Computational Finance

Christian Bayer, Antonis Papapantoleon, Raul Tempone

Version of July 26, 2021

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 [23]). 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 $\big(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in[0,T]}, P\big)$ 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 $V, V_1, \dots, V_d : \mathbb{R}^n \to \mathbb{R}^n$ 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 5 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

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

$s \in \mathbb{R}^n$, 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)

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

[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!

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 6.2. 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 [10] and [42], 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

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

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.

The book of Glasserman [23] is a wonderful text book on Monte Carlos based methods in computational finance, i.e., it covers Chapter 2 and Chapter 5 in great detail. On the other hand, Seydel [55] does also treat Monte Carlo methods, but concentrates more on finite difference and element methods. Wilmott [60] is a very popular, easily 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 [46] for an introduction of SDEs driven by Brownian motion. Cont and Tankov [9] is the text book of choice for Lévy processes, and Protter [47] treats stochastic integration and SDEs in full generality.

[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 7.

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 (RNG) 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 $u_1, u_2, \ldots$ is good, if and only if 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 section, 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 as follows, see L'Ecuyer [34]:

Definition 2.2. A random number generator is a structure $(X, x_0, T, G, \{0, 1, \ldots, m-1\})$ where X is a finite set (the *state space*), $x_0 \in X$ is the initial state (the *seed*), $T : X \to X$ is a *transition*

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

function, and $G : X \to \{0, \dots, m-1\}$ is the *output function*. Given a random number generator, the pseudorandom numbers are computed via the recursion

$$x_l = T(x_{l-1}) \quad \text{and} \quad i_l := G(x_l) \quad \text{for} \quad l = 1, 2, \dots.$$

Remark 2.3. There is an immediate unfortunate consequence of this definition: since X is finite, the sequence of random numbers (i_l) must be periodic. Indeed, there must exist 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, Definition 2.2 arguably contains all possible candidates for good random number generators.

The following criteria for goodness have evolved in the literature on random number generators, see L'Ecuyer [34], L'Ecuyer et al. [36], and Glasserman [23]:

Statistical uniformity: The sequence of random numbers $i_1, i_2, \dots$ produced by the generator for a given seed should be hard to distinguish from truly random samples from the uniform distribution on $\{0, \dots, m-1\}$. 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.) The requirement of statistical uniformity basically means that we cannot guess the next number i_{l+1} given only the previously realized numbers $i_1, \dots, i_l$, at least not better than by choosing at random among $\{0, \dots, m-1\}$, if we assume that *we do not know the algorithm*.[2] Note that by statistical uniformity we require more than just uniformity of the one-dimensional marginals. Indeed, for any dimension d we require that sequences of d-dimensional outputs are difficult to distinguish from truly random sequences according to the uniform distribution on $\{0, \dots, m-1\}^d$. Of course, this property would be a consequence of independence of the numbers $i_1, i_2, \dots$.

Theoretical support: Many properties of random number generators, like the period length and the lattice structure (or hyperplane property), can be studied at a theoretical level; see e.g the remarks below about linear congruential generators). RNGs with strong theoretical support should be used and the others should be avoided. In principle, the optimal approach in choosing random number generators is to first screen their theoretical properties and then submit to empirical tests those with convincing theoretical support.

Speed: In modern applications, a lot of random numbers are needed. In molecular dynamics simulations for example, up to 10^{18} random numbers might be used (during several months of computer time). In finance, most applications do not require more than, say, 10^6 random numbers. However, the generation of random numbers is often 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 one order of magnitude larger than the square of the number of values used; cf. Ripley [49].

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

Portability: The RNG should be portable to different computers. Reliable implementations should be available for different operating systems and various programming languages.

[2] 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, cryptographic security thus means that we cannot compute the state x_l from $i_1, \dots, i_l$. While this property is essential in cryptography, it is not important for Monte Carlo simulations.

Jumping ahead: By "jumping ahead" we mean the possibility to quickly get to the state x_{l+n} given the state x_l for large n (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* (LCG). In the class of LCGs, the state space is $X = \{0, \dots, m-1\}$, the output function is the identity function $x_l = i_l$ and the transition map is provided by

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

Remark 2.4. Linear congruential generators are very well analyzed from a theoretical point of view, see Knuth [32]. 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 4 then so is $a - 1$.

Nonetheless, it should be stressed that a high period is only one of the many requirements identified above. In particular, the requirement of statistical uniformity is very hard to analyse by theoretical tools alone. The choice of parameters a, c, m of an LCG is a largely empirical task, where suites of statistical tests are run on large sequences of pseudo-random numbers.

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
Java	2^{48}	25214903917	11

Table 2.1: List of linear congruential RNGs as reported in [59].

Table 2.1 presents 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 conclude this discussion by pointing out a common weakness of all linear congruential RNGs. Fix $d \geqslant 1$ 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 distributed 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 for an example in $d = 2$. It has been proved that they can lie at most on $(d!m)^{1/d}$ hyperplanes, but often the actual figure is much smaller.

One of the most popular modern random number generators as of today is the Mersenne Twister algorithm[3]. This RNG produces 32-bit integers, the state space is $\mathbb{F}_2^{19968}$ (in its most popular version), where $\mathbb{F}_2$ denotes the finite field of size two, and the period is $2^{19937} - 1$. It is not a linear congruential generator, but the basis of the transformation map T is a linear map in X – with additional transformations, though. Note that in this case, the size of the state space (2^{19968}) is much larger than the $m = 2^{32}$.

Let us finally comment on the parallel generation of random numbers. As we shall see later in Chapter 2, it is often desirable or even necessary to have the possibility to generate random numbers on many cores in parallel. Indeed, as a general trend in computing one can observe that computers are generally no longer accelerated by making processors ever faster, but instead by adding multiple

[3]Available at http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html.

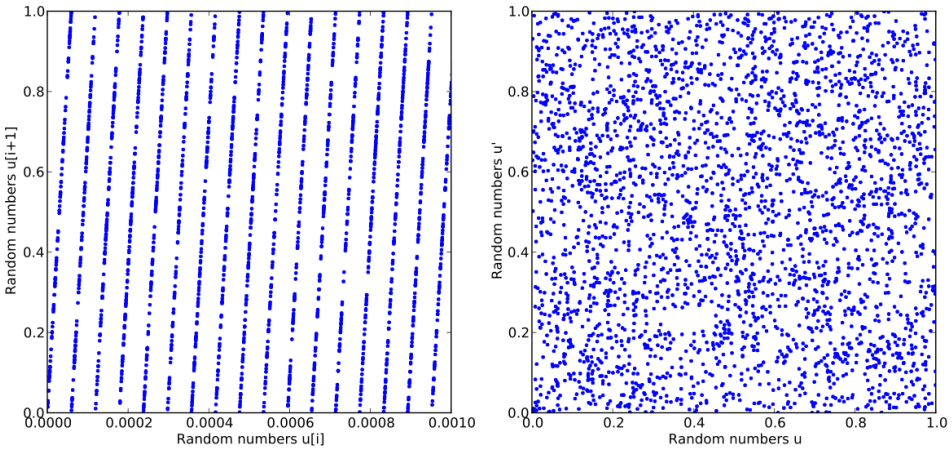


Figure 2.1: Hyperplane property for the linear congruential generator with $a = 16807$, $c = 0$, $m = 2^{31} - 1$. On the left, we have plotted 2 000 000 points (u_i, u_{i+1}), on the right 3000 pairs (i.e., 6000 random numbers plotted as pairs).

cores. This is especially true in graphics processors, where typical GPUs (graphical processing units) installed on average computers have dozens or even hundreds of cores, which are increasingly used also for general numerical purposes. In fact, vendors of GPUs are actively promoting these new applications; see e.g. NVIDIA [45]. [4] To cite L'Ecuyer et al. [36]:

> In highly parallel systems, one may need thousands or even millions of virtual RNGs which [...] run in parallel without exchanging data between one another, and behave from the user's viewpoint just like independent RNGs.

Before continuing, let us have a very cursory look at parallelization in general. Let us consider a simple program, which runs as a single process on the computer. Such a process can now start different *threads* which behave like processes of their own in as much as they can be executed on different cores in parallel, but have the big distinction that all the threads within a process share the same memory. This allows them to work with the same data and even use the output of other threads. As a simple example, think of a for-loop adding all the numbers stored in a very large array a (of size n): a natural parallelization would be to start l threads each summing up n/l (distinct) numbers (say, thread 1 computes $a[0] + \cdots + a[n/l-1]$, thread 2 computes $a[n/l] + \cdots + a[2n/l-1]$, ...), which are finally added to form the total sum. Hence, shared memory is necessary for successful parallelization, but it comes with a danger as different threads my end up using the same chunk of memory in incompatible ways. In general, problems come in the form of a *race-condition*, when the output of a process depends on the timing of threads within it, which produces a bug when this timing is different from the one anticipated by the programmer.

Example 2.5. As an example we consider the following highly simplified (and artificial) example in the context of RNGs. Let us assume we have one thread (thread 1), which runs an RNG and puts a random number into a double variable x. Whenever another thread accesses x, thread 1 will produce a new random number, which is again stored in x. We further have two threads (thread 2 and 3) which use random numbers produced by thread 1 for simulation. Now the intended sequence of events is that, say, thread 2 picks up the random number stored in x, then thread 1 updates x, and then thread 3 picks up the updated number in x. But in the absence of safety mechanisms, it could be that thread 3 is too fast, i.e., it accesses x already when thread 1 has not yet updated x, resulting in threads 2 and 3 using the same random number instead of independent ones.

[4] One limitation of GPUs as compared to classical CPUs is the rather small amount of rapidly accessible memory, which puts real restraints on the size of the seed or the dimensionality of the state space in an RNG context.

These problems mainly occur because developers for many decades were not concerned with parallel execution of code, which only became mainstream in the '90s. *Thread-safety* is the absence of any kind of race conditions, guaranteeing the safe execution of parallelized code. It is always important to check whether libraries or other pieces of code used in a parallel program are thread-safe!

Now, how can we generate parallel streams of random numbers? Let us describe several possible ways, along with their advantages and drawbacks:

- Use a central source of randomness for all threads, i.e., one thread produces all the random numbers for all other threads. As random number generation is often a bottleneck for applications (especially in a Monte Carlo framework) *and* data exchange between different threads is often the bottleneck in parallelization, this simple method is typically not acceptable.

- Use different RNGs for different threads, i.e., either truly different RNGs or the same class of RNGs but with different parameters. This requires one to have many good parameters / good RNGs available, and, besides, even if parameters / RNGs are individually good, their combination may fail the independence requirement. Hence, any such combination needs to be tested statistically, which makes it cumbersome to use this method for an arbitrary (high) number of streams.

- Use a single RNG split into equally-spaced blocks. Say we know that we have n threads which all may require (at most) ν random numbers. We use our favorite RNG with seed x_0 for thread 1. We jump ahead to step ν and use the RNG with seed x_ν for thread 2. In the same way, each thread uses the same RNG with seeds obtained by jumping ahead ν steps from the seed used by the previous thread. From a theoretical point of view, this method is most satisfactory, since good statistical properties of the RNG used imply good statistical properties of the sequence of streams constructed in that way. However, good RNGs can only be used for this method when they allow for rapid jumping-ahead. As in most varieties of RNGs the transition function T has the form of a matrix multiplication (say with a matrix A), this means that there must be a rapid way of computing A^ν, which is often not possible, especially if the state space X is extremely high-dimensional, such as in the case of the Mersenne Twister. Hence, it may be simpler to use an RNG constructed by the combination of two simpler RNGs defined on relatively low-dimensional state spaces. We refer to L'Ecuyer et al. [36] for references on good RNGs and suitable implementations for this purpose.

- Use one RNG with random seeds. If we have a good RNG with very high period, but bad jumping-ahead capability like the Mersenne Twister, then we may want to use n copies of the RNG with n seeds drawn from the state space X with the help of another RNG. While overlaps between the different streams are possible, they are extremely unlikely. Indeed, if the period of the RNG is ρ, then the probability of an overlap is approximately $(1-n\nu/\rho)^{n-1}$. For instance, L'Ecuyer et al. [36] report that this probability is close to 2^{-964} when $l = \nu = 2^{20}$ and $\rho = 2^{1024}$. An added benefit of this method is that it is applicable when the number of random streams is not known beforehand, for instance because new random streams need to be generated depending on random events.

Finally, let us note that reproducibility may become an issue with parallelization, as the organization of threads and the assignment of tasks to a thread may be determined at execution time and may differ between two different executions. Hence, it may be advisable to assign streams at an abstract level, i.e., to distinct computational tasks instead of individual threads, the number and speed of execution of which may be hard to predict for the programmer.

Non-uniform random numbers

In many applications, we do not need uniform random numbers, but random numbers from another distribution. In the Black-Scholes model for instance, the stock price has the following dynamics:

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

Therefore, the stock price S_T has a log-normal distribution, while B_T has a normal distribution. Thus, there are two ways to sample the stock price: 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 $\mathcal{U}(]0,1[)$-distributed random numbers. We will present some 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 [13].

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

Proposition 2.6. *Let F be a cumulative distribution function and define*

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

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

Proof. By definition of F^{-1} we have $F^{-1}(u) \leqslant x \iff F(x) \geqslant u$, therefore

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

Proposition 2.6 is the basis of Algorithm 2.7.

Algorithm 2.7 (Inversion method)**.** *Given F^{-1} and $U \sim \mathcal{U}(]0,1[)$, return $X = F^{-1}(U)$.*

Example 2.8. The exponential distribution with parameter $\lambda > 0$ has the distribution function $F(x) = 1 - \mathrm{e}^{-\lambda x}$, 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 2.9. If an explicit formula for the distribution function F is available, but not for its inverse F^{-1}, 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.[5] 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 Glasserman [23].

Remark 2.10. The transparent relation between the uniform random numbers $U_1, \ldots, U_l$ and the transformed random numbers $X_1, \ldots, X_l$ (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 $X_1, \ldots, X_l$ to be correlated, we can choose the uniforms to be correlated. Another example is the generation of the maximum $X^* := \max(X_1, \ldots, X_l)$. Apart from the obvious solution (generating $X_1, \ldots, X_l$ 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 $U^* = \max(U_1, \ldots, U_l)$. Then, using the monotonicity of F^{-1}, $X^* = F^{-1}(U^*)$. Since we only have to do one inversion instead of l, this method is usually much more efficient than the obvious method.

[5] 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 2.8 suffers from this defect.

- Combining both approaches, we see that the c.d.f. of U^* is given by x^l, $0 \leqslant x \leqslant 1$, with inverse function $x^{1/l}$. So we obtain one sample from the distribution of U^* simply by $U_1^{1/l}$, and X^* has the same distribution as $F^{-1}(U_1^{1/l})$.

Next we present another 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 $c \geqslant 1$ such that

$$f(x) \leqslant cg(x), \quad x \in \mathbb{R}^d. \tag{2.2}$$

Algorithm 2.11 (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 $U \leqslant f(X)/(cg(X))$ return X;*[6] *else go back to 1.*

Proposition 2.12. *Let Y be the outcome of Algorithm 2.11. Then Y has the distribution given by the density f. Moreover, the loop in the algorithm has to be traversed c times on average.*

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

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

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

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

On the other hand, a similar computation shows that $P\left(U \leqslant \frac{f(X)}{cg(X)}\right) = \frac{1}{c}$, 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, the expectation of a geometric distribution with parameter $1/c$. □

Naturally, we want c to be as small as possible. That is, in fact, the tricky part of the endeavour. Exercise 2.2 asks for a method to sample normal random variables starting from the exponential distribution, which we can sample by Example 2.8.

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

Example 2.13. 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

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

Although the acceptance-rejection algorithm is a very general and exact transformation algorithm, i.e., if fed with truly random numbers it will produce random numbers which are exactly distributed according to the desired density, it can be quite inefficient if the parameter c is large. Marsaglia and Tsang [41] have constructed a fast and efficient variant of the acceptance-rejection algorithm, which is still applicable in the majority of cases. For reasons to become clear later, they call their algorithm *Ziggurat* algorithm.

Like in the acceptance-rejection algorithm, the fundamental idea of the Ziggurat algorithm is based on the principle that sampling from the distribution given by a (say, univariate) density f is equivalent to sampling a point from the (say, bi-variate) uniform distribution in the area between 0 and the graph of f. The situation would be especially simple if this area was "Ziggurat" shaped, i.e., had the form of rectangles (parallel to the abscissa) put on top of each other. In this case, we could first choose the rectangle at random (according to their respective volumes) and then we would only have to sample a uniform random number on the lower side of the rectangle – note that the second coordinate of the chosen random number in $\mathbb{R}^2$ does not really matter for the acceptance-rejection method, as long as it is guaranteed that the two-dimensional random variate is below the graph of the density. Now the idea of the Ziggurat algorithm is simply to approximate the area under the graph by such a Ziggurat-shaped polygon using tabulated values for the respective density, and accompany this by a classical acceptance-rejection method for the "remainder" of the area.

More precisely, assume we are given a density $f : [0, \infty[\to [0, \infty[$ which is monotonically decreasing like the density of the exponential distribution. Moreover, fix some $n \in \mathbb{N}$ and assume we aregiven a sequence $0 = x_0 < x_1 < \cdots < x_n$ such that the following condition holds with $y_i := f(x_i)$, $i = 0, \ldots, n$:

$$x_i(y_{i-1} - y_i) = x_n y_n + \int_{x_n}^{\infty} f(x)dx =: v, \quad i = 1, \ldots, n-1. \tag{2.3}$$

Obviously, the values $x_0, \ldots, x_n$ depend on the distribution under consideration and on the numerical parameter n. Hence, these values are best treated as pre-computed, tabulated parameters; we will comment further below.

