed lines: MC simulation; Red: normal simulation, blue: antithetic variates, green: control variates, black: references line proportional to $1/M$.

cases, the "low-dimensionality" of a function f can be improved by applying suitable transformations, thus improving the accuracy of the Quasi Monte Carlo method.

Randomized QMC

We have seen before that the QMC (Quasi Monte Carlo) method generally converges faster than plain Monte Carlo simulation, but lacks good error control. On the other hand, the Monte Carlo method allows for very good error controls (with only very little before-hand information necessary), even though these are only random. So why note combine Monte Carlo and Quasi Monte Carlo?

Let $x = (x_i)_{i\in\mathbb{N}}$ denote a sequence of low discrepancy in dimension d. We can randomize this sequence, e.g., by applying a random shift, i.e., for a d-dimensional uniform random variable U consider

$$X := (x_i + U \pmod 1)_{i\in\mathbb{N}}\,. \tag{2.14}$$

(For other possible randomizations see [26].) Let $J_M[f;X]$ denote the QMC estimate (2.12) based on the randomized sequence X. Now fix a number $m \in \mathbb{N}$ and generate m independent realizations X_l, $1 \le l \le m$, of X (by sampling m independent realizations U_l of U). Then we estimate $I[f]$ by the *randomized Quasi Monte Carlo* estimate

$$J^R_{M;m}[f] := \frac{1}{m}\sum_{l=1}^{m} J_M[f;X_l]. \tag{2.15}$$

Now we can use the error estimate of Proposition 2.17 based on $\mathrm{var}(J_M[f;X])$. By the good convergence of the QMC estimator $J_M[f]$, we can expect $J_M[f]$ to be close to $I[f]$ for most realizations X. Thus, $\mathrm{var}(J_M[f;X])$ will be small. This means, from the point of view of the Monte Carlo method, RQMC can be seen as another variance reduction technique! (L'Ecuyer [26] reports tremendous improvements of the variance as compared with plain MC or even MC with traditional variance reduction.)

Remark 2.34. How should we divide the computational work between m and M? The purpose of m is mostly to compute the error estimate, whereas M controls the error itself. Therefore, in applications m should be chosen quite small, L'Ecuyer suggests $m \le 25$. On the other hand, for theoretical purposes, e.g., for comparison of RQMC to other methods, the error control might be more important and might require higher m.

Exercise 2.35. Solve the Exercises 2.19 and 2.23 using RQMC. Report the results and the reduction in the variance.

2.4 Pricing American options with Monte Carlo

American options are fundamentally different from European options in that they allow the holder of the option to exercise it at any given time between today and the expiry date T of the option. Thus, the holder of the option needs to choose the best time to exercise the option, which mathematically translates to an optimal stopping problem. Therefore, one can show that an arbitrage-free price of the American option is given by

$$\sup_{\tau\le T} E\left[e^{-r\tau} f(X_\tau)\right], \tag{2.16}$$

where the expectation is understood under a risk neutral measure and the sup ranges over all stopping times τ bounded by T. f denotes the payoff function of the option,

e.g., $f(x) = (K - x)_+$ in the case of a put option. In most real life situations, one cannot continuously exercise the option, instead, exercising is only possible at a finite number of times $t_1, \dots, t_m$. In the case of an American option, these dates correspond to all the trading dates between today (time 0) and the maturity T. On the other hand, we can also think of options that can only be exercised once per week or at any other collection of trading days. In the literature, those options are known as *Bermudan options*, since they are lying "between" European and American options.

In the last section, we have already seen one possible way to obtain the price of an American option, namely by solving the corresponding partial differential inequality, which has the form of a free boundary problem, see (4.6). In this section, we present a Monte Carlo based method, based on random trees. We are going to work under the following basic assumptions:

1. The option can only be exercised at the times $t_1 < t_2 < \cdots < t_m = T$. (Henceforth, we will identify times with indices, i.e., we will talk about "times" $i = 0, \dots, m$.)

2. The stock price process $X_i := X_{t_i}$, $0 \le i \le m$ (with $t_0 := 0$), is a Markov chain in discrete time with state space contained in $\mathbb{R}^n$. Moreover, we assume that we can exactly sample from X. All these properties hold under a fixed martingale measure P.

3. The interest rate is equal to $r = 0$.

Remark 2.36. In the case of an American option (with $T > m$ trading days left), condition 1 means that we discretize the problem. However, in the following we are not going to discuss the approximation properties for $m \to \infty$. Instead, we assume that the option is of Bermudan form. If the stock price is modeled by an SDE, we usually do not know the distribution of X_{t_i}, see Chapter 3. Again, in this section we assume that the random variables X_i are exactly given. From the point of view of an SDE, this might mean that we treat the approximation as our true model. (We are only interested in the Monte Carlo and optimal stopping part here.) Finally, note that we can treat stochastic interest rates by enhancing the state space of X_t to include a stochastic interest r_t and changing the payoff.

In what follows, we closely follow Glasserman [16, Chapter 8]. We also refer to the original papers of Broadie and Glasserman, e.g., [3], and to Longstaff and Schwartz [28].

Dynamic programming and stopping rules

Let $f_i(x)$ denote the payoff function of the option at time i for the stock price x and let $V_i(x)$ the value of the option at time i given $X_i = x$. (We implicitly assume that the option has not been exercised before time i.) Obviously, we have $V_m(x) = f_m(x)$. By *dynamic programming* (*Bellman's equation*), we have the backward recursion

$$V_{i-1}(x) = \max\left(f_{i-1}(x),\, E[V_i(X_i)|X_{i-1} = x]\right), \quad i = 1, \dots, m. \tag{2.17}$$

Since we know $V_m(x) = f_m(x)$, (2.17) determines the price $V_0(x)$ of the option, but requires us to compute conditional expectations at each step. Note that, similar to PDE methods, Bellman's equation requires us to compute the values $V_i(x)$ for all times i

and all stock prices x, since we need all the values of V_i to get the *continuation value* $C_{i-1}(x)$ defined by

$$C_{i-1}(x) := E[V_i(X_i)|X_{i-1} = x], \tag{2.18}$$

which is the value of the option, if we decide note to exercise at time $i-1$. Thus, the value of the option at time $i-1$ is the maximum of the continuation value $C_{i-1}(x)$ and the exercise value $f_{i-1}(x)$.

Of course, we can also obtain prices by using particular stopping strategies. Let τ be a candidate for an optimal stopping time (i.e., a stopping time with values in $\{1,\dots,m\}$), then the solution $V_0^{(\tau)}$ of the corresponding stopping problem is, by (2.16), a lower estimate for the option price:

$$V_0^{(\tau)}(X_0) = E[f_\tau(X_\tau)] \le \sup_{\tau \le T} E[f_\tau(X_\tau)] = V_0(X_0).$$

On the other hand, having solved the dynamic program (2.17), we can immediately construct the optimal stopping time τ^* by defining

$$\tau^* := \min\{\, i \in \{1,\dots,m\} \mid f_i(X_i) \ge V_i(X_i) \,\}. \tag{2.19}$$

This stopping time satisfies $V^{(\tau^*)}(X_0) = V_0(X_0)$. In (2.19), we could have replaced the option price $V_i(X_i)$ by the continuation value $C_i(X_i)$. Now assume that we are only given estimate values $\overline{V}_i(x)$ for the option prices (or the continuation values). Inserting them into (2.19) gives a stopping time $\overline{\tau}$ and a corresponding option price $V_0^{(\overline{\tau})}(X_0) \le V_0(X_0)$. (Note that we *cannot* conclude that $V_0^{(\overline{\tau})}(X_0) = \overline{V}_0(X_0)$. This is only true for the optimal stopping time!). On the other hand, many estimate values $\overline{V}_i(x)$ obtained by approximately solving the Bellman equation (2.17) will tend to overestimate the value of the American option. Indeed, assume that we construct our approximate solution using an *unbiased* approximation of the conditional expectation operator denoted by $I[Y|X]$ (or $I[Y|X = x]$) in the sense that $E[I[g(X_i)|X_{i-1}]] = E[E[g(X_i)|X_{i-1}]] = E[g(X_i)]$. (One such choice would be to base the approximation on a sample average.) This means, the Bellman equation (2.17) is solved by setting $\overline{V}_m(x) = f_m(x)$ and then recursively

$$\overline{V}_{i-1}(x) := \max\left(f_{i-1}(x),\, I[\overline{V}_i(X_i)|X_{i-1} = x]\right). \tag{2.20}$$

Notice that $\overline{V}_i(x)$ will be a random variable, containing information about the future distributions of the stock price, unless it is a true conditional expectation.) Then we have the following result:

Lemma 2.37. *The approximate solution $\overline{V}$ defined in* (2.20) *is biased high, i.e.,*

$$E[\overline{V}_i(X_i)|X_i] \ge V_i(X_i), i = 0,\dots,m.$$

Proof. The statement holds for $i = m$ with equality. We prove the statement for $i = 0,\dots,m-1$ by backward induction. So, assume that the inequality holds for $i, i+1,\dots,m$. Then, Jensen's inequality and the induction hypothesis imply that

$$\begin{aligned}
E[\overline{V}_{i-1}(X_{i-1})|X_{i-1}] &= E\left[\max\left(f_{i-1}(X_{i-1}), I[\overline{V}_i(X_i)|X_{i-1}]\right)\middle| X_{i-1}\right] \\
&\ge \max\left(f_{i-1}(X_{i-1}), E[I[\overline{V}_i(X_i)|X_{i-1}]|X_{i-1}]\right) \\
&= \max\left(f_{i-1}(X_{i-1}), E[E[\overline{V}_i(X_i)|X_{i-1}]|X_{i-1}]\right) \\
&= \max\left(f_{i-1}(X_{i-1}), E[E[\overline{V}_i(X_i)|X_i]|X_{i-1}]\right) \\
&\ge \max\left(f_{i-1}(X_{i-1}), E[V_i(X_i)|X_{i-1}]\right) = V_{i-1}(X_{i-1}).
\end{aligned}$$

□

Remark 2.38. Assume we have an approximation technique for solving Bellman's equation (2.17) as above. By Lemma 2.37, the corresponding option prices $\overline{V}_i$ will be biased high. Now use the approximate option prices to construct a stopping time $\overline{\tau}$ by (2.19). The corresponding option prices $V_i^{(\overline{\tau})}$ are biased low. Thus, we assume that the true price is below $\overline{V}_0$ and above $V_0^{(\overline{\tau})}$.[8] In practice, this information is very useful, especially if the two values are close to each other. On the other hand, one can also try to combine both techniques into one numerical method, thus correcting the bias already on a local level. These methods can be much more precise, but on the other hand it is usually no longer possible to identify, whether the method has a high or low bias – if the bias has a common sign at all.

Random tree method

Probably the simplest method for pricing American options is by approximating the conditional expectations in the Bellman equation (2.17) using Monte Carlo simulation, i.e., by the sample average of b independent copies of $V_i(X_i)$ simulated conditioned on the value $X_{i-1} = x$. This leads to a random tree with branching parameter b. Indeed, assume we want to compute the price of the option given $X_0 = x$, i.e., $V_0(x)$. Then we simulate b independent samples from the conditional distribution of X_1 given $X_0 = x$ – recall that X_i is assumed to be a Markov chain. We denote these samples by $X_1^1, \dots, X_1^b$. Now pick any $1 \le j \le b$ and sample b independent simulations of X_2 using the conditional distribution of X_2 given that $X_1 = X_1^j$ and denote these samples by $X_1^{j,1}, \dots, X_2^{j,b}$. After having sampled all the values $X_2^{i,j}$, $i \le i, j \le b$, we have sampled a tree over two time periods, where each node at time 0 or 1 is connected with b daughter nodes at the next time. We continue in this way until time m, sampling simulations denoted by $X_m^{j_1,\dots,j_m}$ under the conditional distribution of X_m given $X_{m-1} = X_{m-1}^{j_1,\dots,j_{m-1}}$ in the last step. Now we are ready to solve the dynamic program (2.17) approximately along the tree. First we set $V_m^{j_1,\dots,j_m} := f_m(X_m^{j_1,\dots,j_m})$, and then we define approximate solutions by backward recursion

$$V_i^{j_1,\dots,j_i} := \max\left(f_i(X_i^{j j_1,\dots,j_i}),\ \frac{1}{b}\sum_{j=1}^{b} V_{i+1}^{j_1,\dots,j_i,j} \right), \tag{2.21}$$

i.e., by approximating the conditional expectation by the sample average among all daughter nodes.

Remark 2.39. The random tree method should not be confused with a binomial or b-ary tree method. For a b-ary tree, the distribution of X_i given X_{i-1} is discrete, with exactly b possible values, usually given in the form $u_1 X_{i-1}, \dots, u_b X_{i-1}$. For a random tree in the above sense, this is not true. In fact, the distribution can be continuous, and very different nodes can appear on the tree. As a consequence, the expectation computed in (2.21) is only an approximation, whereas the b-ary tree allows to compute true conditional expectation, however in a much simpler model.

By Lemma 2.37, the estimator $V_0^{\emptyset}$ is biased high. Indeed, using the approximation (2.21), we can give a more intuitive reason for this fact: the bias is high, because

[8] In fact, our results are random variables. This is obvious for $\overline{V}_0$ and also true for $V_0^{(\overline{\tau})}$, since we cannot compute the corresponding expectation explicitly, albeit by using an approximate expectation operator I as above. Thus, we can only claim that the true value is contained in a *confidence interval* around $[V_0^{(\overline{\tau})}, \overline{V}_0]$ with high probability.

the sample average *does* depend on the future realisations, and not only on their distribution. Thus, in the case of an American put-option, it might be advantageous to continue holding the option, even if the current stock price has actually crossed the exercise threshold, if we see (from the average) that the future stock prices will be particularly low given the current value.

An elegant low biased estimator can be constructed as follows: in node $X_i^{j_1,\dots,j_i}$ compute the estimator for $E[V_{i+1}(X_{i+1})|X_i]$ using all of the daughter nodes except for one, say except for $X_{i+1}^{j_1,\dots,j_i,b}$. Using that estimate, determine whether to exercise or to continue holding the option. Then compute the value *based on the decision but using the unused* value $X_{i+1}^{j_1,\dots,j_i,b}$, i.e., the option value at $X_i^{j_1,\dots,j_i}$ is set to $f_i(X_i^{j_1,\dots,j_i})$ or to $V_{i+1}^{j_1,\dots,j_i,b}$ depending on the decision. Finally, we take the average of the option values based on leaving out all possible daughter nodes. Thus, the low-biased estimator is defined as follows: $v_m^{j_1,\dots,j_m} := f_m(X_m^{j_1,\dots,j_m})$ at time m and then

$$v_{i,k}^{j_1,\dots,j_i} := \begin{cases} f_i(X_i^{j_1,\dots,j_i}), & \frac{1}{b-1}\sum_{j=1,j\neq k}^{b} v_{i+1}^{j_1,\dots,J_i,j} \leq f_i(X_i^{j_1,\dots,j_i}), \\ v_{i+1}^{j_1,\dots,j_i,k}, & \text{else,} \end{cases}$$

$$v_i^{j_1,\dots,j_i} := \frac{1}{b}\sum_{k=1}^{b} v_{i,k}^{j_1,\dots,j_i}.$$

A relatively straightforward argument shows that $v_0^{\emptyset}$ is indeed biased low.

Obviously, we expect the results of the random tree method to improve by increasing the branching number b. Indeed, Broadie and Glasserman [3] show convergence to the true price $V_0(X_0)$ for $b \to \infty$. On the other hand, note that size of the tree is m^b, and thus depends exponentially on the branching number. This causes severe restrictions in the possible size of the branching number (and of m) for actual computations. In particular, relatively small choices of b can already exhaust the memory of modern computers. While these memory requirements can be overcome by clever programming (reducing the number of nodes to be kept in memory to $mb+1$), the exponential growth in the computational time cannot be resolved easily. Thus, the random tree method is a good "toy method" because of its clarity and ease, but it is not really suitable for computations.

Pricing based on regression

A different method proposed in the literature is to estimate the conditional expectation in Bellman's equation (2.17) by a linear regression, instead of sample averages. Use the ansatz

$$E[V_{i+1}(X_{i+1})|X_i = x] = \sum_{j=1}^{N} \beta_{i,j}\psi_j(x) \tag{2.22}$$

for given basis functions $\psi_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1,\dots,N$. The unknown and time-dependent coefficients $\beta_{i,j}$, $j = 1,\dots,N$, then need to be estimated from simulations. From (2.22) we obtain (denoting $\psi(x) = (\psi_1(x),\dots,\psi_N(x))^T$ and $\beta_i = (\beta_{i,1},\dots,\beta_{i,N})^T$)

$$E[\psi(X_i)V_{i+1}(X_{i+1})] = E[\psi(X_i)\beta_i^T\psi(X_i)] = E[\psi(X_i)\psi(X_i)^T]\beta_i,$$

implying that

$$\beta_i = \left(E[\psi(X_i)\psi(X_i)^T]\right)^{-1} E[\psi(X_i)V_{i+1}(X_{i+1})] =: M_\psi^{-1}M_{\psi V}, \tag{2.23}$$

assuming that the matrix M_ψ is non-singular. In order to obtain the matrix M_ψ and the vector $M_{\psi V}$, we use sample averages. Summarizing, the whole regression-based algorithm looks like this:

Algorithm 2.40. *Simulate b independent paths of the Markov chain $X_1, \dots, X_m$ starting from $X_0 = x$. We denote the simulated values by $X_i^{(j)}$, $i = 1, \dots, m$, $j = 1, \dots, b$. Set $V_{m,j} \coloneqq f_m(X_m^{(j)})$ and proceed for $i = m-1, \dots, 0$ (backwards in time):*

(i) Compute matrices $\hat{M}_\psi$ and $\hat{M}_{\psi V}$ by

$$(\hat{M}_\psi)_{l,k} \coloneqq \frac{1}{b}\sum_{j=1}^{b} \psi_l(X_i^{(j)})\psi_k(X_i^{(j)}), \quad (\hat{M}_{\psi V})_k \coloneqq \frac{1}{b}\sum_{j=1}^{b} \psi_k(X_i^{(j)})V_{i+1,j}.$$

(ii) Set the regression coefficient $\hat{\beta}_i \coloneqq \hat{M}_\psi^{-1}\hat{M}_{\psi V}$.

(iii) Obtain the new option price estimates by

$$V_{i,j} \coloneqq \max\left(f_i(X_i^{(j)}), \hat{\beta}_i^T \psi(X_i^{(j)})\right). \tag{2.24}$$

If the regression (2.22) holds exactly, then it is not difficult to show that the algorithm converges to the true option price as $b \to \infty$. Longstaff and Schwartz [28] have introduced a modification, which provides a low-biased estimator.

As compared to the random tree method, the regression based method clearly is much less restrictive regarding memory and speed. A main difference is that all the future samples $X_{i+1}^{(1)}, \dots, X_{i+1}^{(b)}$ are used to provide the estimate at $X_i^{(j)}$, because they all enter into the regression (they are used to determine β_i). In contrast, in the random tree method only information coming from daughter nodes enter into the computation at a particular node. This points to a different class of algorithms, known as *stochastic mesh methods*, see Glasserman [16, Section 8.5].

It is well known that the choice of basis functions ψ is of vital importance for the accuracy of the algorithm. This seems clear from the onset, because in general, we would only expect the regression-based algorithm to converge for $b \to \infty$ and $N \to \infty$, assuming that $\psi_1(X_i), \dots, \psi_N(X_i)$ are chosen from a basis of $L^2(\Omega)$. In particular, practical applications show that it is not enough to use polynomial basis functions only, and that the accuracy increases when the payoff functions f_i are also included – in particular when they are non-differentiable, as in the cases of put-options or digital options.

Chapter 3

Discretization of stochastic differential equations

3.1 Generating sample paths

A stochastic differential equation describes the dynamics of a stochastic process in terms of a generating signal, usually a Brownian motion or, more generally, a Lévy process. Before actually solving SDEs, we are first going to discuss, how to (effectively) sample from the driving signal.

Brownian motion

In what follows, B_t denotes a one-dimensional Brownian motion. This restriction is imposed purely for convenience: all the methods hold, mutatis mutandis, also for a multi-dimensional Brownian motion.

Clearly, we cannot sample the full path $(B_t)_{t\in[0,T]}$, since it is an infinite-dimensional object. Instead, we concentrate on a finite-dimensional "skeleton" $(B_{t_1}, \dots, B_{t_n})$ based on a partition $0 = t_0 < t_1 < \cdots < t_n = T$ of the interval $[0, T]$. If we need an approximation to the true sample path, we can interpolate – note that interpolation makes the path non-adapted! For instance, if we want to simulate the payoff of a path-dependent option (in the Black-Scholes model), we can use interpolation of the sample path of the underlying Brownian motion to compute the *exact* payoff given by the interpolated finite-dimensional sample, or we can compute an *approximate* payoff directly from the sample. (In many cases, the two alternatives will actually coincide, think on Asian options, where the first method using linear interpolation of the finite sample coincides with a trapezoidal approximation of the integral.)

Cholesky factorization

Luckily, we know that

$$(B_{t_1}, \dots, B_{t_n}) \sim \mathcal{N}(0, \Sigma), \qquad \text{with } \Sigma^{i,j} = \min(t_i, t_j), \quad 1 \le i,\ j \le n.$$

Moreover, in Remark 2.15 we have indicated how to sample from a general, multi-dimensional Gaussian distribution: given n independent one-dimensional normal random variables $X = (X_1, \dots, X_n)$, we a n-dimensional normal random vector with co-

variance matrix Σ by AX, where $\Sigma = AA^T$. In this particular case, it is easy to find the Cholesky factorization A by

$$A = \begin{pmatrix} \sqrt{t_1} & 0 & \dots & 0 \\ \sqrt{t_1} & \sqrt{t_2 - t_1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{t_1} & \sqrt{t_2 - t_1} & \dots & \sqrt{t_n - t_{n-1}} \end{pmatrix}. \tag{3.1}$$

Random walk approach

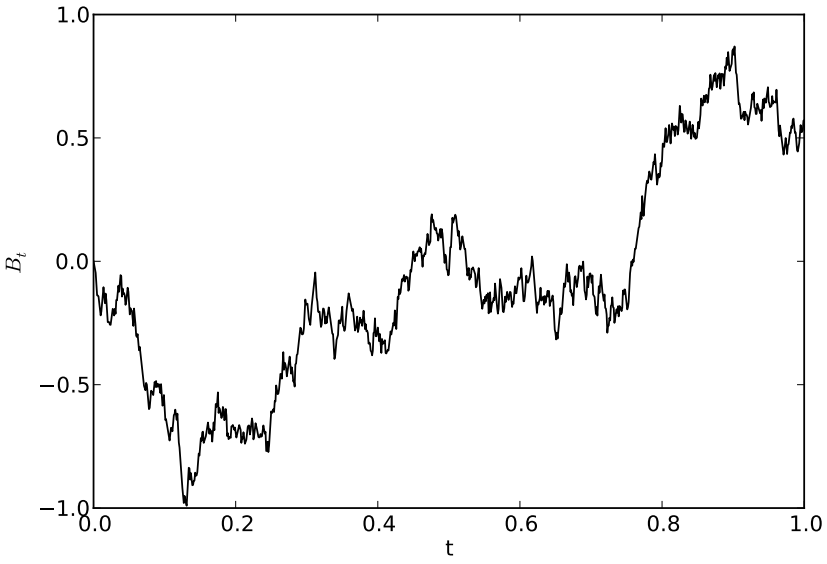

Figure 3.1: Brownian motion simulated using the random walk approach

An alternative way to sample $(B_{t_1}, \dots, B_{t_n})$ is by using the independence of the increments of the Brownian motion. Indeed, B_{t_1} can be directly sampled. Given B_{t_1}, we have $B_{t_2} = B_{t_1} + (B_{t_2} - B_{t_1})$, where the two summands B_{t_1} and $B_{t_2} - B_{t_1}$ are independent of each other and both have a normal distribution. We continue iteratively until we reach $B_{t_n} = B_{t_{n-1}} + (B_{t_n} - B_{t_{n-1}})$. Thus, we have seen that we only have to sample the increments $\Delta B_1 := B_{t_1} (= B_{t_1} - B_{t_0})$, $\Delta B_2 := B_{t_2} - B_{t_1}, \dots \Delta B_n := B_{t_n} - B_{t_{n-1}}$. Denoting $\Delta t_1 := t_1, \Delta t_2 := t_2 - t_1, \dots, \Delta t_n := t_n - t_{n-1}$, this is achieved by

$$\Delta B_1 = \sqrt{\Delta t_1} X_1, \dots, \Delta B_n = \sqrt{\Delta t_n} X_n, \tag{3.2}$$

where X again denotes an n-dimensional standard normal random variable. A closer look at the simulation using the Cholesky factorization (3.1) and the simulation of the increments (3.2) shows that both method give exactly the same samples from the Brownian motion if we start with the same standard normal sample X. Thus, (3.2) (with the additional summation of the increments ΔB) can be seen as an *efficient* implementation of the matrix multiplication AX.