Equation (2.3) means that the areas of the $n-1$ rectangles with corners $(0, y_i)$, (x_i, y_i), (x_i, y_{i-1}) and $(0, y_{i-1})$ (denoted by R_i), $i = 1, \ldots, n-1$, are all equal to v, just as the area of the last rectangle with corners $(0, 0)$, $(x_n, 0)$, (x_n, y_n) and $(0, y_n)$ together with the area below the graph f on $[x_n, \infty[$. This surface will be denoted by R_n. Moreover, the area below the graph of f is contained in the surface $\bigcup_{i=1}^{n} R_i$ composed of all the surfaces described above. Furthermore, we assume that we have a specialized algorithm for sampling from the tail distribution $X \sim f$ conditioned on $X > x_n$. See also Figure 2.2.

Algorithm 2.14 (Ziggurat algorithm)**.** *Goal: Sample a random variable $X \sim f$.*

1. *Generate i uniform in $\{1, \ldots, n\}$.*
2. *If $i = n$, go to (6).*
3. *Generate $U_1 \sim \mathcal{U}(0, 1)$ and set $x := U_1 x_i$.*
4. *If $x < x_{i-1}$ return x.*
5. *Otherwise, generate $U_2 \sim \mathcal{U}(0, 1)$ and set $y := y_i + U_2(y_{i-1} - y_i)$. If $y \leqslant f(x)$ return x; else go back to (1).*
6. *Generate $U_1 \sim \mathcal{U}(0, 1)$ and set $x := vU_1/y_n$.*

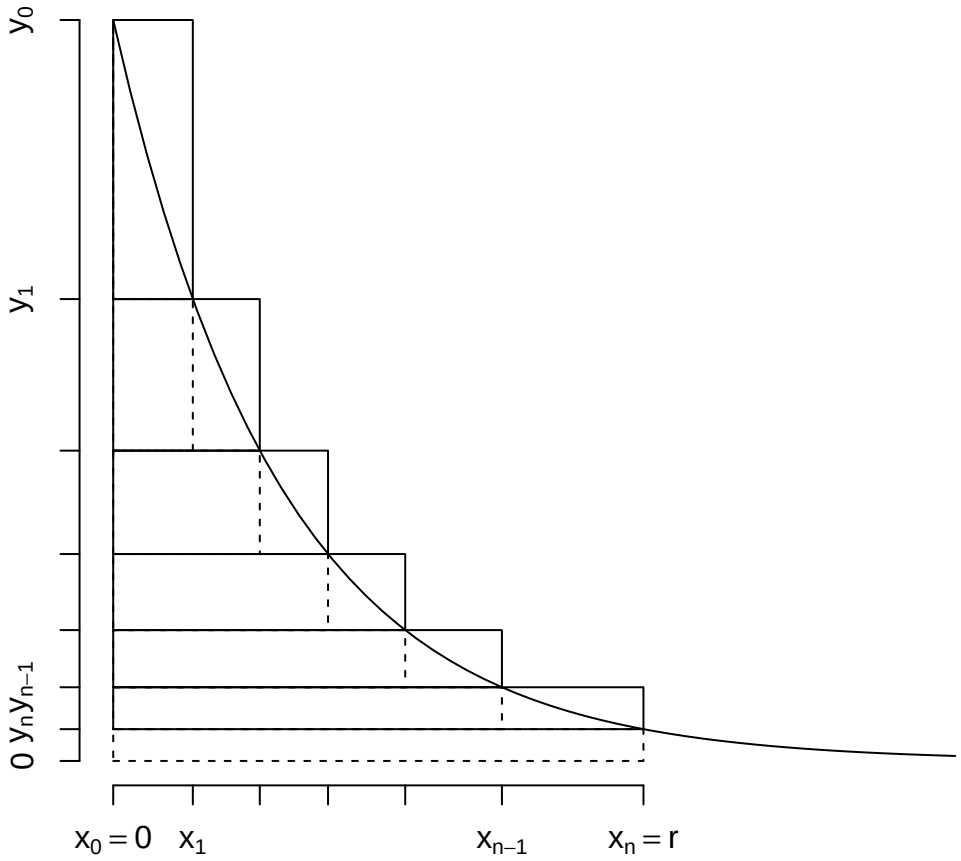


Figure 2.2: Ziggurat algorithm

7. If $x < x_n$, return x. Otherwise, return a sample from the tail distribution $X|X > x_n$.

Remark 2.15. Most of the time the algorithm stops in step (4), in which case we save one uniform random number generation and do not even need to evaluate f once. Moreover, it is obvious how to extend the algorithm to a symmetric or uni-modal distribution.

We round up this discussion with two examples, namely the Ziggurat algorithm for the exponential and the standard normal distributions. In both cases, the Ziggurat algorithm is highly competitive in speed.

Example 2.16. For the exponential distribution $Exp(1)$, we can use as tail sample $x_n - \log U$ for $U \sim \mathcal{U}(0,1)$. For $n = 255$ (a typical value), a possible choice of x_n is $7.697\ldots$ implying $v = 0.0039\ldots$, which results in an efficiency of 98.9%, i.e., the probability of needing only one iteration of the algorithm to produce one sample of the target distribution is 98.9%.

Example 2.17. For the standard normal distribution $\mathcal{N}(0,1)$, for $n = 255$, a possible choice of x_n

is $3.65\ldots$ implying $v = 0.0049\ldots$, which results in an efficiency of 99.33%. [41] suggest the following algorithm for sampling from the tail distribution:

1. Generate $U_1, U_2 \sim \mathcal{U}(0,1)$.
2. Set $x := -\log(U_1)/x_n$, $y := -\log U_2$.
3. If $2y > x^2$, return $x + x_n$, else go back to (1).

We conclude this section by presenting two methods designed specifically for generating standard normal random numbers. The *Box–Muller method* and the *polar method* are probably two of the simplest such methods, although not the most efficient ones. A comprehensive list of random number generators specifically available for Gaussian random numbers is available in the survey article by Thomas et al. [57].

Algorithm 2.18 (Box–Muller method)**.** *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 $X_1 = \rho\cos(\theta)$, $X_2 = \rho\sin(\theta)$.

Algorithm 2.19 (Polar method)**.** *1. Generate two independent uniform random numbers U_1, U_2 from the interval $]-1, 1[$;*

2. Set $S = U_1^2 + U_2^2$;

3. If $S < 1$, return the independent standard normals

$$Y_1 = U_1\sqrt{\frac{-2\ln(S)}{S}} \quad \textit{and} \quad Y_2 = U_2\sqrt{\frac{-2\ln(S)}{S}};$$

else, return to 1.

The polar method is more efficient than the Box–Muller algorithm, because it avoids the evaluation of the computationally expensive trigonometric functions.

Remark 2.20. In order to generate 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 2.1. Explain why c in (2.2) can only be greater than or equal to 1. What does $c = 1$ imply?

Exercise 2.2. Provide a method for generating double exponential random variables using only *one* uniform random number per output. Moreover, justify the bound c in Example 2.13.

Exercise 2.3. Show that (X_1, X_2) generated by the Box–Muller method 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.

Exercise 2.4. Show that (Y_1, Y_2) generated by the polar method have the two-dimensional standard normal distribution.

Exercise 2.5. Implement the different methods for generating Gaussian random numbers and compare their efficiency.

2.2 Monte Carlo simulation

The Monte Carlo simulation method is one of the most important numerical methods available. It was developed by giants of mathematics and physics like J. von Neumann, E. Teller, S. Ulam and N. Metropolis during the development of the H-bomb. A short account of the origins of Monte Carlo simulation can be found in Metropolis [43]. Today, it is widely used in fields like statistical mechanics, particle physics, computational chemistry, molecular dynamics, computational biology and, of course, computational finance! An overview of the mathematics behind the Monte Carlo method is available, for instance, in the survey paper of Caflisch [6] or, as usual, in Glasserman [23].

The Monte Carlo method

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

$$I[f;X] \coloneqq \mathbb{E}\left[f(X)\right], \tag{2.4}$$

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 $X_1, X_2, \dots$ of independent realizations of X, the law of large numbers implies that

$$I[f;X] = \lim_{M\to\infty} \frac{1}{M}\sum_{i=1}^{M} f(X_i), \quad \mathbb{P}-\text{a.s.} \tag{2.5}$$

However, in numerics we are usually not quite satisfied with a mere convergence statement like in (2.5). 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 also like to know how fast the error goes to zero if we increase M. Before continuing the discussion, let us formally introduce the Monte Carlo integration error ϵ_M by

$$\epsilon_M = \epsilon_M(f;X) \coloneqq I[f;X] - I_M[f;X], \quad \text{where} \quad I_M[f;X] \coloneqq \frac{1}{M}\sum_{i=1}^{M} f(X_i) \tag{2.6}$$

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.* $\mathbb{E}\left[I_M[f;X]\right] = I[f;X]$, implying $\mathbb{E}\left[\epsilon_M\right] = 0$. Let us also introduce the *mean square error* and its square root, the error in L^2, via

$$\mathrm{MSE}[I_M] = \mathbb{E}\left[\epsilon_M(f;X)^2\right] \quad \text{and} \quad \mathrm{RMSE}[I_M] = \mathbb{E}\left[\epsilon_M(f;X)^2\right]^{1/2}. \tag{2.7}$$

The *central limit theorem* immediately implies both an error bound and a convergence rate provided that $f(X)$ is square integrable.

Proposition 2.21. *Let $\sigma = \sigma(f;X) < \infty$ denote the standard deviation of the random variable $f(X)$. Then the root mean square error satisfies*

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

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

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

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

Proof. Using the independence of the X_i's and the fact that $I_M[f;X]$ is an unbiased estimator of $I[f;X]$, we get

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

In addition, from the central limit theorem we know that

$$\frac{\sum_{i=1}^{M} f(X_i) - M \cdot I[f;X]}{\sigma\sqrt{M}} \xrightarrow[M\to\infty]{} \mathcal{N}(0,1)$$

which yields the asymptotic normality of the error. □

Remark 2.22. Proposition 2.21 has two important implications:

1. The error is *probabilistic*: there is no deterministic error bound. In other words, for a particular simulation and a given sample size M, the error of the simulation can be arbitrarily large. 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 thus say that the Monte Carlo method *converges with rate* $1/2$.

Let us now discuss the merits of Monte Carlo simulation. We assume, for simplicity, that X is a d-dimensional uniform random variable, *i.e.*,

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

Observe that the dimension of the space did not enter into our discussion of the convergence rate and of the 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 $0 \leqslant x_1 < x_2 < \cdots < x_N \leqslant 1$ of arbitrary length N. The corresponding d-dimensional grid is simply given by $\{x_1, \ldots, x_N\}^d$, a set of size $n \coloneqq N^d$. 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 then proportional to $(1/N)^k$. However, we have to evaluate the function on n points, implying that the total computational work is proportional to n rather then N. Therefore, the accuracy in terms of the complexity n, the ratio of the error relative to the computational work, behaves like $n^{-k/d}$. Thus, the rate of convergence in terms of the computational cost is only k/d, which rapidly decreases in the dimension d. This phenomenon is known as the *curse of dimensionality*: 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}!

Error control and confidence intervals

Next, we discuss 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

$$\mathbb{P}\big(\,|\epsilon_M(f;X)| > \varepsilon\big) < \delta.$$

Fortunately, this question is already almost answered in Proposition 2.21. Indeed, it implies that

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

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

$$M = \left(\Phi^{-1}\left(\frac{2-\delta}{2}\right)\right)^2 \frac{\sigma^2}{\varepsilon^2}. \tag{2.8}$$

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

Remark 2.23. This analysis tacitly assumed that we know $\sigma = \sigma(f; X)$. Since we started the whole endeavor in order to compute the mean of $f(X)$, $I[f; X]$, 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. See Exercise 2.6 for a sample estimator of σ. (This is not unproblematic: what about the Monte Carlo error for the approximation of $\sigma(f; X)$?)

In addition, since the Monte Carlo estimator is a random variable, when computing expectations via this method it is not very helpful to report just the value $I_M[f; X]$. This estimator is a function of the sample size M and we do not know how accurate the estimation is unless we also have information about the sample size. Therefore, it is more meaningful to report the estimator and some *confidence interval*.

Definition 2.24. Let Z be a random variable and consider some level $\alpha \in (0, 1)$. The $1-\alpha$-level *confidence interval* is defined by

$$[-z_{1-\frac{\alpha}{2}}, z_{1-\frac{\alpha}{2}}]$$

such that the *critical number* $z_{1-\frac{\alpha}{2}}$ satisfies:

$$\mathbb{P}\Big(|Z| \leqslant z_{1-\frac{\alpha}{2}}\Big) = 1 - \alpha. \tag{2.9}$$

The critical number $z_{1-\frac{\alpha}{2}}$ for a given level $1-\alpha$ can be computed from the inverse cumulative distribution function. Consider, for example, the normal distribution; then we get that

$$z_{1-\frac{\alpha}{2}} = \Phi^{-1}\Big(1 - \frac{\alpha}{2}\Big).$$

In particular, using the inverse cdf of the normal distribution, we get that for $\alpha = 5\%$ the critical number equals 1.96, while for $\alpha = 1\%$ it equals 2.58.

Now, we can use the asymptotic normality of the Monte Carlo error ϵ_M to derive confidence intervals for $I_M[f; X]$. Indeed, using Proposition 2.21 and denoting $\epsilon_M = I - I_M$, we have

$$\begin{aligned} 1 - \alpha &\approx \mathbb{P}\left(-\frac{\sigma z_{1-\frac{\alpha}{2}}}{\sqrt{M}} \leqslant \epsilon_M \leqslant \frac{\sigma z_{1-\frac{\alpha}{2}}}{\sqrt{M}}\right) \\ &= \mathbb{P}\left(I_M - \frac{\sigma z_{1-\frac{\alpha}{2}}}{\sqrt{M}} \leqslant I \leqslant I_M + \frac{\sigma z_{1-\frac{\alpha}{2}}}{\sqrt{M}}\right). \end{aligned}$$

Thus, the $1-\alpha$-level confidence interval for $I = I[f; X]$ is

$$\mathrm{CI}_\alpha[I_M] := \left[I_M - \frac{\sigma z_{1-\frac{\alpha}{2}}}{\sqrt{M}}, I_M + \frac{\sigma z_{1-\frac{\alpha}{2}}}{\sqrt{M}}\right]. \tag{2.10}$$

Example 2.25. We consider the Black–Scholes–Samuelson model, where the dynamics of the underlying asset have the form

$$\mathrm{d}S_t = rS_t\mathrm{d}t + \sigma S_t \mathrm{d}W_t, \quad S_0 = s \in \mathbb{R}_+, \tag{2.11}$$

where W is a standard Brownian motion, while we assume we are already under the martingale measure. We want to compute the price of a European call option with payoff function

$$f(S_T) = (S_T - K)^+, \tag{2.12}$$

together with the 95% and 99% confidence intervals. Algorithm 1 contains pseudo-code for the Black–Scholes formula for a European call, while Algorithm 2 contains pseudo-code for the computation of the Monte Carlo price and the RMSE. An outcome of this example is shown in Figure 2.3, where the Monte Carlo price for different sample sizes together with the corresponding confidence intervals are plotted together with the Black–Scholes price. One should notice how the Monte Carlo price converges to the Black–Scholes price and how the confidence intervals shrink as the sample size M increases.

Algorithm 1 Pseudo-code for the Black–Scholes formula

input: S_0, K, T, r, σ
$d_1 \leftarrow (\log(S_0/K) + (r + \sigma^2/2) \cdot T)/(\sigma \cdot \sqrt{T})$
$d_2 \leftarrow (\log(S_0/K) + (r - \sigma^2/2) \cdot T)/(\sigma \cdot \sqrt{T})$
price $\leftarrow S_0 \cdot \Phi(d_1) - K \cdot \mathrm{e}^{-r \cdot T} \cdot \Phi(d_2)$
output: price

Algorithm 2 Pseudo-code for MC simulation in the Black–Scholes model

input: S_0, K, T, r, σ, M
$W \leftarrow M$ independent samples from the standard normal distribution
$S \leftarrow S0 \cdot \exp(\sigma \cdot \sqrt{T} \cdot W + (r - \sigma^2/2) \cdot T)$
$C \leftarrow \exp(-r \cdot T) \cdot \max\{S - K, 0\}$
price $\leftarrow \mathrm{sum}\{C\}/M$
varest $\leftarrow \mathrm{sum}\{(\mathrm{price} - C)^2\}/(M - 1)$
rmse $\leftarrow \sqrt{\mathrm{varest}/M}$
output: price, rmse

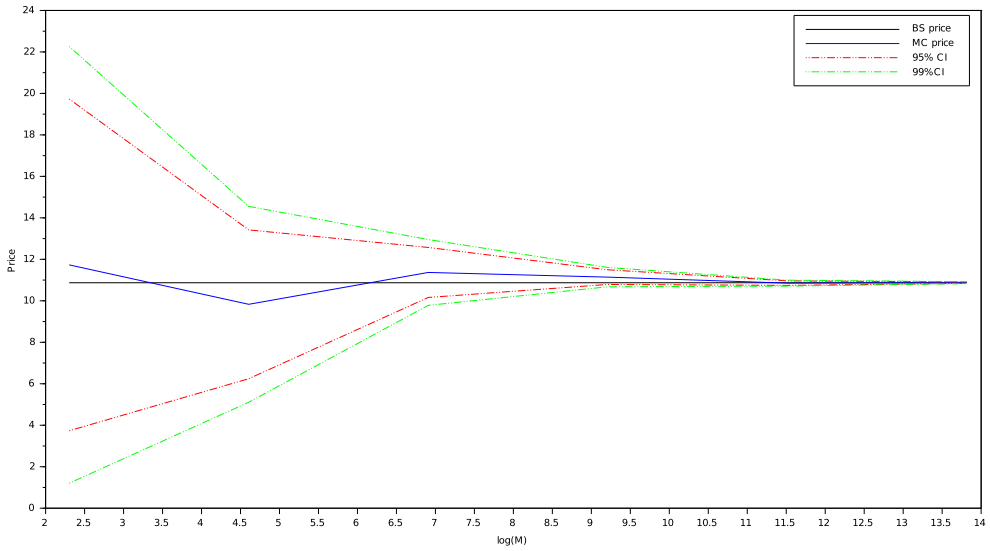


Figure 2.3: Convergence of the Monte Carlo price of a European call option to the Black–Scholes price as a function of the sample size M, together with the 95% and 99% confidence intervals.