Brownian bridge construction

Instead of starting with the first random variable B_{t_1}, let us start with the last one, $B_{t_n} = B_T \sim \mathcal{N}(0, T)$. Obviously, we can directly sample from this random variable. Next fix some k such that $t_k \approx T/2$. We want to continue by sampling B_{t_k}. But how? We cannot proceed by considering the corresponding increment, as before. However, the conditional distribution of B_{t_k} given B_{t_n} is well known as *Brownian bridge*[1]. Indeed, let $u < s < t$, then the conditional distribution of B_s given that $B_u = x$ and $B_t = y$ is

$$(B_s|B_u = x,\ B_t = y) \sim \mathcal{N}\left(\frac{(t-s)x + (s-u)y}{t-u},\ \frac{(s-u)(t-s)}{t-u}\right). \tag{3.3}$$

Thus, starting with B_T, we can sample the remaining values $B_{t_1}, \ldots, B_{t_{n-1}}$ iteratively and in any order. For instance, we could sample the value of the Brownian motion at time t_k closest to $T/2$ first, then continue with the values closest to $T/4$ and $3T/4$, respectively. While we can still represent the final sample $(B_{t_1}, \ldots, B_{t_n})$ as a deterministic function of an n-dimensional standard normal random variable X, this time the functional will not coincide with the functionals in the first two methods. However, the sampling is still exact, i.e., the sample $(B_{t_1}, \ldots, B_{t_n})$ constructed by Brownian bridges has the correct distribution.

Remark 3.1. Why should we use this complicated approach instead of the much simpler construction based on the increments? Note that the Brownian bridge construction starts by a very coarse approximation, which is more and more refined. Therefore, in many applications the final value of the quantity of interest (e.g., of the payoff of an option) depends much stronger on the coarse structure of the underlying path then on the details – think of a barrier option in the Black-Scholes model. Thus, if we write our option payoff as a functional $f(X_1, X_2, \ldots, X_n)$ of the normal random variables used for the Brownian bridge construction of the Brownian path (in the right order, i.e., X_1 is used to sample B_T and so on), then f will typically vary much stronger in the first variables then in the variables with high index. Thus, the Brownian bridge construction can be seen as a dimension-reduction technique, as discussed in the context of QMC.

Karhunen-Loève expansion

The Karhunen-Loève expansion is a type of Fourier expansion of the Brownian motion. Thus, it differs from the previous approximations by actually giving a sequence of continuous processes in time. Consider the eigenvalue problem for the covariance operator of the Brownian motion on the interval $[0, 1]$, i.e.,

$$\int_0^1 \min(s, t)\psi(s)ds = \lambda\psi(t). \tag{3.4}$$

Let λ_i denote the sequence of eigenvalues and ψ_i the corresponding sequence of eigenfunctions. Then we have the equality

$$B_t = \sum_{i=0}^{\infty} \sqrt{\lambda_i}\psi_i(t)Z_i, \tag{3.5}$$

[1] More precisely, the Brownian bridge is a Brownian motion on the interval $[0, 1]$ conditioned on $B_1 = 0$. It is a simple exercise to express the above conditional distribution in terms of the distribution of a Brownian bridge.

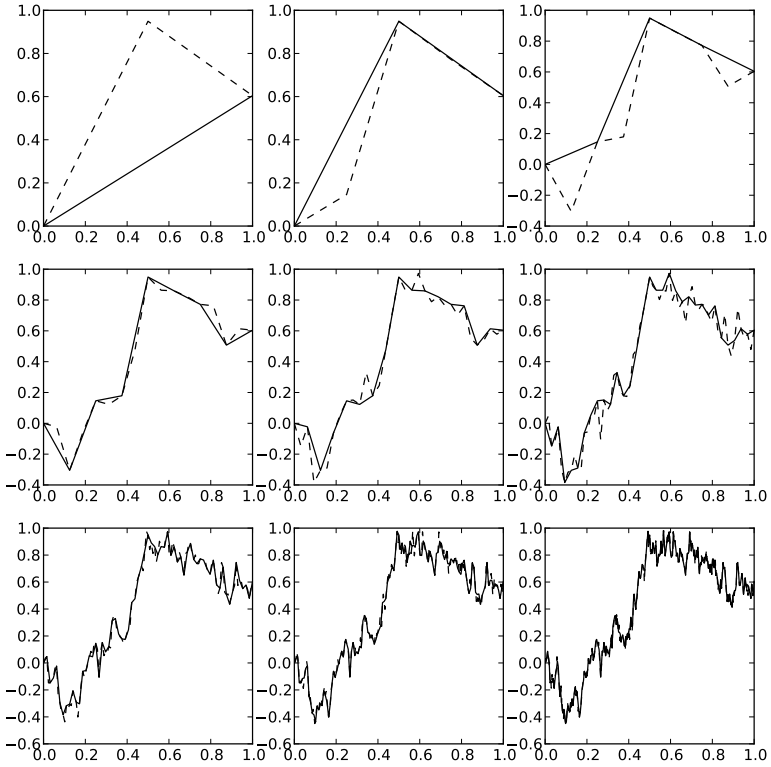

Figure 3.2: Brownian motion constructed by the Brownian bridge approach. Dashed lines correspond to the newly inserted Brownian bridge

with Z_i denoting a sequence of independent standard normal random variables. Since we can solve the eigenvalue problem explicitly, with

$$\lambda_i = \left(\frac{2}{(2i+1)\pi}\right)^2, \ \psi_i(t) = \sqrt{2}\sin\left(\frac{(2i+1)\pi t}{2}\right),$$

this leads to an exact approximation of Brownian motion by (random) trigonometric polynomials.

Remark 3.2. A similar expansion of Brownian motion is related to the Brownian bridge construction. It was, in fact, first used by P. Lévy for his construction of Brownian motion.

Example 3.3. We can, of course, sample from the paths of the stock prices S_t in the Samuelson model by applying any of the sampling techniques for the Brownian motion and then using

$$S_t = S_0 \exp\left(\sigma B_t + \left(\mu - \frac{\sigma^2}{2}\right)t\right).$$

The Poisson process

Many models in mathematical finance include jump processes, which are usually Lévy processes. The numerical treatment of these jump components is quite simple, provided that the have *finite activity*, i.e., only finitely many jumps in compact intervals. In this case, they are, in fact, *compound Poisson processes*, i.e., processes of the form

$$Z_t = Z_0 + \sum_{i=1}^{N_t} X_i, \tag{3.6}$$

where N_t denotes a (homogeneous) Poisson process and $(X_i)_{i=1}^{\infty}$ are independent samples of the jump distribution. This motivates the need to sample trajectories of the Poisson process. For what follows, N_t will denote a Poisson process with parameter $\lambda > 0$.

Sampling values of a Poisson process

We have (at least) two different possibilities if we want to sample the vector $(N_{t_1}, \dots, N_{t_n})$. In full analogy to the first method for sampling a Brownian motion, we can use independence of the increments of a Poisson process: N_{t_1} has a Poisson distribution with parameter λt_1, $N_{t_2} - N_{t_1}$ has a Poisson distribution with parameter $\lambda(t_2 - t_1)$ and is independent of N_{t_1} and so forth. Note that samples from a Poisson distribution can be generated using the inversion method.

On the other hand, there is also a *Poisson bridge*. Indeed, given $N_t = n$, we know that N_s has a binomial distribution with parameters n and $p = s/t$, $0 < s < t$.

Sampling the true trajectory

Unlike in the case of a Brownian motion, we can actually sample the true trajectory of a Poisson process on an interval $[0, t]$. Indeed, the trajectory is piecewise constant, so it suffices to sample the jump times within the interval, which is easily possible since there can only be finitely many such jumps. Again, two methods exist for sampling the jump times of a Poisson process. Let us denote the jump times of the Poisson process by T_n, $n \geq 1$. Thus, we have to construct the finite sequence $(T_1, \dots, T_{N_t})$.

(i) Note that the inter-arrival times $\tau_n \coloneqq T_n - T_{n-1}$ (with $T_0 \coloneqq 0$) of the jumps are independent of each other and have an exponential distribution with parameter λ. Therefore, we can start with $T_0 = 0$ and can iteratively produce τ_n and set $T_n = T_{n-1} + \tau_n$ and stop when $T_n > t$. Obviously, the algorithm stops in finite time with probability one.

(ii) Given $N_t = n$, the jump times $(T_1, \dots, T_n)$ are uniformly distributed on the interval $[0, t]$. More precisely, they are the *order statistics* of n independent uniforms on $[0, t]$. Thus, we can sample the jump times of the Poisson process by first sampling the number of jumps N_t, then taking a sequence of independent uniforms $(tU_1, \dots, tU_{N_t})$ (the U_ns are from a uniform distribution on $[0, 1]$) and finally ordering them in the sense that $T_1 = \min(tU_1, \dots, tU_{N_t})$, …, $T_{N_t} = \max(tU_1, \dots, tU_{N_t})$.

Example 3.4. Already in the seventies Merton introduced a jump diffusion into financial modelling. He proposed to model the stock price process by the SDE

$$dS_t = \mu S_{t-} dt + \sigma S_{t-} dB_t + S_{t-} dJ_t, \tag{3.7}$$

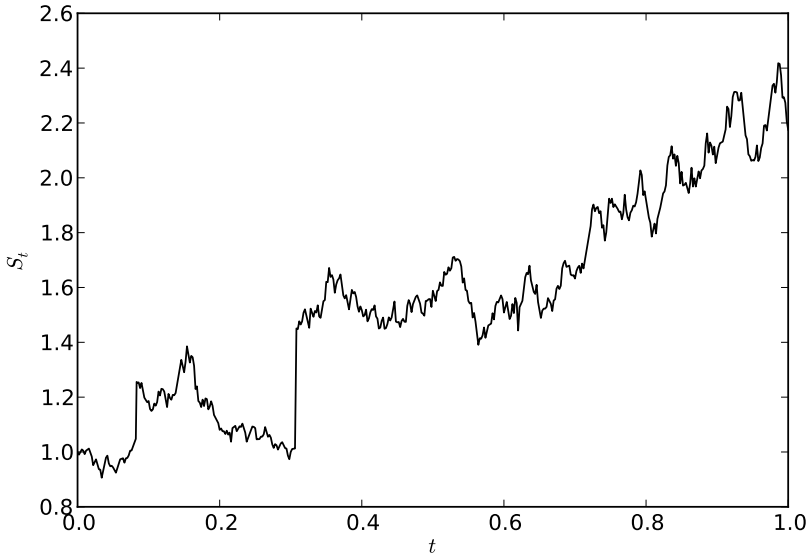

Figure 3.3: Trajectory of Merton's jump diffusion, see Example 3.4.

where J_t denotes a compound Poisson process which we denote by

$$J_t = \sum_{j=1}^{N_t} (X_j - 1),$$

where the X_j are independent samples from a distribution supported on the positive half-line. Moreover, we assume that the Poisson process N is independent of the Brownian motion B. In this case, it is possible to understand the SDE (3.7) without appealing to general stochastic integration. Indeed, between two jump times T_n and T_{n+1} of the underlying Poisson process, the stock price evolves according to the SDE of a geometric Brownian motion, i.e.,

$$S_t = S_{T_n} \exp\left(\sigma(B_t - B_{T_n}) + \left(\mu - \frac{\sigma^2}{2}\right)(t - T_n)\right), \quad T_n \le t < T_{n+1}.$$

At the time of the jump of the Poisson process, the stock price jumps as well. By convention, we require S to be right-continuous, i.e., we assume that S_t is the value of S just after the jump occurs, if there is a jump at time t. Now at time $t = T_{n+1}$, we read (3.7) to mean that

$$S_t - S_{t_-} = S_{t_-}(X_{n+1} - 1),$$

i.e., S jumps at time t and the value after the jump is given by $S_t = S_{t_-} X_{n+1}$. Summarising, we see that we can actually solve (3.7) explicitly:

$$S_t = S_0 \exp\left(\sigma B_t + \left(\mu - \frac{\sigma^2}{2}\right)t\right) \prod_{j=1}^{N_t} X_j.$$

If we want to sample trajectories of the Merton jump diffusion, we need to combine the sampling techniques for the Brownian motion and the Poisson process – of course,

we also need to sample the jumps X_j. Since these three components are assumed to be independent, no special care is necessary. We can sample $(S_{t_1}, \dots, S_{t_n})$ on a predefined grid by sampling the Brownian motion $(B_{t_1}, \dots, B_{t_n})$ and the Poisson process $(N_{t_1}, \dots, N_{t_n})$ along the grid and additionally sampling $(X_1, \dots, X_{N_{t_n}})$ from the jump distribution. Or we can sample the stock prices on a random grid containing the jump times. Note that in the original model by Merton, the jump heights X_j were assumed to have a log-normal distribution.

The variance gamma model

In mathematical finance, a very popular class of models for the stock price are the *exponential Lévy processes*, i.e., the stock price is given by $S_t = S_0 \exp(Z_t)$ for some Lévy process Z_t. By the very definition of a Lévy process as a process with stationary, independent increments, we know that the general strategy for sampling used for Brownian motion can also be applied for more general Lévy processes, i.e., if we want to sample $(Z_{t_1}, \dots, Z_{t_n})$, we can do so by sampling the increments $(Z_{t_1}, Z_{t_2} - Z_{t_1}, \dots, Z_{t_n} - Z_{t_{n-1}})$, which are independent. Moreover, in the case of a homogeneous grid $\Delta t_1 = \cdots = \Delta t_n$, we also know that, in fact, all the increments $\Delta Z_i = Z_{t_i} - Z_{t_{i-1}}$ have the same distribution.

Moreover, any Lévy process Z can be decomposed into a sum of a deterministic drift, a Brownian motion (in fact, a Brownian motion multiplied with a constant) and a pure jump process independent of the Brownian motion. If the process has *finite activity*, i.e., jumps only finitely often in each finite interval, then the pure jump process is a compound Poisson process. This case was, in fact, already treated in Example 3.4. However, in many popular models, the Lévy process has infinite activity, and is, in fact, a pure jump process, without Brownian component. One of these models will be presented a bit more detailed in this section.

One particular pure-jump exponential Lévy model is the *variance gamma model*. In this model, Z is the difference of two independent *gamma processes*, $Z_t = U_t - D_t$. A gamma process is a Lévy processes, whose increments satisfy the gamma distribution.[2] More precisely, a gamma process is a Lévy process whose marginals satisfy the gamma distribution with constant scale parameter θ and linear shape parameter, i.e., $Z_t \sim \Gamma_{kt,\theta}$, $k \in \mathbb{R}_{>0}$. Therefore, also the increments satisfy $Z_t - Z_s \sim \Gamma_{k(t-s),\theta}$. Notice that the gamma process is a *subordinator* (i.e., a process with non-decreasing sample paths) of infinite activity.

Obviously, sampling from the variance gamma process is easy once we can sample the gamma process – after all, U and D are independent. In order to sample trajectories of the gamma process, we sample the increments, which have the gamma distribution. Sampling from the gamma distribution can be done by the acceptance-rejection method. The density of a $\Gamma_{k,\theta}$-distribution is

$$f(x) = x^{k-1} \frac{e^{-x/\theta}}{\theta^k \Gamma(k)}, \quad x > 0.$$

Various complimentary distributions have been suggested. First of all note that we may assume that $\theta = 1$: if $X \sim \Gamma_{k,1}$, then $\theta X \sim \Gamma_{k,\theta}$. Then [10, Theorem IX.3.2] shows that the density of the $\Gamma_{k,1}$-distribution converges to a standard Gaussian density. Therefore, for the sampling algorithm to work equally well for all values of k, the complimentary

[2] Recall that the sum of n independent gamma-distributed random variables $X_i \sim \Gamma_{k_i,\theta}$ has a gamma distribution $\Gamma_{\sum_i k_i,\theta}$. Thus, the gamma distribution is infinitely divisible, which implies that there is a Lévy process with gamma distributed marginals.

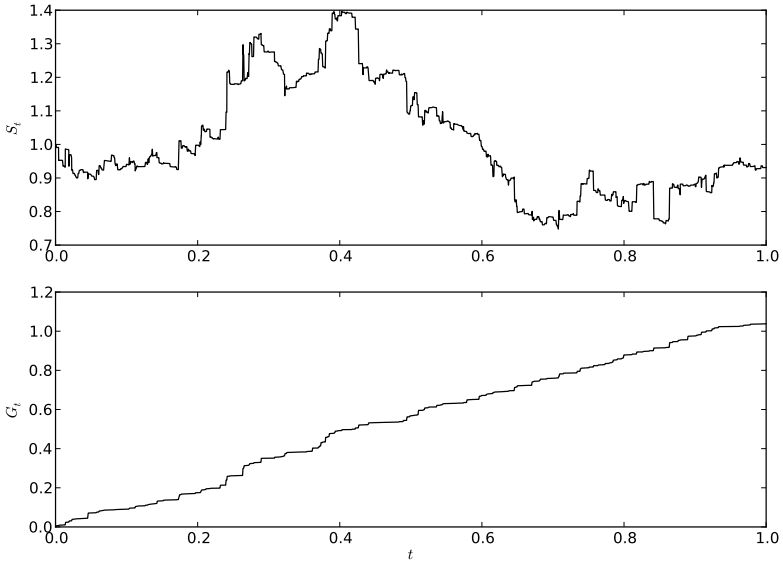

Figure 3.4: Trajectory of the variance-gamma process

density g should be close to a normal density. On the other hand, the gamma distribution has fatter tails than the normal distribution, i.e., the value of the density converges much slower to 0 for $x \to \infty$ than for the normal density. Therefore, we cannot choose a normal distribution as complimentary distribution. By this reasoning, combinations of the densities of normal and exponential distributions have been suggested, as well as many other distributions. (Note that we will usually only need small values of k if we sample the increments.)

Remark 3.5. If the scale and shape parameters θ_U, θ_D and k_U, k_D of the two gamma processes U and D satisfy $k_U = k_D =: 1/\theta$, then we can represent the variance gamma process $Z_t = U_t - D_t$ as

$$Z_t = W_{G_t},$$

where G is a gamma process with parameters θ and $k = 1/\theta$ and W is a Brownian motion with drift, more precisely

$$W_t = \mu t + \sigma B_t, \quad \mu = \frac{\theta_U - \theta_D}{\theta},\ \sigma^2 = 2\frac{\theta_U \theta_D}{\theta}$$

for a standard Brownian motion B independent of G. This gives another method of sampling the variance gamma process: instead of sampling from two gamma processes, we can also sample from one gamma process and one Brownian motion. Note that this representation motivates the name "variance gamma process": conditional on G_t, Z_t is Gaussian with variance $\sigma^2 G_t$. Moreover, this type of construction (log-stock-price as a random time-change (or subordination) of a Brownian motion) is often used in financial modelling.

Approximation of Lévy processes

In the previous sections, we have seen how to sample from compound Poisson processes (or, more generally, jump diffusions, i.e., finite activity Lévy processes). Moreover, we have also seen that we can sample the gamma process (and variants like the variance gamma process), a special example of an infinite activity Lévy process. However, in general we do not know how to sample increments of a Lévy process, if we only know its characteristic triple. In the case of a finite Lévy measure ν, we know that the Lévy process is a compound Poisson process (modulo a Brownian motion), and then the problem is reduced to the problem of sampling random variables with distribution $\nu(\cdot)/\nu(\mathbb{R})$ – which might be easy or not. In this section, we concentrate on the case of infinite activity.

For the rest of this section, let us assume that the Lévy process Z under consideration does not have a Brownian component, i.e., that it has the characteristic triple $(\gamma, 0, \nu)$. By Theorem B.4, we can write Z as a sum of a compound Poisson process and a process of (compensated) jumps of size smaller than ϵ. In fact, we have

$$Z_t = \gamma t + \sum_{0<s\le t} \Delta Z_s \mathbf{1}_{|\Delta Z_s|\ge 1} + \lim_{\epsilon\to 0} N_t^\epsilon, \quad N_t^\epsilon := \sum_{0<s\le t} \Delta Z_s \mathbf{1}_{\epsilon\le|\Delta Z_s|<1} - t\int_{\epsilon\le|z|\le 1} z\nu(dz).$$

Thus, we may approximate Z by fixing a finite ϵ in the above formula, i.e., by discarding all jumps smaller than ϵ:

$$Z_t^\epsilon := \gamma t + \sum_{0<s\le t} \Delta Z_s \mathbf{1}_{|\Delta Z_s|\ge 1} + N_t^\epsilon \tag{3.8}$$

for some fixed $\epsilon > 0$. Obviously, Z^ϵ is a compound Poisson process with drift, therefore we can – in principle – sample from this process (even the paths). It is not surprising that the error of the approximation depends on the Lévy measure ν. Indeed, one can show (see Cont and Tankov [7, Section 6.3, 6.4]) that

$$\operatorname{var}[Z_t - Z_t^\epsilon] = t\int_{|z|<\epsilon} z^2\nu(dz) =: t\sigma(\epsilon)^2. \tag{3.9}$$

This is also relevant for weak approximation in the following sense: assume that f is a differentiable function whose derivative f' is bounded by a constant C. Then one can show([7, Proposition 6.1]) that

$$\left|E[f(Z_t)] - E[f(Z_t^\epsilon)]\right| \le C\sigma(\epsilon)\sqrt{t}.$$

The error $Z_t - Z_t^\epsilon$ consists of all small jumps of Z. It seems naturally to suggest that these small jumps might, in turn, be approximated by a Brownian motion. This is indeed the case, but only under certain assumptions on the Lévy measures. Asmussen and Rosinski [1] show that $\sigma(\epsilon)^{-1}(Z - Z^\epsilon)$ converges to a Brownian motion if and only if

$$\frac{\sigma(\epsilon)}{\epsilon} \xrightarrow[\epsilon\to 0]{} \infty$$

(provided that ν has no atoms in a neighborhood of 0). This leads to a jump diffusion approximation

$$Z_t \approx Z_t^\epsilon + \sigma(\epsilon)B_t, \tag{3.10}$$

which also improves the weak convergence. Let us conclude with a few examples taken from [7].

Example 3.6. *Symmetric stable* Lévy processes are one-dimensional pure jump processes with Lévy measure $\nu(dx) = C/|x|^{1+\alpha}$ for some $0 < \alpha < 2$. (Their characteristic function is then $\exp(-\sigma^\alpha |u|^\alpha)$ at $t = 1$ for some positive constant σ.) In this case, $\sigma(\epsilon) \sim \epsilon^{1-\alpha/2}$. Moreover, the intensity λ_ϵ of the approximating compound Poisson process Z^ϵ satisfies $\lambda_\epsilon \sim \epsilon^{-\alpha}$. This in particular implies that here the approximation can be further improved by adding a Brownian motion $\sigma(\epsilon)B$, since the error of the approximation is asymptotically a Brownian motion.

These results can be extended to *tempered stable processes*, i.e., pure jump processes with Lévy measure

$$\nu(dx) = \frac{C_- e^{-\lambda_-|x|}}{|x|^{1+\alpha_-}} \mathbf{1}_{x<0} dx + \frac{C_+ e^{-\lambda_+|x|}}{|x|^{1+\alpha_+}} \mathbf{1}_{x>0} dx.$$

In finance, $S_t = \exp(Z_t)$ is often used as model for stock prices, when Z is a tempered stable process. In particular, the prominent CGMY-model, see Carr, Geman, Madan and Yor [5], is a special case with $C_- = C_+$ and $\alpha_- = \alpha_+$. Note that in for stable or tempered stable processes simulation of the compound Poisson process Z^ϵ is straightforward, by the acceptance-rejection method, while simulation of the increments of the true process Z is difficult.

Example 3.7. In the case of the gamma process, we have $\sigma(\epsilon) \sim \epsilon$. This means on the one hand, that the quality of the approximation by the compound Poisson process Z^ϵ is already very good. On the other hand, the error does not converge to a Brownian motion, thus the jump diffusion approximation will not improve the quality even more. Here, the intensity of Z^ϵ satisfies $\lambda_\epsilon \sim -\log(\epsilon)$.

3.2 The Euler method

Many financial models are (entirely or partly) determined in terms of a *stochastic differential equation*. Therefore, a major area of computational finance is the numerical approximation of solutions of SDEs. To fix ideas, let us start with a general n-dimensional SDE driven by a d-dimensional Brownian motion B, i.e.,

$$dX_t = V(X_t)dt + \sum_{i=1}^{d} V_i(X_t) dB_t^i, \tag{3.11}$$

for some vector fields $V, V_1, \ldots, V_d : \mathbb{R}^n \to \mathbb{R}^n$, which we assume to be uniformly Lipschitz and linearly bounded (with the same constant K) – these are the usual assumptions for existence and uniqueness of the solution of (3.11). Notice that the above formulation includes non-autonomous SDEs, i.e., SDEs where the vector fields depend explicitly on time. However, since regularity requirements are usually less stringent on the time-dependence than on the space-dependence, this formulation will not yield sharp results for the non-autonomous case. See Appendix A for a collection of basic facts and examples of SDEs in finance. Moreover, we shall assume that the initial value $X_0 = x \in \mathbb{R}^n$ is a constant. This is mainly for convenience, the theory is not more difficult as long as the random initial value X_0 is independent of the noise.

Of course, we can also consider SDEs driven by more general processes than Brownian motion, for instance an SDE driven by a Lévy process,

$$dX_t = V(X_t)dt + \sum_{i=1}^{d} V_i(X_t) dZ_t^i$$

for a d-dimensional Lévy process Z, or even by a general semi martingale. We will not treat the case of a semimartingale noise, but we will give some results for SDEs driven by Lévy noise. The main focus – and also the main theoretical difficulty – is however on diffusions of the type (3.11).

So, the goal of the next part of the course is to derive, for a fixed time interval $[0, T]$, approximations $\overline{X}$ to the solution X. These approximations will be based on a time grid $\mathcal{D} = \{0 = t_0 < t_1 < \cdots < t_N = T\}$ with size N. We denote

$$|\mathcal{D}| \coloneqq \max_{1 \le i \le N} |t_i - t_{i-1}|$$

the *mesh* of the grid, and we define the increments of time and of any process Y (which will usually be either X, B, or Z) along the grid by

$$\Delta t_i \coloneqq t_i - t_{i-1}, \ \Delta Y_i \coloneqq Y_{t_i} - Y_{t_{i-1}}, \quad 1 \le i \le N.$$

Moreover, for $t \in [0, T]$ we set $\lfloor t \rfloor = \sup\{ t_i \mid 0 \le i \le N, \ t_i \le t \}$. We will define the approximation along the grid, i.e., we will define the random variables $\overline{X}_i = \overline{X}_{t_i}$, $0 \le i \le N$. We will write $\overline{X}^{\mathcal{D}}$ if we want to emphasise the dependence on the grid. The first natural question arising from this program is in which sense $\overline{X}$ should be an approximation to X. The two most important concepts are *strong* and *weak approximation*.

Definition 3.8. The scheme $\overline{X}^{\mathcal{D}}$ *converges strongly* to X if

$$\lim_{|\mathcal{D}| \to 0} E\left[\left|X_T - \overline{X}_T^{\mathcal{D}}\right|\right] = 0.$$

Moreover, we say that the scheme $\overline{X}^{\mathcal{D}}$ has *strong order* γ if (for $|\mathcal{D}|$ small enough)

$$E\left[\left|X_T - \overline{X}_T^{\mathcal{D}}\right|\right] \le C\,|\mathcal{D}|^{\gamma}$$

for some constant $C > 0$, which does not depend on $\gamma > 0$.

Definition 3.9. Given a suitable class $\mathcal{G}$ of functions $f : \mathbb{R}^n \to \mathbb{R}$, we say that the scheme $\overline{X}^{\mathcal{D}}$ *converges weakly* (with respect to $\mathcal{G}$) if

$$\forall f \in \mathcal{G} : \ \lim_{|\mathcal{D}| \to 0} E\left[f\left(\overline{X}_T^{\mathcal{D}}\right)\right] = E[f(X_T)].$$

Moreover, we say that $\overline{X}^{\mathcal{D}}$ has weak order $\gamma > 0$ if for every $f \in \mathcal{G}$ there is a constant C (not depending on $|\mathcal{D}|$) such that

$$\left|E\left[f\left(\overline{X}_T^{\mathcal{D}}\right)\right] - E[f(X_T)]\right| \le C\,|\mathcal{D}|^{\gamma}$$

provided that $|\mathcal{D}|$ is small enough.

The class of functions $\mathcal{G}$ in Definition 3.9 should reflect the applications we have in mind. Of course, there is a strong link between strong and weak convergence. For instance, if a scheme converges strongly with order γ, then we can immediately conclude that it will also converge weakly with order γ, provided that all the functions in $\mathcal{G}$ are uniformly Lipschitz. In principle, however, there is a big difference between these concepts: for instance, a strong scheme must be defined on the same probability space as the true solution X, which is clearly not necessary in the weak case. Moreover, since most approximation problems in finance are of the weak type, this notion seems to be the more relevant to us.

The classical reference for approximation of SDEs is the book by Kloeden and Platen [21].

The Euler-Maruyama method

Fix a grid $\mathcal{D}$ and an SDE driven by a Brownian motion, i.e., of type (3.11). We hope to get some insight into how to approximate the solution by taking a look at a deterministic ODE

$$\dot{x}(t) = V(x(t)), \quad x(0) = x_0 \in \mathbb{R}^n. \tag{3.12}$$

The simplest method of approximating a value $x(t_i)$ given the value $x(t_{i-1})$ is by doing a first order Taylor expansion around $x(t_{i-1})$, giving

$$x(t_i) = x(t_{i-1}) + \dot{x}(t_{i-1})\Delta t_i + O(\Delta t_i^2) = x(t_{i-1}) + V(x(t_{i-1}))\Delta t_i + O(\Delta t_i^2).$$

So, the Euler scheme for SDEs is defined by $\overline{x}_0 = x_0$ and $\overline{x}_i = V(\overline{x}_{i-1})\Delta t_i$, $1 \le i \le N$. Since we have to add up the individual error contributions, we get the global error

$$|x(T) - \overline{x}_N| = \sum_{i=1}^{N} O(\Delta t_i^2) \le O(|\mathcal{D}|)T.$$

Therefore, the deterministic Euler scheme has order one.

The Euler scheme for SDEs (also known as *Euler-Maruyama scheme*) is defined in complete analogy, i.e., we set $\overline{X}_0 = x$ and then continue by

$$\overline{X}_i = \overline{X}_{i-1} + V(\overline{X}_{i-1})\Delta t_i + \sum_{j=1}^{d} V_j(\overline{X}_{i-1})\Delta B_i^j, \quad 1 \le i \le N. \tag{3.13}$$

Moreover, we extend the definition of $\overline{X}_i = \overline{X}_{t_i}$ for all times $t \in [0, T]$ by some kind of stochastic interpolation between the grid points, more precisely by

$$\overline{X}_t = \overline{X}_{\lfloor t \rfloor} + V(\overline{X}_{\lfloor t \rfloor})(t - \lfloor t \rfloor) + \sum_{i=1}^{d} V_i(\overline{X}_{\lfloor t \rfloor})(B_t^i - B_{\lfloor t \rfloor}^i). \tag{3.14}$$

Notice, however, that we should not expect the Euler scheme to converge with order one as in the ODE setting: the increments of a Brownian motion are much bigger than the increment of time, since $\Delta B_i^j \sim \sqrt{\Delta t_i}$, and this is indeed the correct strong order of convergence.

Theorem 3.10. *Suppose that the coefficients of the SDE* (3.11) *have a uniform Lipschitz constant $K > 0$ and satisfy the linear growth condition with the same constant. Then the Euler-Maruyama approximation $\overline{X}$ satisfies*

$$E\left[\sup_{0 \le t \le T} \left|X_t - \overline{X}_t\right|\right] \le C\sqrt{|\mathcal{D}|}$$

for some constant C only depending on the coefficients, the initial value and the time horizon T. In particular, the Euler-Maruyama method has strong order $1/2$.

Proof. In this proof, C denotes a constant that may change from line to line, but never in a way depending on the partition. Moreover, *for this proof only*, we set $V_0 := V$, $B_t^0 := t$.

We know from the existence and uniqueness proof of the SDE (3.11) that $E\left[\sup_{0\le t\le T}|X_t|^2\right] \le C(1+|x|^2)$, and in the same fashion we can prove the analogous inequality for X replaced by $\overline{X}$. Now fix some $0 \le t \le T$. We want to estimate

$$e_t \coloneqq E\left[\sup_{0\le s\le t}\left|X_s - \overline{X}_s\right|^2\right].$$

First note that we have the representation

$$\begin{aligned} X_s - \overline{X}_s &= \int_0^s \left(V(X_u) - V(\overline{X}_{\lfloor u\rfloor})\right)du + \sum_{i=1}^d \int_0^s \left(V_i(X_u) - V_i(\overline{X}_{\lfloor u\rfloor})\right)dB_u^i \\ &= \sum_{i=0}^d \int_0^s \left(V_i(X_u) - V_i(\overline{X}_{\lfloor u\rfloor})\right)dB_u^i \\ &= \sum_{i=0}^d \left\{\int_0^s \left(V_i(X_u) - V_i(X_{\lfloor u\rfloor})\right)dB_u^i + \int_0^s \left(V_i(X_{\lfloor u\rfloor}) - V_i(\overline{X}_{\lfloor u\rfloor})\right)dB_u^i\right\}. \end{aligned}$$

Therefore, we can bound e_t by

$$\begin{aligned} e_t &\le C\sum_{i=0}^d \left\{E\left[\sup_{0\le s\le t}\left|\int_0^s \left(V_i(X_u) - V_i(X_{\lfloor u\rfloor})\right)dB_u^i\right|^2\right] + E\left[\sup_{0\le s\le t}\left|\int_0^s \left(V_i(X_{\lfloor u\rfloor}) - V_i(\overline{X}_{\lfloor u\rfloor})\right)dB_u^i\right|^2\right]\right\} \\ &=: C\sum_{i=0}^d (c_t^i + d_t^i). \end{aligned}$$

Using the Ito isometry and Lipschitz continuity of the coefficients, we get

$$d_t^i = E\left[\sup_{0\le s\le t}\left|\int_0^s \left(V_i(X_{\lfloor u\rfloor}) - V_i(\overline{X}_{\lfloor u\rfloor})\right)dB_u^i\right|^2\right] \le \begin{cases} K^2T\int_0^t e_s ds, & i = 0, \\ K^2\int_0^t e_s ds, & 1 \le i \le d. \end{cases} \tag{3.15}$$

For c_t^0, we get

$$c_t^0 \le K^2T\int_0^t E\left[\left|X_u - X_{\lfloor u\rfloor}\right|^2\right]du \quad = K^2T\int_0^t E\left[\left|\int_{\lfloor u\rfloor}^u V(X_s)ds + \sum_{i=1}^d \int_{\lfloor u\rfloor}^u V_i(X_s)dB_s^i\right|^2\right]du,$$

which can be estimated (using the uniform bound on the second moment of X) by

$$\begin{aligned} c_t^0 &\le K^2C(1+|x|^2)\int_0^t \left((u - \lfloor u\rfloor)^2 + d(u - \lfloor u\rfloor)\right)du \\ &\le TK^2C(1+|x|^2)(|\mathcal{D}| + d)\,|\mathcal{D}|\,. \end{aligned}$$

A similar computation for c_t^i gives the common bound

$$c_t^i \le \begin{cases} TK^2C(1+|x|^2)\,|\mathcal{D}|\,, & i = 0, \\ K^2C(1+|x|^2)\,|\mathcal{D}|\,, & 1 \le i \le d. \end{cases} \tag{3.16}$$

Combining the bounds (3.15) and (3.16), we obtain

$$e_t \le C\,|\mathcal{D}| + C\int_0^t e_s ds,$$

and Gronwall's inequality implies

$$e_t \leq C\,|\mathcal{D}|\,,$$

giving the statement of the theorem by taking the square root and applying the Hölder inequality. □

Weak convergence of the Euler method

Next we discuss the weak convergence of the Euler method. While the strong convergence problem might seem more natural to consider, in most applications we are actually mainly interested in weak convergence. This is especially true for mathematical finance, where the option pricing problem is precisely of the form introduced in Definition 3.9. Moreover, weak approximation of SDEs can be used as a numerical method for solving linear parabolic PDEs. Indeed,

$$u(t,x) \coloneqq E\left[\, f(X_T) | \, X_t = x\right] \tag{3.17}$$

satisfies the *Kolmogorov backward equation* associated to the generator $L = V_0 + \frac{1}{2}\sum_{i=1}^{d} V_i^2$, i.e., the Cauchy problem

$$\begin{cases} \dfrac{\partial}{\partial t}u(t,x) + Lu(t,x) = 0, \\ \qquad\qquad u(T,x) = f(x), \end{cases} \tag{3.18}$$

a PDE known as Black-Scholes PDE in finance. (For details and more precise statements see Appendix A.) Note that similar *stochastic representations* also exist for the corresponding Dirichlet and Neumann problems.

On the other hand, strong convergence implies weak convergence. Indeed, assume that f is Lipschitz, with Lipschitz constant denoted by $\|\nabla f\|_\infty$. Then we have

$$\left| E\left[f\left(\overline{X}_T^{\mathcal{D}}\right)\right] - E\left[f\left(X_T\right)\right]\right| \leq \|\nabla f\|_\infty \, E\left[\left| X_T - \overline{X}_T^{\mathcal{D}}\right|\right] \leq C\,\|\nabla f\|_\infty \, \sqrt{|\mathcal{D}|}, \tag{3.19}$$

by Theorem 3.10. Thus, the Euler scheme has (at least) weak order $1/2$ for all Lipschitz functions f – which includes most, but not all the claims used in finance. However, in many situations the weak order is actually better than the strong order. In the following, we shall first present (and prove) "the typical situation" under unnecessarily restrictive regularity assumptions, before we state sharper results (without proofs). Our presentation is mainly based on Talay and Tubaro [37]. For our discussion we assume that the grids $\mathcal{D}$ are homogeneous, i.e., $\Delta t_i = h \coloneqq T/N$ for every i. Of course, the results hold (with minor corrections) also in the general case, with h being replaced by $|\mathcal{D}|$.

Theorem 3.11. *Assume that the vector fields $V, V_1, \ldots, V_d$ are C^∞-*bounded*, i.e., they are smooth and the vector fields together with all there derivatives are bounded functions. Moreover, assume that $\mathcal{G}$ consists of smooth, polynomially bounded functions. Then the Euler method has weak order one. Moreover, the error*

$$e(T,h,f) \coloneqq E\left[f\left(\overline{X}_T^{\mathcal{D}}\right)\right] - u(0,x)$$

for the weak approximation problem started at $t = 0$ at $X_0 = \overline{X}_0 = x \in \mathbb{R}^n$ has the representation

$$e(T,h,f) = h\int_0^T E[\psi_1(s,X_s)]ds + h^2 e_2(T,f) + O(h^3), \tag{3.20}$$

where ψ_1 is given by

$$\begin{aligned}\psi_1(t,x) = \frac{1}{2}\sum_{i,j=1}^{n} V^i(x)V^j(x)\partial_{(i,j)}u(t,x) + \frac{1}{2}\sum_{i,j,k=1}^{n} V^i(x)a_k^j(x)\partial_{(i,j,k)}u(t,x) + \\ + \frac{1}{8}\sum_{i,j,k,l=1}^{n} a_j^i(x)a_l^k(x)\partial_{(i,j,k,l)}u(t,x) + \frac{1}{2}\frac{\partial^2}{\partial t^2}u(t,x) + \\ + \sum_{i=1}^{n} V^i(x)\frac{\partial}{\partial t}u(t,x)\partial_i u(t,x) + \frac{1}{2}\sum_{i,j=1}^{d} a_j^i(x)\frac{\partial}{\partial t}u(t,x)\partial_{(i,j)}u(t,x),\end{aligned}$$

where $\partial_I = \frac{\partial^k}{\partial x^{i_1}\cdots\partial x^{i_k}}$ for a multi-index $I = (i_1,\dots,i_k)$ and $a_j^i(x) = \sum_{k=1}^{d} V_k^i(x)V_k^j(x)$, $1 \le i, j \le n$.

Remark 3.12. The result also holds for the non-autonomous case, i.e., for $f = f(t,x)$ and the vector fields also depending on time.

We will prove the theorem by a succession of lemmas, starting by a lemma whose proof is obvious by differentiating inside the expectation (3.17).

Lemma 3.13. *Under the assumptions of Theorem 3.11, the solution u of* (3.18) *is smooth and all its derivatives have polynomial growth.*

In the next lemma, we compute the *local error* of the Euler scheme, i.e., the weak error coming from one step of the Euler scheme.

Lemma 3.14. *Again under the assumptions of Theorem 3.11, we have*

$$E\left[u(t_{i+1},\overline{X}_{i+1})\middle|\overline{X}_i = x\right] = u(t_i,x) + h^2\psi_1(t_i,x) + O(h^3).$$

Proof. Obviously, we may restrict ourselves to $i = 0$, i.e., we only need to show that

$$E\left[u(h,\overline{X}_1)\middle|\overline{X}_0 = x\right] = u(0,x) + h^2\psi_1(0,x) + O(h^3),$$

since the general situation works precisely the same way. Taylor expansion of $u(h, x + \Delta x)$ in h and Δx around $u(0,x)$ gives

$$\begin{aligned}u(h,x+\Delta x) = u(0,x) + h\partial_t u(0,x) + \frac{1}{2}h^2\partial_{tt}u(0,x) + h\sum_{i=1}^{n}\Delta x^i\partial_t u(0,x)\partial_i u(0,x) + \\ + \frac{1}{2}h\sum_{i,j=1}^{n}\Delta x^i\Delta x^j\partial_t u(0,x)\partial_{(i,j)}u(0,x) \\ + \sum_{k=1}^{4}\frac{1}{k!}\sum_{i_1,\dots,i_k=1}^{n}\Delta x^{i_1}\cdots\Delta x^{i_k}\partial_{(i_1,\dots,i_k)}u(0,x) + O(h\Delta x^3) + O(\Delta x^5),\end{aligned}$$

where $O(\Delta x^k)$ means that the term is $O(\Delta x^{i_1}\cdots\Delta x^{i_k})$ for any multi-index $(i_1,\dots,i_k)$. Now insert

$$\Delta\overline{X} = V(x)h + \sum_{i=1}^{d} V_i(x)\Delta B_1^i$$

in place of Δx and take the expectation. First we note that there are no terms of order $k/2$ for odd numbers k, because they can only appear as odd moments of the Brownian

increment $\Delta B_1 \sim \mathcal{N}(0, hI_d)$, which vanish. Moreover, $E\left[O\left(h\Delta\overline{X}^3\right)\right] = O(h^3)$, since $\Delta B_1^i \sim \mathcal{N}(0,h)$, and, similarly, $E\left[O\left(\Delta\overline{X}^5\right)\right] = O(h^3)$. Let us know collect all the terms of order one in h. Apart from the deterministic term $h\partial_t u(0,x) = -hLu(0,x)$ (since u solves (3.18)), we have the drift term from the first order Taylor term (in Δx) (note that the diffusion part in the first order term vanishes since $E[\Delta B_1^i] = 0$), and the diffusion terms from the second order Taylor term, more precisely, the term of order h is given by

$$-hLu(0,x) + h\sum_{i=1}^{n} V^i(x)\partial_i u(0,x) + \frac{1}{2}h\sum_{i,j=1}^{n}\sum_{k=1}^{d} V_k^i(x)V_k^j(x)\partial_{(i,j)}u(0,x) = 0,$$

by the definition of the partial differential operator L. Here, we only used that

$$E[\Delta\overline{X}^i\Delta\overline{X}^j] = h^2V^i(x)V^j(x) + h\sum_{k=1}^{d} V_k^i(x)V_k^j(x).$$

This shows the main point of the lemma, namely that the local error is of order two in h. Figuring out the precise form of the leading order error term as given above (i.e., figuring out ψ_1) is done by computing all the expectations of the terms of the above Taylor expansion using the moments of $\Delta\overline{X}$, and is left to the reader. □

Proof of Theorem 3.11. By the final condition of (3.18), we may express the error of the Euler scheme (for approximating $u(0,x)$) as

$$\begin{aligned} E\left[f(\overline{X}_N)\right] - u(0,x) &= E\left[u(T,\overline{X}_N) - u(0,x)\right] \\ &= \sum_{i=0}^{N-1} E\left[u(t_{i+1},\overline{X}_{i+1}) - u(t_i,\overline{X}_i)\right] \\ &= \sum_{i=1}^{N-1}\left\{h^2E\left[\psi_1(t_i,\overline{X}_i)\right] + O(h^3)\right\}. \end{aligned} \tag{3.21}$$

Therefore, we have reduced the global error to the sum of the local errors, whose leading order terms are given by the expectations of ψ_1. By Lemma 3.13, ψ_1 has polynomial growth. Moreover, we know that $\overline{X}$ has bounded moments – see the proof of Theorem 3.10. This implies the bound

$$\left|E[\psi_1(t_i,\overline{X}_i)]\right| \le C$$

by a constant C only depending on the problem and on T, but not on h. Thus, we have

$$\left|E\left[f(\overline{X}_N)\right] - u(0,x)\right| \le C\sum_{i=0}^{N-1}(h^2 + O(h^3)) = CN(h^2 + O(h^3)) = CT(h + O(h^2)),$$

implying that the Euler method has weak order one.

All that is left to prove for the error representation is an integral representation for the error term (3.21). Consider

$$\begin{aligned} \left|h\sum_{i=1}^{N-1} E\left[\psi_1(t_i,\overline{X}_i)\right] - \int_0^T E[\psi_1(t,X_t)]dt\right| &\le h\sum_{i=0}^{N-1}\left|E\left[\psi_1(t_i,\overline{X}_i)\right] - E[\psi_1(t_i,X_{t_i})]\right| + \\ &+ \left|h\sum_{i=0}^{N-1} E[\psi_1(t_i,X_{t_i})] - \int_0^T E[\psi_1(t,X_t)]dt\right|. \end{aligned}$$

For the first term, note that $\left|E\left[\psi_1(t_i, \overline{X}_i)\right] - E[\psi_1(t_i, X_{t_i})]\right| = O(h)$ for each $0 \le i \le N-1$, because $\psi_1(t_i, \cdot)$ satisfies the assumptions imposed on the function f, therefore we can use the already proved first order weak convergence for $f = \psi_1(t_i, \cdot)$. Thus, the first term is $O(h)$. For the second term, note that the function $t \mapsto g(t) := E[\psi_1(t, X_t)]$ is continuously differentiable, and it is a simple calculus exercise to show that

$$\left| h \sum_{i=0}^{N-1} g(t_i) - \int_0^T g(t)dt \right| = O(h)$$

for C^1-functions g. Therefore, also the second term can be bounded by $O(h)$. Inserting these results into (3.21), we indeed obtain

$$E\left[f(\overline{X}_N)\right] - u(0,x) = E\left[u(T, \overline{X}_N) - u(0,x)\right] = h \int_0^T E[\psi_1(t, X_t)]dt + O(h^2).$$

The higher order expansion can now be obtained by continuing the Taylor expansion of Lemma 3.14 to higher order terms. □

Remark 3.15. The error expansion of Theorem 3.11 now allows us to use *Richardson extrapolation* (also known as *Romberg extrapolation*). Given a numerical method for approximating a quantity of interest denoted by A producing approximations $A(h)$ based on steps of size h such that we have an error expansion of the form

$$A - A(h) = a_n h^n + O(h^m), \quad a_n \neq 0,\ m > n.$$

Then we can define an approximation $R(h)$ to A by

$$R(h) = A(h/2) + \frac{A(h/2) - A(h)}{2^n - 1} = \frac{2^n A(h/2) - A(h)}{2^n - 1},$$

leading to a new error $A - R(h) = O(h^m)$.

In the case of the Euler method, this means that we can obtain a method of order two by combining Euler estimates based on step-size h and $h/2$. Indeed, in the setting of Theorem 3.11 even more is true: we could iterate the Richardson extrapolation similar to Romberg's integration rule and obtain numerical methods of arbitrary order. However, higher order extrapolation is usually not considered practical.

Remark 3.16. In the derivation of Theorem 3.11, we have never relied on the fact that the increments ΔB_i^j of the Brownian motion have a normal distribution. All we used to get the first order error representation (and thus the weak order one) was that the first five (mixed) moments of $(\Delta B_j^i : 1 \le i \le d,\ 1 \le j \le N)$ coincide with those of the increments of a Brownian motion, i.e., with a collection of $d \times N$ independent Gaussian random variables with mean zero and variance h. Therefore, we could choose any such sequence of random variables ΔB_j^i, in particular we could use independent discrete random variables such that ΔB_j^i has the same first five moments as $\mathcal{N}(0, h)$. The simplest possible choice is $\Delta B_j^i = \sqrt{h} Y_j^i$, where the Y_j^i are independent copies of the random variable Y defined by

$$Y = \begin{cases} \sqrt{3}, & \text{with probability } 1/6, \\ 0, & \text{with probability } 2/3, \\ -\sqrt{3}, & \text{with probability } 1/6. \end{cases}$$

While this remark also holds true under the assumptions of Theorem 3.17, it is not true for Theorem 3.18, which does depend on particular properties of the normal distribution.

Notice that our proof of Theorem 3.11 mainly relied on smoothness of the solution $u(t, x)$ of the Kolmogorov backward equation. (More precisely, we used that the solution was twice differentiable in time and four times differentiable in space and that theses derivatives are polynomially bounded in order to show that the Euler scheme has weak order one.) In Theorem 3.11, these properties were verified by direct differentiation inside the expectation – using smoothness of f and of the coefficients, via existence of the first and higher variations of the SDE. Of course, this approach can still be done under weaker assumptions. Kloeden and Platen [21, Theorem 14.5.1] is based on this type of arguments:

Theorem 3.17. *Assume that f and the coefficients of the SDE are four times continuously differentiable with polynomially bounded derivatives. Then the Euler method has weak order one.*

It is clear that this method of proof must fail if the payoff function f does not satisfy basic smoothness assumption as in Theorem 3.17. However, there is a second method to get smoothness of u, based on the smoothing property of the heat kernel, see Section A.4. The following result is [2, Theorem 3.1].