Variance reduction

Although there are no obvious handles on how to increase the convergence rate in Proposition 2.21, we might be able to improve the constant factor in the RMSE by reducing the variance $\sigma(f;X)^2 = \mathrm{var}(f(X))$. The idea is to obtain, in a systematic way, random variables Y and functions g such that $\mathbb{E}[g(Y)] = \mathbb{E}[f(X)]$, but with smaller variance $\mathrm{var}(g(Y)) < \mathrm{var}(f(X))$. Inserting $\sigma(g;Y) = \sqrt{\mathrm{var}(g(Y))}$ into (2.8) shows that such an approach will decrease the computational work—proportional to the number of trajectories—provided that the generation of samples from $g(Y)$ is not prohibitively more expensive than the generation of samples from $f(X)$. This leads then to a faster numerical scheme, since the same error can be achieved with fewer samples.

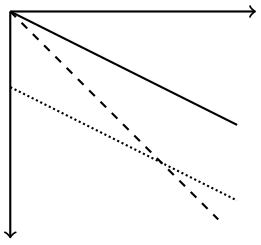


Figure 2.4: Improved convergence rate vs. improved constant.

A pictorial representation of the potential improvement is available in Figure 2.4, where the log-error (y-axis) is plotted against the log-number of samples (x-axis). The convergence rate of the Monte Carlo method is depicted with the solid line with slope $\frac{1}{2}$. An improved convergence rate would lead to a line with different slope, *e.g.* the dashed line with slope 1 in the figure above. On the other hand, an improved constant leads to a parallel shift of the line with slope $\frac{1}{2}$, see the dotted line in the figure above.

Antithetic variates

Let us start with the following simple observation: If U has the uniform distribution, then the same is true for $1-U$. Similarly, if B has the d-dimensional normal distribution, then so does $-B$. Therefore, these transformations do not change the expected value $\mathbb{E}[f(X)]$, if $X = U$ or $X = B$.[7] In general, assume there exists 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 variates* Monte Carlo estimate by

$$I_M^A[f;X] = \frac{1}{M}\sum_{i=1}^{M} \frac{f(X_i) + f(\widetilde{X}_i)}{2}. \tag{2.13}$$

Since $\mathbb{E}[(f(X_i) + f(\widetilde{X}_i))/2] = \mathbb{E}[f(X)]$, (2.13) is another unbiased estimator for $I[f;X]$. If we assume that the actual simulation of $(f(X_i) + f(\widetilde{X}_i))/2$ takes at most two times the computing 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 $I_{2M}[f;X]$.[8] Then, the application of antithetic variates makes sense if the mean square error of $I_M^A[f;X]$ is smaller than the MSE of $I_{2M}[f;X]$, *i.e.* if

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

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

$$\mathrm{cov}\left(f(X), f(\widetilde{X})\right) < 0. \tag{2.14}$$

In other words, the antithetic variates Monte Carlo method should be used when the negative dependence between the input variables X and $\widetilde{X}$ (think of U and $1-U$ or B and $-B$) produces

[7] 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.

[8] 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 $(f(X_i) + f(\widetilde{X}_i))/2$ than to compute two realizations of $f(X_i)$.

also negative dependence between the output variables $f(X)$ and $f(\widetilde{X})$. A simple, sufficient condition for the latter is the monotonicity of the function f that maps inputs to outputs.

The calculations above yield also the following decomposition for the MSE of the antithetic variates Monte Carlo method:

$$\mathrm{MSE}\big[I_M^A\big] = \mathrm{MSE}\big[I_{2M}\big] + \frac{\mathrm{cov}\big(f(X), f(\widetilde{X})\big)}{2M}. \tag{2.15}$$

In other words, the improvement over the standard Monte Carlo method, if any, comes in the form of an *additive* factor (which obviously tends to zero as $M \to \infty$). The larger the negative dependence between $f(X)$ and $f(\widetilde{X})$, the larger this factor as well, for fixed M.

Remark 2.26. Exercise 2.9 asks the reader to justify the application of the antithetic variates Monte Carlo method for pricing a European call option in the Black–Scholes model, theoretically and by computing the sample covariance. Once the sample estimator for the covariance is coded, one could notice that the speed-up factor depends on the strike K (all other parameters equal), and is larger for deep-in-the-money options, *i.e.* for $K \to 0$.

Control variates

Assume there exists a random variable Y and a functional g such that we know the exact value of $I[g;Y] = \mathbb{E}[g(Y)]$. (Note that we allow for $Y = X$.) Then obviously

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

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

$$I_M^{C,\lambda}[f;X] := \frac{1}{M}\sum_{i=1}^{M}\Big(f(X_i) - \lambda g(Y_i)\Big) + \lambda I[g;Y], \tag{2.16}$$

where (X_i, Y_i) are independent realizations of (X, Y). Similar to the situation with antithetic variates, we may assume that the simulation of $I_M^{C,\lambda}[f]$ takes at most c times the computing time of the simulation of $I_M[f]$, where $c > 1$ often is quite small, especially if $X = Y$. We are going to choose the parameter λ such that $\mathrm{var}(f(X) - \lambda g(Y))$ is minimized. A simple calculation yields that

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

which is minimized by choosing λ to be equal to

$$\lambda^\star = \frac{\mathrm{cov}(f(X), g(Y))}{\mathrm{var}(g(Y))}. \tag{2.17}$$

Applying Proposition 2.21, we get that the mean square error for the standard and the control variates Monte Carlo simulations compare as follows:

$$\mathrm{MSE}\big[I_M^{C,\lambda^\star}\big] = \frac{\mathrm{var}(f(X))}{M}(1-\rho^2) \leqslant \frac{\mathrm{var}(f(X))}{M} = \mathrm{MSE}\big[I_M\big], \tag{2.18}$$

where ρ denotes the correlation coefficient between $f(X)$ and $g(Y)$. In other words, the improvement of the control variates Monte Carlo simulation over the standard Monte Carlo method comes in the form of a *multiplicative* factor. Assuming that the computational work per realization is c times higher using control variates, (2.8) implies that the control variates technique is $1/(c(1-\rho^2))$-times faster than standard Monte Carlo. In particular, the improvement in speed from the use of control variates is larger as the correlation between $f(X)$ and $g(Y)$ becomes higher. If, for example, $\rho = 0.8$ and $c = 2$ the speed-up factor equals 1.38, while if $\rho = 0.95$ the speed-up factor equals 5.

Remark 2.27. We can determine the optimal factor $\lambda^\star$ only 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—see Exercise 2.6—obtained by (standard) Monte Carlo simulations with a smaller sample size.

A natural question now is how to find, or construct, good control variates. There does not exist a general answer since these are typically specified by the problem at hand. However, in option pricing the underlying asset provides a virtually universal source of control variates, because

$$e^{-rT}\mathbb{E}[S_t] = S_0 \tag{2.19}$$

for every $t \geqslant 0$, assuming that $\mathbb{E}$ denotes the expectation with respect to a martingale measure. Moreover, simple options that admit a closed-form solution can be used as control variates for the pricing of more complex derivatives, see *e.g.* Exercise ... with geometric and arithmetic Asian options. In additions, simple models can be used as control variates for option pricing in more advanced models; for example, the Black–Scholes model can serve as control variate for stochastic volatility models.

Example 2.28. Assume we want to compute the price of an option with payoff $f(S_T)$ and we are given a sample $S_T^1, \dots, S_T^M$ from the law of S_T. The control variates Monte Carlo estimator takes the form

$$I_M^{C,\lambda^\star}[f; S_T] = \frac{1}{M}\sum_{i=1}^{M}\left\{f(S_T^i) - \lambda^\star S_T^i\right\} + \lambda^\star S_0, \tag{2.20}$$

where $\lambda^\star$ can be also replaced by the sample estimator $\lambda_M^\star$. The interest rate is set to zero, for simplicity. If $f(S_T) = (S_T - K)^+$, *i.e.* we are pricing a call option, then

$$\lambda^\star = \frac{\operatorname{cov}\left((S_T - K)^+, S_T\right)}{\operatorname{var}(S_T)}, \tag{2.21}$$

and the efficiency of the control variate depends, essentially, on the strike K (all other parameters equal). In particular, for $K = 0$ we obviously have perfect correlation and the method is very effective. On the other hand, for deep out-of-the-money options (*i.e.* for large K) the correlation becomes quite low and the effectiveness of the method deteriorates.

Stratified sampling

The main principle of stratified sampling is to partition the sample space into disjoint subsets, called *strata*, and to constrain the number of samples selected from each stratum. Let $A_1, \dots, A_L$ be disjoint subsets of $\mathbb{R}^d$ such that $\mathbb{P}(X \in \cup_l A_l) = 1$. Then, using the law of total probability, we can estimate $f(X)$ as follows

$$\mathbb{E}[f(X)] = \sum_{l=1}^{L}\mathbb{E}[f(X)|X \in A_l]\mathbb{P}(X \in A_l) = \sum_{l=1}^{L} p_l \mathbb{E}[f(X)|X \in A_l], \tag{2.22}$$

where $p_l = \mathbb{P}(X \in A_l)$. In the standard Monte Carlo method, we generate $X_1, \dots, X_M$ which are independent and distributed identically to X, and the fraction of samples that belong to each stratum A_l is in general not equal to p_l, although it converges to p_l as $M \to \infty$. In contrast, in stratified sampling we preselect what fraction of samples should belong to each stratum, and every sample drawn from A_l has the distribution of X *conditional on* $X \in A_l$.

Let M denote the total size of the sample. For every $l = 1, \dots, L$, let $q_l = \frac{M_l}{M}$ denote the fraction of observations from the stratum A_l, and X_{lk}, $k = 1, \dots, M_l$, be i.i.d. realizations from the distribution of X conditional on $X \in A_l$. An unbiased estimator for the expectation in the RHS of (2.22) is provided by the sample average, *i.e.* by $\frac{1}{M_l}\sum_{k=1}^{M_l} f(X_{lk})$. Therefore, the *stratified sampling estimator* takes the form

$$I_M^{ST}[f; X] = \sum_{l=1}^{L} p_l \frac{1}{M_l}\sum_{k=1}^{M_l} f(X_{lk}) = \frac{1}{M}\sum_{l=1}^{L}\frac{p_l}{q_l}\sum_{k=1}^{M_l} f(X_{lk}). \tag{2.23}$$

Remark 2.29. The strata can also depend on another variable Z, called the *stratifying variable*, which is possibly dependent on X. In that case, the estimator has the same form, *i.e.*

$$I_M^{ST}[f;X] = \frac{1}{M}\sum_{l=1}^{L}\frac{p_l}{q_l}\sum_{k=1}^{M_l} f(X_{lk}), \tag{2.24}$$

where now $p_l = \mathbb{P}(Z \in A_l)$ and $(X_{lk})_k$ are i.i.d. realizations from the distribution of X conditional on $Z \in A_l$. We will use this more general formulation from now on.

Therefore, in order to effectively implement a stratified sampling estimator we should select and optimize the following variables: the stratification variable Z, the strata $A_1, \dots, A_L$ and the allocations $M_1, \dots, M_L$. Moreover, we should also know how to efficiently sample from the law of (X, Z) conditional $Z \in A_l$.

Let us now compare the variance of the stratified sampling estimator with the variance of the standard Monte Carlo estimator. We will use the following notation:

$$\mu_l = \mathbb{E}[f(X_{lk})] = \mathbb{E}[f(X)|Z \in A_l] \quad \text{and} \quad \sigma_l^2 = \mathrm{var}[f(X_{lk})] = \mathrm{var}[f(X)|Z \in A_l], \tag{2.25}$$

and then the variance of the stratified sampling estimator, using the *proportional allocation* $q_l = p_l$, is provided by

$$\mathrm{var}\left(I_M^{ST}\right) = \frac{1}{M}\sum_{l=1}^{L} p_l\sigma_l^2. \tag{2.26}$$

On the other hand, the variance of the standard Monte Carlo estimator equals $\mathrm{var}(I_M) = \mathrm{var}(f(X))/M$, where

$$\begin{aligned}\mathrm{var}(f(X)) &= \mathbb{E}[f(X)^2] - \mathbb{E}[f(X)]^2 \\ &= \sum_{l=1}^{L} p_l\mathbb{E}[f(X)^2|Z \in A_l] - \left(\sum_{l=1}^{L} p_l\mathbb{E}[f(X)|Z \in A_l]\right)^2 \\ &= \sum_{l=1}^{L} p_l(\sigma_l^2 + \mu_l^2) - \left(\sum_{l=1}^{L} p_l\mu_l\right)^2. \qquad (2.27)\end{aligned}$$

Therefore, the MSE of the stratified sampling Monte Carlo estimator admits the following decomposition:

$$\mathrm{MSE}\big[I_M^{ST}\big] = \mathrm{MSE}\big[I_M\big] + \frac{1}{M}\sum_{l=1}^{L} p_l\mu_l^2 - \frac{1}{M}\left(\sum_{l=1}^{L} p_l\mu_l\right)^2, \tag{2.28}$$

therefore any potential improvement over the standard Monte Carlo method comes in the form of an additive factor again. Now, Jensen's inequality yields that

$$\sum_{l=1}^{L} p_l\mu_l^2 \geqslant \left(\sum_{l=1}^{L} p_l\mu_l\right)^2,$$

therefore stratified sampling Monte Carlo with proportional allocation leads to a reduction of the variance of the estimator.

One can achieve a further reduction of the variance by optimizing the allocations, *i.e.* by selecting the fractions q_l such that the variance of the estimator is minimized. The variance of the stratified sampling estimator in general has the form

$$\mathrm{var}\left(I_M^{ST}\right) = \frac{1}{M}\sum_{l=1}^{L}\frac{p_l^2}{q_l}\sigma_l^2,$$

and minimizing this quantity subject to the constraints $q_l \in (0,1)$ and $\sum_l q_l = 1$ leads to the optimal allocation provided by

$$q_l^\star = \frac{p_l \sigma_l}{\sum_k p_k \sigma_k}.$$

The variance of the estimator with the optimal allocation equals then

$$\operatorname{var}\left(I_M^{ST,\star}\right) = \frac{1}{M}\left(\sum_{l=1}^{L} p_l \sigma_l\right)^2.$$

Using Jensen's inequality once again and comparing with (2.26) we observe that optimizing the allocations leads to a further reduction of the variance.

Remark 2.30. Similar to other methods, the variances σ_l are typically not known explicitly. One could then use sample estimators with a smaller sample size to compute $q_l^\star$ and then use the estimated optimal allocations in a second simulation run.

Importance sampling

Importance sampling is related to the acceptance-rejection method and also to Girsanov's theorem (or changes of measures). The idea is to sample more often in regions where the variance is higher, thus increasing the sampling efficiency. 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. The quantity p/q is called the likelihood ratio or the Radon–Nikodym derivative. Thus, a Monte Carlo estimate for $I[f]$ is given by

$$I_M^{IS}[f;X] = \frac{1}{M}\sum_{i=1}^{M} f(Y_i)\frac{p(Y_i)}{q(Y_i)} = I_M\left[f\frac{p}{q};Y\right]. \tag{2.29}$$

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

$$\operatorname{var}\left(f(Y)\frac{p(Y)}{q(Y)}\right) + I[f;X]^2 = E\left[\left(f(Y)\frac{p(Y)}{q(Y)}\right)^2\right] = E\left[f(X)^2\frac{p(X)}{q(X)}\right]. \tag{2.30}$$

So how do we have to choose q? Assume for a moment that $f \geqslant 0$ 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 the 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). However, this also means that there is no general way to implement control variates or importance sampling.

Exercise 2.6. Show that an unbiased estimator of $\sigma^2(f;X)$ is

$$\sigma_M^2(f;X) = \frac{1}{M-1}\sum_{i=1}^{M}\Big(f(X_i) - I_M[f;X]\Big)^2 \tag{2.31}$$

and an unbiased estimator of $\operatorname{cov}(f(X),g(Y))$ is

$$\operatorname{cov}_M(f(X),g(Y)) = \frac{1}{M-1}\sum_{i=1}^{M}\Big(f(X_i) - I_M[f;X]\Big)\Big(g(Y_i) - I_M[g;Y]\Big). \tag{2.32}$$

Exercise 2.7. Compute the price of a European call option in the Black–Scholes model using Monte Carlo simulation, as well as the 95% and 99% confidence intervals. Study the convergence and the asymptotic normality of the error. Then, use (2.8) for a more systematic approach.

Exercise 2.8. Compute the expected value of $1/\sqrt{U}$ for a uniform random variable U using Monte Carlo simulation. Study the speed of convergence and whether the errors are still asymptotically normal.

Hint: This exercise shows that if we want to compute the expected value of an integrable random variable, which is not square integrable, the above analysis does not apply.

Exercise 2.9. Compute the price of a European call option in the Black–Scholes model using the antithetic variates Monte Carlo method. Justify why the method works

(i) numerically, by computing the sample covariance;

(ii) theoretically, by showing that the map from inputs to outputs is monotone.

Exercise 2.10. Compute the price of a European call option in the Black–Scholes model using the control variates Monte Carlo method where the underlying price is the control. Study how the efficiency of the method depends on the strike price and compare the convergence rates with Exercises 2.7 and 2.9.