Theorem 3.18. *Assume that the vector fields are smooth and all their derivatives, but not necessarily the vector fields themselves, are bounded. Moreover, assume they satisfy the uniform Hörmander condition, cf. Definition A.9. Then, for any bounded measurable function f, the Euler scheme converges with weak order one. Indeed, the error representation* (3.20) *holds with the same definition of the function ψ_1.*

Comparing Theorem 3.18 and Theorem 3.17, we see that the latter has some smoothness assumptions on both the vector fields and the functional f, whereas the former does not impose any smoothness assumption on f, while imposing quite severe assumptions on the vector fields.

Example 3.19. Let us consider an example, where the Euler method actually only converges with order $1/2$ – as guaranteed by the strong convergence. Let the vector fields in Stratonovich formulation be linear, $V_i(x) = A_i x$, $i = 0, 1, 2$, with

$$A_0 = 0, \quad A_1 = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & 0 \end{pmatrix}.$$

Note that the matrices are antisymmetric, i.e., $A_i^T + A_i = 0$, implying that the vector fields are tangent to the unit sphere $D = \left\{ x \in \mathbb{R}^3 \;\middle|\; |x| = 1 \right\}$ in $\mathbb{R}^3$. Since we are using the Stratonovich formulation, this means that the solution X_t will always stay on the unit sphere provided that the starting value x is chosen from D. Now consider $f(x) = (|x|-1)^+$, clearly a Lipschitz continuous but otherwise non-smooth function. The vector fields, on the other hand, are smooth, all derivatives are bounded, but they do not satisfy the uniform Hörmander condition. Take the starting value $x = (1, 0, 0)$, time horizon $T = 1$. Then the exact value is $E[f(X_T)] = 0$. The weak error from the Euler scheme (together with the Milstein scheme treated later in these notes) is plotted in Figure 3.5. We clearly see the order of convergence $1/2$.

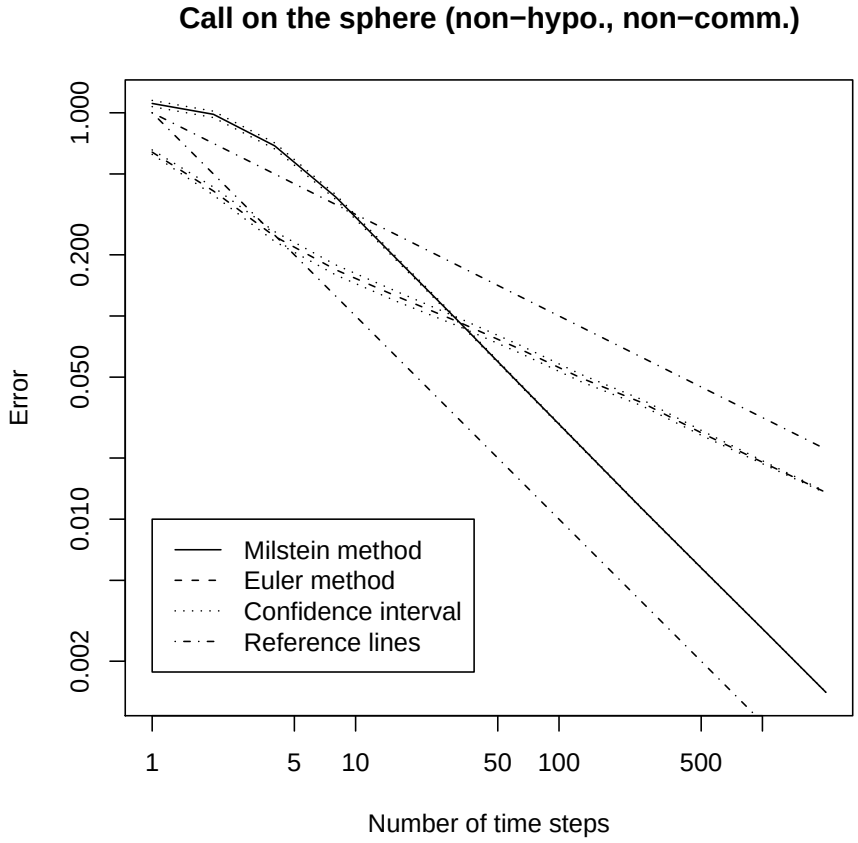

Figure 3.5: Weak error for Example 3.19

In many situations, we can expect the Euler scheme to converge with weak order one, even if the assumptions of neither Theorem 3.17 nor Theorem 3.18 are satisfied. This is especially true, if the process "does not see" the singularities, e.g., because they are only met with probability zero. This is the case in many financial applications, involving standard payoffs like the call or put options. The point of Example 3.19 is that here the functional f is non-smooth on the unit sphere, i.e., the set of points, where f is not smooth has probability one under the law of the solution of the SDE.

The Euler scheme for SDEs driven by Lévy processes

Let Z_t denote a d-dimensional Lévy process and consider the stochastic differential equation

$$dX_t = \sum_{i=1}^{d} V_i(X_{t-})dZ_t^i, \tag{3.22}$$

for vector fields $V_1, \ldots, V_d : \mathbb{R}^n \to \mathbb{R}^n$. For a given function $f : \mathbb{R}^n \to \mathbb{R}$, we want to approximate $u(t, x) := E[f(X_t)|X_0 = x]$ – assuming, of course, that the expectation ex-

ists. We know that u satisfies the partial integro-differential equation (B.3). The Euler scheme for the SDE (3.22) is defined in the same way as the Euler scheme (3.13), but the increments $B_{t_{i+1}} - B_{t_i}$ of the Brownian motion have to be replaced by the corresponding increments $Z_{t_{i+1}} - Z_{t_i}$ of the Lévy process Z. By the properties of a Lévy process, these increments are independent of each other, and, if the grid is homogeneous, identically distributed. Nevertheless, sampling the increments might be difficult, depending on the particular Lévy process. We assume here, that exact simulation of the increments is possible. Then, Protter and Talay [33] have shown weak convergence of the Euler scheme. However, it turns out that the rate of convergence is smaller than in the Brownian case (and there might even be no such rate), unless the Lévy measure has finite moments up to some order.

Theorem 3.20. *Assume that the vector fields $V_1, \dots, V_d$ and the functional f are C^4 and that their derivatives of order up to four are bounded. Then the Euler scheme converges in the weak sense to the solution of the SDE* (3.22)*.*
If, moreover, the Lévy measure ν associated to Z has bounded moments up to order eight, then the Euler scheme has weak rate one (among all functions f satisfying the above conditions).

In the proof of the theorem, the Lévy process is approximated by Lévy processes Z_t^m, where jumps with size larger than m are excluded. Then one obtains an error estimate based on the grid size $|\mathcal{D}|$ and on m. In the case of a Lévy measure admitting finite moments of high enough order, this intermediate step can be avoided.

In many cases, we cannot sample the increments of a Lévy process, because we do not know their distribution. Indeed, assume that a Lévy process Z only by its characteristic triple (α, Σ, ν). In general, there is no feasible way to obtain the distribution of the increments ΔZ – the possibility to compute the characteristic function using numerical integration and to apply an inverse Fourier transformation for every point in space is not considered feasible here – see, however, Section 4.2 for a related numerical method. We have seen, however, that every (pure jump) Lévy process can be approximated by compound Poisson processes with drift, see (3.8), and that this approximation can even be improved by adding an (additional) Brownian component in some cases. Assuming that we can sample the increments of all compound Poisson processes, this means that we can sample approximations ΔZ_i^ϵ of the increments ΔZ_i of the Lévy process along a grid $\mathcal{D}$. Since the proof of Theorem 3.20 is based on such approximations, it is natural to suspect that we can still obtain convergence of the Euler scheme using approximate samples from the increments ΔZ_i. The following theorem by Jacod, Kurtz, Méléard and Protter [18] confirms this suspicion.

Theorem 3.21. *Let X be the solution of the SDE* (3.22) *and let $(\zeta_i^h)_{i\in\mathbb{N}}$ be a sequence of i.i.d. random variables satisfying*

$$\forall g \in C_0^4(\mathbb{R}^d): \quad \left|E[g(\zeta_1^h)] - E[g(Z_h)]\right| \le K h u_h \|g\|_{0,4},$$

where $u_h > 0$ (for every h), K is a constant and C_0^4 denotes the space of four times differentiable bounded functions with bounded derivatives. Moreover, assume that the driving vector fields are contained in $C_0^4(\mathbb{R}^n;\mathbb{R}^n)$ and that Z has finite moments of order up to eight. Define the approximate Euler scheme for the partition $\mathcal{D} = \{0, h, 2h, \dots, Nh\}$, $T = Nh$, by $\overline{X}_0 = X_0$ and then

$$\overline{X}_{n+1} = \overline{X}_n + \sum_{i=1}^{d} V_i(\overline{X}_n)\zeta_{n+1}^h, \quad n = 0, \dots, N-1. \tag{3.23}$$

Then for functions $f \in C_0^4(\mathbb{R}^n)$ the weak error satisfies

$$\left|E[f(X_T)] - E[f(\overline{X}_N)]\right| \leq C \max(u_h, h) \|f\|_{0,4} .$$

Remark 3.22. Under even stronger assumptions, one can get an error expansion similar to Theorem 3.11 for the approximate Euler scheme (3.23). Note that in the case of exact simulation of the increments of the Lévy process Z, we can choose $u_h = 0$, which implies that we obtain the same results as in Theorem 3.20 – but under stronger conditions.

A convenient way to obtain approximations ζ_i^h of ΔZ_i is to use the approximating compound Poisson process (with drift) Z^ϵ obtained by removing all jumps smaller than ϵ, see (3.8), and the setting

$$\zeta_i^h := Z_{ih}^\epsilon - Z_{(i-1)h}^\epsilon = \Delta Z_i^\epsilon, \quad i = 1, \ldots, N.$$

Obviously, we would expect u_h to depend on ϵ. Let us assume that Z satisfies the moment conditions of Theorem 3.21. Define

$$\lambda_\epsilon := \nu(\{ z \in \mathbb{R}^n \mid |z| > \epsilon \}) . \tag{3.24}$$

Let us assume that there is a number $\gamma \in [0, 2]$ such that $\lambda_\epsilon \leq C/\epsilon^\gamma$, $\epsilon \leq 1$. Note that this condition is always satisfied for $\gamma = 2$ because of the integrability condition of the Lévy measure. Then one can show (see [18, (3.9)]) that

$$u_h = \epsilon^{3-\gamma} \tag{3.25}$$

is a possible choice. In order to get $u_h \sim h$ (implying weak order one in Theorem 3.21) we thus have to choose $\epsilon \sim h^{1/(3-\gamma)}$.

Under some conditions, the quality of approximation can be further improved by the jump-diffusion approximation (3.10), see Kohatsu-Higa and Tankov [23].

The Euler-Monte-Carlo method

The Euler method only solves half the problem in determining the quantity $E[f(X_T)]$, when X_T is given as the solution of an SDE. Indeed, it replaces the unknown random variable X_T by a known random variable $\overline{X}_N$, which we can sample in a straightforward way – assuming that we can sample the increments of the driving Lévy process. Therefore, we want to approximate $E[f(X_T)]$ by $E[f(\overline{X}_N)]$. This leaves us with an integration problem as treated in Chapter 2. Of course, in most cases we cannot integrate f explicitly with respect to the law of $\overline{X}_N$, so we will use (Quasi) Monte Carlo simulation.

Remark 3.23. Given an SDE driven by a d-dimensional Lévy process (assuming that no component is deterministic), $\overline{X}_N$ is a function of the increments $(\Delta Z_n^i)_{i=1,\ldots,d;n=1,\ldots,N}$. Thus, the integration problem to compute $E\left[f\left(\overline{X}_N\right)\right]$ presents itself naturally as an integral on $\mathbb{R}^{Nd}$ (with respect to the law of (ΔZ_n^i). Therefore, the dimension of the integration problem can be large, even if the dimension of the model itself is small, if we have to choose N large.

In the end, we approximate our quantity of interest $E[f(X_T)]$ by a weighted average of copies of $f(\overline{X}_N)$, which are either chosen to be random, independent of each other in the case of Monte Carlo simulation, or deterministic according to a sequence of

low discrepancy in the case of Quasi Monte Carlo. Of course, this gives us a natural decomposition of the (absolute) computational error into two parts:

$$\text{Error} = \left| E[f(X_T)] - \frac{1}{M}\sum_{i=1}^{M} f\left(\overline{X}_N^{(i)}\right)\right| \tag{3.26}$$
$$\leq \left|E[f(X_T)] - E\left[f\left(\overline{X}_N\right)\right]\right| + \left|E\left[f\left(\overline{X}_N\right)\right] - \frac{1}{M}\sum_{i=1}^{M} f\left(\overline{X}_N^{(i)}\right)\right|.$$

The first part captures the error caused by the approximation method to the SDE, therefore, we call it the *discretization error*. The second part corresponds to the error of our numerical integration method used to integrate f with respect to the law of $\overline{X}_N$. Therefore we call it *integration error*. (If we use the Monte Carlo method, we might also think about the second part as a *statistical error*. For the Quasi Monte Carlo method, this name would not make much sense, however.) Having fixed the discretization method (Euler or higher order as presented below) and the integration method (MC or QMC), the *Euler Monte Carlo* scheme has only two parameters left: the number of paths M for the integration part and the time grid for the discretization of the SDE. For simplicity, let us work with homogeneous grids only. Then the time grid is uniquely specified by the grid size N (in the sense that the corresponding grid is $\{0 = t_0 < t_1 = T/N < \cdots < t_N = T\}$). Ignoring possible cancellation effects, it is clear that the computational error will be decreased by increasing M (reducing the integration error) and N (reducing the discretization error). On the other hand, it would not be efficient, say, to choose N very large, if M is chosen comparatively small, so that the discretization error is completely overshadowed by the integration error: in an efficient setup, both error contributions should have the same order of magnitude. This suggests that we should not choose M and N independent of each other.

Let us make a more careful analysis. Depending on whether we use MC or QMC, the integration error satisfies

$$\text{Error}_{\text{Int}}(M) \leq C_I M^{-q}, \quad q \in \left\{\frac{1}{2}, 1-\delta\right\},$$

for any $\delta > 0$. Moreover, assume that the discretization error is bounded by

$$\text{Error}_{\text{Disc}}(N) \leq C_D N^{-p}.$$

For the Euler method, p is either one or $1/2$. In the sequel, we will also present other discretization methods with higher order p. A priori, C_I will depend on N – in the case of the Monte Carlo simulation, it is the standard deviation of $f(\overline{X}_N)$. However, asymptotically it is equal to a constant independent of N, namely the standard deviation of $f(X_T)$. So we assume that both C_I and C_D are independent of N and M. In the following, "$\approx$" will mean equality up to a constant. In a real life computation, we want to obtain the quantity of interest $E[f(X_T)]$ with an error tolerance ϵ. (In many cases, the error tolerance would be understood with respect to the relative error, not the absolute one. On the other hand, these two concepts are roughly equivalent, if we know the order of magnitude of the quantity of interest before hand, an assumption which we make here.) On the other hand, we want to reach this objective using as little computer time as possible. Obviously, the computational work for the Euler Monte Carlo method is proportional to MN. These considerations have, thus, led us to a constraint optimization problem of finding

$$\min\{\, MN \;\big|\; C_I M^{-q} + C_D N^{-p} \leq \epsilon \,\}. \tag{3.27}$$

The Lagrangian of this optimization problem is given by

$$F(M, N, \lambda) = MN + \lambda(C_I M^{-q} + C_D N^{-p} - \epsilon).$$

The condition $\frac{\partial F}{\partial N} = 0$ leads to $M \approx \lambda N^{-(p+1)}$. In order to obtain λ, we set $\frac{\partial F}{\partial M} = 0$, giving us

$$\lambda \approx N^{p+1+p/q}, \quad M \approx N^{p/q}.$$

Inserting this in the error bound, reveals that both the integration and the discretization error or of order N^{-p}, as we have already hinted above. More precisely, we see that $\epsilon \approx N^{-p}$, implying that we need to choose $N \approx \epsilon^{-1/p}$ and $M \approx \epsilon^{-1/q}$. Then, the computational cost to compute the quantity of interest with a error bounded by ϵ is proportional to $\epsilon^{-(1/p+1/q)}$. We summarize our results as a proposition.

Proposition 3.24. *Given a discretization scheme with weak order p and an integration method with order q the optimal choice of the number of timesteps N and the number of paths M is to choose M (asymptotically) proportional to $N^{p/q}$. Moreover, the computational cost for obtaining the quantity of interest with a computational error bounded by a tolerance ϵ is (asymptotically) proportional to $\epsilon^{-\left(\frac{1}{p}+\frac{1}{q}\right)}$.*

If the work in order to guarantee an error bounded by ϵ is proportional to ϵ^{-k}, then one might call k the order of complexity of the problem. The consequence is clear: in order to reduce the computational error by a factor c, the computational cost will grow by a factor c^k. In Table 3.2 we have collected the order of complexity for certain

Problem description	p	q	$M(N)$	k
Euler (Lipschitz) + MC	$1/2$	$1/2$	N	4
Euler (Lipschitz) + QMC	$1/2$	$1-\delta$	$N^{1/2+\delta}$	$3+\delta$
Euler (regular) + MC	1	$1/2$	N^2	3
Order p + MC	p	$1/2$	N^{2p}	$2+1/p$

Table 3.1: Complexity of the Euler Monte Carlo method

scenarios. For instance, if the payoff and/or the vector fields are so irregular that the Euler method only has weak order $1/2$, and we use the MC simulation for integration, then M and N should be chosen proportionally to each other and the overall order of complexity is four. In the generic case, i.e., when the Euler method has weak order one, the order of complexity is three and M is chosen to be proportional to N^2. The table also shows that higher order discretization schemes for the SDE cannot really improve the overall computational cost significantly, when combined with a low order integration method. For instance, if we use Monte Carlo simulation, then increasing the weak order from $1/2$ to 1 decreases the order of complexity from 4 to 3. But then a further increase of the weak order to 2, 3 and 4 will only lead to decreases of the order of complexity to 2.5, 2.33 and 2.25, respectively. Given that higher order methods are usually more difficult to implement and computationally more costly (thereby possibly increasing the constant in the complexity), it might not be worthwhile to implement such methods, if we are only using Monte Carlo simulation. (In principle, the same holds true for QMC, but then second order methods might still be a good choice.). Of course, this is only a very rough comparison, and in special applications we might get a completely different picture.

3.3 Advanced methods

Stochastic Taylor schemes

Stochastic Taylor expansion

The Euler discretization of the SDE (3.11) (driven by a Brownian motion) was inspired by a Taylor expansion of the solution of an ODE, giving the deterministic Euler method. In the deterministic case, higher order Taylor expansions are obviously possible, thus leading to higher order numerical schemes for ODEs. However, it is not obvious how to extend these schemes to the stochastic situation: after all, the Brownian motion behaves fundamentally different than bounded variation functions, especially for small times. Fortunately, there is a remedy in the form of a genuinely *stochastic Taylor expansion*. Because of the easier form of the chain rule, we prefer to work with the Stratonovich formulation

$$dX_t = V_0(X_t)dt + \sum_{i=1}^{d} V_i(X_t) \circ dB_t^i$$

instead of the Ito formulation. Recall that Ito's formula implies that for every function f smooth enough

(3.28)

$$f(X_t) = f(X_0) + \int_0^t V_0 f(X_s)ds + \sum_{i=1}^{d} \int_0^t V_i f(X_s) \circ dB_s^i = f(X_0) + \sum_{i=0}^{d} \int_0^t V_i f(X_s) \circ dB_s^i$$

by appending time as an additional component to the Brownian motion, formally $B_t^0 := t$ (with the corresponding integral being the usual Lebesgue integral). For simplicity, assume that both f and the vector fields $V_0, \dots, V_d$ are smooth. Then, we can also apply the Ito formula to the function $x \mapsto V_i f(x)$, giving

$$V_i f(X_s) = V_i f(X_0) + \sum_{j=0}^{d} \int_0^s V_j V_i f(X_u) \circ dB_u^j.$$

Inserting this into (3.28), we get

$$f(X_t) = f(X_0) + \sum_{i=0}^{d} V_i f(X_0) \int_0^t \circ dB_t^i + \sum_{i,j=0}^{d} \int_0^t \int_0^s V_j V_i f(X_u) \circ dB_u^j \circ dB_s^i.$$

Iterating this procedure, gives an expansion of $f(X_t)$ in terms of the iterated integrals of the Brownian motion. Before we turn to the exact statement, we also need to think about the *order* of our expansion. Clearly, B^0 and B^i, $i = 1, \dots, d$, have a different scaling. Indeed, $B_t^0 = t = tB_1^0$, whereas $B_t^i \sim \sqrt{t}B_1^i$, "$\sim$" meaning equality in law. In fact, this scaling propagates to the iterated integrals. For some multi-index $I = (i_1, \dots, i_k) \in \{0, \dots, d\}^k$, $k \in \mathbb{N}$, denote

$$B_t^I := \int_{0 \le t_1 \le \cdots \le t_k \le t} \circ dB_{t_1}^{i_1} \cdots \circ dB_{t_k}^{i_k}.$$

Then we have the scaling

$$B_t^I \sim t^{\deg(I)/2} B_1^I, \tag{3.29}$$

where $\deg(I) := k + \#\left\{ 1 \le j \le k \,\middle|\, i_j = 0 \right\}$ for $I = (i_1, \dots, i_k)$, i.e., we have to count zeros twice, in order to account for the higher order of B^0 compared to the Brownian motion.

The scaling of the iterated Brownian integrals suggests that we should only include terms with degree $\deg(I) \le m$ for a stochastic Taylor expansion of order m (or rather $m/2$) instead of including all terms with multi-indices of length smaller or equal m, because the latter expansion would include terms of much higher degree of the scaling in t. This leads to the following

Theorem 3.25. *Assume that the function f and the vector fields are C^{m+1}, bounded with bounded derivatives for some $m \in \mathbb{N}$, and that $X_0 = x \in \mathbb{R}^n$. Then*

$$f(X_t) = f(x) + \sum_{\substack{I=(i_1,\dots,i_k)\in\{0,\dots,d\}^k \\ \deg(I)\le m,\ k\le m}} V_{i_1} \cdots V_{i_k} f(x) B_t^I + R_m(t, x, f)$$

for an error term R_m satisfying (for $t < 1$)

$$\sup_{x\in\mathbb{R}^n} E\left[R_m(t,x,f)^2\right]^{1/2} \le C t^{(m+1)/2} \sup_{m<\deg(I)\le m+2} \left\| V_{i_1} \cdots V_{i_k} f \right\|_\infty .$$

Proof. Iterate the procedure as described above until you get the expansion of $f(X_t)$ in terms of iterated integrals up to order m with remainder terms of the form

$$\int_{0\le t_1\le\cdots\le t_k\le t} V_{i_1} \cdots V_{i_k} f(X_{t_1}) \circ dB_{t_1}^{i_1} \cdots \circ dB_{t_k}^{i_k},$$

with $\deg(I) > m$. To show the error bound stated above, we have to transform the Stratonovich integrals to Ito integrals, and then repeatedly apply the Ito isometry. □

Remark 3.26. Re-doing the stochastic Taylor expansion of Theorem 3.25 in terms of Ito integrals gives a similar expansion in terms of iterated Ito integrals. More precisely, let $\widetilde{B}_t^I$ denote the iterated Ito integral of the Brownian motion B with respect to the multi-index I. Moreover, let $\widetilde{V}_0$ be the partial differential operator defined by

$$\widetilde{V}_0 f(x) = \nabla f(x) \cdot V(x) + \frac{1}{2} \sum_{i=1}^{d} V_i(x)^T H f(x) V_i(x),$$

where Hf denotes the Hessian matrix of f. Note that $\widetilde{V}_0$, contrary to V_0, is a second order differential operator, and therefore not a vector field. Then the stochastic Taylor expansion in Ito calculus is obtained by replacing every "B_t^I" with "W_t^I" and every "V_0" with "$\widetilde{V}_0$". For simplicity, we shall also denote $\widetilde{V}_i := V_i$, $i = 1, \dots, d$.

Stochastic Taylor schemes

For $m \in \mathbb{N}$ set

$$\mathcal{A}_m := \left\{ (i_1, \dots, i_k) \in \{0, \dots, d\}^k \,\middle|\, 1 \le k \le m,\ \deg((i_1, \dots, i_k)) \le m \right\},$$

$$\mathcal{A}_m^* := \left\{ (i_1, \dots, i_k) \in \{0, \dots, d\}^k \,\middle|\, 1 \le k \le m,\ \deg((i_1, \dots, i_k)) \le m \text{ or } k = \#\left\{ j | i_j = 0 \right\} = (m+1)/2 \right\}.$$

Moreover, we write $V_I = V_{i_1} \cdots V_{i_k}$ for $I = (i_1, \ldots, i_k)$ and likewise for the Ito expansion. Then we may write the stochastic Taylor expansion of Theorem 3.25 more compactly as

$$f(X_t) = f(x) + \sum_{I \in \mathcal{A}_m} V_I f(x) B_t^I + R_m(t, f, x) = f(x) + \sum_{I \in \mathcal{A}_m} \widetilde{V}_I f(x) \widetilde{B}_t^I + \widetilde{R}_m(t, f, x),$$

depending on whether we use the Ito or Stratonovich version. For our numerical scheme, we rather want to approximate the solution X_t itself, instead of $f(X_t)$. To this end, use the identity function, i.e., $\mathrm{id} : x \mapsto x$ and insert it into the above equation. Then we get, defining $V_I \coloneqq V_I \,\mathrm{id}$,

$$X_t = x + \sum_{I \in \mathcal{A}_m} V_I(x) B_t^I + R_m(t, \mathrm{id}, x) = x + \sum_{I \in \mathcal{A}_m} \widetilde{V}_I(x) \widetilde{B}_t^I + \widetilde{R}_m(t, \mathrm{id}, x).$$

(Strictly speaking, we should insert all the coordinate functions $x \mapsto x^i$, $i = 1, \ldots, n$, and then combine these n components again.)

Given a time grid $\mathcal{D} = \{0 = t_0 < t_1 < \cdots < t_N = T\}$, we thus define the general *Taylor schemes* as follows.

Definition 3.27. The *strong stochastic Taylor scheme* of order $m \in \mathbb{N}$ for the SDE (3.11) is defined by $\overline{X}_0 = X_0$ and

$$\overline{X}_{j+1} \coloneqq \overline{X}_j + \sum_{I \in \mathcal{A}_m^*} \widetilde{V}^I(x) \Delta \widetilde{B}_j^I,$$

where

$$\Delta \widetilde{B}_j^I \coloneqq \int_{t_j \le s_1 \le s_2 \le \cdots \le s_k \le t_{j+1}} dB_{s_1}^{i_1} \cdots dB_{s_k}^{i_k},$$

when $I = (i_1, \ldots, i_k)$.

Remark 3.28. Note that $\Delta \widetilde{B}_j^I \sim \widetilde{B}_{\Delta t_j}^I \sim \sqrt{\Delta t}^{\deg(m)} \widetilde{B}_1^I$.

Remark 3.29. We can also define a stochastic Taylor scheme based on the stochastic Taylor expansion in terms of Stratonovich integrals.

For the proof of the following Theorem we refer to Kloeden and Platen [21, Theorem 10.6.3]

Theorem 3.30. *Given $m \in \mathbb{N}$ and assume that all the functions $\widetilde{V}^I$ for $I \in \mathcal{A}_m$ are twice differentiable. Then the strong stochastic Taylor scheme converges strongly to the solution X_T of the SDE with strong order $m/2$.*

Example 3.31. The Euler scheme is the strong Taylor scheme for $m = 1$.

The relevant indices which need to be included in the scheme if we are interested in weak convergence are different, thereby leading to a different set of schemes.

Definition 3.32. The *weak stochastic Taylor scheme* of order $m \in \mathbb{N}$ for the SDE (3.11) is defined by $\overline{X}_0 = X_0$ and

$$\overline{X}_{j+1} \coloneqq \overline{X}_j + \sum_{I \in \bigcup_{1 \le k \le m} \{0, \ldots, d\}^k} \widetilde{V}^I(x) \Delta \widetilde{B}_j^I.$$

Example 3.33. The Euler scheme is also a weak Taylor scheme of order $m = 1$. (Note that the Hessian matrix of the function id is zero, thus the second order part of $\widetilde{V}^0$ vanishes when applied to id.)

The weak convergence proof can be found in Kloeden and Platen [21, Theorem 14.5.1].

Theorem 3.34. *Let the vector fields $V, V_1, \dots, V_d$ be $2(m+1)$-times continuously differentiable. Then the weak stochastic Taylor scheme converges with weak order m for every $2(m+1)$-times continuously differentiable function, which is together with its derivative of polynomial growth.*

Remark 3.35. Define a *weak Stratonovich Taylor scheme* by $\overline{X}_0 = X_0$ and

$$\overline{X}_{j+1} \coloneqq \overline{X}_j + \sum_{I \in \mathcal{A}_m} V^I(\overline{X}_j) \Delta B_j^I.$$

This scheme converges with weak order $(m-1)/2$, provided that the functions and vector fields are regular enough. Indeed, Theorem 3.25 basically shows that the local error is of order $(m+1)/2$, and the usual summation trick gives the global order $(m-1)/2$.

Example 3.36. The *Milstein scheme* is a scheme of weak and strong order one. In fact, it is the strong scheme of order $m = 2$. Thus, it is given by

$$\overline{X}_{j+1} = \overline{X}_j + V(\overline{X}_j)\Delta t_j + \sum_{i=1}^{d} V_i(\overline{X}_j)\Delta B_j^i + \sum_{(i_1,i_2)\in\{1,\dots,d\}^2} V^{(i_1,i_2)}(\overline{X}_j)\Delta \widetilde{B}_j^{(i_1,i_2)}.$$

Here, $V^{(i_1,i_2)}(x) = DV_{i_2}(x) \cdot V_{i_1}(x)$ and

$$\Delta \widetilde{B}_j^{(i_1,i_2)} = \int_{t_j}^{t_{j+1}} B_s^{i_1} dB_s^{i_2}.$$

In the case $i_1 = i_2$, we can easily compute $\Delta B_j^{(i,i)} = (\Delta B_j^i)^2 - \Delta t_j$. However, for $i_1 \neq i_j$, there is no explicit formula in terms of the Brownian increments. Notice that we do not need to sample the iterated integrals if the vector fields commute, i.e., if $[V_{i_1}, V_{i_2}] = 0$ for all choices of i_1 and i_2. See Figure 3.5 for an example of the strong order convergence of the Milstein scheme.

Sampling of iterated integrals

Apart from the higher complexity of the higher order (strong or weak) Taylor schemes, the biggest obstacle for a successful implementation is certainly the need to sample the iterated integrals of Brownian motion. Indeed, every step of the higher order Taylor schemes requires a sample of

$$(B_1^I)_{I\in\Gamma}$$

for some set Γ of multi-indices. (Of course, one could replace B^I by $\widetilde{B}^I$). Note that the increments used in Definition 3.27 and Definition 3.32 are scaled independent copies of the above random variables. The simples example is provided by the Milstein scheme for an SDE driven by a two-dimensional Brownian motion. Then for one step of the scheme, we need to sample

$$(B_1^1, B_1^2, A_1), \quad A_t \coloneqq \int_0^t B_s^1 dB_s^2 - \int_0^t B_s^2 dB_s^1.$$

A_t is called *Lévy's area*. Lévy's area is one of the fundamental processes of stochastic analysis, thus it is a very well-studied object. For instance, there is an explicit formula for the characteristic function of (B^1, B^2, A). On the other hand, there is no exact way to sample from its distribution. In particular, it is not possible to obtain a density of (B_1^1, B_1^2, A_1), which would allow one to apply acceptance-rejection. (Note that an explicit density for A_t itself is known, see for instance Protter [34, Theorem II.43]. But we really need the joint distribution here.)

Of course, there are some inexact sampling techniques. The easiest is probably to approximate the Ito integrals in the definition of A by Riemann sums, i.e., for a partition $\mathcal{D} = \{0 = t_0 < \cdots < t_N = 1\}$ we can approximate

$$B_1^1 = \sum_{j=1}^N \Delta B_j^1,\ B_1^2 = \sum_{j=1}^N \Delta B_j^2,\ A_1 \approx \sum_{i=1}^N \underbrace{B_{t_{j-1}}^1}_{=\sum_{l=1}^{j-1} \Delta B_l^1} \Delta B_j^2 + \sum_{i=1}^N \underbrace{B_{t_{j-1}}^2}_{=\sum_{l=1}^{j-1} \Delta B_l^2} \Delta B_j^1.$$

However, this sampling method would certainly not lead to a competitive numerical method.

Remark 3.37. For weak approximation, we could replace (B_1^1, B_1^2, A_1) by a different distribution with the same moments up to order two. Then, we still obtain the correct weak convergence, and can sample the increments. This trick can, naturally, also be applied to higher order Taylor schemes. However, in many applications one wants to avoid these moment matched increments, and prefers sampling the true increments.

Multilevel Monte Carlo simulation

In typical situations, the computational work necessary to achieve an (absolute) error bounded by ϵ using the Euler Monte Carlo method is of order $O(\epsilon^{-3})$, as we have seen in Proposition 3.24. Giles [14], [15] has constructed a method, which leads to a considerably smaller order of complexity, by a clever combination of simulation of the Euler scheme (or more general schemes) at different time grids. More precisely, fix a time horizon T and consider homogeneous grids given by the time increment $h = \Delta t$. Let X_t denote the solution of an SDE (3.11) driven by a Brownian motion. We want to compute $E[f(X_T)]$ for a given functional of the solution of the SDE. We approximate X by approximations $\overline{X}^{(h)}$ based on the grid with increments h. Instead of simply applying the Monte Carlo method for the random variable $f(\overline{X}^{(h)})$, our estimate for $E[f(X_T)]$ will be based on a combination of samples from the random variables $\overline{X}^{(h_1)}, \ldots, \overline{X}^{(h_L)}$ for a sequence $h_1 > \cdots > h_L$, in such a way that the bias of the estimate, i.e., the discretization error, is given by the discretization error on the finest level, i.e., the discretization error corresponding to h_L, whereas the computational work is some average of the computational works associated to the different grids. This should give the same error as the method based on h_L, whereas the computational work is strongly reduced.

In order to understand the idea of multilevel Monte Carlo, let us remember the control variates technique for reducing the variance in an ordinary Monte Carlo problem (to compute $E[f(X)]$). There the idea was to find a random variable Y which is similar to X and a function g such that $I[g; Y] = E[g(Y)]$ is explicitly known. (It turned out that "similarity" meant that the correlation of $f(X)$ and $g(Y)$ was high.) Then $f(X)$ is replaced by $f(X) - \lambda(g(Y) - I[g; Y])$, which has the same expected value, but much smaller variance, if Y and g were wisely chosen. In our case, we want to compute

the expectation of $f(\overline{X}^{(h_L)})$ – which is itself a biased estimate of $E[f(X_T)]$. How can we find another random variable Y "close" to X with known expectation $E[f(Y)]$? If we believe in the (strong) convergence of our method, we also believe that $\overline{X}^{(h_L)}$ and $\overline{X}^{(h_{L-1}}$ should be close, which implies that the covariance of $f(\overline{X}^{(h_L)})$ and $f(\overline{X}^{(h_{L-1})})$ is high, but this choice does not seem to qualify since we do not know the expectation of $f(\overline{X}^{(h_{L-1})})$. Notice, however, that it is much cheaper to sample $f(\overline{X}^{(h_{L-1})})$ as opposed to $f(\overline{X}^{(h_L)})$, since the grid corresponding to h_L contains h_{L-1}/h_L more points than the grid corresponding to h_{L-1}. Therefore, Monte Carlo simulation to get a good estimate of the expectation of $f(\overline{X}^{(h_{L-1})})$ is much cheaper. Therefore, the first step for multilevel Monte Carlo is:

1. Compute an estimate of $E[f(\overline{X}^{(h_{L-1})})]$ using Monte Carlo simulation.
2. Compute an estimate for $E[f(\overline{X}^{(h_L)})]$ using variance reduction based on $f(\overline{X}^{(h_{L-1})})$.

Methods of this form are also known as "quasi control variates". Now we iterate the idea, by using variance reduction based on $f(\overline{X}^{(h_{L-2})})$ in order to compute $E[f(\overline{X}^{(h_{L-1})})]$, which we need for the computation of $E[f(\overline{X}^{(h_L)})]$. We shall see below that this method is, indeed, more efficient than simple Monte Carlo simulation at the finest grid.

Before we go on, let us first reflect for a moment on the relation between $\overline{X}^{(h_L)}$ and $\overline{X}^{(h_{L-1})}$. Usually, we only cared about the law of our approximations, not on the approximations as actual random variables. Here we have to treat them as random variables, because we need to sample $\overline{X}^{(h_L)}(\omega)$ and $\overline{X}^{(h_{L-1})}(\omega)$ for the same ω in the control variates technique. This can be easily achieved in the following way: sample the Brownian motion on the finer grid and compute $\overline{X}^{(h_L)}$ based on the sampled Brownian increments. If the coarser grid is actually contained in the finer grid (as will be the case below), then add the Brownian increments along the fine grid to obtain the corresponding increments on the coarse grid, and use them to obtain $\overline{X}^{(h_{L-1})}$. Otherwise, we need to use a Brownian bridge construction to obtain the Brownian increments on the coarse grid based on those along the fine grid.

Before finally formulating the main result of multilevel Monte Carlo, let us first introduce some notation. Fix some $N \in \mathbb{N}$, $N > 1$, and define the step sizes $h_l := N^{-l}T$, $l = 0, \dots, L$. Let $P_l := f(\overline{X}^{(h_l)})$ denote the payoff given by the numerical approximation along the grid with step-size h_l. Moreover, let I_l denote the Monte Carlo estimator based on M_l samples $P_l^{(i)} - P_{l-1}^{(i)}$ of $P_l - P_{l-1}$ for $l > 0$ and on P_0 for $l = 0$, i.e.,

$$I_l := I_{M_l}[P_l - P_{l-1}] = \frac{1}{M_l} \sum_{i=1}^{M_l} \left(P_l^{(i)} - P_{l-1}^{(i)} \right).$$

We assume the estimators I_l to be independent of each other.

Theorem 3.38. *Assume that there are constants $\alpha \geq 1/2$, $C_1, C_2, \beta > 0$ such that $E[f(X_T) - P_l] \leq C_1 h_l^\alpha$ and $\operatorname{var}[I_l] \leq C_2 h_l^\beta M_l^{-1}$. Then there is $L \in \mathbb{N}$ and there are choices $M_0, \dots, M_L$ such that the multilevel estimator $I := \sum_{l=0}^{L} I_l$ satisfies*

$$\sqrt{E\left[(I - E[f(X_T)])^2\right]} \leq \epsilon$$

and the computational work C is bounded by

$$C \leq \begin{cases} C_3\epsilon^{-2}, & \beta > 1, \\ C_3\epsilon^{-2}(\log \epsilon)^2, & \beta = 1, \\ C_3\epsilon^{-2-(1-\beta)/\alpha}, & 0 < \beta < 1. \end{cases}$$

Corollary 3.39. *Assume that the Euler method has weak order* 1 *and strong order* 1/2 *for the problem at hand. Choose* $L = \frac{\log(\epsilon^{-1})}{\log N} + O(1)$ *in* ϵ *and choose* M_l *proportional to* $\epsilon^{-2}(L+1)h_l$*. Then the multilevel estimator has computational error* $O(\epsilon)$*, while the computational cost is* $O(\epsilon^{-2}(\log \epsilon)^2)$*.*

Proof. Note that the corollary follows from the theorem by choosing $\alpha = 1$ and $\beta = 1$. However, for simplicity we only give (sketch of) a proof of the corollary, but not of the theorem.

Let L be defined by

$$L := \left\lfloor \frac{\log(\sqrt{2}C_1 T\epsilon^{-1})}{\log N} \right\rfloor$$

implying that $\epsilon/(\sqrt{2}M) < C_1 h_L \leq \epsilon/\sqrt{2}$, and thus

$$(E[I] - E[f(X_T)])^2 \leq \frac{\epsilon^2}{2}.$$

Moreover, choosing

$$M_l := \left\lceil 2\epsilon^{-2}(L+1)C_2 h_l \right\rceil, \quad l = 0, \ldots, L,$$

we have

$$\mathrm{var}[I] = \sum_{l=0}^{L} \mathrm{var}[I_l] \leq \sum_{l=0}^{L} C_2 \frac{h_l}{M_l} \leq \frac{1}{2}\epsilon^2.$$

Thus, the means square error satisfies

$$\begin{aligned} E\left[(I - E[f(X_T))^2]\right] &= E[I^2] - 2E[I]E[f(X_T)] + E[f(X_T)]^2 \\ &= \mathrm{var}[I] + (E[I] - E[f(X_T)])^2 \leq \epsilon^2, \end{aligned}$$

and we are only left to compute the computational cost C.

We assume ϵ to be small enough. Then $L + 1 \leq C\log(\epsilon^{-1})$ for some constant C varying from line to line. Moreover, we bound $M_l \leq 2\epsilon^{-2}(L+1)C_2 h_l + 1$. Then

$$\begin{aligned} C \leq C\sum_{l=0}^{L} \frac{M_l}{h_l} \leq \sum_{l=0}^{L} \left(2\epsilon^{-2}(L+1)C_2 + h_l^{-1}\right) &\leq 2\epsilon^{-2}(L+1)^2 C_2 + \sum_{l=0}^{L} h_l^{-1} \\ &\leq 2\epsilon^{-2}\log(\epsilon^{-1})^2 C_2 + \sum_{l=0}^{L} h_l^{-1}. \end{aligned}$$

By the geometric series, an elementary inequality and the definition of L, we have

$$\begin{aligned} \sum_{l=0}^{L} h_l^{-1} = h_L^{-1}\sum_{l=0}^{L} N^{-l} = h_L^{-1}\frac{N^{-(L+1)} - 1}{N^{-1} - 1} &< h_L^{-1}\frac{N}{N-1} \\ &\leq \frac{N^2}{N-1}\sqrt{2}C_1\epsilon^{-1} \leq \frac{N^2}{N-1}\sqrt{2}C_1\epsilon^{-2}, \end{aligned}$$

provided that $N > 1$. This implies that

$$C \leq C\epsilon^{-2} \log(\epsilon^{-1})^2. \qquad \square$$

Chapter 4

Deterministic methods

4.1 The finite difference method

The Black-Scholes PDE

Before going to the numerical treatment of option pricing by solving the associated partial differential equations, let us first recapitulate these PDEs. in what follows we mainly follow Seydel [36]. For the theoretical part see also the notes of Kohn on PDEs for finance.

By the Feynman-Kac formula, see (A.11) for the diffusion case and (B.3) for the case of an SDE driven by a Lévy process the price of a European option $u(t,x)$ as a function of calendar time t and stock price $S_t = x$ satisfies a parabolic partial differential equation. In fact, similar relations also hold for more exotic options, like path dependent options – by enhancing the state space – and American options. For simplicity, let us work in the simplest possible stock model, the Black-Scholes model

$$dS_t = rS_t dt + \sigma S_t dB_t.$$

Then, by (A.11), the price $u(t,x) = E\left[e^{-r(T-t)} f(S_T)\middle| S_t = x\right]$ of a European option with payoff function f satisfies

$$\frac{\partial}{\partial t}u(t,x) + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2}{\partial x^2}u(t,x) + rx\frac{\partial}{\partial x}u(t,x) - ru(t,x) = 0, \tag{4.1}$$

with terminal value $u(T,x) = f(x)$. In the following we assume that f is a call or put option with strike price K.

One of the advantages of the PDE point of view is that it is relatively straightforward to treat American options. Indeed, consider an American put option (in our setting without dividends the American call option would coincide with the European one). Then its price $\widetilde{u}(t,x)$ (again, at time t with $S_t = x$, provided that the option has not been exercised before) satisfies the following conditions:

$$\frac{\partial}{\partial t}\widetilde{u}(t,x) + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2}{\partial x^2}\widetilde{u}(t,x) + rx\frac{\partial}{\partial x}\widetilde{u}(t,x) - r\widetilde{u}(t,x) \le 0, \tag{4.2a}$$

$$\widetilde{u}(t,x) \ge (K-x)_+, \tag{4.2b}$$

$$\widetilde{u}(T,x) = (K-x)_+, \tag{4.2c}$$

where we have equality in (4.2a) whenever there is a strict inequality in (4.2b). It can be shown that problem (4.1) is a *free boundary problem*, i.e., there is an (unknown) value $x_0 = x_0(t)$ such that $\widetilde{u}$ solves the PDE (4.2a) with equality (i.e., the classical Black-Scholes PDE) on the domain $]x_0, \infty[$ and $\widetilde{u}(t,x) = (K-x)_+$ whenever $x \le x_0$. Thus, it is optimal to exercise the American option iff $x < x_0(t)$, and to wait in the other case. If we are above the exercise boundary x_0, the American option (locally) behaves like a European option, and thus also satisfies the Black-Scholes PDE.

If we want to solve the problems (4.1) or (4.2) numerically, we should first try to simplify the PDEs. Introduce some new variables, namely $y = \log(x/K)$ (the *log-moneyness*), $\tau = \frac{1}{2}\sigma^2(T-t)$, $q = 2r/\sigma^2$ and

$$v(\tau, y) \coloneqq \frac{1}{K} \exp\left(\frac{1}{2}(q-1)y + \left(\frac{1}{4}(q-1)^2 + q\right)\tau\right) u(t,x), \tag{4.3}$$

and obtain $\widetilde{v}(\tau, y)$ in the same way from $\widetilde{u}(t,x)$. It is easy to see that the transformed European option price v now satisfies the *heat equation* and to figure out the new boundary condition. For a European put option they read:

$$\frac{\partial}{\partial \tau} v(\tau, y) = \frac{\partial^2}{\partial y^2} v(\tau, y), \quad v(y,0) = \left(e^{\frac{1}{2}(q-1)y} - e^{\frac{1}{2}(q+1)y}\right)_+ . \tag{4.4}$$

Moreover, one can see that (again for a put option)

$$v(\tau, y) = \exp\left(\frac{1}{2}(q-1)y + \frac{1}{4}(q-1)^2\tau\right) \text{ for } y \to -\infty, \quad v(\tau, y) = 0 \text{ for } y \to \infty. \tag{4.5}$$

In the case of an American put option one can show that $\widetilde{v}(\tau, y)$ is solution to the following problem: let $g(y,\tau) \coloneqq \exp\left(\frac{1}{4}(q+1)^2\tau\right)\left(e^{\frac{1}{2}(q-1)y} - e^{\frac{1}{2}(q+1)y}\right)_+$, then

$$\left(\frac{\partial}{\partial \tau}\widetilde{v}(\tau, y) - \frac{\partial^2}{\partial y^2}\widetilde{v}(\tau, y)\right)(\widetilde{v}(\tau, y) - g(\tau, y)) = 0, \quad \frac{\partial}{\partial \tau}\widetilde{v}(\tau, y) - \frac{\partial^2}{\partial y^2}\widetilde{v}(\tau, y) \ge 0, \tag{4.6a}$$

$$\widetilde{v}(\tau, y) \ge g(\tau, y), \quad \widetilde{v}(0, y) = g(0, y), \tag{4.6b}$$

$$\widetilde{v}(\tau, y) = g(\tau, y) \text{ for } y \to -\infty, \quad \widetilde{v}(\tau, y) = 0 \text{ for } y \to \infty. \tag{4.6c}$$

Moreover, one needs to require $\widetilde{v}$ to be continuously differentiable.

Explicit finite differences

For the rest of this section, we change back notation to the more familiar $u(t,x)$ – instead of $v(\tau, y)$. That is, we consider the heat equation

$$\begin{cases} \dfrac{\partial}{\partial t}u(t,x) = \dfrac{\partial^2}{\partial x^2}u(t,x), \quad 0 < t \le T,\ x \in \mathbb{R}, \\ u(0,x) = \left(e^{\frac{1}{2}(q-1)x} - e^{\frac{1}{2}(q+1)x}\right)_+, \quad x \in \mathbb{R}, \\ u(t,x) \sim \exp\left(\dfrac{1}{2}(q-1)x + \dfrac{1}{4}(q-1)^2 t\right) \text{ for } x \to -\infty, \quad u(t,x) \sim 0 \text{ for } x \to \infty, \end{cases}$$

i.e., we consider the transformed European put-option as described in the last subsection. The general idea of the finite difference method is to replace partial derivative by finite difference quotients along a grid, thereby transforming a PDE into a difference equation.

Therefore, we need to discretize time and space, i.e., we need to have a time grid and a space grid. For simplicity, let us work with homogeneous grids only. Then the time grid is determined by its size N, i.e., we set $\Delta t := T/N$ and define the grid points $t_i := i\Delta t$, $i = 0, \dots, N$. For the space grid, we first have to turn our infinite domain $\mathbb{R}$ into a finite domain $[a, b]$. Then the grid is again determined by its size M by setting $\Delta x := (b-a)/M$ and then $x_j := a + j\Delta x$, $j = 0, \dots, M$. The goal of the finite difference method is to determine approximations $v_{i,j}$, $0 \le i \le N$, $0 \le j \le M$, of the values $u_{i,j} := u(t_i, x_j)$.

Remark 4.1. Note that the values of u for large values of $|x|$ will be necessary to set the (approximately) correct boundary conditions at $x = a$ and $x = b$. They are not necessary for the PDE on the domain $\mathbb{R}$.

Remark 4.2. In a multi-dimensional setting, the same construction applies. Note, however, that a grid in $\mathbb{R}^n$ with the same mesh Δx has N^n nodes. Therefore, we need to compute MN^n values $u_{i,j_1,\dots,j_n}$. This is the curse of dimensionality: the computational work for the same accuracy grows exponentially fast in the dimension.