Chapter 3

Deterministic integration techniques

3.1 Quasi Monte Carlo simulation

Monte Carlo simulation provides a method to compute numerically integrals of the form

$$I[f] := \int_{[0,1]^d} f(x)dx. \tag{3.1}$$

In fact, by composition with the inverse of the distribution function, all the integration problems in the previous section were of the form (3.1). This means that we use the approximation

$$J_M[f] := \frac{1}{M}\sum_{i=1}^{M} f(x_i), \tag{3.2}$$

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. However, they are, in fact, still deterministic. The idea of Quasi Monte Carlo (QMC) simulation is to choose instead a deterministic sequence $x_i \in [0,1]^d$ which is especially "evenly distributed" in $[0,1]^d$. Figure 3.1 shows samples in $[0,1]^2$ as generated from a

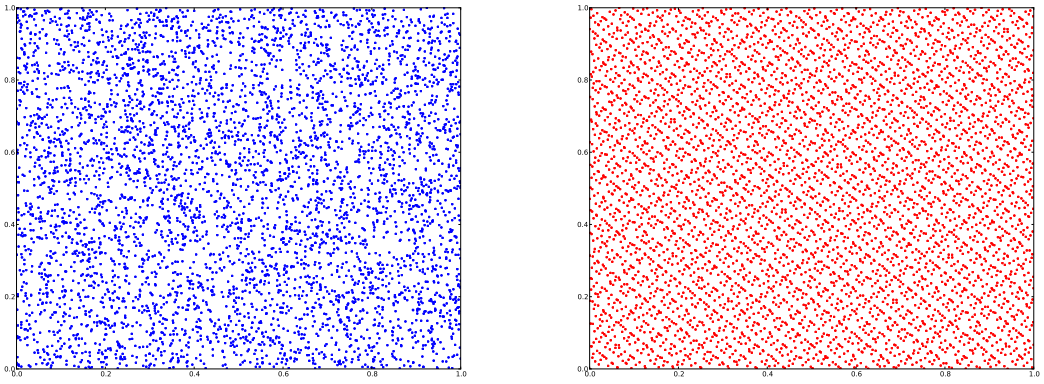


Figure 3.1: Pseudo random samples in $[0,1]^2$ (left picture) versus quasi random ones (right picture)

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 3.1. So, in some sense the idea is to replace pseudo random numbers by "more evenly distributed" but deterministic sequences.

For more information on Quasi Monte Carlo methods, we refer to Glasserman [23] and the survey articles by Caflisch [6] and L'Ecuyer [35].

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

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

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

Definition 3.1. 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

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

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.,

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

The quality of the quadrature rule (3.2) will depend both on the uniformity of the sequence (measured by some form of discrepancy) and on 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 simulation will generally require much more regularity. One typical measure of regularity is the following.

Definition 3.2. The *variation in the sense of Hardy-Krause* is recursively defined as follows: for a one-dimensional function $f : [0,1] \to \mathbb{R}$

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

and[1] for a function $f : [0,1]^d \to \mathbb{R}$

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

where $f_1^{(j)}$ denotes the restriction of f to the boundary $x^j = 1$, i.e.

$$f_1^{(j)} : [0,1]^{d-1} \to \mathbb{R} \;\text{ s.t. }\; x \mapsto f(x_1, \cdots, x_j = 1, \cdots, x_d), \quad j \in \{1, \ldots, d\}$$

The convergence of the Monte Carlo estimator to the true value follows from the law of large numbers, however the same argumenation does not apply for the Quasi Monte Carlo method since the sequence (x_i) is deterministic. The following result, combined with sequences of low discrepancy, justifies the convergence of $J_M[f]$ in (3.2) to $I[f]$.

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

Theorem 3.3. *For any integrable function* $f : [0,1]^d \to \mathbb{R}$ *the* Koksma-Hlawka inequality *holds:*

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

Remark 3.4. The Koksma-Hlawka inequality is a deterministic upper bound for the integration error, a worst case bound. In contrast, for the Monte-Carlo method, we only got probabilistic bounds (see Proposition 2.21), which could be seen as bounds for the average case. However, 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 L'Ecuyer [35] 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 3.3 for $d = 1$. Assume that $f \in C^1([0,1])$. Then for any $0 \leqslant x \leqslant 1$ 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

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

□

Sequences of low discrepancy

According to Theorem 3.3, the QMC method will perform well when the star discrepancy of the quasi-random sequence is small and converges to zero as the sample gets larger, which leads to *sequences of low discrepancy*.

Definition 3.5. We say that a sequence $(x_i)_{i \in \mathbb{N}}$, $x_i \in [0,1]^d$, has low discrepancy, if

$$D_M^* \leqslant c \frac{\log(M)^d}{M^{-1}}.$$

We give a few examples of sequences of low discrepancy.

Example 3.6. Choose a prime number p (or more generally, an integer $p \geqslant 2$); this is the *basis*. Compute, for every $k \in \mathbb{N}_0$, the coefficients $a_j(k)$, $j \geqslant 0$, in basis p, i.e. the p-ary expansion of k

$$k = \sum_{j=0}^{\infty} a_j(k) p^j.$$

Define the map $\psi_p : \mathbb{N}_0 \to [0,1[$ by

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

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

Example 3.7. Let the basis $p = 2$ and consider the integer $k = 7$. The 2-ary expansion of 7 is $7 = 2^0 + 2^1 + 2^2$, thus the coefficients are $a_0(7) = a_1(7) = a_2(7) = 1$. Then, using the map ψ_2 we get that

$$\psi_2(7) = \sum_{j \geqslant 0} \frac{a_j(7)}{2^{j+1}} = \frac{7}{8} \in [0,1].$$

Example 3.8. 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 $x_i^j = \psi_{p_j}(i)$, $j = 1, \dots, d$.

Additionally, there are several other prominent families of sequences with low discrepancy, like the *Sobol* or *Faure* sequences.

Remark 3.9. 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

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

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 low discrepancy, see Figure 3.2.

Remark 3.10. Clearly, a very evenly spaced (finite) sequence is given by taking all the $(n+1)^d$ points $\big\{0, \frac{1}{n}, \frac{2}{n}, \dots, 1\big\}^d$ 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 $J_{M+1}[f]$ 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 $M < (n+1)^d$ 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.

Consider a sequence of low discrepancy. Then the Koksma-Hlawka inequality, when applicable, implies that the quadrature error satisfies

$$\big|I[f] - J_M[f]\big| \leqslant \frac{V[f] c \log(M)^d}{M}, \tag{3.3}$$

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 statement 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; see also Figure 3.3. For instance, in dimension $d = 8$, we only have

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

Remarks on Quasi Monte Carlo

Low dimensionality

It is generally difficult to construct good sequences of low discrepancy in high dimensions, i.e. for $d \gg 1$. 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 on the first two

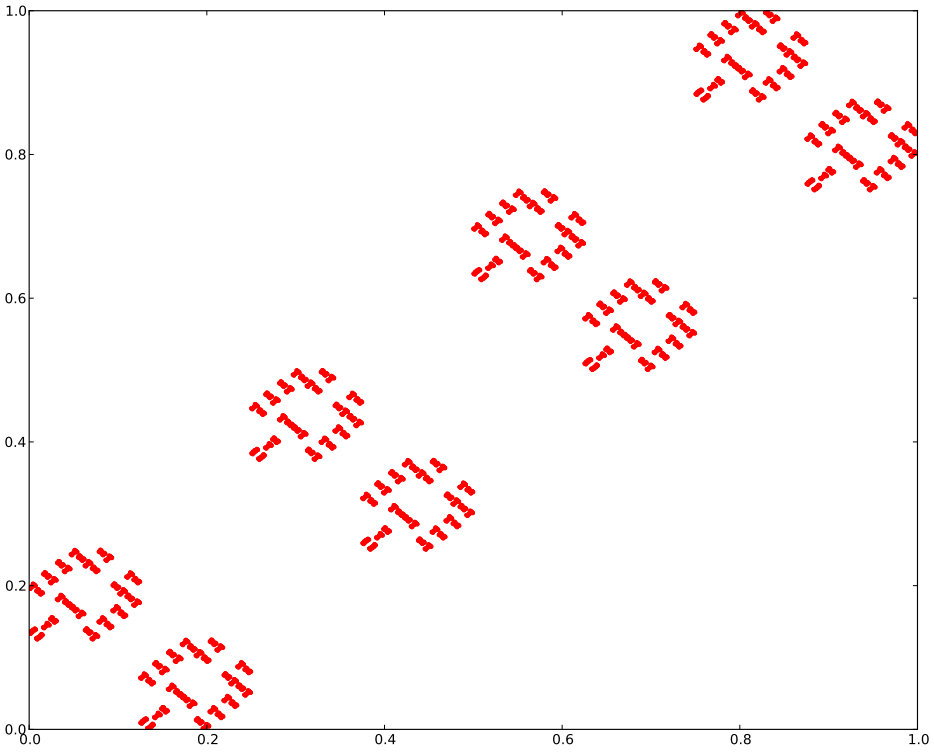


Figure 3.2: Pairs of one-dimensional Sobol numbers

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 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 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 not 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, for example by applying a random shift, i.e., for a d-dimensional uniform random variable

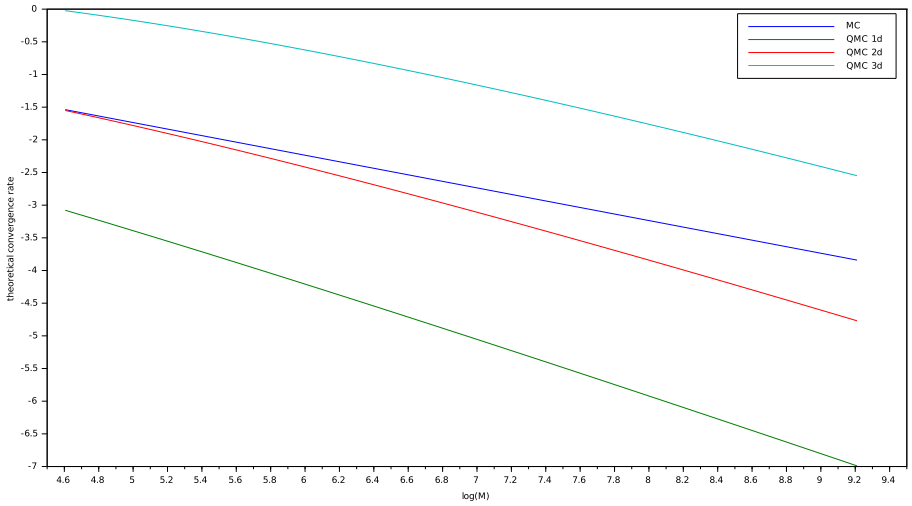


Figure 3.3: Comparison of the theoretical convergence rate for the Monte Carlo and Quasi Monte Carlo methods, up to dimension three. We can observe that, for a relatively small number of simulations (up to 10^4), MC converges faster (in theory) than QMC in dimension 3.

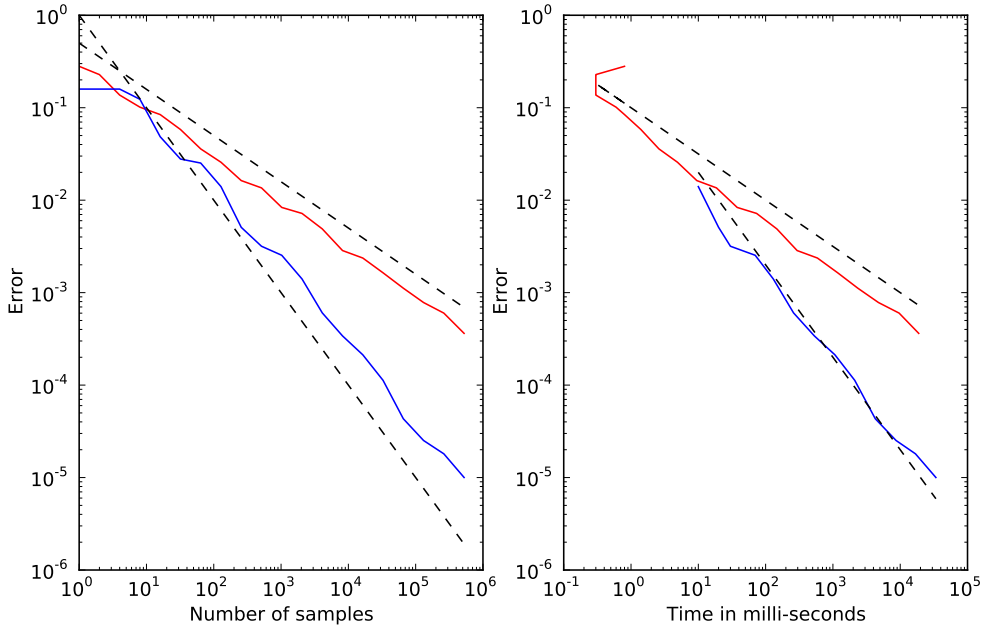


Figure 3.4: A call option in the Black-Scholes model using Monte Carlo and Quasi Monte Carlo simulation. Red: MC simulation, blue: QMC simulation, black: Reference lines proportional to $1/M$ and $1/\sqrt{M}$.

U consider

$$X := (x_i + U \pmod 1)_{i\in\mathbb{N}}\,. \tag{3.4}$$

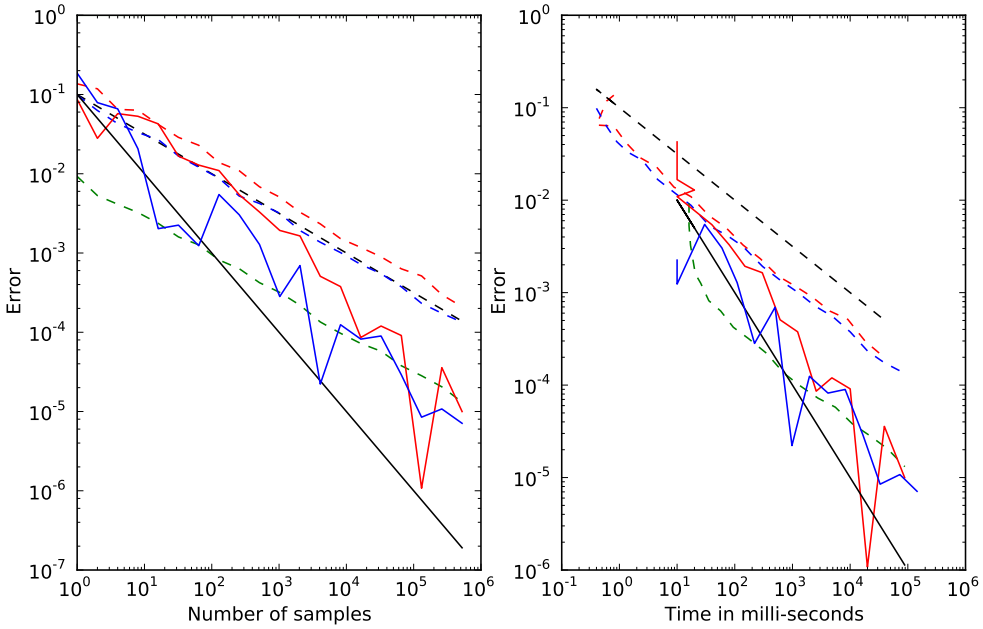


Figure 3.5: An Asian option using Monte Carlo and Quasi Monte Carlo simulation (Solid lines: QMC simulation, dashed lines: MC simulation; Red: normal simulation, blue: antithetic variates, green: control variates, black: references line proportional to $1/M$.

Other possible randomizations are presented in L'Ecuyer [35]. Let $J_M[f;X]$ denote the QMC estimate (3.2) based on the randomized sequence X, that is

$$J_M[f;X] = \frac{1}{M}\sum_{i=1}^{M} f(x_i + U \pmod 1).$$

Now fix a number $m \in \mathbb{N}$ and generate m independent realizations X_l of X by sampling m independent realizations U_l of U, $1 \leqslant l \leqslant m$. Then we estimate $I[f]$ by the *randomized Quasi Monte Carlo* (RQMC) estimate

$$J^R_{M;m}[f] := \frac{1}{m}\sum_{l=1}^{m} J_M[f;X_l]. = \frac{1}{m}\sum_{l=1}^{m}\frac{1}{M}\sum_{i=1}^{M} f(x_i + U_l \pmod 1). \tag{3.5}$$

Now we can use the sharp error estimate of Proposition 2.21 based on $\operatorname{var}(J_M[f;X])$, that is

$$\operatorname{RMSE}[J^R_{M;m}] = \frac{\operatorname{var}(J_M[f;X])}{\sqrt{m}}.$$

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, $\operatorname{var}(J_M[f;X])$ will be small. This means that, from the point of view of the Monte Carlo method, RQMC can be seen as another variance reduction technique! L'Ecuyer [35] reports tremendous improvements of the variance as compared to the plain MC or even MC with traditional variance reduction.

Remark 3.11. 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 $m \leqslant 25$. 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 3.1. Solve Exercises 2.7, 2.9, 2.10 using Quasi Monte Carlo simulation. Report the results and compare the speed of convergence with the one obtained by Monte Carlo simulation.

Exercise 3.2. Solve Exercise 2.7 using RQMC. Report the results and the reduction in the variance.

Chapter 4

Sample path generation

4.1 Brownian motion

The generation of sample paths from a stochastic process is the topic of the next section. On the one hand, a stochastic differential equation (SDE) describes the dynamics of a stochastic process in terms of a generating signal, usually a Brownian motion or, more generally, a Lévy process. Thus, in order to solve SDEs numerically we shall first discuss how to (effectively) sample from the driving signal. On the other hand, the generation of sample paths is important for the pricing of path-dependent options, e.g. Asian or barrier options.

In what follows, B 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 of Asian options, where the first method using linear interpolation of the finite sample coincides with a trapezoidal approximation of the integral.)