On the other hand, during our finite difference calculation, we compute the option prices $u(t, x)$ for all times t_i and all stock prices x_j, not just the price for one particular time t and one particular stock price x as in the Euler Monte Carlo scheme. It depends on the application, whether this is constitutes a (possibly big) advantage or not.

Next we replace all derivatives in (4.4) by difference quotients. Note that there are many different choices, which will lead to different finite difference schemes. For instance, we can approximate

$$\frac{\partial}{\partial t}u(t_i, x_j) = \frac{u_{i+1,j} - u_{i,j}}{\Delta t} + O(\Delta t) = \frac{u_{i,j} - u_{i-1,j}}{\Delta t} + O(\Delta t). \tag{4.7}$$

For our first method we choose the former approximation of the time derivative. For the space derivative, we choose the approximation

$$\frac{\partial^2}{\partial x^2}u(t_i, x_j) = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta x^2} + O(\Delta x^2). \tag{4.8}$$

Combining these approximations and solving for $u_{i+1,j}$ (or rather its approximation $v_{i+1,j}$), we obtain

$$v_{i+1,j} = v_{i,j} + \frac{\Delta t}{\Delta x^2}(v_{i,j+1} - 2v_{i,j} + v_{i,j-1}).$$

Thus, we use the approximations at time t_i to compute the approximations at time t_{i+1}, and we do so in an *explicit* and linear way. Note that the approximations at time $t_0 = 0$ are given by the initial condition of the PDE, i.e., we set $v_{0,j} := u(0, x_j)$, with $u(0, x)$ given by (4.4). Obviously, the above iteration is not well defined for $j = 0$, since this would require us to use a value $v_{i,-1}$ outside of our grid. Here the boundary conditions (4.5) come into play: we treat a as being close to $-\infty$, and use the corresponding boundary value. We obtain $v_{i,M}$ in a similar way by treating $b = x_M$ as being close to $+\infty$. Combining these considerations using the notation $\lambda := \Delta t/(\Delta x)^2$, we obtain:

$$v_{0,j} = \left(e^{\frac{1}{2}(q-1)x_j} - e^{\frac{1}{2}(q+1)x_j}\right)_+, \quad j = 0, \dots, M, \tag{4.9a}$$

$$v_{i+1,j} = v_{i,j} + \lambda(v_{i,j+1} - 2v_{i,j} + v_{i,j-1}), \quad i = 0, \dots, N-1,\ j = 1, \dots, M-1, \tag{4.9b}$$

$$v_{i+1,0} = \exp\left(\frac{1}{2}(q-1)a + \frac{1}{4}(q-1)^2 t_{i+1}\right), \quad v_{i+1,M} = 0, \quad i = 0, \dots, N-1. \tag{4.9c}$$

For numerical analysis, it is useful to obtain a more "compact" notation for the scheme (4.9). To this end, let us ignore the boundary conditions (4.9c) and just implement the iterations step (4.9b). Let $v^{(i)} = (v_{i,1}, \dots, v_{i,M-1})$ denote the vector of values along the whole space grid (except for the boundary points) for one fixed time node t_i. Then we can express the iteration as

$$v^{(i+1)} = Av^{(i)}, \quad A := \begin{pmatrix} 1-2\lambda & \lambda & 0 & \cdots & 0 \\ \lambda & 1-2\lambda & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \lambda \\ 0 & \cdots & 0 & \lambda & \ddots \end{pmatrix}. \tag{4.10}$$

Therefore, the bulk in the computations of the explicit finite difference scheme (4.9) consists of matrix multiplications $v^{(i+1)} = Av^{(i)}$ with a tridiagonal matrix A. (Strictly speaking, this is only true for zero boundary conditions. However, the analysis remains correct even for our non-trivial boundary conditions (4.9c).)

Example 4.3. Consider the following problem (Seydel [36], Beispiel 4.1): let u solve the heat equation with $u(0, x) = \sin(\pi x)$ on the space domain $[0, 1]$ with boundary condition $u(t, 0) = u(t, 1) = 1$. It is easy to see that the explicit solution for this problem is

$$u(t, x) = \sin(\pi x)e^{-\pi^2 t}.$$

In particular, we obtain $u(0.5, 0.2) = 0.004227$. Next we are going to calculate this value using the finite difference scheme. We fix the space grid by $\Delta x = 0.1$. First we choose a time grid $\Delta t = 0.0005$, i.e., $u(0.5, 0.2) = u_{1000,2}$, and we obtain a reasonably good approximations $v_{1000,2} = 0.00435$. Next, we choose a smaller time grid given by $\Delta t = 0.01$. In this case, we have $u(0.5, 0.2) = u_{50,2}$ and the explicit finite difference scheme gives a value $v_{50,2} = -1.5 \times 10^8$.

Obviously, the second choice of parameters makes the explicit finite difference scheme (4.9) *unstable*, i.e., round-off errors propagate and explode by iterated multiplication with the matrix A. (In this case, the boundary values are in fact trivial.)

It is easy to see that the map $x \mapsto Ax$ is stable in the sense that round-off errors fade out iff the spectral radius of A is smaller than one. By a tedious calculation, one can show that the eigenvalues of A have the form

$$\sigma_k = 1 - 2\lambda\left(1 - \cos\left(\frac{k\pi}{M}\right)\right), \quad k = 1, \dots, M-1. \tag{4.11}$$

Thus, the spectral radius is smaller than one if $\lambda \leq 1/2$. Thus, we have (partially) proved the following

Theorem 4.4. *If we choose the time mesh Δt and the space mesh Δx in such a way that $\Delta t \leq \frac{1}{2}\Delta x^2$, then the explicit finite difference method is stable and converges with error $O(\Delta t) + O(\Delta x^2)$, provided that the given boundary conditions are exact.*

Remark 4.5. Given $N \approx M^2$, we have an error proportional to M^{-2} and the computational work is proportional to M^3. Thus, the computational work needed to get the result with error tolerance ϵ is proportional to $\epsilon^{-3/2}$, which is much better than any

of the complexity estimates given in Table 3.2 for the Euler method or even the complexity estimate in Theorem 3.38 for the multi-level Monte Carlo method. However, the picture changes dramatically in dimension $n > 1$. In this case, the error is still proportional to M^{-2}, but the work is now proportional to M^{2+n}. Thus, we obtain a complexity

$$\text{Work} \approx \epsilon^{-(2+n)/2}.$$

One can see that already in dimension $n > 4$ this crude estimate is much worse than plain Euler Monte Carlo.

Crank-Nicolson

The right hand side of (4.7) an be interpreted both as a forward difference quotient for $\frac{\partial}{\partial t}u(t_i, x_j)$, involving the values $u_{i+1,j}$ and $u_{i,j}$ and as a backward difference quotient,

$$\frac{\partial}{\partial t}u(t_{i+1}, x_j) = \frac{u_{i+1,j} - u_{i,j}}{\Delta t} + O(\Delta t)$$

for $\frac{\partial}{\partial t}u(t_{i+1}, x_j)$. Both of them agree. If we use the central difference quotient (4.8) for the second derivative of u at (t_i, x_j) and (t_{i+1}, x_j), equate them to the respective forward and backward difference quotients and average these two equations, we obtain the *Crank-Nicolson scheme*

$$\frac{v_{i+1,j} - v_{i,j}}{\Delta t} = \frac{1}{2\Delta x^2}\left(v_{i,j+1} - 2v_{i,j} + v_{i,j-1} + v_{i+1,j+1} - 2v_{i+1,j} + v_{i+1,j-1}\right). \tag{4.12}$$

Note that this scheme is not explicit anymore: values of v at time t_{i+1} appear on both sides of the equation, in fact, on the right hand side we even have the three different values $v_{i+1,j-1}$, $v_{i+1,j}$ and $v_{i+1,j+1}$. As a consequence, (4.12) should be understood as a linear equation for $(v_{i+1,j})_{j=1}^{M-1}$ given all the values of $v_{i,j}$. (Schemes like this, where equations have to be solved for every time step, are called *implicit* schemes.) By similar methods as for the explicit finite difference method, one can prove the following theorem (see Seydel [36, Satz 4.4]).

Theorem 4.6. *Assume that the solution u of the heat equation with the given initial and boundary conditions is four times continuously differentiable. Then the solution of the Crank-Nicolson method is stable for every choice of Δx and Δt. Moreover, the solution converges approximation error of the solution of the Crank-Nicolson method is $O(\Delta t^2) + O(\Delta x^2)$.*

4.2 Fourier methods

So far, we have explored several possible ways to compute option prices $e^{-rT}E[g(S_T)]$ using sophisticated techniques like Monte Carlo simulation and finite difference methods. We have not, however, commented on the seemingly most straightforward approach, namely to just compute an integral with respect to the density of the price (under the risk neutral measure). Indeed, let us change variables to the log-spot-price $s_T := \log(S_T)$ and assume that s_T has a density denoted by q_T. Then, the option price is given by

$$e^{-rT}\int_{\mathbb{R}} f(s)q_T(s)ds,$$

where $f(s) := g(\exp(s))$. In most models, the density q_T is not known, but in surprisingly many cases, its Fourier transform $\hat{q}_T$, which coincides with the characteristic function of s_T is known.

Example 4.7. In the Black-Scholes model, we have $s_T = \log(S_0) + \sigma B_T + (r - \frac{1}{2}\sigma^2)T$, and thus the characteristic function of the log-price is given by

$$\phi_T(u) = \exp\left(iu\left(\log(S_0) + (r - \frac{1}{2}\sigma^2)T\right) - \frac{1}{2}\sigma^2 T u^2\right).$$

In order to simplify notations, we will often (but not always) set $S_0 = 1$ and $r = 0$ in the following considerations.

Example 4.8. In Section 3.1 we have introduced a wide class of models called exponential Lévy models, where the log-price is given by a Lévy process X, i.e., $s_T = X_T$. For most Lévy processes, the density is not known explicitly. However, due to the Lévy-Khintchine formula, see Theorem B.5, there is a fairly explicit formula for the characteristic function ϕ_T in terms of the characteristic triple. When the Lévy process id defined in terms of the characteristic triple, it is therefore often possible to get (semi) explicit formulas for the characteristic function of the log-price.

Example 4.9. If the log-price process is given by an *affine model*, see Appendix C, the generalized Fourier transform of the log-price process solves a (generalized) system of Riccati ODEs, see (C.2) and (C.3). Many stochastic volatility models are in affine form, i.e., the log-price is given as the first component of an n-dimensional affine process, where the other components are interpreted as some sort of volatility, e.g., volatility, volatility of volatility and so on. For the interpretation of the Heston model as an affine process see Example C.4.

Remark 4.10. Fourier methods for certain path-dependent and Bermudan options also exist. We will, however, concentrate on the European framework.

Now assume that the payoff function f also has a Fourier transform

$$\hat{f}(z) = \int_{\mathbb{R}} f(s)e^{izs}ds.$$

Then, by Plancharel's theorem, we can express the option price by integrating the Fourier transforms, i.e.,

$$\int_{\mathbb{R}} \overline{f(s)} q_T(s)ds = \frac{1}{2\pi}\int_{\mathbb{R}} \overline{\hat{f}(z)}\phi_T(-z)dz, \tag{4.13}$$

where ϕ_T is the Fourier transform of q_T, i.e., the characteristic function of the log-price s_T, and $\bar{z}$ denotes the conjugate of a complex number z. Of course, any sensible option payoff f is real-valued, thus $\overline{f} = f$. In typical situations, $\hat{f}$ will not exist, because the payoff function is not square integrable (e.g., the payoff function of a call option). This problem can be avoided, however, by either *dampening* the payoff function or by introducing some generalized Fourier transform instead. In this case, the corresponding integrand $\hat{f}$ often is of exponential form, and thus the above integral can, in turn, be interpreted as an inverse Fourier transform, allowing us to use numerical methods for computing Fourier transforms like the *Fast Fourier Transform* (FFT). As references we use the seminal paper of Carr and Madan [6] and the clarifying work by Lewis [27]. See also [13] for a very fast alternative.

Contours and damping

The following discussion is mainly taken from Lewis [27]. Take $f(s) = (e^s - K)_+$, the payoff function of a European call option. This function is clearly not square integrable, implying that the Fourier transform $\hat{f}(z)$ does not exist for $z \in \mathbb{R}$. It does, however, exist for $\Im(z) > 1$:

$$\hat{f}(z) = \int_{\mathbb{R}} f(x)e^{izx}dx = -\frac{K^{1+iz}}{z^2 - iz}, \quad \Im(z) > 1. \tag{4.14}$$

Indeed, for general $z \in \mathbb{C}$, we have

$$\int_{\log(K)}^{\infty} (e^x - K)e^{ixz}dx = \left(\frac{\exp((iz+1)x)}{iz+1} - K\frac{\exp(izx)}{iz}\right)\Bigg|_{x=\log(K)}^{x=\infty},$$

which exists (and simplifies to (4.14)) if and only if $\Im(z) > 1$. We call $\hat{f}(z)$ for general $z \in \mathbb{C}$ the *generalized Fourier transform* of f.

Thus, we cannot use the Plancherel identity (4.13) directly, since it requires us to integrate along the contour $\{\Im(z) = 0\} \subset \mathbb{C}$, which is not contained in the domain of definition $\{\Im(z) > 1\}$ of $\hat{f}$ (in the case of a European call option). Our task is therefore to extend the Plancherel theorem to other domains of the form $\{\Im(z) = \nu\}$.

Assumption 4.11. The payoff function $f(x)$ (as a function of the log-stock-price x) has a regular (i.e., holomorphic and single valued) generalized Fourier transform $\hat{f}(z)$ on a domain $\mathcal{S}_f = \{ z \mid a < \Im(z) < b \}$. Moreover, we assume that f is locally bounded.

Remark 4.12. The assumption basically means that f does not grow faster than an exponential function for $x \to \pm\infty$. As we have seen above, the condition is satisfied for the payoff function of a European call option (with $a = 1$ and $b = \infty$).

The generalized Fourier transform can be inverted along any straight line in the complex plain parallel to the real axis, i.e.,

$$f(x) = \frac{1}{2\pi}\int_{i\nu-\infty}^{i\nu+\infty} e^{-izx}\hat{f}(z)dz, \tag{4.15}$$

$a < \nu < b$. Moreover, we can generalize the Plancherel equality to hold for complex integration along straight lines parallel to the real axis, as long as both (generalized) Fourier transforms are regular along the line of integrations, i.e., assume that both generalized Fourier transforms $\overline{\hat{f}}$ and $\hat{g}$ are regular around the strip $\{\Im(z) = \nu\}$, then

$$\int_{\mathbb{R}} f(x)g(x)dx = \frac{1}{2\pi}\int_{i\nu-\infty}^{i\nu+\infty} \overline{\hat{f}(z)}\hat{g}(z)dz. \tag{4.16}$$

This leaves us with the task of determining the regions of regularity of the characteristic function $\phi_T(z)$, $z \in \mathbb{C}$.

Assumption 4.13. The characteristic function $\phi_T(z)$ of the log-price process s_T exists and is regular on the strip $\mathcal{S}_X := \{ z \in \mathbb{C} \mid \alpha < \Im z < \beta \}$ with $\alpha < -1$ and $\beta > 0$.

As a justification, Lewis [27] shows that this is true for a large class of exponential Lévy models.

Lemma 4.14. *Assume that s_T is a Lévy process such that S_T is a true martingale and such that the moment generating function exists in a complex neighborhood of* 1. *Then Assumption 4.13 holds and the characteristic function ϕ_T is given by the Lévy-Khintchine formula (Theorem B.5), i.e., $\phi_T(z) = \exp(-T\psi(z))$, with ψ defined as in the real case.*

Proof. Note that $\phi_T(z)$ exists for $\Im z = 0$ and $z = -i$ (by the martingale property) and is regular in a neighborhood of 0. By a theorem of Lukacs, this implies that the characteristic function is regular in a strip parallel to the real axis and can be represented as a Fourier integral there – thus the Lévy-Khintchine formula holds. Moreover, the maximal strip of regularity is either the whole complex domain $\mathbb{C}$, or it has one or two horizontal boundary lines. In the latter case, the purely imaginary points on these lines are points of singularity. Thus, the maximal domain of regularity must contain the purely imaginary points $z = 0$ and $z = -i$, establishing the result. □

Finally, under Assumptions 4.11 and 4.13, the following option valuation formula using contour integrals can be given.

Theorem 4.15. *Let $C = e^{-rT}E[f(s_T)]$ denote the price of the option f satisfying Assumption 4.11 in a model satisfying Assumption 4.13 and assume that $\mathcal{S}_f \cap \overline{\mathcal{S}_X} \neq \emptyset$. Then the option price is given by*

$$C = \frac{e^{-rT}}{2\pi} \int_{i\nu-\infty}^{i\nu+\infty} e^{-izy}\phi_T(-z)\hat{f}(z)dz,$$

where $y \coloneqq \log S_0 + rT$ and $i\nu \in \mathcal{S}_f \cap \overline{\mathcal{S}_X}$.

Proof. This is a consequence of Plancherel's formula (4.16) together with the assumptions imposed in the theorem, noting that

$$\overline{\phi_T(u+i\nu)} = \overline{E[e^{i(u+i\nu)s_T}]} = E[\overline{e^{i(u+i\nu)s_T}}] = E[e^{-i(-u+i\nu)s_T}] = \phi_T(-(-u+i\nu)),$$

implying that we can replace $\overline{\phi(z)}$ by $\phi_T(-z)$ by implicitly changing the direction of the integration. □

By equation (4.14), we see that $\mathcal{S}_f \cap \overline{\mathcal{S}_X}$ is indeed non-empty for European call and put options. Thus, we obtain the following formula for the price of a European call-option (with strike-price K, using the notation $k \coloneqq \log(S_0/K) + rT$:

$$C(S_0, K, T) = -\frac{Ke^{-rT}}{2\pi} \int_{i\nu-\infty}^{i\nu+\infty} e^{-izk} \frac{\phi_T(-z)}{z^2 - iz} dz, \tag{4.17}$$

valid for $1 < \nu < \beta$, where β is the upper bound of the stripe of regularity in Assumption 4.13.

From (4.14), we see that the integrand in (4.17) has two singularities, namely at $z = 0$ and $z = i$. Complex analysis shows us that contour integrals can be moved over singularities of the integrand by subtracting the residue[1] multiplied by $2\pi i$ at the

[1]The residue of a holomorphic function f with an isolated singularity at a point $a \in \mathbb{C}$ is given by the coefficient a_{-1} of $(z-a)^{-1}$ in the Laurent series expansion of $f(z)$. It is also given as the value of the integral $\frac{1}{2\pi i}\int_\gamma f(z)dz$ for any curve γ around a but no other singularity of f.

singularity. Moving the contour over the first singularity $z = i$ (with residue $iS_0/(2\pi)$) gives the formula

$$C(S_0, K, T) = S_0 - \frac{Ke^{-rT}}{2\pi} \int_{i\nu-\infty}^{i\nu+\infty} e^{-izk} \frac{\phi_T(-z)}{z^2 - iz} dz, \quad 0 < \nu < 1, \tag{4.18}$$

which is preferable to (4.17) for numerical reasons.

Remark 4.16. The residue at $z = 0$ is $-iKe^{-rT}/(2\pi)$, which means that the total residual term when moving past both singularities has the form $S_0 - Ke^{-rT}$. This term reminds us of the put-call-parity and, indeed, the remaining integral along the contour $\{\Im z = \nu\}$ for $\nu < 0$ is the price of the European put-option, whose payoff has the same generalized Fourier transform (4.14) as the call-option, but only valid for $\Im z < 0$. Thus, we get the put-call-parity

$$C(S_0, K, T) = P(S_0, K, T) + S_0 - Ke^{-rT}$$

by moving the contour over both singularities. In fact, we can even get a generalized Black-Scholes formula, namely by moving the contour *exactly over the singularities* and combining the results. Indeed, Lewis [27] shows that this gives the formula

$$C(S_0, K, T) = S_0\Pi_1 + Ke^{-rT}\Pi_2, \tag{4.19}$$

where

$$\Pi_1 = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \Re\left(\frac{e^{iuk}\phi_T(u - i)}{iu}\right) du, \quad \Pi_2 = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \Re\left(\frac{e^{iuk}\phi_T(u)}{iu}\right) du.$$

Moreover, note that Π_1 is the option-delta, whereas $\Pi_2 = P[S_T > K]$.

In Theorem 4.15, we have to integrate along a contour parallel, but not equal to the real axis, because option payoff functions like the one of the call-option are not square integrable, thus they do not have a classical Fourier transform. For the generalized Fourier transform we only needed an exponential bound condition, imposed in Assumption 4.11. Under that condition, we could, on the other hand, also multiply the payoff by an exponential function in such a way that the product is square integrable and then apply the classical Fourier transform to the product. This approach of *dampening* the payoff function is proposed in the seminal paper of Carr and Madan [6]. Theoretically, these approaches are, of course, equivalent, because the classical Fourier transform of the dampened payoff is nothing but a generalized Fourier transform, it leads, however, to different computational formulations.

For simplicity we will only consider put and call options in the sequel. Let $k := \log(K)$. In contrast to the previous approach we will apply the Fourier transform with respect to the variable k, which, by (4.14), is essentially equivalent to transforming the stock price. Then the price of a European call option with maturity T satisfies

$$C(S_0, T, k) = \int_k^\infty e^{-rT}(e^s - e^k) q_T(s) ds,$$

which tends to S_0 for $k \to -\infty$, and consequently is not square integrable in k. For some positive choices of α, we might hope that $c(S_0, T, k) := e^{\alpha k} C(S_0, T, k)$ is square integrable, implying that its Fourier transform denoted by $\psi_T(u)$ exists. In this case, we

obviously get back the option price C by inverting the Fourier transform and multiplying by $e^{-\alpha k}$, i.e.,

$$C(S_0, T, k) = \frac{e^{-\alpha k}}{2\pi} \int_{-\infty}^{\infty} e^{-iuk} \psi_T(u) du = \frac{e^{-\alpha k}}{\pi} \int_0^{\infty} e^{-iuk} \psi_T(u) du, \tag{4.20}$$

which leaves us with the task of computing ψ_T. Changing the order of integration, we obtain

$$\psi_T(u) = \int_{-\infty}^{\infty} e^{iuk} \int_k^{\infty} e^{\alpha k} e^{-rT} (e^s - e^k) q_T(s) ds dk = \frac{e^{-rT} \phi_T(u - (\alpha + 1)i)}{\alpha^2 + \alpha - u^2 + i(2\alpha + 1)u}. \tag{4.21}$$

Note that the inverse Fourier transform (4.20) applied to the formula (4.21) gives the same expression as the contour integral (4.17) under the change of variables $z = (\alpha + 1)i - u$ for $\nu = \alpha$. Clearly, we need here that $\phi_T(-(\alpha + 1)i)$ is defined, which leads us back to our Assumption 4.13. Thus, we see that using contour integrals is really equivalent to using a damping term.

Remark 4.17. Since we can bound $|\psi_T(u)|^2 \leq A/u^4$, one can bound the tail of the integral in (4.20) by

$$\int_a^{\infty} |\psi_T(u)| \, du \leq \frac{\sqrt{A}}{a}.$$

Thus, if we want the error from integrating on a finite domain only to be less than ϵ, we have to choose $a \geq e^{-\alpha k} \sqrt{A}/(\epsilon\pi)$.

Remark 4.18. Carr and Madan [6] note that the integrands in the representations of option prices can have large oscillations, leading to low accuracy. Indeed, consider for instance the price $Z(S_0, T, k)$ of a function with payoff $\max(S_T - K, K - S_T)$. The Fourier transform $\zeta_T(u)$ of Z can be shown to be

$$\zeta_T(u) = e^{-rT} \left(\frac{1}{1 + iu} - \frac{e^{rT}}{iu} - \frac{\phi_T(u - i)}{u^2 - iu} \right),$$

which has large oscillations in u when T is small. In this case, the authors recommend to multiply the price Z with the function $\sinh(\alpha k)$ for some constant α. Then the Fourier transform γ_T of the modified option price is given by

$$\gamma_T(u) = \frac{\zeta_T(u - i\alpha) - \zeta_T(u + i\alpha)}{2},$$

which mollifies the oscillations.

Fast Fourier Transform

In the previous part we derived concrete formulas for option prices in terms of an inverse Fourier transform. After cutting off to obtain an integral on a finite domain, see Remark 4.17, we are left with a problem of the form

$$C(k) \approx \frac{e^{-\alpha k}}{\pi} \int_0^a e^{-iuk} \psi(u) du \tag{4.22}$$

for some complex function $\psi = \psi_T$. In the first step, let us apply the trapezoidal rule to the above integral, using a uniform grid $u_l := \eta(l-1)$ for a constant η and for $1 \leq l \leq N$,

implying that $a = N\eta$. This means we approximate

$$\int_0^a e^{-iuk}\psi(u)du \approx \eta\left(\frac{\psi(0)}{2} + \sum_{l=2}^{N} e^{-iu_l k}\psi(u_l) + \frac{e^{-iak}\psi(a)}{2}\right) \approx \eta\sum_{l=1}^{N} e^{-iu_l k}\psi(u_l).$$

In general, the last term is in order two approximation (in terms of η) of the integral only if $\psi(0) = \psi(a)$, i.e., in the periodic case. Otherwise, it reduces the order of the trapezoidal rule. *For ease of notation* we will still use the second approximation in the following, even though we suggest to use the true trapezoidal rule in implementations. As it is, this approximation requires a computational work proportional to N.

Now assume that we do not only want to compute the price at one log-strike k, but for a whole variety of log-strikes – as it is the case in a typical calibration situation. We again choose a uniform grid in the log-strike domain, i.e., we set $k_j := -b + \lambda(j-1)$ where $b := N\lambda/2$. Thus, we want to compute the values

$$\sum_{l=1}^{N} e^{-i\eta\lambda(j-1)(l-1)} e^{ibu_l}\psi(u_l)\eta, \quad j = 1, \dots, N.$$

Next, choose the grid parameters η and λ such that the *Nyquist relation* $\lambda\eta = 2\pi/N$ holds. Then, the computational problem can be written as

$$X_j = \sum_{l=1}^{N} e^{-i\frac{2\pi}{N}(l-1)(j-1)} x_l, \quad j = 1, \dots, N, \tag{4.23}$$

where $x_l := e^{ibu_l}\psi(u_l)$ and X_j is an approximation of the option price with strike price k_j in the sense that

$$C(k_j) \approx \frac{e^{-\alpha k_j}}{\pi}\eta X_j + \frac{e^{-\alpha k_j}}{\pi} O\left(\frac{\sqrt{A}}{a} + \eta^2\right)$$

(provided that we are, in fact, using the true trapezoidal rule). We used all these assumptions and notations, because the vector X defined in (4.23) is the *discrete Fourier transform* of the vector x, and there is a very good numerical algorithm for computing discrete Fourier transforms. The computational cost of a usual implementation of (4.23) is proportional to N^2, but the so-called *Fast Fourier Transform* (FFT) reduces the work to $N\log_2(N)$.

Let $\omega_N := e^{-2\pi i/N}$ and define the $N \times N$-matrix T_N by

$$T_N := \begin{pmatrix} \omega_N^0 & \omega_N^0 & \omega_N^0 & \cdots & \omega_N^0 \\ \omega_N^0 & \omega_N^1 & \omega_N^2 & \cdots & \omega_N^{N-1} \\ \omega_N^0 & \omega_N^2 & \omega_N^4 & \cdots & \omega_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega_N^0 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \cdots & \omega_N^{(N-1)(N-1)} \end{pmatrix}.$$

Then we can obviously express the discrete Fourier transform (4.23) as $X = T_N x$.

Lemma 4.19. *For $x \in \mathbb{C}^{2N}$ let $X := T_{2N}x$ denote its discrete Fourier transform. Write $x' := (x_1, x_3, \dots, x_{2N-1})$ and $x'' := (x_2, x_4, \dots, x_{2N})$, but (differently) $X' := (X_1, \dots, X_N)$ and $X'' := (X_{N+1}, \dots, X_{2N})$. Moreover, denote $D_N := \operatorname{diag}(\omega_{2N}^0, \dots, \omega_{2N}^{N-1})$ and $c := T_N x'$, $d := D_N T_N x''$. Then*

$$X' = c + d, \quad X'' = c - d.$$

Proof. Simple calculation using $\omega_N^{jl} = \omega_{2N}^{2jl}$. □

Lemma 4.19 forms the basis of a classical *divide-and-conquer* algorithm, the celebrated FFT.

Algorithm 4.20. *Assume that $N = 2^J$, $J \geq 1$. Given $x \in \mathbb{C}^N$, apply the following recursive algorithm to compute its discrete Fourier transform $X = T_N x$:*

1. *If $N = 2$ go to 2, otherwise: split x into x' and x'' like in Lemma 4.19, apply the FFT to compute $c = T_{N/2}x'$, $d = D_{N/2}T_{N/2}x''$ and return $X = (X', X'')$ given by $X' = c + d$ and $X'' = c - d$.*
2. *If $N = 2$ compute $X = T_2 x$ directly.*

It can be easily shown that the computational effort to compute the discrete Fourier transform using Algorithm 4.20 is, indeed, proportional to $N \log_2(N)$.

Remark 4.21. In the same way, we can compute the inverse discrete Fourier transform.

Remark 4.22. Many variants of FFT exist. While most variants assume N to be a power of 2 (or even 4 or 8 for higher efficiency), there are also other variants without these requirements. Historically, the FFT was invented and implemented or used by many people, the first one probably being Gauss in 1805. However, it only became popular and widely used after the re-discovery of Cooley and Tukey [9]. Today, there are many different variants and even more different implementations. It is probably one of the most important algorithms, widely used in signal analysis, electrical engineering and even algebra (fast evaluation of polynomials).

Remark 4.23. While Carr and Madan [6] use FFT for evaluating the option price formula based on the Fourier transform, other authors like Lord and Kahl [11] advocate alternative specialized algorithms or classical quadrature because of usual non-uniform arrangements of strike-prices in practical calibration scenarios.

Cosine-series expansions

For even functions f, the Fourier transform specializes to the *cosine transform*,

$$\hat{f}(z) = 2 \int_0^\infty f(x) \cos(xz) dx.$$

In particular, by shifting variables, the Fourier transform of any function with bounded support can be expressed by its cosine transform. Since the density of log-spot prices s_T usually decays very fast to zero when the log-spot price approaches $\pm\infty$, we may assume that this is the case for the European option pricing problem. Starting from this idea, Fang and Oosterlee [13] have constructed a very fast method based on cosine expansions.

Remark 4.24. Before going into details, let us present the idea of Fang and Oosterlee in an abstract form. Assume that the density $q = q_T$ of the log-spot price decays very fast to 0, so that we may truncate it and treat it as a function with compact support, w.l.o.g., $\text{supp}(q) \subset [0, \pi]$ with $q(\pi) = 0$. Now, *Pontryagin duality*, as a starting point see [39], tells us that the "right" notion of a Fourier transform of a function defined on a finite subset of the real line is the Fourier series.

Consider a locally compact abelian group G. Then the *dual group* $\hat{G}$ is the set of all *characters* of G, i.e., of all continuous group homomorphisms from G with values in $\mathbb{T}$, the unit circle of $\mathbb{C}$. Here we are interested in two special cases:

1. if $G = \mathbb{R}$, then $\hat{G} \simeq \mathbb{R}$ and the characters take the form $\chi(x) = e^{iux}$, for $u \in \mathbb{R}$;
2. if $G = [-\pi, \pi]$ (which is isomorphic to $\mathbb{T}$), then $\hat{G} \simeq \mathbb{Z}$ and characters take the form $\chi(x) = e^{inx}$, $n \in \mathbb{Z}$.

Let μ denote the Haar measure of the group G. Then the Fourier transform $\hat{f}$ of an integrable function $f : G \to \mathbb{C}$ is a bounded continuous function on $\hat{G}$ defined by

$$\hat{f}(\chi) = \int_G f(x)\overline{\chi(x)}\mu(dx).$$

Inserting the representations of characters for the groups $\mathbb{R}$ and $[-\pi, \pi]$ as seen above, we see that the abstract Fourier transform boils down to the following special cases:

1. if $G = \mathbb{R}$, the Haar measure is the Lebesgue measure and we obtain the classical Fourier transform $\hat{f}(u) = \int_{\mathbb{R}} e^{-iux} f(x)dx$;
2. if $G = [-\pi, \pi]$, the Haar measure is again the Lebesgue measure, possibly with normalization, and $\hat{f}$ is the sequence of classical Fourier coefficients $c_n := \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-inx}dx$, $n \in \mathbb{Z}$.

Finally, note that the Fourier series of an even function $f : [-\pi, \pi] \to \mathbb{R}$ actually is a cosine series, i.e., all the sine-parts vanish. Thus, we may represent a function $f : [0, \pi] \to \mathbb{R}$ as a cosine series, under some mild regularity conditions.

Let $q : [0, \pi] \to \mathbb{R}$. Then, under certain conditions, q is represented by its *cosine expansion*

$$q(\theta) = \sum_{k=0}^{\infty}{}' A_x \cos(k\theta), \quad A_k := \frac{2}{\pi}\int_0^{\pi} q(\theta)\cos(k\theta)d\theta,$$

where $\sum'$ signifies that the first summand is taken with weight $\frac{1}{2}$. For entire functions, the convergence of the cosine series is exponential. If the function f is defined on a finite interval $[a, b]$, then the cosine expansion instead reads (by a change of variables)

$$q(x) = \sum_{k=0}^{\infty}{}' A_k \cos\left(k\pi\frac{x-a}{b-a}\right), \quad A_k := \frac{2}{b-a}\int_a^b q(x)\cos\left(k\pi\frac{x-a}{b-a}\right)dx. \tag{4.24}$$

Now, let us suppose that we know the Fourier transform $\phi = \hat{q}$ of q but not necessarily q itself – as is the case in many financial models, when q represents the density of the log-spot price. We want to express the coefficients A_k of the cosine expansion in terms of ϕ. In the first step, we need to replace the infinite domain of q by a finite domain, i.e., we consider

$$\phi_1(u) := \int_a^b e^{iux}q(x)dx \approx \phi(u).$$

Taking real parts, we immediately obtain

$$A_k = \frac{2}{b-a}\Re\left(\phi_1\left(\frac{k\pi}{b-a}\right)\exp\left(-i\frac{ka\pi}{b-a}\right)\right) \approx \frac{2}{b-a}\Re\left(\phi\left(\frac{k\pi}{b-a}\right)\exp\left(-i\frac{ka\pi}{b-a}\right)\right) =: F_k. \tag{4.25}$$

Numerically, we cannot add infinitely many numbers, thus we have to truncate the summation after N summands. Hence, we approximate

$$q(x) \approx q_1(x) := \sum_{k=0}^{N-1}{}' F_k \cos\left(k\pi\frac{x-a}{b-a}\right). \tag{4.26}$$

Note that q_1 is explicitly available if ϕ is explicitly given.

Remark 4.25. There are three different approximation errors in (4.26). First, we have truncated the integral, i.e., the domain of the density, in order to be able to do the cosine expansion in the first place. Then we replaced the Fourier transform of the truncated density by the Fourier transform of the true density and used this to obtain the coefficients of the cosine expansion. Finally, we replaced the infinite sum by a finite sum.

In the next step, we truncate the domain of integration in the option valuation formula

$$C(S_0, T) = e^{-rT} \int_{-\infty}^{\infty} f(x) q_T(x) dx$$

and then replace q_T by its approximation (4.26) (where we drop the subscript T). Thus, we obtain the approximation

$$C(S_0, T) \approx C_1(S_0, T) := e^{-rT} \sum_{k=0}^{N-1}{}' \Re\left(\phi_T\left(\frac{k\pi}{b-a}\right) e^{-ik\pi\frac{a}{b-a}}\right) C_k, \tag{4.27}$$

where

$$C_k := \frac{2}{b-a} \int_a^b f(x) \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \tag{4.28}$$

Notice that ϕ_T is the Fourier transform of s_T given that the spot-price at time 0 is S_0.

If we want to use the approximation (4.27) for pricing option, we only have to compute the coefficients C_k of the cosine expansion of the payoff function f. Fortunately, these are known explicitly for vanilla option.

Example 4.26. Consider a call option with payoff function $f(x) = (K(e^x - 1))_+$ in terms of log-moneyness $x = \log(S_T/K)$. Then the corresponding coefficient C_k is given by

$$C_k^{\text{call}} = \frac{2}{b-a} K(\chi_k(0, b) - \psi_k(0, b)),$$

with

$$\chi_k(c, d) := \frac{1}{1 + \left(\frac{k\pi}{b-a}\right)^2} \left[\cos\left(k\pi \frac{d-a}{b-a}\right) e^d - \cos\left(k\pi \frac{c-a}{b-a}\right) e^c + \right.$$
$$\left. + \frac{k\pi}{b-a} \sin\left(k\pi \frac{d-a}{b-a}\right) e^d - \frac{k\pi}{b-a} \sin\left(k\pi \frac{c-a}{b-a}\right) e^c \right]$$

and

$$\psi_k(c, d) := \begin{cases} \left(\sin\left(k\pi\frac{d-a}{b-a}\right) - \sin\left(k\pi\frac{c-a}{b-a}\right)\right) \frac{b-a}{k\pi}, & k \neq 0, \\ d - c, & k = 0. \end{cases}$$

For the put-option, we obtain

$$C_k^{\text{put}} = \frac{2}{b-a} K(\psi_k(a, 0) - \chi_k(a, 0)).$$

We remark here that these formulas are valid for the call and put options written in log-moneyness. Thus, we also have to use the density $\psi_T(x)$ of log-moneyness, and likewise for the characteristic function ϕ_T.

Fang and Oosterlee [13] also analyse the error of the approximation (4.27), and find that the error mostly depends on the smoothness of the density. While this does not effect two of the error terms (corresponding to truncation of the integration domain and replacing A_k by F_k), the error of the truncation of the infinite series converges exponentially, i.e., like $e^{-(N-1)\nu}$ for some ν, if the truncated density is smooth on $[a, b]$, or it converges algebraically, i.e., like $(N-1)^{-\beta}$ with β larger or equal to the order of the first derivative of the density with a discontinuity on $[a, b]$. Thus, at least for smooth densities, we have rapid convergence of the expansion (4.27), implying that we only need to compute a few of the coefficients. In fact, in the numerical experiments presented in the paper, they observe that $N \approx 60$ is usually enough to get a relative error of around 10^{-3} even in cases where FFT requires many more grid points due to high oscillations.

Fang and Oosterlee also comment on the truncation domain $[a, b]$, and suggest to choose it depending on the cumulants c_n of the distribution. More precisely, they suggest

$$a = c_1 - L\sqrt{c_2 + \sqrt{c_4}}, \quad b = c_1 + L\sqrt{c_2 + \sqrt{c_4}} \tag{4.29}$$

with $L = 10$.

Appendix A

Stochastic differential equations

A.1 Existence and uniqueness

We start by a very general existence and uniqueness result for SDEs driven by general semimartingales, which, in particular, covers the case of SDEs driven by Lévy processes. The following theorem is a special case of Protter [34, Theorem V.7].

Theorem A.1. *Let Z be a d-dimensional càdlàg semimartingale with $Z_0 = 0$ and let $F : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \to \mathbb{R}^{n\times d}$ be Lipschitz in the sense that for every $t \geq 0$ there is a constant K_t such that*

$$\forall x, y \in \mathbb{R}^n : \ |F(t,x) - F(t,y)| \leq K_t \, |x - y| \, .$$

Then the stochastic differential equation

$$X_t = X_0 + \int_0^t F\,(s, X_{s-})\, dZ_s$$

admits a unique solution X which is again a semimartingale.

We can also formulate everything in terms of the Stratonovich integral. Recall that for two given semimartingales H and Z, the the quadratic covariation satisfies

$$[H,Z]_t = H_0 Z_0 + \lim_{|\mathcal{D}|\to 0} \sum_{t_i \in \mathcal{D}} (H_{t_{i+1}} - H_{t_i})(Z_{t_{i+1}} - Z_{t_i}).$$

Let $[H,Z]^c$ denote the continuous part of the quadratic covariation. Then the *Stratonovich integral* of H with respect to Z is defined by

$$\int_0^t H_{s-} \circ dZ_s \coloneqq \int_0^t H_{s-} dZ_s + \frac{1}{2}[H,X]^c_t. \tag{A.1}$$

The advantage of the Stratonovich integral is that Ito's formula holds in a much simpler form: let $f : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \to \mathbb{R}^n$ be C^1 in the first and C^2 in the second component. Then

$$\begin{aligned} f(t, Z_t) = f(0, Z_0) + \int_0^t \partial_t f(s, Z_{s-}) ds + \int_{0+}^t \nabla f(Z_{s-}) \cdot \circ dZ_s \\ + \sum_{0<s\leq t} \left(f(Z_s) - f(Z_{s-}) - \nabla f(Z_{s-}) \cdot \Delta Z_s \right). \end{aligned} \tag{A.2}$$

The following existence and uniqueness result for Stratonovich SDEs is a special case of Protter [34, Theorem V.22].

Theorem A.2. *Assume that $F : \mathbb{R}_{\geq 0} \times \mathbb{R}^{n\times d} \to \mathbb{R}^n$ satisfies the following conditions: $F = F(t,x)$ is C^1 in t, F is C^1 in x and the Jacobian DF is C^1 in t and for every t both $x \mapsto F(t,x)$ and $x \mapsto DF_i(t,x)F_i(t,x)$ are Lipschitz, $i = 1,\dots,d$, where $F_i(t,x) = (F(t,x)_i^j)_{j=1}^n$. Then there is a unique semimartingale X solving*

$$X_t = X_0 + \int_0^t F(s, X_{s-}) \circ dZ_s.$$

Moreover, X is also the unique solution of the Ito SDE

$$X_t = X_0 + \int_0^t F(s, X_{s-})dZ_s + \frac{1}{2}\sum_{i=1}^d \int_0^t DF_i(s, X_{s-})F_i(s, X_{s-})d[Z, Z^i]_s^c,$$

where $[Z, Z^i] = ([Z^j, Z^i])_{j=1}^d$.

We will mostly consider SDEs driven by a d-dimensional Brownian motion B, i.e., SDEs of the form

$$X_t = X_0 + \int_0^t V(X_s)ds + \sum_{i=1}^d \int_0^t V_i(X_s)dB_s^i, \tag{A.3}$$

where $V, V_1, \dots, V_d : \mathbb{R}^n \to \mathbb{R}^n$ are vector fields and we have restricted ourselves to the autonomous case for simplicity. In this case, the change from the Ito formulation to the Stratonovich formulation corresponds to a change of the drift from V to

$$V_0(x) \coloneqq V(x) - \frac{1}{2}\sum_{i=1}^d DV_i(x)V_i(x), \tag{A.4}$$

i.e., X solves the Stratonovich equation

$$X_t = X_0 + \int_0^t V_0(X_s)ds + \sum_{i=1}^d \int_0^t V_i(X_s) \circ dB_s^i. \tag{A.5}$$

In the Brownian case we also have that the solution to the SDE will have finite pth moments provided that X_0 already has them.

Example A.3. The *Heston model* is a stochastic volatility model, i.e., the volatility of the the stock price is itself the solution of a stochastic differential equation. Since the volatility must be positive (or at least non-negative), we either have to choose an SDE for the volatility that is guaranteed to stay positive, or the volatility can be given as a deterministic, positive function of the solution of an SDE. A popular choice of a diffusion (i.e., a solution of an SDE driven by Brownian motion alone) that stays positive is the *square root process* (in finance well known as Cox-Ingersoll-Ross model for the short interest rate), and the corresponding stochastic volatility model is the Heston model, see Heston [17]. More precisely, the stock price and its instantaneous variance solve the following two-dimensional SDE

$$dS_t = \mu S_t dt + \sqrt{V_t}S_t dB_t^1 \tag{A.6a}$$

$$dV_t = \kappa(\theta - V_t)dt + \xi\sqrt{V_t}\left(\rho dB_t^1 + \sqrt{1-\rho^2}dB_t^2\right), \tag{A.6b}$$

with parameters $\kappa, \theta, \xi > 0$. The correlation ρ is typically negative. Obviously, this SDE fails to satisfy the Lipschitz condition of the existence and uniqueness theorem. More sophisticated, but still standard techniques (Feller's test of explosions, see Karatzas and Shreve [20, Theorem 5.5.29]) show that a unique solution does, indeed, exist. Under the obvious condition $V_0 > 0$, the variance component V_t stays non-negative, and it even stays strictly positive if $2\kappa\theta \geq \xi^2$, a condition that is often assumed for Heston's model. Positivity of the stock price is obvious.

Example A.4. The *SABR* model is similar to Heston's model. More precisely, we have

$$dS_t = V_t S_t^{\beta} dB_t^1, \tag{A.7a}$$

$$dV_t = \alpha V_t \left(\rho dB_t^1 + \sqrt{1-\rho^2} dB_t^2 \right). \tag{A.7b}$$

Example A.5. The *Stein-Stein* model is more regular than Heston's model or the SABR model. Here, positivity of the stochastic volatility is simply assured by taking the absolute value (of an Ornstein-Uhlenbeck process). More precisely, the model satisfies

$$dS_t = \mu S_t dt + |V_t| S_t dB_t^1, \tag{A.8a}$$

$$dV_t = q(m - V_t)dt + \sigma dB_t^2. \tag{A.8b}$$

Example A.6. A different class of models are *local volatility models*. The idea is that the volatility smile can be exactly reproduced by choosing a peculiar state dependence of the volatility in the Black-Scholes model, i.e., choose some function $\sigma(t, x)$ and let the stock price be given as solution to

$$dS_t = rS_t dt + \sigma(t, S_t) S_t dB_t. \tag{A.9}$$

Let $C(T, K)$ denote the price of a European call option as a function of the strike price K and the time to maturity T. If the local volatility σ satisfies *Dupire's formula*

$$\frac{\partial C}{\partial T} = \frac{1}{2}\sigma^2(T, K) K^2 \frac{\partial^2 C}{\partial K^2} - rK \frac{\partial C}{\partial K}, \tag{A.10}$$

then the local volatility model (A.9) produces the right prices for these call options, thus reproduces the volatility surface. Of course, one might also impose a local volatility function $\sigma(t, x)$ for more fundamental modelling purposes.

A.2 The Feynman-Kac formula

Assume that the vector fields $V, V_1, \ldots, V_d$ driving the SDE (A.3) are uniformly Lipschitz. Given three continuous and polynomially bounded functions $f : \mathbb{R}^n \to \mathbb{R}$, $g : [0, T] \times \mathbb{R}^n \to \mathbb{R}$ and $k : [0, T] \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, consider the Cauchy problem

$$\begin{cases} \dfrac{\partial}{\partial t} u(t, x) + Lu(t, x) + g(x) = k(t, x)u(t, x), & (t, x) \in [0, T) \times \mathbb{R}^n, \\ u(T, x) = f(x), & x \in \mathbb{R}^n. \end{cases} \tag{A.11}$$

Here, L denotes the second order linear partial differential operator defined by $L = V_0 + \frac{1}{2}\sum_{i=1}^{d} V_i^2$, with the usual identification of vector fields V with linear first order

differential operators via $Vf(x) = \nabla f(x) \cdot V(x)$. Assuming that a $C^{1,2}$ and polynomially bounded solution u of (A.11) exists, then it can be expressed as

(A.12)

$$u(t,x) = E\left[f(X_T) \exp\left(- \int_t^T k(s, X_s) ds \right) + \int_t^T g(s, X_s) \exp\left(- \int_t^s k(v, X_v) dv \right) ds \,\middle|\, X_t = x \right].$$

Similar *stochastic representations* exist for the corresponding Dirichlet and Neumann problems.

A.3 The first variation

Let X_t^x, $x \in \mathbb{R}^n$, $t \geq 0$, denote the solution to the Brownian stochastic differential equation (A.3) started at $X_0^x = x$. As indicated by the notation, we now consider X_t^x as a function of its initial value x. Under the assumptions of the existence and uniqueness Theorem A.1, for almost all $\omega \in \Omega$ and all $t \geq 0$, the map $x \mapsto X_t^x(\omega)$ is a homeomorphism of $\mathbb{R}^n \to \mathbb{R}^n$ – see [34, Theorem V.46]. In particular, the map is bijective. Thus X^x gives a *flow of homeomorphisms* of $\mathbb{R}^n$ (indexed by t). If we impose more smoothness on the driving vector fields, then the map $x \mapsto X_t^x(\omega)$ is differentiable (for almost all ω) and the Jacobian can be obtained by solving an SDE. This Jacobian is known as the *first variation*, and we will denote it by $J_{0\to t}(x)(\omega)$. More precisely, assume that the vector fields $V, V_1, \ldots, V_d$ are C^1 with bounded and uniformly Lipschitz derivatives. Then the first variation process exists and is the unique solution of the SDE

$$dJ_{0\to t}(x) = DV(X_t^x) J_{0\to t}(x) dt + \sum_{i=1}^{d} DV_i(X_t^x) J_{0\to t}(x) dB_t^i, \tag{A.13}$$

with initial value $J_{o\to 0}(x) = I_n$, the n-dimensional unit matrix. Notice that (A.13) alone does not fully specify an SDE, only an SDE along X^x. To get a true SDE, we have to consider the system consisting of (A.13) together with (A.3). Further note that $J_{0\to t}(x)$ is an invertible matrix, and the inverse also solves an SDE, which can be easily obtained by Ito's formula.

If, moreover, the vector fields $V, V_1, \ldots, V_d$ are smooth (with bounded first derivative), then one can show that $x \mapsto X_t^x$ even gives (almost surely) a diffeomorphism, i.e., a bijective smooth map, with smooth inverse.

If we replace the driving Brownian motion by a continuous semimartingale, then the above results remain true without any necessary modifications. If we use a general semimartingale with jumps as our driving signal, however, then the results only remain true as regards differentiability of the flow. If we want $x \mapsto X_t^x$ to be bijective, we would have to add more conditions on the vector fields. For more information, see Protter [34, Section V.7 – V.10].