Example 4.1. We can, of course, sample from the paths of the stock prices S in the Samuelson model by applying any of the sampling techniques for the Brownian motion and then transforming via

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

4.1.1 Cholesky factorization

The first method for generating sample paths from the finite dimensional skeleton $(B_{t_1}, \dots, B_{t_n})$ is based on the following property of Brownian motion:

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

Moreover, in Remark 2.20 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 obtain an n-dimensional normal random vector with covariance matrix Σ by AX, where $\Sigma =$

AA^T. In this particular case, it is easy to find the Cholesky factorization A by

$$A = \begin{pmatrix} \sqrt{t_1} & 0 & \dots & 0 \\ \sqrt{t_1} & \sqrt{t_2 - t_1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{t_1} & \sqrt{t_2 - t_1} & \dots & \sqrt{t_n - t_{n-1}} \end{pmatrix}. \tag{4.1}$$

4.1.2 Random walk approach

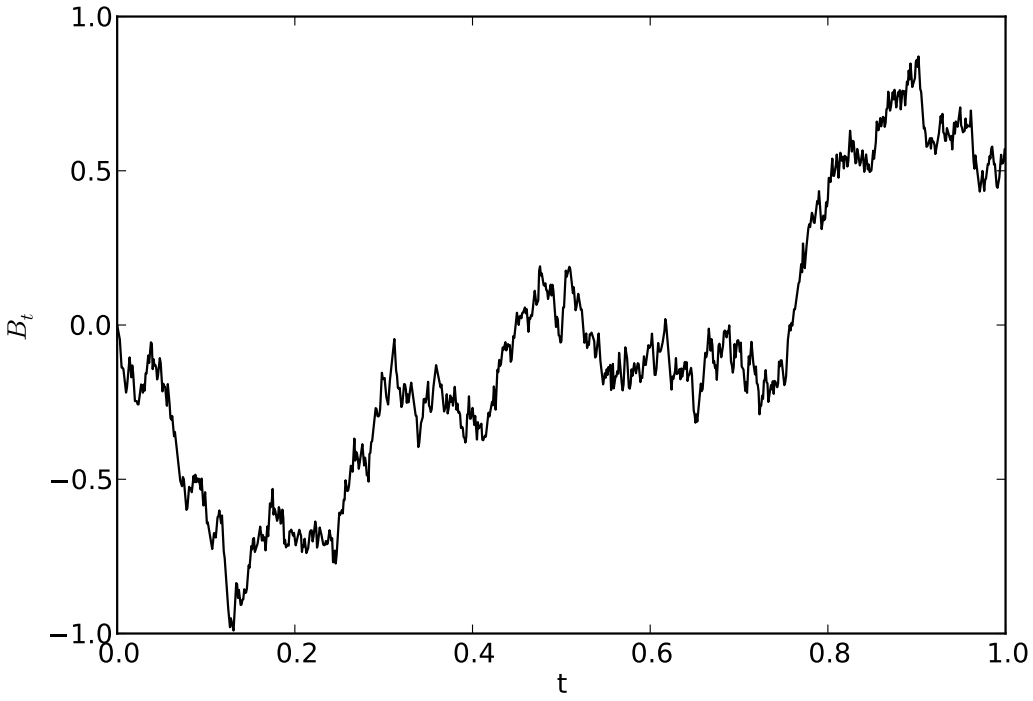


Figure 4.1: Brownian motion simulated using the random walk approach

An alternative way to sample $(B_{t_1}, \dots, B_{t_n})$ is using the independence of the increments of 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

$$\Delta B_1 = \sqrt{\Delta t_1} X_1, \dots, \Delta B_n = \sqrt{\Delta t_n} X_n, \tag{4.2}$$

where X again denotes an n-dimensional standard normal random variable. A closer look at the simulation using the Cholesky factorization (4.1) and the simulation of the increments (4.2) shows that both methods give exactly the same samples from the Brownian motion if we start with the same standard normal sample X. Thus, (4.2) (with the additional summation of the increments ΔB) can be seen as an *efficient* implementation of the matrix multiplication AX.

4.1.3 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

$$(B_s | B_u = x,\, B_t = y) \sim \mathcal{N}\left(\frac{(t-s)x + (s-u)y}{t-u},\, \frac{(s-u)(t-s)}{t-u}\right), \tag{4.3}$$

see e.g. Karatzas and Shreve [30, §5.6B]. 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 4.2. 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 than 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.

4.1.4 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.,

$$\int_0^1 \min(s,t)\psi(s)ds = \lambda\psi(t). \tag{4.4}$$

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

$$B_t = \sum_{i=0}^{\infty} \sqrt{\lambda_i}\psi_i(t)Z_i, \tag{4.5}$$

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.

[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.

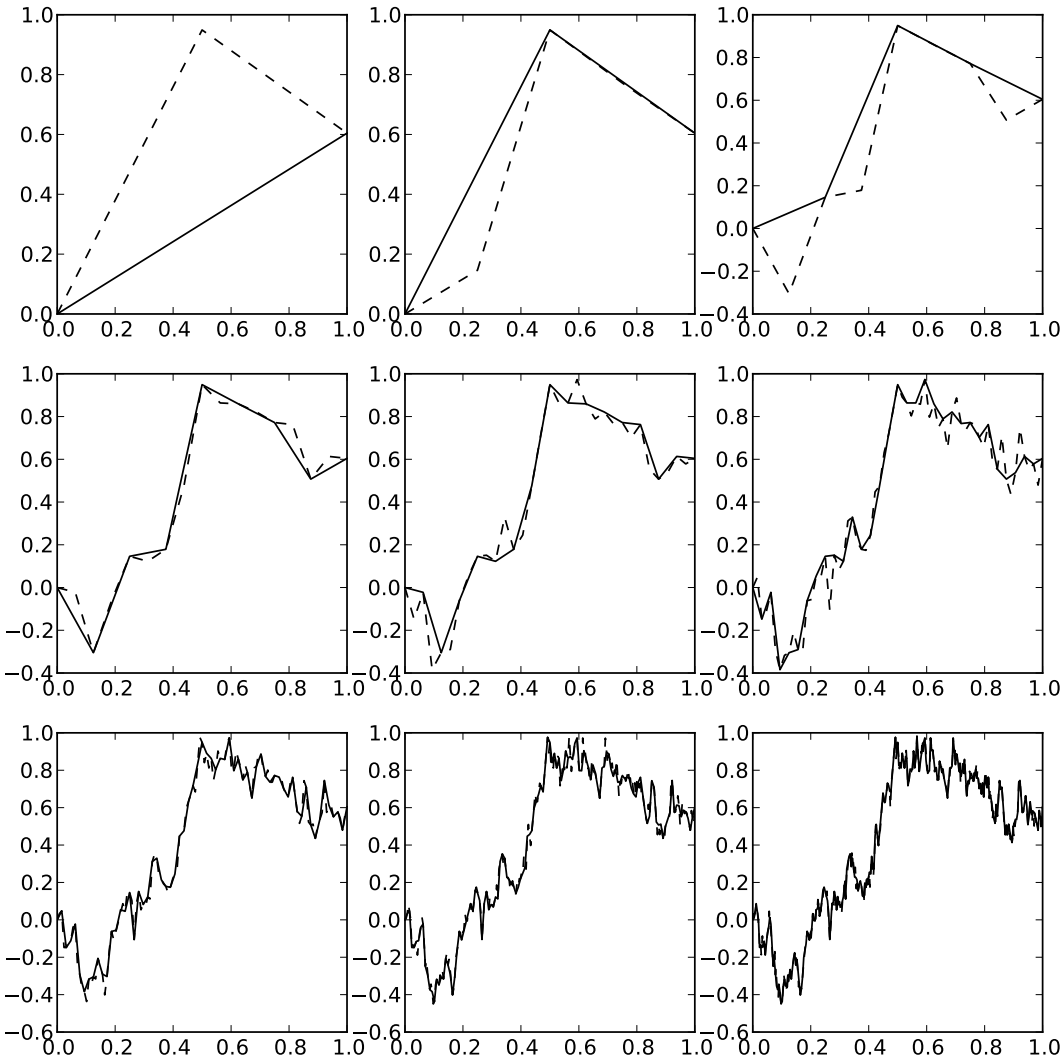


Figure 4.2: Brownian motion constructed by the Brownian bridge approach. Dashed lines correspond to the newly inserted Brownian bridge

4.1.5 Wavelet constructions

Finally, we present a family of constructions which are both general and easy to implement based on wavelets. Note that the previous Karhunen-Loève construction was based on an orthonormal basis of $L^2([0,1])$, specifically the eigenbasis of the covariance operator associated to the Brownian motion. However, similar constructions work for any orthonormal basis.

For sake of concreteness, let us employ the arguably simplest such basis, comprising of Haar wavelets. Those are defined in terms of a single "mother" wavelet

$$\psi(t) \coloneqq \begin{cases} 1, & 0 \leqslant t < \frac{1}{2}, \\ -1, & \frac{1}{2} \leqslant t < 1, \\ 0, & \text{else.} \end{cases} \tag{4.6}$$

The basis functions are now defined by shifting and rescaling the mother wavelet ψ. More specifically,

fix $n \in \mathbb{N} \cup \{0\}$ and $0 \leqslant k \leqslant 2^n - 1$ and define

$$\psi_{n,k}(t) := 2^{n/2}\psi\left(2^n t - k\right). \tag{4.7}$$

Note that the support of $\psi_{n,k}$ is of size 2^n whereas the amplitude is of size $2^{n/2}$. For fixed n and different ks, the Haar functions have disjoint support. it is easy to see that the set of functions $\psi_{n,k}$ forms an orthonormal basis of $L^2([0,1])$, as claimed.

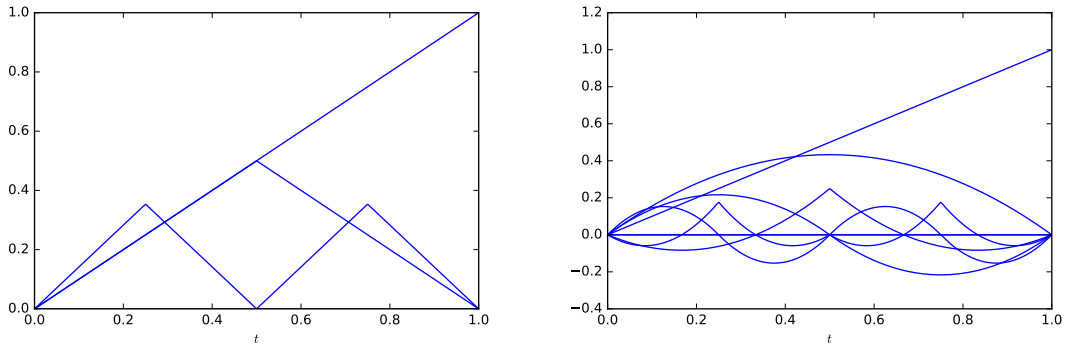
Starting from Lévy [37], this basis has been used to give a very explicit and simple construction of the Brownian motion. Indeed, let X_0 and $X_{n,k}$, $n \geqslant 0$, $0 \leqslant k \leqslant 2^n - 1$, be a sequence of i.i.d. standard normal random variables, then

$$B_t := X_0 t + \sum_{n=0}^{\infty} \sum_{k=0}^{2^n-1} X_{n,k}\Psi_{n,k}(t), \quad t \in [0,1], \tag{4.8}$$

defines a Brownian motion, where

$$\Psi_{n,k}(t) := \int_0^t \psi_{n,k}(s)ds = 2^{-n/2}\Psi(2^n t - k), \tag{4.9}$$

are tent functions. Here, $\Psi(t) = \int_0^t \psi(s)ds$. The interpretation is clear: the terms with low n describe the "macroscopic" behaviour of the Brownian motion, whereas the terms with large n provide fine resolution, see Figure 4.3 for a plot of the integrated Haar basis of orders 0 and 1 and Figure 4.4 for paths generated by wavelets at different resolutions.



(a) Haar basis up to order $n = 1$

(b) Alpert-Rokhlin basis of degree $q = 2$ up to order $n = 1$, see Example 4.3

Figure 4.3: Integrated wavelet basis functions

Of course, for actual applications, we have to truncate (4.8) at some point, giving us an approximation

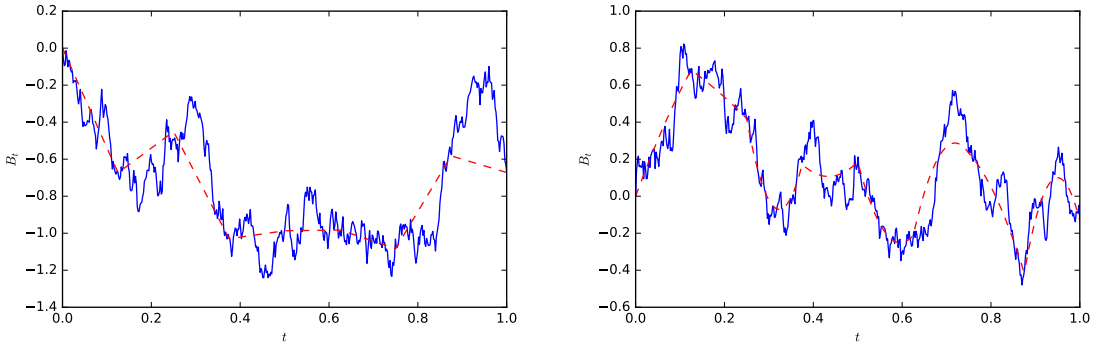
$$B_t^{(N)} := X_0 t + \sum_{n=0}^{N} \sum_{k=0}^{2^n-1} X_{n,k}\Psi_{n,k}(t), \quad t \in [0,1]. \tag{4.10}$$

Note that by construction we have that

$$B_t^{(N)} = B_t \text{ for } t \in \mathcal{D}^{(N)} := \left\{ k2^{-N-1} \;\middle|\; 0 \leqslant k \leqslant 2^{N+1} \right\},$$

i.e., the approximate Brownian motion $B_t^{(N)}$ is exact at all points contained in the dyadic grid of level N. This fact can be easily seen noting that $\Psi_{n,k}(t) = 0$ for $t \in \mathcal{D}^{(N)}$ and $n > N$.

The Haar basis is probably the simplest and most popular wavelet basis in the context of the construction of Brownian motion, but other wavelets can be used, as well. In particular, the recent review article [24] advocates the use of the (higher order) Alpert-Rokhlin multiwavelet basis. For any $q \in \mathbb{N}$, this basis consists of piecewise polynomial functions of order $q-1$ generated by q mother wavelets $\psi^{q,1}, \ldots, \psi^{q,q}$, which are polynomials of order $q-1$ on $[0,1/2]$ and $[1/2,1]$, respectively.



(a) Haar wavelets

(b) Alpert-Rokhlin wavelets of degree $q = 2$, see Example 4.3

Figure 4.4: Approximate Brownian motion $B_t^{(N)}$, $0 \leqslant t \leqslant 1$, for $N = 10$ (blue) and $N = 2$ (red) superimposed

Example 4.3. At order two, the Alpert-Rokhlin multiwavelet basis consists of piecewise linear functions. The mother wavelets are given by

$$\psi^{2,1}(t) := \begin{cases} \sqrt{3}(1-4t), & 0 \leqslant t < \frac{1}{2}, \\ \sqrt{3}(4t-3), & \frac{1}{2} \leqslant t < 1, \\ 0, & \text{else}, \end{cases} \quad \psi^{2,2}(t) := \begin{cases} 6t-1, & 0 \leqslant t < \frac{1}{2}, \\ 6t-5, & \frac{1}{2} \leqslant t < 1, \\ 0, & \text{else}. \end{cases}$$

The anti-derivatives are then given by the piecewise quadratic polynomials

$$\Psi^{2,1}(t) := \begin{cases} \sqrt{3}t(1-2t), & 0 \leqslant t < \frac{1}{2}, \\ \sqrt{3}(1-t)(1-2t), & \frac{1}{2} \leqslant t < 1, \\ 0, & \text{else}, \end{cases} \quad \Psi^{2,2}(t) := \begin{cases} t(3t-1), & 0 \leqslant t < \frac{1}{2}, \\ (t-1)(3t-2), & \frac{1}{2} \leqslant t < 1, \\ 0, & \text{else}. \end{cases}$$

Similarly to the Haar case, the Brownian motion is given by

$$B_t = X_0 t + X_1 \sqrt{3}t(1-t) + \sum_{n=0}^{\infty} \sum_{k=0}^{2^n-1} \sum_{p=1}^{2} X_{n,k,p} \Psi_{n,k}^{2,p}(t),$$

where the random variables $X_0, X_1, X_{n,k,p}$, $n =\in \mathbb{N} \cup \{0\}$, $k = 0, \ldots, 2^n - 1$, $p = 1, 2$, are independent standard normals and

$$\Psi_{n,k}^{2,p}(t) := 2^{-n/2} \Psi^{2,p}(2^n t - k).$$

Remark 4.4. Different wavelets basis can lead to very different approximation qualities of the corresponding approximate Brownian paths $B^{(N)}$ at a resolution N to the true path B, as can be seen from Figure 4.4. However, in terms of numerical analysis, this comparison is questionable, as the computational work needs to be considered, as well. From that side, the simplicity of the basis function, as well as the actual number of terms involved at scale N matters, and the optimal choice of wavelet basis will depend on the problem at hand.

4.2 Lévy processes

A Lévy process is a process with stationary and independent increments, and the simplest way to sample its trajectory is the *random walk approach* used for Brownian motion, see 4.1.2, where the normal distribution shall now be replaced by the corresponding infinitely divisible distribution. This is not as straightforward forward as it sounds, since infinitely divisible distributions may not

be closed under convolution; in other words, while the law of X_t follows a given law, the law of the increment $X_{tk/n} - X_{t(k-1)/n}$ will not necessarily follow the same law. A popular distribution that can be simulated using the random walk approach is the normal inverse Gaussian (NIG) distribution.

Exercise 4.1. Show that the NIG distribution is closed under convolution and simulate a trajectory of the NIG Lévy process.

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

$$Z_t = Z_0 + \sum_{i=1}^{N_t} X_i, \tag{4.11}$$

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, $n \geqslant 1$. 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 $T_1 = \min(tU_1, \dots, tU_{N_t})$, $\dots$, $T_{N_t} = \max(tU_1, \dots, tU_{N_t})$.

Example 4.5. Already in the seventies Merton introduced a jump diffusion into financial modelling. He proposed to model the stock price process by the SDE

$$dS_t = \mu S_{t\_}\, dt + \sigma S_{t\_}\, dB_t + S_{t\_}\, dJ_t, \tag{4.12}$$

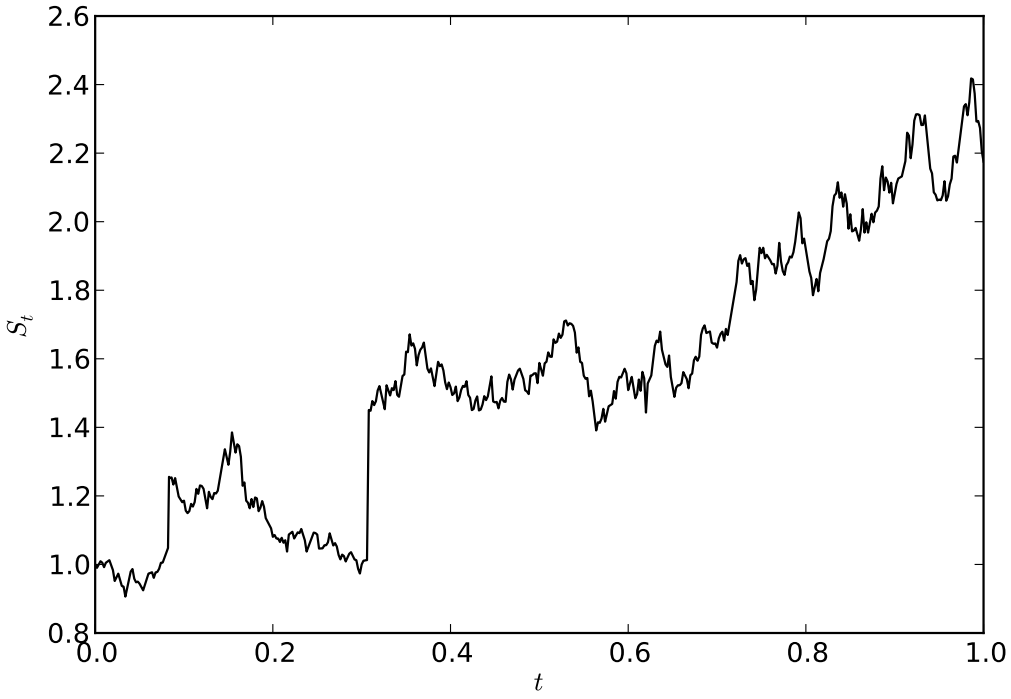


Figure 4.5: Trajectory of Merton's jump diffusion, see Example 4.5.

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 (4.12) 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.,

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

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 (4.12) 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 (4.12) 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 pre-defined 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 4.5. 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 [13, 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 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.)

[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.

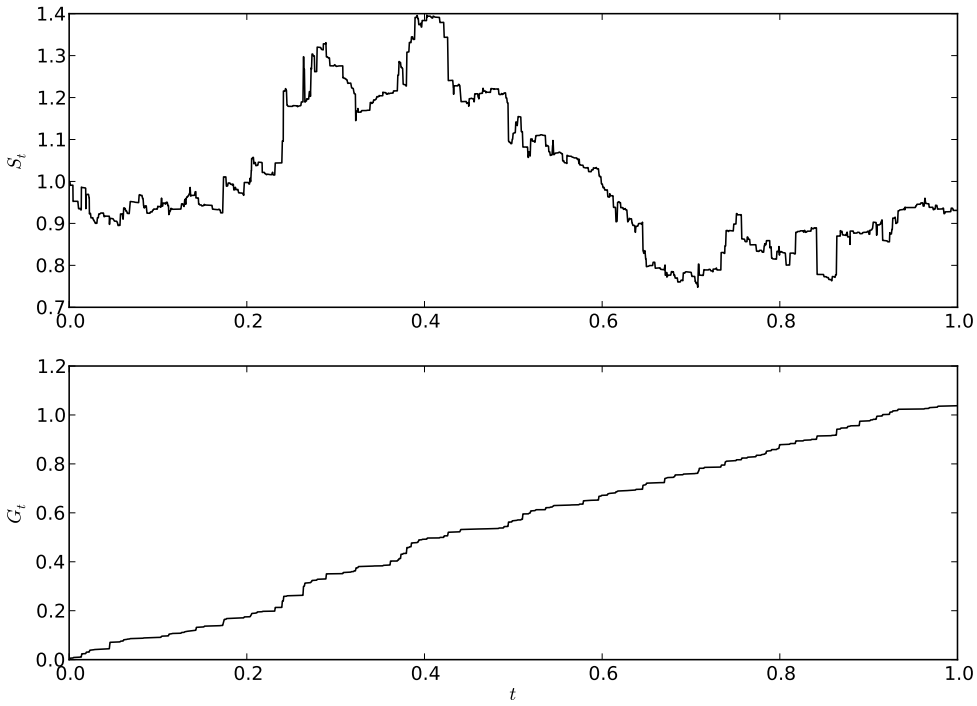


Figure 4.6: Trajectory of the variance-gamma process

Remark 4.6. 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

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

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

$$Z_t^\epsilon := \gamma t + \sum_{0<s\leqslant t} \Delta Z_s \mathbf{1}_{|\Delta Z_s|\geqslant 1} + N_t^\epsilon \tag{4.13}$$

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 [9, Section 6.3, 6.4]) that

$$\operatorname{var}[Z_t - Z_t^\epsilon] = t\int_{|z|<\epsilon} z^2\nu(dz) =: t\sigma(\epsilon)^2. \tag{4.14}$$

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([9, Proposition 6.1]) that

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

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

$$Z_t \approx Z_t^\epsilon + \sigma(\epsilon)B_t, \tag{4.15}$$

which also improves the weak convergence. Let us conclude with a few examples taken from [9].

Example 4.7. *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 [8], 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 4.8. 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 $\lambda_\epsilon \sim -\log(\epsilon)$.

Chapter 5

Discretization of stochastic differential equations

5.1 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.,

$$dX_t = V(X_t)dt + \sum_{i=1}^{d} V_i(X_t)dB_t^i, \tag{5.1}$$

for some vector fields $V, V_1, \dots, V_d : \mathbb{R}^n \to \mathbb{R}^n$, 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 (5.1). 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 (5.1).

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

$$|\mathcal{D}| := \max_{1 \leqslant i \leqslant N} |t_i - t_{i-1}|$$

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

$$\Delta t_i := t_i - t_{i-1}, \ \Delta Y_i := Y_{t_i} - Y_{t_{i-1}}, \quad 1 \leqslant i \leqslant N.$$

Moreover, for $t \in [0,T]$ we set $\lfloor t \rfloor = \sup\{\, t_i \mid 0 \leqslant i \leqslant N,\ t_i \leqslant t \,\}$. We will define the approximation along the grid, i.e., we will define the random variables $\overline{X}_i = \overline{X}_{t_i}$, $0 \leqslant i \leqslant N$. 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 5.1. 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)

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

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

Definition 5.2. 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

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

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

The class of functions $\mathcal{G}$ in Definition 5.2 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 [31].

The Euler-Maruyama method

Fix a grid $\mathcal{D}$ and an SDE driven by a Brownian motion, i.e., of type (5.1). We hope to get some insight into how to approximate the solution by taking a look at a deterministic ODE

$$\dot{x}(t) = V(x(t)), \quad x(0) = x_0 \in \mathbb{R}^n. \tag{5.2}$$

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

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

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

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

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

$$\overline{X}_i = \overline{X}_{i-1} + V(\overline{X}_{i-1})\Delta t_i + \sum_{j=1}^{d} V_j(\overline{X}_{i-1})\Delta B_i^j, \quad 1 \leqslant i \leqslant N. \tag{5.3}$$

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

$$\overline{X}_t = \overline{X}_{\lfloor t \rfloor} + V(\overline{X}_{\lfloor t \rfloor})(t - \lfloor t \rfloor) + \sum_{i=1}^{d} V_i(\overline{X}_{\lfloor t \rfloor})(B_t^i - B_{\lfloor t \rfloor}^i). \tag{5.4}$$

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. Note that the heuristic convergence argument for the Euler method for ODEs would, in fact, not give convergence for a scheme with local order $\mathcal{O}(\sqrt{\Delta t_i})$. Convergence is rather obtained as the "offending" term driven by Brownain motion is a *martingale*.

Before presenting the main theorem, we recall two fundamental lemmas from analysis and probability theory essential for the proof.

Lemma 5.3 (Grönwall's inequality)**.** *Let u be a real-valued, continuous function and let $\alpha, \beta \geqslant 0$. Assuming that for all $0 \leqslant t \leqslant T$ we have*

$$u(t) \leqslant \alpha + \beta \int_0^t u(s)ds,$$

then we get

$$u(t) \leqslant \alpha e^{\beta t}, \quad 0 \leqslant t \leqslant T.$$

Lemma 5.4 (Doob's inequality, L^2 case)**.** *Let $(M_t)_{t\in[0,T]}$ be a square integrable martingale. Then we have*

$$E\left[\sup_{0\leqslant t\leqslant T} |M_t|^2\right] \leqslant 4E\left[|M_T|^2\right].$$

Theorem 5.5. *Suppose that the coefficients of the SDE (5.1) 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*

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

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

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

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

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

Let us start with bounding the term d^0. Note that Jensen's inequality implys for any integrable function g that

$$\left| \int_0^s g(u) du \right|^2 = s^2 \left| \frac{1}{s} \int_0^s g(u) du \right|^2 \leqslant s \int_0^s |g(u)|^2 \, du.$$

Applying this inequality to the integral inside d^0 and using the Lipschitz continuity of V, we obtain

$$\begin{aligned} d_t^0 &\leqslant E\left[\sup_{0 \leqslant s \leqslant t} s \int_0^s \left| V(X_{\lfloor u \rfloor}) - V(\overline{X}_{\lfloor u \rfloor}) \right|^2 du \right] \\ &\leqslant K^2 T E\left[\int_0^t \left| X_{\lfloor u \rfloor} - \overline{X}_{\lfloor u \rfloor} \right|^2 du \right] \\ &\leqslant K^2 T \int_0^t e_s ds. \end{aligned}$$

On the other hand, for $1 \leqslant i \leqslant d$ we can directly appeal to Doob's inequality (Lemma 5.4) together with the Itô isometry, obtaining

$$\begin{aligned} d_t^i &\leqslant 4E\left[\left| \int_0^t \left(V_i\left(X_{\lfloor u \rfloor}\right) - V_i\left(\overline{X}_{\lfloor u \rfloor}\right)\right) dB_u^i \right|^2 \right] \\ &= 4E\left[\int_0^t \left| V_i\left(X_{\lfloor u \rfloor}\right) - V_i\left(\overline{X}_{\lfloor u \rfloor}\right) \right|^2 du \right] \\ &\leqslant 4K^2 \int_0^t e_s ds. \end{aligned}$$

Summarizing, we get

$$d_t^i = E\left[\sup_{0 \leqslant s \leqslant t} \left| \int_0^s \left(V_i(X_{\lfloor u \rfloor}) - V_i(\overline{X}_{\lfloor u \rfloor})\right) dB_u^i \right|^2 \right] \leqslant \begin{cases} K^2 T \int_0^t e_s ds, & i = 0, \\ 4K^2 \int_0^t e_s ds, & 1 \leqslant i \leqslant d. \end{cases} \tag{5.5}$$

For c_t^0, a similar calculation as before (using Jensen's inequality once more) leads us to

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

Noting that for any vectors $|x_1 + \cdots + x_n|^2 \leqslant n\left(|x_1|^2 + \cdots + |x_n|^2\right)$, we further get

$$\begin{aligned}
c_t^0 &\leqslant K^2(d+1)\int_0^t \left(E\left[\left|\int_{\lfloor u\rfloor}^u V(X_s)ds\right|^2\right] + \sum_{i=1}^d E\left[\left|\int_{\lfloor u\rfloor}^u V_i(X_s)dB_s^i\right|^2\right]\right) du \\
&\leqslant C\int_0^t \left(E\left[(u-\lfloor u\rfloor)\int_{\lfloor u\rfloor}^u |V(X_s)|^2\, ds\right] + \sum_{i=1}^d E\left[\int_{\lfloor u\rfloor}^u |V_i(X_s)|^2\, ds\right]\right) du \\
&\leqslant CK^2\int_0^t \left((u-\lfloor u\rfloor)\int_{\lfloor u\rfloor}^u E\left[(1+|X_s|^2)\right] ds + d\int_{\lfloor u\rfloor}^u E\left[(1+|X_s|^2)\right] ds\right) du \\
&\leqslant C(1+|x|^2)\int_0^t \left((u-\lfloor u\rfloor)^2 + d(u-\lfloor u\rfloor)\right) du \\
&\leqslant TC(1+|x|^2)(|\mathcal{D}|+d)\,|\mathcal{D}|\,.
\end{aligned}$$

A similar computation for c_t^i gives the common bound

$$c_t^i \leqslant \begin{cases} TC(1+|x|^2)\,|\mathcal{D}|\,, & i=0, \\ C(1+|x|^2)\,|\mathcal{D}|\,, & 1\leqslant i\leqslant d. \end{cases} \tag{5.6}$$

Combining the bounds (5.5) and (5.6), we obtain

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

and Grönwall's inequality (Lemma 5.3) implies

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

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

Remark 5.6. Note that the constant C in Theorem 5.5 depends exponentially on T—as becomes apparent in the proof.

In fact, the proof of Theorem 5.5 can be turned into a proof of existence and uniqueness of solutions of SDEs. We give a sketch of the argument. First of all, we shall assume without proof that there is a constant $C>0$ (independent of the grid $\mathcal{D}$) such that

$$E\left[\sup_{0\leqslant t\leqslant T}\left|\overline{X}_t^{\mathcal{D}}\right|^2\right] \leqslant C\left(1+|x|^2\right).$$

This estimate can be proved by localization and a careful estimation along the iterative construction of $\overline{X}^{\mathcal{D}}$.

i) Uniqueness: Theorem 5.5 directly implies uniqueness of solution (within the class of square-integrable processes).

ii) An estimate: A simple adaptation of the above proof shows that for two grids $\mathcal{D}$ and $\mathcal{D}'$ we have

$$E\left[\sup_{0\leqslant t\leqslant T}\left|\overline{X}_t^{\mathcal{D}} - \overline{X}_t^{\mathcal{D}'}\right|^2\right] \leqslant C\max\left(|\mathcal{D}|\,,|\mathcal{D}'|\right).$$

Indeed, we may assume that $\mathcal{D}\subset\mathcal{D}'$, i.e., $\mathcal{D}'$ is finer than $\mathcal{D}$. Then we repeat the argument, replacing the true solution by $\overline{X}^{\mathcal{D}'}$.

iii) Existence of a limit: It follows that there is a unique limit process

$$\widetilde{X} := \lim_{|D|\to 0}\overline{X}^{\mathcal{D}}$$

in the above sense.
iv) The limit solves the SDE: Using similar estimations as in the proof, for any $\overline{X} = \overline{X}^{\mathcal{D}}$ we see that

$$E\left[\sup_{0\leqslant t\leqslant T}\left|\int_0^t \left(V_i(\overline{X}_{\lfloor s\rfloor}) - V_i(\widetilde{X}_s)\right) dB_s^i\right|^2\right] \leqslant C\,|\mathcal{D}|\,, \quad i = 1,\dots,d,$$

$$E\left[\sup_{0\leqslant t\leqslant T}\left|\int_0^t \left(V(\overline{X}_{\lfloor s\rfloor}) - V(\widetilde{X}_s)\right) ds\right|^2\right] \leqslant C\,|\mathcal{D}|\,.$$

Hence, we obtain

$$\begin{aligned}
&E\left[\sup_{0\leqslant t\leqslant T}\left|\left(\int_0^t V(\overline{X}_{\lfloor s\rfloor})ds + \sum_{i=1}^d\int_0^t V_i(\overline{X}_{\lfloor s\rfloor})dB_s^i\right) - \left(\int_0^t V(\widetilde{X}_s)ds + \sum_{i=1}^d\int_0^t V_i(\widetilde{X}_s)dB_s^i\right)\right|^2\right] \\
&\leqslant C\left\{E\left[\sup_{0\leqslant t\leqslant T}\left|\int_0^t \left(V(\overline{X}_{\lfloor s\rfloor}) - V(\widetilde{X}_s)\right) ds\right|^2\right] + \sum_{i=1}^d E\left[\sup_{0\leqslant t\leqslant T}\left|\int_0^t \left(V_i(\overline{X}_{\lfloor s\rfloor}) - V_i(\widetilde{X}_s)\right) dB_s^i\right|^2\right]\right\} \\
&\leqslant C\,|\mathcal{D}|\,.
\end{aligned}$$

Finally, note that for any grid $\mathcal{D}$,

$$\begin{aligned}
&E\left[\sup_{0\leqslant t\leqslant T}\left|\widetilde{X}_t - \left(x + \int_0^t V(\widetilde{X}_s)ds + \sum_{i=1}^d\int_0^t V_i(\widetilde{X}_s)dB_s^i\right)\right|\right] \\
&\qquad\leqslant C\Bigg\{E\left[\sup_{0\leqslant t\leqslant T}\left|\widetilde{X}_t - \overline{X}_t\right|\right] + \\
&+ E\left[\sup_{0\leqslant t\leqslant T}\left|\left(\int_0^t V(\overline{X}_{\lfloor s\rfloor})ds + \sum_{i=1}^d\int_0^t V_i(\overline{X}_{\lfloor s\rfloor})dB_s^i\right) - \left(\int_0^t V(\widetilde{X}_s)ds + \sum_{i=1}^d\int_0^t V_i(\widetilde{X}_s)dB_s^i\right)\right|^2\right]\Bigg\} \\
&\qquad\leqslant C\,|\mathcal{D}|\,,
\end{aligned}$$

implying that

$$\widetilde{X}_t = x + \int_0^t V(\widetilde{X}_s)ds + \sum_{i=1}^d\int_0^t V_i(\widetilde{X}_s)dB_s^i.$$

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 5.2. Moreover, weak approximation of SDEs can be used as a numerical method for solving linear parabolic PDEs. Indeed,

$$u(t,x) \coloneqq E\left[\,f(X_T)\middle|\,X_t = x\right] \tag{5.7}$$

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

$$\begin{cases} \dfrac{\partial}{\partial t}u(t,x) + Lu(t,x) = 0, \\ \qquad\qquad u(T,x) = f(x), \end{cases} \tag{5.8}$$

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

$$\left|E\left[f\left(\overline{X}_T^{\mathcal{D}}\right)\right] - E\left[f\left(X_T\right)\right]\right| \leqslant \|\nabla f\|_\infty E\left[\left|X_T - \overline{X}_T^{\mathcal{D}}\right|\right] \leqslant C\|\nabla f\|_\infty \sqrt{|\mathcal{D}|}, \tag{5.9}$$

by Theorem 5.5. 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 [56]. For our discussion we assume that the grids $\mathcal{D}$ are homogeneous, i.e., $\Delta t_i = h := T/N$ for every i. Of course, the results hold (with minor corrections) also in the general case, with h being replaced by $|\mathcal{D}|$.