A.4 Hörmander's theorem

Hörmander's theorem is a result on the smoothness of the transition density of the solution of an SDE – at least, that is the probabilistic interpretation of the result. For more information see the book of Nualart [31]. For the application to numerics of SDEs we refer to Bally and Talay [2].

Consider the SDE (A.3) and assume that the vector fields $V, V_1, \ldots, V_d : \mathbb{R}^n \to \mathbb{R}^n$ are smooth and all their derivatives are bounded functions (but not necessarily the

vector fields themselves). Given two smooth vector fields V and W, recall that the *Lie bracket* is the vector field defined by

$$[V, W] = DV \cdot W - DW \cdot V,$$

where DV denotes the Jacobian matrix of V. Moreover, for a multi-index $I = (i_1, \dots, i_k) \in \{0, 1, \dots, d\}^k$, $|I| := k \in \mathbb{N}$, we define the iterated Lie brackets for $|I| = 1$ by $V_{[(i)]} = V_i$ if $i \neq 0$ and $V_{[(0)]} = V$, and recursively for $I = (i_1, \dots, i_{k+1})$ by

$$V_{[I]} = \begin{cases} [V_{i_1}, V_{[(i_2,\dots,i_{k+1})]}], & i_1 \neq 0, \\ [V, V_{[(i_2,\dots,i_{k+1})]}], & i_1 = 0. \end{cases}$$

Definition A.7. The vector fields $V, V_1, \dots, V_d$ satisfy *Hörmander's condition* at a point $x \in \mathbb{R}^n$ if the vector space generated by the set of n dimensional vectors

$$\bigcup_{k \in \mathbb{N}} \left\{ V_{[I]}(x) \;\middle|\; I \in \{0, 1, \dots, d\}^k,\ i_k \neq 0 \right\}$$

is equal to $\mathbb{R}^n$.

Note that the drift vector field plays a special role here, as it does not appear in the start ($|I| = 1$) of the recursive construction of the above set, but only by taking Lie brackets. The reason for this is that only the diffusion vector fields contribute to the smoothing effect.

Let $p_t(x, y)$ denote the transition probability density of the solution X_t of the SDE, i.e., $p_t(x, \cdot)$ is the density of X_t conditioned on $X_0 = x$.

Theorem A.8 (Hörmander's theorem)**.** *If the driving vector fields satisfy Hörmander's condition at a point $x \in \mathbb{R}^n$, then the transition probability density $p_t(x, \cdot)$ is smooth.*

In its probabilistic proof, the theorem is obtained by showing that X_t is smooth in the sense of *Malliavin derivatives*. In fact, one can even get further by imposing a uniform version of Hörmander's condition.

Definition A.9. For $K \in \mathbb{N}$ and $\eta \in \mathbb{R}^n$ define the quantities

$$C_K(x, \eta) := \sum_{k=1}^{K} \sum_{I \in \{0,\dots,d\}^k,\, i_k \neq 0} (V_{[I]}(x) \cdot \eta)^2,\ C_K(x) := \inf_{|\eta|=1} C_K(x, \eta),\ C_K := \inf_{x \in \mathbb{R}^n} C_K(x).$$

We say that the *uniform Hörmander condition* (UH) holds if there is a $K \in \mathbb{N}$ such that $C_K > 0$.

Remark A.10. Note that the uniform Hörmander condition is considerably weaker than *uniform ellipticity*, a condition often imposed in PDE theory. Uniform ellipticity for a linear parabolic operator $Lf(x) = \sum_{k,j} a_{k,j}(x) \frac{\partial^2}{\partial x^k \partial x^j} f(x) + \sum_j b^j(x) \frac{\partial}{\partial x^j} f(x)$ means that there is a constant $C > 0$ such that

$$\sum_{k,j=1}^{n} a_{k,j}(x) \eta^k \eta^j \geq C\, |\eta|^2$$

for every $\eta \in \mathbb{R}^n$. But the relation between a and the vector fields is given by $a_{j,k}(x) = \sum_{i=1}^{d} V_i^j(x) V_i^k(x)$, therefore the above bound means that

$$\sum_{i=1}^{d} (V_i(x) \cdot \eta)^2 \geq C\, |\eta|^2,$$

which is satisfied iff $C := C_1 > 0$.

Under the UH condition, there is an explicit exponential bound on the derivatives of any order of $p_t(x, y)$ in all the variables t, x, y (provided that $t > 0$ of course), see Kusuoka and Stroock [24].

Appendix B

Lévy processes

We cite a few facts about Lévy processes. For more information about Lévy processes and their stochastic analysis we refer to Cont and Tankov [7] and Protter [34].

Definition B.1. A stochastic process $(X_t)_{t\geq 0}$ is called a *Lévy process* if

(i) X has independent increments, i.e., $X_t - X_s$ is independent of $\mathcal{F}_s$, the natural filtration of X,

(ii) X has stationary increments, i.e., $X_{t+h} - X_t$ has the same distribution as X_h, $h > 0$,

(iii) X is continuous in probability, i.e., $\lim_{s\to t} X_s = X_t$, if the limit is understood in probability.

Example B.2. If a Lévy process X is even continuous almost surely, then it is a Brownian motion with drift (i.e., $X_t = \mu + \sigma B_t$ for a standard Brownian motion B). On the other hand, every Lévy process has a càdlàg modification.

Example B.3. If X is a Lévy process, then the law of X_t is *infinitely divisible* for every t, i.e., for every $n \in \mathbb{N}$ we can find independent and identically distributed random variables $Y_1, \dots, Y_n$ such that X_t has the same distribution as $Y_1 + \cdots + Y_n$. Conversely, given any infinitely divisible distribution μ, there is a Lévy process X such that μ is the law of X_1. This gives rise to plenty of examples. Since the Poisson distribution is infinitely divisible, there is a Lévy process N_t such that N_1 has the Poisson distribution P_λ. Indeed, since the sum of n independent random variables $Y_i \sim P_{\lambda_i}$ is again Poisson distributed with parameter $\lambda_1 + \cdots + \lambda_n$, we have $N_t \sim P_{\lambda t}$, implying that N is the Poisson process.

The last example shows that Lévy processes actually can have jumps. We say that a Lévy process has *finite activity* if only finitely many jumps occur in every bounded interval with probability one, and *infinite activity* in the contrary case. The *Lévy-Ito decomposition* is a decomposition of a Lévy process into a diffusion, a process of finite activity, and a process of infinite activity. More precisely, we have

Theorem B.4. *Given a Lévy process X, we can find three independent Lévy processes $X^{(1)}$, $X^{(2)}$ and $X^{(3)}$ such that $X = X^{(1)} + X^{(2)} + X^{(3)}$ and*

- *$X^{(1)}$ is a Brownian motion with drift,*
- *$X^{(2)}$ is a compound Poisson process (the finite activity part),*

- *$X^{(3)}$ is a pure jump martingale, with jumps bounded by a fixed number $\epsilon > 0$ (the infinite activity part).*

So we can approximate Lévy processes by sums of a Brownian motion with drift and a compound Poisson process.

Theorem B.5 (Lévy-Khintchine formula)**.** *Given a (d-dimensional) Lévy process X. Then there is an $\alpha \in \mathbb{R}^d$, a positive semi-definite matrix $\Sigma \in \mathbb{R}^{d\times d}$ and a measure ν satisfying $\nu(\{0\}) = 0$, $\nu(A) < \infty$, $\int_{B(0,1)} |x|^2 \, \nu(dx) < \infty$ ($B(0,1)$ denotes the unit ball) such that*

$$E[\exp(iu \cdot X_t)] = \exp(-t\psi(u)),$$

where

$$\psi(u) = -iu \cdot \alpha + \frac{1}{2}\Sigma u \cdot u - \int_{\mathbb{R}^d} \left(\exp(iu \cdot x) - 1 - iu \cdot x\mathbf{1}_{|x|\leq 1}\right) \nu(dx).$$

We call (α, Σ, ν) the characteristic triplet of X.

Conversely, for every such characteristic triplet, there exists a corresponding Lévy process X.

Any Lévy process is a Markov process and the generator $Lf(x) = \lim_{t\to 0} \frac{P_t f(x) - f(x)}{t}$ for $P_t f(x) = E[f(X_t)|X_0 = x]$ is given (for bounded C^2-functions f on $\mathbb{R}^d$) by

$$\text{(B.1)} \quad Lf(x) = \nabla f(x) \cdot \alpha + \frac{1}{2} \sum_{j,k=1}^{d} \Sigma_{j,k} \frac{\partial^2}{\partial x^j \partial x^k} f(x) + \\ + \int_{\mathbb{R}^d} \left(f(x + y) - f(x) - \nabla f(x) \cdot y\mathbf{1}_{|y|\leq 1}\right) \nu(dy).$$

Notice that L is an integro-differential operator. Indeed, if f is constant around x, then

$$Lf(x) = \int_{\mathbb{R}^d} (f(x + y) - f(x))\nu(dy).$$

This formula has a very intuitive meaning, noting that ν describes the distribution of jumps of a Lévy process (in the sense that the jumps form a Poisson point process with intensity measure ν). If f is constant around x, then it can change values within an infinitesimal time interval only by an instantaneous jump out of the region where $f(y) = f(x)$. Therefore, the Kolmogorov backward equation

$$\frac{\partial}{\partial t} u(t, x) = Lu(t, x)$$

for $u(t, x) = P_t f(x)$ is a PIDE (partial integro-differential equation).

Note that if the Lévy measure ν is a finite measure (with $\lambda := \nu(\mathbb{R}^d)$), then Z_t is the sum of a Brownian motion (with drift) and a compound Poisson process with intensity λ and jump distribution $\frac{1}{\lambda}\nu$.

In Theorem A.1 we have formulated the existence and uniqueness statement for SDEs driven by general semimartingales. This, of course, also includes Lévy processes as drivers. Let $\sigma : \mathbb{R}^n \to \mathbb{R}^{n\times d}$ satisfy the assumptions of Theorem A.1 and let Z_t denote a d-dimensional Lévy process with characteristic triplet given in Theorem B.5, and consider the SDE

$$\text{(B.2)} \qquad dX_t = \sigma(X_{s-})dZ_s.$$

Then, given some boundedness and regularity conditions on $f : \mathbb{R}^n \to \mathbb{R}$, $u(t,x) \coloneqq E\left[f(X_T)|X_0 = x\right]$ satisfies the PIDE

$$\text{(B.3)} \quad \frac{\partial}{\partial t}u(t,x) = Au(t,x) + \\ + \int_{\mathbb{R}^d} \left(u(t, x+\sigma(x)z) - u(t,x) - (\sigma(x)z)\cdot\nabla u(t,x)\mathbf{1}_{|\sigma(x)z|\leq 1}\right)\nu(dz),$$

where

$$Ag(x) = \nabla g(x)\cdot(\sigma(x)\alpha) + \frac{1}{2}\sum_{i,j=1}^{n}\frac{\partial^2}{\partial x^i \partial x^j}g(x)(\sigma(x)\Sigma\sigma(x)^T)^{i,j}.$$

Appendix C

Affine processes

Definition C.1. A stochastically continuous Markov process X_t with state space $D = \mathbb{R}^m_{\geq 0} \times \mathbb{R}^n$ is called *affine* iff the logarithm of the characteristic function is affine in the initial state $X_0 = x$, more precisely if there are functions $\phi(t,u)$ taking values in $\mathbb{C}$ and $\psi(t,u)$ taking values in $\mathbb{C}^d$, $d = m + n$, such that

$$E[e^{u \cdot X_t}] = \exp\left(\phi(t,u) + x \cdot \psi(t,u)\right)$$

for all $u \in \mathbb{C}^d$ such that the expectation is finite. Here, $x \cdot y = \sum_{j=1}^d x_j y_j$ for $x, y \in \mathbb{C}^d$.

Affine processes are often used in finance, not at least because they allow for efficient computations using Fourier methods, see Section 4.2. The authorative reference article on affine processes is Duffie, Filipovic and Schachermayer [12].

The infinitesimal generator L of an affine process X has the form

$$\begin{aligned} Lf(x) = \frac{1}{2} \sum_{k,l=1}^d \left(a_{kl} + \sum_{i=1}^m \alpha^i_{kl} x_i \right) \frac{\partial^2}{\partial x_k \partial x_l} f(x) + \left(b + \sum_{i=1}^d \beta^i x_i \right) \cdot \nabla f(x) + \\ + \int_{D \setminus \{0\}} \left(f(x+\xi) - f(x) - h_F(\xi) \cdot \nabla f(x) \right) m(d\xi) + \\ + \sum_{i=1}^m x_i \int_{D \setminus \{0\}} \left(f(x+\xi) - f(x) - h^i_R(\xi) \cdot \nabla f(x) \right) \mu^i(d\xi), \end{aligned} \tag{C.1}$$

where the parameters satisfy the *admissibility conditions*: Let $I = \{1, \ldots, m\}$, $J = \{m+1, \ldots, d\}$ and write $x = (x_I, x_J)$ for $x \in \mathbb{R}^d$. The parameters are admissible if

- a, α^i are positive semi-definite $d \times d$ matrices, $b, \beta^i \in \mathbb{R}^d$, $c, \gamma_i \geq 0$, m and μ^i are Lévy measures on D;
- $a_{kk} = 0$ for $k \in I$, $\alpha^j = 0$ for $j \in J$, $\alpha^i_{kl=0}$ whenever $i \in I$ and $k \in I \setminus \{i\}$ or $l \in I \setminus \{i\}$;
- $b \in D$, $\beta^i_k \geq 0$ for $i \in I$ and $k \in I \setminus \{i\}$, $\beta^j_k = 0$ for $j \in J$ and $k \in I$;
- $\gamma^j = 0$ for $j \in J$;
- $\int_{D \setminus \{0\}} \min\left(|x_I| + |x_J|^2, 1 \right) m(dx) < \infty$, $\mu^j = 0$ for $j \in J$;
- $\int_{D \setminus \{0\}} \min\left(\left|x_{I \setminus \{i\}}\right| + \left|x_{J \cup \{i\}}\right|^2, 1 \right) \mu_i(dx) < \infty$ for $i \in I$.

Moreover, h_F and h_R^i are cut-off functions like in the Lévy-Khintchine formula.

Conversely, given admissible parameters, there is an affine process with generator (C.1).

Remark C.2. These admissibility conditions mainly serve to make the affine process well-defined in the sense that the process exists and does not leave the state space $\mathbb{R}^m_{\geq 0} \times \mathbb{R}^n \subset D$.

Even more important for numerical applications is the fact that there are tractable equations for the characteristic exponent, i.e., for the functions ϕ and ψ in terms of the parameters. They are solutions to the *generalized Riccati equations*

$$\frac{\partial}{\partial t}\phi(t,u) = F(\psi(t,u)), \quad \phi(0,u) = 0, \tag{C.2a}$$

$$\frac{\partial}{\partial t}\psi(t,u) = R(\psi(t,u)), \quad \psi(0,u) = u, \tag{C.2b}$$

where the right hand side is given by

$$F(u) = \frac{1}{2}(au)\cdot u + b\cdot u - c + \int_D \left(e^{\xi\cdot u} - 1 - h_F(\xi)\cdot u\right) m(d\xi), \tag{C.3a}$$

$$R_i(u) = \frac{1}{2}(\alpha^i u)\cdot u + \beta^i \cdot u - \gamma_i + \int_D \left(e^{\xi\cdot u} - 1 - h_R^i(\xi)\cdot u\right)\mu_i(d\xi), \tag{C.3b}$$

$i = 1,\dots,d$, with $F : \mathbb{C}^d \to \mathbb{C}$ and $R = (R_1,\dots,R_d) : \mathbb{C}^d \to \mathbb{C}^d$.

Remark C.3. Another characterization of affine processes as semi-martingales can be given in terms of the (local) semi-martingale characteristics, see Kallsen [19].

Example C.4. Recall the Heston model from Example A.3. Here we consider the model under a risk neutral measure, and we set $r = 0$, i.e., we set $\mu = 0$ in (A.6a). We change variables to the log-price $X_t := \log S_t$, which satisfies the SDE

$$\begin{aligned} dX_t &= -\frac{1}{2}V_t dt + \sqrt{V_t}dB_t^1, \\ dV_t &= \kappa(\theta - V_t)dt + \xi\sqrt{V_t}\left(\rho dB_t^1 + \sqrt{1-\rho^2}dB_t^2\right). \end{aligned}$$

By looking at the formula (C.1) for the generator of an affine process, we easily see that this two-dimensional SDE is, in fact, affine. Indeed, there are no jumps, so the integral terms vanish. The drift term of the generator of the (log-) Heston model is clearly affine in x, and so is the diffusion term, since the volatilities are linear in the square root of the state. The right hand sides of the Riccati equations (C.2) for the log-spot price X_t alone are given by

$$\begin{aligned} F(u,w) &= \kappa\theta w, \\ R(u,w) &= -\frac{1}{2}u - \kappa w + \frac{1}{2}u^2 + \frac{1}{2}\xi^2 w^2 + \rho\xi w u, \end{aligned}$$

i.e., $E[\exp(uX_t)] = \exp(\phi(t,u) + x\psi(t,u))$. Several generalizations of the Heston model like the *Bates model*, a stochastic volatility model with jumps, are affine processes, too.

Example C.5. Here we consider a relatively general jump-diffusion model. Let Z be a pure-jump semi-martingale with state-dependent intensity $\lambda(x)$ and compensator measure ν. Consider the SDE

$$dX_t = \mu(X_t)dt + \sigma(X_t)dB_t + dZ_t,$$

with $\mu : \mathbb{R}^n \to \mathbb{R}^n$ and $\sigma : \mathbb{R}^n \to \mathbb{R}^{n\times n}$, both smooth enough. Thus, the generator of the Markov process X is given by

$$Lf(x) = \mu(x) \cdot \nabla f(x) + \frac{1}{2} \operatorname{trace}\left(\sigma(x)\sigma(x)^T Hf(x)\right) + \\ + \lambda(x) \int_{\mathbb{R}^n} (f(x+\xi) - f(x) - h_F(\xi)\nabla f(x))\, \nu(d\xi),$$

where Hf denotes the Hessian matrix of f. Comparing the generator with the generic generator of an affine process given in (C.1), we see that X is affine if and only if the drift $\mu(x)$ is an affine function in x, the jump intensity $\lambda(x)$ is an affine function in x and the diffusion matrix is such that $\sigma(x)\sigma(x)^T$ is an affine function in x.

Bibliography

[1] Søren Asmussen and Jan Rosiński. Approximations of small jumps of Lévy processes with a view towards simulation. *J. Appl. Probab.*, 38(2):482–493, 2001.

[2] Vlad Bally and Denis Talay. The law of the Euler scheme for stochastic differential equations. I: Convergence rate of the distribution function. *Probab. Theory Relat. Fields*, 104(1):43–60, 1996.

[3] Mark Broadie and Paul Glasserman. Pricing American-style securities using simulation. *J. Econom. Dynam. Control*, 21(8-9):1323–1352, 1997. Computational financial modelling.

[4] Russel E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. In *Acta numerica, 1998*, volume 7 of *Acta Numer.*, pages 1–49. Cambridge Univ. Press, Cambridge, 1998.

[5] Peter Carr, Hélyette Geman, Dilip B. Madan, and Marc Yor. The fine structure of asset returns: An empirical investigation. *Journal of Business*, 75(2):305–332, 2002.

[6] Peter Carr and Dilip Madan. Option valuation using the Fast Fourier Transform. *Journal of Computational Finance*, 2(4):61–73, 1999.

[7] Rama Cont and Peter Tankov. *Financial modelling with jump processes*. Chapman & Hall/CRC Financial Mathematics Series. Chapman & Hall/CRC, Boca Raton, FL, 2004.

[8] Rama Cont and Ekaterina Voltchkova. A finite difference scheme for option pricing in jump diffusion and exponential Lévy models. *SIAM J. Numer. Anal.*, 43(4):1596–1626 (electronic), 2005.

[9] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comp.*, 19:297–301, 1965.

[10] Luc Devroye. *Nonuniform random variate generation*. Springer-Verlag, New York, 1986. Available online from http://cg.scs.carleton.ca/~luc/rnbookindex.html.

[11] Jin-Chuan Duan, James E. Gentle, and Wolfgang Härdle, editors. *Handbook of Computational Finance*, chapter Fourier Inversion Methods in Finance. Springer, forthcoming.

[12] D. Duffie, D. Filipović, and W. Schachermayer. Affine processes and applications in finance. *Ann. Appl. Probab.*, 13(3):984–1053, 2003.

[13] F. Fang and C. W. Oosterlee. A novel pricing method for European options based on Fourier-cosine series expansions. *SIAM J. Sci. Comput.*, 31(2):826–848, 2008/09.

[14] Michael B. Giles. Multilevel Monte Carlo path simulation. *Oper. Res.*, 56(3):607–617, 2008.

[15] Michael B. Giles, Desmond J. Higham, and Xuerong Mao. Analysing multi-level Monte Carlo for options with non-globally Lipschitz payoff. *Finance Stoch.*, 13(3):403–413, 2009.

[16] Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, 2004. Stochastic Modelling and Applied Probability.

[17] Steven L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6:327–343, 1993.

[18] Jean Jacod, Thomas G. Kurtz, Sylvie Méléard, and Philip Protter. The approximate Euler method for Lévy driven stochastic differential equations. *Ann. Inst. H. Poincaré Probab. Statist.*, 41(3):523–558, 2005.

[19] Jan Kallsen. A didactic note on affine stochastic volatility models. In *From stochastic calculus to mathematical finance*, pages 343–368. Springer, Berlin, 2006.

[20] Ioannis Karatzas and Steven E. Shreve. *Brownian motion and stochastic calculus*, volume 113 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1991.

[21] Peter E. Kloeden and Eckhard Platen. *Numerical solution of stochastic differential equations*, volume 23 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin, 1992.

[22] Donald E. Knuth. *The art of computer programming. Vol. 2*. Addison-Wesley Publishing Co., Reading, Mass., second edition, 1981. Seminumerical algorithms, Addison-Wesley Series in Computer Science and Information Processing.

[23] Arturo Kohatsu-Higa and Peter Tankov. Jump-adapted discretization schemes for lévy-driven sde. Preprint.

[24] Shigeo Kusuoka and Daniel Stroock. Applications of the Malliavin calculus, part II. *J. Fac. Sci. Univ. Tokyo*, 32:1–76, 1985.

[25] Pierre L'Ecuyer. Uniform random number generation. *Ann. Oper. Res.*, 53:77–120, 1994. Simulation and modeling.

[26] Pierre L'Ecuyer. Quasi-Monte Carlo methods with applications in finance. *Finance Stoch.*, 13(3):307–349, 2009.

[27] Alan L. Lewis. A simple option formula for general jump-diffusion and other exponential Lévy processes. Working paper, 2001.

[28] F.A. Longstaff and E.S. Schwartz. Valuing American options by simulation: a simple least-squares approach. *Review of Financial Studies*, 14:113–147, 2001.

[29] A.-M. Matache, P.-A. Nitsche, and C. Schwab. Wavelet Galerkin pricing of American options on Lévy driven assets. *Quant. Finance*, 5(4):403–424, 2005.

[30] N. Metropolis. The beginning of the Monte Carlo method. *Los Alamos Sci.*, 15, Special Issue:125–130, 1987. Stanislaw Ulam 1909–1984.

[31] David Nualart. *The Malliavin calculus and related topics. 2nd ed.* Probability and Its Applications. Berlin: Springer. xiv, 382 p., 2006.

[32] Bernt Øksendal. *Stochastic differential equations*. Universitext. Springer-Verlag, Berlin, sixth edition, 2003. An introduction with applications.

[33] Philip Protter and Denis Talay. The Euler scheme for Lévy driven stochastic differential equations. *Ann. Probab.*, 25(1):393–423, 1997.

[34] Philip E. Protter. *Stochastic integration and differential equations*, volume 21 of *Stochastic Modelling and Applied Probability*. Springer-Verlag, Berlin, 2005. Second edition. Version 2.1, Corrected third printing.

[35] Brian D. Ripley. *Stochastic simulation*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons Inc., New York, 1987.

[36] Rüdiger Seydel. *Introduction to numerical computation of finance derivates. Computational finance. (Einführung in die numerische Berechnung von Finanz-Derivaten. Computational finance.)*. Berlin: Springer. xii, 154 p., 2000.

[37] Denis Talay and Luciano Tubaro. Expansion of the global error for numerical schemes solving stochastic differential equations. *Stochastic Anal. Appl.*, 8(4):483–509 (1991), 1990.

[38] David B. Thomas, Wayne Luk, Philip H. W. Leong, and John D. Villasenor. Gaussian random number generators. *ACM Comput. Surv.*, 39(4), 2007.

[39] Wikipedia. Harmonic analysis — wikipedia, the free encyclopedia, 2010. [Online; accessed 9-July-2010].

[40] Wikipedia. Linear congruential generator — wikipedia, the free encyclopedia, 2010. [Online; accessed 22-March-2010].

[41] Paul Wilmott. *Paul Wilmott on quantitative finance. 3 Vols. With CD-ROM. 2nd ed.* Chichester: John Wiley & Sons. lxxv, 1379 p., 2006.