Theorem 5.7. *Assume that the vector fields $V, V_1, \dots, V_d$ 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*

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

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

$$e(T,h,f) = h\int_0^T E[\psi_1(s,X_s)]ds + h^2 e_2(T,f) + \mathcal{O}(h^3), \tag{5.10}$$

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 $a_j^i(x) = \sum_{k=1}^d V_k^i(x)V_k^j(x)$, $1 \leqslant i,j \leqslant n$.

Remark 5.8. 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 (5.7).

Lemma 5.9. *Under the assumptions of Theorem 5.7, the solution u of (5.8) 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 5.10. *Again under the assumptions of Theorem 5.7, we have*

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

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

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

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

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

where $\mathcal{O}(\Delta x^k)$ means that the term is $\mathcal{O}(\Delta x^{i_1}\cdots\Delta x^{i_k})$ 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, $E\left[\mathcal{O}\left(h\Delta\overline{X}^3\right)\right] = \mathcal{O}(h^3)$, since $\Delta B_1^i \sim \mathcal{N}(0,h)$, and, similarly, $E\left[\mathcal{O}\left(\Delta\overline{X}^5\right)\right] = \mathcal{O}(h^3)$. 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 (5.8)), 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 5.7. By the final condition of (5.8), we may express the error of the Euler scheme (for approximating $u(0,x)$) as

$$\begin{aligned} E\left[f(\overline{X}_N)\right] - u(0,x) &= E\left[u(T,\overline{X}_N) - u(0,x)\right] \\ &= \sum_{i=0}^{N-1} E\left[u(t_{i+1},\overline{X}_{i+1}) - u(t_i,\overline{X}_i)\right] \\ &= \sum_{i=1}^{N-1}\left\{h^2 E\left[\psi_1(t_i,\overline{X}_i)\right] + \mathcal{O}(h^3)\right\}. \end{aligned} \tag{5.11}$$

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 5.9, ψ_1 has polynomial growth. Moreover, we know that $\overline{X}$ has bounded moments – see the proof of Theorem 5.5. This implies the bound

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

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

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

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 (5.11). Consider

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

For the first term, note that $\left|E\left[\psi_1(t_i, \overline{X}_i)\right] - E[\psi_1(t_i, X_{t_i})]\right| = \mathcal{O}(h)$ for each $0 \leqslant i \leqslant N-1$, 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 $\mathcal{O}(h)$. 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

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

for C^1-functions g. Therefore, also the second term can be bounded by $\mathcal{O}(h)$. Inserting these results into (5.11), we indeed obtain

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

The higher order expansion can now be obtained by continuing the Taylor expansion of Lemma 5.10 to higher order terms. □

Remark 5.11. The error expansion of Theorem 5.7 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

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

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 $A - R(h) = \mathcal{O}(h^m)$.

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 5.7 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 5.12. In the derivation of Theorem 5.7, 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 $(\Delta B_j^i : 1 \leqslant i \leqslant d,\ 1 \leqslant j \leqslant N)$ 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 5.13, it is not true for Theorem 5.14, which does depend on particular properties of the normal distribution.

Notice that our proof of Theorem 5.7 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 5.7, 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 [31, Theorem 14.5.1] is based on this type of arguments:

Theorem 5.13. *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 5.13. 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 5.14. *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 (5.10) holds with the same definition of the function ψ_1.*

Comparing Theorem 5.14 and Theorem 5.13, 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 5.15. 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 5.1. We clearly see the order of convergence $1/2$.

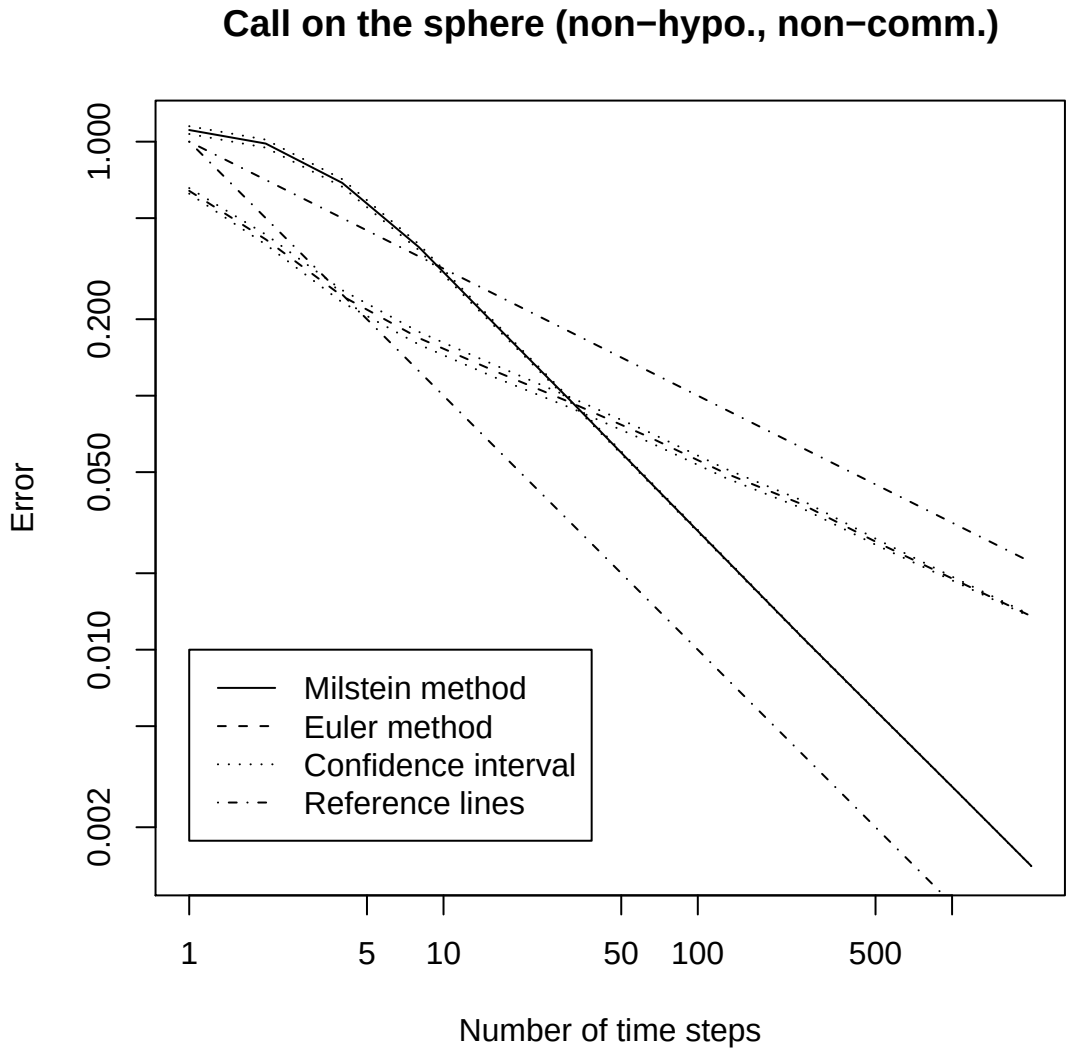


Figure 5.1: Weak error for Example 5.15

In many situations, we can expect the Euler scheme to converge with weak order one, even if the assumptions of neither Theorem 5.13 nor Theorem 5.14 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 5.15 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-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 5.16. 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 $(\Delta Z_n^i)_{i=1,\dots,d;n=1,\dots,N}$. 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:

$$\begin{aligned}\text{(5.12)}\quad \text{Error} &= \left|E[f(X_T)] - \frac{1}{M}\sum_{i=1}^{M} f\left(\overline{X}_N^{(i)}\right)\right| \\ &\leqslant \left|E[f(X_T)] - E\left[f\left(\overline{X}_N\right)\right]\right| + \left|E\left[f\left(\overline{X}_N\right)\right] - \frac{1}{M}\sum_{i=1}^{M} f\left(\overline{X}_N^{(i)}\right)\right|.\end{aligned}$$

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

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

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

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

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

$$\min \left\{ MN \;\middle|\; C_I M^{-q} + C_D N^{-p} \leqslant \epsilon \right\}. \tag{5.13}$$

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 5.17. *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 5.1

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 5.1: Complexity of the Euler Monte Carlo method

we have collected the order of complexity for certain 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.

5.2 Advanced methods

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 $\mathcal{O}(\epsilon^{-3})$, as we have seen in Proposition 5.17. Giles [21], [22] 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 (5.1) 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 $f\big(\overline{X}^{(h)}\big)$, our estimate for $E[f(X_T)]$ will be based on a combination of samples from the random variables $\overline{X}^{(h_1)}, \dots, \overline{X}^{(h_L)}$ 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 $f\big(\overline{X}^{(h_L)}\big)$ – 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 $f\big(\overline{X}^{(h_L)}\big)$ and $f\big(\overline{X}^{(h_{L-1})}\big)$ is high, but this choice does not seem to qualify since we do not know the expectation of $f\big(\overline{X}^{(h_{L-1})}\big)$. Notice, however, that it is much cheaper to sample $f\big(\overline{X}^{(h_{L-1})}\big)$ as opposed to $f\big(\overline{X}^{(h_L)}\big)$, 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 $f\big(\overline{X}^{(h_{L-1})}\big)$ is much cheaper. Therefore, the first step for multilevel Monte Carlo is:

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

Methods of this form are also known as "quasi control variates". Now we iterate the idea, by using variance reduction based on $f\big(\overline{X}^{(h_{L-2})}\big)$ in order to compute $E\big[f\big(\overline{X}^{(h_{L-1})}\big)\big]$, which we need for the computation of $E\big[f\big(\overline{X}^{(h_L)}\big)\big]$. 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 5.18. *Assume that there are constants $\alpha \geqslant 1/2$, $C_1, C_2, \beta > 0$ such that $E[f(X_T) - P_l] \leqslant C_1 h_l^\alpha$ and $\operatorname{var}[I_l] \leqslant C_2 h_l^\beta M_l^{-1}$. 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*

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

and the computational work C is bounded by

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

Corollary 5.19. *Assume that the Euler method has weak order 1 and strong order 1/2 for the problem at hand. Choose $L = \frac{\log(\epsilon^{-1})}{\log N} + \mathcal{O}(1)$ in ϵ and choose M_l proportional to $\epsilon^{-2}(L+1)h_l$. Then the multilevel estimator has computational error $\mathcal{O}(\epsilon)$, while the computational cost is $\mathcal{O}(\epsilon^{-2}(\log \epsilon)^2)$.*

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

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

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

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

Moreover, choosing

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

we have

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

Thus, the means square error satisfies

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

and we are only left to compute the computational cost $\mathcal{C}$.

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

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

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

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

provided that $N > 1$. This implies that

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

Chapter 6

Numerical methods for PDEs

6.1 The Black–Scholes PDE

We start by revisiting the partial differential equations (PDEs) associated with option pricing in the Black–Scholes model, before proceeding to their numerical solution. In what follows, we mainly follow Seydel [55]. For the theoretical part see also the notes of Kohn on PDEs for finance.

Using 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 price model, the Black-Scholes model

$$dS_t = rS_t dt + \sigma S_t dB_t, \quad S_0 = s \in \mathbb{R}_+.$$

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

$$\frac{\partial}{\partial t}u(t,x) + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2}{\partial x^2}u(t,x) + rx\frac{\partial}{\partial x}u(t,x) - ru(t,x) = 0, \tag{6.1}$$

with terminal value $u(T,x) = f(x)$. In the sequel, 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:

$$\frac{\partial}{\partial t}\widetilde{u}(t,x) + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2}{\partial x^2}\widetilde{u}(t,x) + rx\frac{\partial}{\partial x}\widetilde{u}(t,x) - r\widetilde{u}(t,x) \leqslant 0, \tag{6.2a}$$

$$\widetilde{u}(t,x) \geqslant (K-x)_+, \tag{6.2b}$$

$$\widetilde{u}(T,x) = (K-x)_+, \tag{6.2c}$$

where we have equality in (6.2a) whenever there is a strict inequality in (6.2b). It can be shown that problem (6.2) is a *free boundary problem*, i.e. there exists an (unknown) value $x_0 = x_0(t)$ such that $\widetilde{u}$ solves the PDE (6.2a) with equality (i.e. the classical Black-Scholes PDE) on the domain $]x_0, \infty[$ and $\widetilde{u}(t,x) = (K-x)_+$ whenever $x \leqslant x_0$. 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 (6.1) or (6.2) 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

$$v(\tau, y) := \frac{1}{K} \exp\left(\frac{1}{2}(q-1)y + \left(\frac{1}{4}(q-1)^2 + q\right)\tau\right) u(t, x), \tag{6.3}$$

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:

$$\frac{\partial}{\partial \tau} v(\tau, y) = \frac{\partial^2}{\partial y^2} v(\tau, y), \quad v(0, y) = \left(e^{\frac{1}{2}(q-1)y} - e^{\frac{1}{2}(q+1)y}\right)_+. \tag{6.4}$$

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

$$v(\tau, y) = \exp\left(\frac{1}{2}(q-1)y + \frac{1}{4}(q-1)^2\tau\right) \text{ for } y \to -\infty, \quad v(\tau, y) = 0 \text{ for } y \to \infty. \tag{6.5}$$

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

$$\left(\frac{\partial}{\partial \tau}\widetilde{v}(\tau, y) - \frac{\partial^2}{\partial y^2}\widetilde{v}(\tau, y)\right)(\widetilde{v}(\tau, y) - g(\tau, y)) = 0, \quad \frac{\partial}{\partial \tau}\widetilde{v}(\tau, y) - \frac{\partial^2}{\partial y^2}\widetilde{v}(\tau, y) \geqslant 0, \tag{6.6a}$$

$$\widetilde{v}(\tau, y) \geqslant g(\tau, y), \quad \widetilde{v}(0, y) = g(0, y), \tag{6.6b}$$

$$\widetilde{v}(\tau, y) = g(\tau, y) \text{ for } y \to -\infty, \quad \widetilde{v}(\tau, y) = 0 \text{ for } y \to \infty. \tag{6.6c}$$

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

Exercise 6.1. Show that indeed (6.4) is equivalent to the Black–Scholes PDE (6.1) under the given transformations.

6.2 The finite difference method

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

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

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 derivatives by finite difference quotients along a grid, thereby transforming a PDE into a difference equation.

There are several different choices of difference quotients that will be used in the sequel. They are motivated by the Taylor expansion, and we have:

$$\text{forward difference:} \qquad f'(x) = \frac{f(x+h) - f(x)}{h} + \mathcal{O}(h)$$

$$\text{backward difference:} \qquad f'(x) = \frac{f(x) - f(x-h)}{h} + \mathcal{O}(h)$$

while combining these two yields

$$\text{central difference:} \qquad f'(x) = \frac{f(x+h) - f(x-h)}{2h} + \mathcal{O}(h^2).$$

Moreover, for the second derivatives we will use the central difference:

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \mathcal{O}(h^2).$$

In addition, 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 $v_{i,j}$, $0 \leqslant i \leqslant N$, $0 \leqslant j \leqslant M$, of the values $u_{i,j} := u(t_i, x_j)$.

Remark 6.1. 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 6.2. In a multi-dimensional setting, the same construction applies. Note, however, that a grid in $\mathbb{R}^n$ with the same mesh Δx has M^n nodes. Therefore, we need to compute NM^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 constitutes a (possibly big) advantage or not.

Explicit finite differences

The next step is to replace all derivatives in (6.4) by difference quotients. In the explicit finite difference scheme we use forward differences to discretize the time derivative, that is we approximate

$$\frac{\partial}{\partial t} u(t_i, x_j) = \frac{u_{i+1,j} - u_{i,j}}{\Delta t} + \mathcal{O}(\Delta t), \tag{6.7}$$

while for the space derivative we choose the approximation

$$\frac{\partial^2}{\partial x^2} u(t_i, x_j) = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta x^2} + \mathcal{O}(\Delta x^2). \tag{6.8}$$

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 (6.4). 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 (6.5) 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 and using the notation $\lambda := \Delta t/(\Delta x)^2$, we obtain:

$$v_{0,j} = \left(e^{\frac{1}{2}(q-1)x_j} - e^{\frac{1}{2}(q+1)x_j}\right)_+, \quad j = 0, \dots, M, \tag{6.9a}$$

$$v_{i+1,j} = v_{i,j} + \lambda(v_{i,j+1} - 2v_{i,j} + v_{i,j-1}), \quad i = 0, \dots, N-1,\ j = 1, \dots, M-1, \tag{6.9b}$$

$$v_{i+1,0} = \exp\left(\frac{1}{2}(q-1)a + \frac{1}{4}(q-1)^2 t_{i+1}\right), \quad v_{i+1,M} = 0, \quad i = 0, \dots, N-1. \tag{6.9c}$$

In order to do numerical analysis, it is useful to obtain a more 'compact' notation for the scheme (6.9). To this end, let us ignore the boundary conditions (6.9c) and just implement the iterations step (6.9b). 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

$$v^{(i+1)} = Av^{(i)}, \quad A := \begin{pmatrix} 1-2\lambda & \lambda & 0 & \cdots & 0 \\ \lambda & 1-2\lambda & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \lambda \\ 0 & \cdots & 0 & \lambda & 1-2\lambda \end{pmatrix}. \tag{6.10}$$

Therefore, the bulk of the computations in the explicit finite difference scheme (6.9) 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 (6.9c).)

Example 6.3. Consider the following problem (Seydel [55], 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 approximation $v_{1000,2} = 0.00435$. Next, we choose a coarser 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 (6.9) *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

$$\sigma_k = 1 - 2\lambda\left(1 - \cos\left(\frac{k\pi}{M}\right)\right), \quad k = 1, \dots, M-1. \tag{6.11}$$

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

Theorem 6.4. *If we choose the time mesh Δt and the space mesh Δx in such a way that $\Delta t \leqslant \frac{1}{2}\Delta x^2$, then the explicit finite difference method is stable and converges with error $\mathcal{O}(\Delta t) + \mathcal{O}(\Delta x^2)$, provided that the given boundary conditions are exact.*

Remark 6.5. 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 5.1 for the Euler method or even the complexity estimate in Theorem 5.18 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.

Implicit finite differences

We can also use the backward difference to disretize the time derivative in (6.4), i.e.

$$\frac{\partial}{\partial t}u(t_i, x_j) = \frac{u_{i,j} - u_{i-1,j}}{\Delta t} + \mathcal{O}(\Delta t), \tag{6.12}$$

while retaining (6.8) for the space derivative. This leads to the following approximation:

$$v_{i-1,j} = v_{i,j} + \frac{\Delta t}{\Delta x^2}\big(- v_{i,j+1} + 2v_{i,j} - v_{i,j-1}\big). \tag{6.13}$$

This scheme is not explicit anymore, because only the value $v_{i-1,j}$ is known from the previous step of the iteration, while the right hand side contains three unknown values that should be computed. This is an example of the *implicit* scheme, where a system of equations has to be solved at each time step. The iteration step of this scheme can be represented in matrix notation as follows:

$$Av^{(i)} = v^{(i-1)}, \quad A \coloneqq \begin{pmatrix} 1+2\lambda & -\lambda & 0 & \cdots & 0 \\ -\lambda & 1+2\lambda & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\lambda \\ 0 & \cdots & 0 & -\lambda & 1+2\lambda \end{pmatrix}. \tag{6.14}$$

Moreover, using similar methods as for the explicit finite difference method, one can prove the following result (see Seydel [55, Example 4.2]).

Theorem 6.6. *The implicit finite difference method is unconditionally stable for $\Delta t > 0$. Moreover, it converges with error $\mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^2)$, provided that the boundary conditions are exact.*

Crank-Nicolson

The right hand side of (6.7) can 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,

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

for $\frac{\partial}{\partial t}u(t_{i+1}, x_j)$. Both of them agree. If we use the central difference quotient (6.8) 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*

$$\frac{v_{i+1,j} - v_{i,j}}{\Delta t} = \frac{1}{2\Delta x^2}\left(v_{i,j+1} - 2v_{i,j} + v_{i,j-1} + v_{i+1,j+1} - 2v_{i+1,j} + v_{i+1,j-1}\right). \tag{6.15}$$

This is again an explicit scheme, since 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 three different values $v_{i+1,j-1}$, $v_{i+1,j}$ and $v_{i+1,j+1}$. As a consequence, (6.15) should be understood as a linear equation for $(v_{i+1,j})_{j=1}^{M-1}$ given all the values of $v_{i,j}$. The iteration step of the Crank-Nicolson scheme can be represented in matrix notation as follows:

$$Av^{(i+1)} = Bv^{(i)}, \tag{6.16}$$

where

$$A \coloneqq \begin{pmatrix} 1+\lambda & -\frac{\lambda}{2} & 0 & \cdots & 0 \\ -\frac{\lambda}{2} & 1+\lambda & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\frac{\lambda}{2} \\ 0 & \cdots & 0 & -\frac{\lambda}{2} & 1+\lambda \end{pmatrix}, \; B \coloneqq \begin{pmatrix} 1-\lambda & \frac{\lambda}{2} & 0 & \cdots & 0 \\ \frac{\lambda}{2} & 1-\lambda & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \frac{\lambda}{2} \\ 0 & \cdots & 0 & \frac{\lambda}{2} & 1-\lambda \end{pmatrix} \tag{6.17}$$

Finally, using similar methods as for the explicit finite difference method, one can prove the following theorem (see Seydel [55, Satz 4.4]).

Theorem 6.7. *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 and the approximation error of the solution of the Crank-Nicolson method is $\mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^2)$.*

6.3 The finite element method

The finite element method (FEM) is an elegant numerical method for solving linear PDEs based on the variational formulation of the PDE. While PDEs in finance, in particular in option pricing, are usually of the time-dependent, parabolic variety, FEM is more transparent for time-independent, elliptic problems. Hence, we shall first consider elliptic problems, before coming back to parabolic problems at the end.

6.3.1 A step-by-step guide to the finite element method

We start by a hands-on introduction to the finite element method based on a simple, one-dimensional example. Consider

$$\big(-a(x)u'(x)\big)' + r(x)u(x) = f(x), \quad x \in (0,1), \tag{6.18a}$$

$$u(x) = 0, \quad x \in \{0,1\}\,. \tag{6.18b}$$

For the equation to be well-defined we assume that $a > 0$ and $r \geqslant 0$. Regularity assumptions will be discussed later.

Step 1: A variational formulation

As already indicated, the finite element method is based on the *variational formulation* of the PDE. To this end, we first have to define the natural space on which to consider the PDE. Formally, (6.18) requires two derivatives to exist. Nonetheless, we shall see that we can make sense of the equation provided that only one (weak) derivative exists. We refer to Appendix D for a short introduction to weak derivatives and Sobolev spaces.

Hence, consider

$$V := H_0^1((0,1)) = \left\{ v : [0,1] \to \mathbb{R} \;\middle|\; \|v\|_V^2 := \int_0^1 \Big(v(x)^2 + \big[v'(x)\big]^2\Big)\,dx < \infty, \quad v(0) = v(1) = 0 \right\}. \tag{6.19}$$

Elements of V are called *test functions*.

In order to get the variational formulation, we now formally multiply the PDE (6.18a) by a test function $v \in V$ and integrate from 0 to 1. Note that test functions are, in fact, regular enough that integration by parts can be justified. Starting on the right hand side, we simply obtain

$$L(v) := \int_0^1 f(x)v(x)dx, \tag{6.20}$$

which we consider as a *linear* functional $L : V \to \mathbb{R}$. For the right hand side, we have

$$\int_0^1 \Big[\big(-a(x)u'(x)\big)' + r(x)u(x)\Big]\,v(x)dx$$

$$= \big[-a(x)u'(x)v(x)\big]_0^1 + \int_0^1 \big[a(x)u'(x)v'(x) + r(x)u(x)v(x)\big]\,dx.$$

As $v \in V$, it vanishes at the boundary of the domain, i.e., at $\{0,1\}$. Hence, the first term above vanishes and the left hand side of (6.18a) corresponds to

$$A(u,v) \coloneqq \int_0^1 \left[a(x)u'(x)v'(x) + r(x)u(x)v(x)\right] dx, \tag{6.21}$$

which we will understand as a bi-linear form $A : V \times V \to \mathbb{R}$. Hence, the *variational formulation* of the elliptic problem (6.18) is

$$\forall v \in V : \ A(u,v) = L(v) \tag{6.22}$$

for a solution $u \in V$.

Remark 6.8. If $u \in C^2([0,1])$ is a (classical) solution of (6.18), then it is also a solution of the variational formulation (6.22). Indeed, first of all such a u is, in fact, an element of V, and the above formal calculation holds. Conversely, suppose that $u \in V \cap C^2([0,1])$ solves the variational formulation (6.22). Then we can undo the integration by parts and obtain

$$\forall v \in V : \ \int_0^1 \left[\left(-a(x)u'(x)\right)' + r(x)u(x) - f(x)\right] v(x)dx = 0.$$

This immediately implies that the integrand vanishes for any $x \in (0,1)$ and, hence, u is a classical solution to (6.18). On the other hand, depending on the regularity of the coefficients a, r, f, there may exist (unique) solutions u of (6.22) which are not twice differentiable and, hence, cannot be classical solutions of (6.18). This means, the solution concept discussed here is *weaker* than the concept of classical solutions.

Remark 6.9. The key point of the above construction is the fact that the partial integration against test functions allows us to define solutions (or candidates for solutions) which are only once (weakly) differentiable. Of course, it would also be possible to define a variational formulation based on the bi-linear form

$$(u,v) \mapsto \int_0^1 \left[\left(-au'\right)' + ru\right] vdx.$$

However, that form requires more smoothness. Moreover, (and more importantly, perhaps) A is *symmetric*, whereas the above form is not.

Remark 6.10. Here, we have a form $A : V \times V \to \mathbb{R}$, i.e., the space of test functions is the same as the space of solutions. This is not always possible, and there are also approaches to allow using different spaces for test functions and solutions.

Step 2: Projection onto a finite-dimensional subspace

In the second step, we choose a subspace $V_h \subset V$ of V with finite dimension. We choose the symbol V_h to allude that this subspace corresponds to the discretization of the problem — and, indeed, V_h usually depends on a real parameter "h" in some form. V_h is the actual collection of "finite elements". In the current context, we choose a space of piece-wise linear functions based on a grid $0 = x_0 < x_1 < \cdots < x_{N+1} = 1$. More precisely, let

$$V_h \coloneqq \left\{ v \in C([0,1]) \,\middle|\, \forall i \in \{0,\dots,N\} : \ v|_{[x_i,x_{i+1}]} \text{ is affine, } v(0) = v(1) = 0 \right\}. \tag{6.23}$$

It is easy to check that $V_h \subset V$ and that $\dim V_h = N$. Other polynomial spline spaces may also be useful.

We now consider the project problem of finding $u_h \in V_h$ such that

$$\forall v \in V_h : \ A(u_h,v) = L(v). \tag{6.24}$$

Clearly, there is no reason why the true solution $u \in V$ of the variational formulation of our PDE (provided it exists, is unique etc.) is contained in V_h. Hence, $u_h \neq u$ in general, and the projection step induces an error in the finite element method.

Step 3: Basis of V_h

Note that (6.24) is linear in both u_h and in v. Hence, we can replace the infinite system of equations (6.24) by a finite system of equations by running only through a set of vs forming a basis of V_h. While there are many possible bases of V_h, it is advantageous to choose a basis which is well-adapted to the problem at hand. For variational problems as considered here, a good choice is one such that the supports of the basis functions have minimal intersection—this will become transparent in the final step below. Hence, we an obvious choice of basis for V_h consists of "tent functions", i.e., functions $\phi_i \in V_h$, $i = 1, \dots, N$, defined by the constraint that

$$\phi_i(x_j) = \delta_{i,j}, \quad j = 0, \dots, N+1, \quad , i = 1, \dots, N.$$

This basis leads to an extremely simple representation of general elements of V_h as linear combination of basis vectors. Indeed, for $v \in V_h$ we have

$$v(x) = \sum_{i=1}^{N} v(x_i)\phi_i(x), \quad x \in [0,1].$$

Hence, we are now left with the finite, linear system of equation to find $u_h \in V_h$ such that

$$\forall i \in \{1, \dots, N\} : \ A(u_h, \phi_i) = L(\phi_i). \tag{6.25}$$

Step 4: Solving the linear system

The last step is now almost trivial: we need to solve the system of equations (6.25). We will describe the procedure in some detail, mostly to introduce some names.

First, we represent the approximate solution u_h in terms of the basis,

$$u_h = \sum_{i=1}^{N} \xi_i \phi_i.$$

Defining the *stiffness matrix* $\overline{A} \in \mathbb{R}^{N \times N}$ and the *load vector* $\overline{L} \in \mathbb{R}^N$ by

$$\overline{A}_{i,j} := A(\phi_i, \phi_j), \quad \overline{L}_i := L(\phi_i),$$

for $i, j \in \{1, \dots, N\}$, we end up with the system

$$\overline{A}\xi = \overline{L}, \quad \xi = (\xi_1, \dots, \xi_N)^\top.$$

Note that the special structure of the basis functions induces sparsity in $\overline{A}$. Indeed, if ϕ_i and ϕ_j have disjoint support, then it is easy to see that $\overline{A}_{i,j} = 0$.

Exercise 6.2. Set up the matrix $\overline{A}$ and the vector $\overline{L}$ in the model problem (6.18). Convince yourself of the sparsity of $\overline{A}$.

6.3.2 Existence and uniqueness of solutions to the variational problem

Here we consider the general problem (6.22). I.e., we assume that we are given a Hilbert space V and a bi-linear form $A : V \times V \to \mathbb{R}$ and a linear form $L : V \to \mathbb{R}$. The existence and uniqueness theorem for such problems is a classical (and surprisingly simple) theorem of functional analysis, which we are going to present below. But first we need an auxiliary result, providing a helpful reformulation of the variational formulation of a PDE in terms of a minimization problem.

Lemma 6.11. *Assume that the bi-linear form A is* symmetric *and* positive semi-definite*, i.e.,*

$$\forall v, w \in V : \ A(v, w) = A(w, v),$$
$$\forall v \in V : \ A(v, v) \geqslant 0.$$

Then $u \in V$ satisfies (6.22) if and only if u is a minimizer of the functional $F(v) := \frac{1}{2}A(v, v) - L(v)$, $v \in V$.

Proof. "⇒:" If u solves the variational problem and $v \in V$ is arbitrary, we need to show that $F(u) \leqslant F(v)$. Let $w := \frac{v-u}{\epsilon} \in V$ for some $\epsilon \in \mathbb{R}$. We have that

$$F(v) = F(u+\epsilon w) = \underbrace{\left(\frac{1}{2}A(u,u) - L(u)\right)}_{=F(u)} + \epsilon \underbrace{\left(A(u,w) - L(w)\right)}_{=0} + \frac{1}{2}\epsilon^2 \underbrace{A(w,w)}_{\geqslant 0} \geqslant F(u).$$

"⇐:" Suppose that u is a minimizer of F and take any $w \in V$. Consider $g(\epsilon) := F(u+\epsilon w)$, $g : \mathbb{R} \to \mathbb{R}$. As g is a smooth function taking its minimum at $\epsilon = 0$, we have

$$0 = g'(0) = A(u,w) - L(w).$$

As w was arbitrary, u solves the variational problem. □

Theorem 6.12 (Lax-Milgram Lemma)**.** *Let V be a Hilbert space with norm $\|\cdot\|_V$. Assume that the bi-linear functional $A : V \times V \to \mathbb{R}$ and the linear functional $L : V \to \mathbb{R}$ satisfy:*

(i) A is symmetric*;*

(ii) A is elliptic*, i.e., $\exists \alpha > 0 \ \forall v \in V : \ A(v,v) \geqslant \alpha \|v\|_V^2$;*

(iii) A is continuous*, i.e., $\exists C > 0 \ \forall v, w \in V : \ |A(v,w)| \leqslant C \|v\|_V \|w\|_V$;*

(iv) L is continuous*, i.e, $\exists \Lambda > 0 \ \forall v \in V : \ |L(v)| \leqslant \Lambda \|v\|_V$.*

Then there is a unique $u \in V$ such that

$$\forall v \in V : \ A(u,v) = L(v).$$

Moreover, we have the a-priori estimate $\|u\|_V \leqslant \frac{\Lambda}{\alpha}$.

Proof. By Lemma 6.11 we have to show that there is a unique $u \in V$ with $F(u) = \inf_{v \in V} F(v)$.

Define the *energy norm* $\|v\| := \sqrt{A(v,v)}$, $v \in V$. We note that $\|\cdot\|$ is equivalent to $\|\cdot\|_V$ as

$$\alpha \|v\|_V^2 \leqslant A(v,v) = \|v\|^2 \leqslant C \|v\|_V^2 .$$

Step 1: The minimum of F is finite.
Let $\beta := \inf_{v \in V} F(v)$. Note that

$$F(v) = \frac{1}{2}\|v\|^2 - L(v) \geqslant \frac{1}{2}\|v\|^2 - \Lambda \|v\| .$$

Hence, $\beta \geqslant \inf_{x \in \mathbb{R}} \frac{1}{2}x^2 - \Lambda x = -\frac{1}{2}\Lambda^2 > -\infty$.
Step 2: Minimizing sequences are Cauchy.
Let $(v_n)_{n \in \mathbb{N}}$ denote a minimizing sequence, i.e., $F(v_n) \to \beta$ as $n \to \infty$. Then

$$\begin{aligned}
\|v_n - v_m\|^2 &= \|v_n\|^2 + \|v_m\|^2 - 2A(v_n, v_m) \\
&= 2\|v_n\|^2 + 2\|v_m\|^2 - \|v_n + v_m\|^2 \\
&= 2\|v_n\|^2 - 4L(v_n) + 2\|v_m\|^2 - 4L(v_m) - 4\left(\left\|\frac{v_n+v_m}{2}\right\|^2 - 2L\left(\frac{v_n+v_m}{2}\right)\right) \\
&= 4F(v_n) + 4F(v_m) - 8\underbrace{F\left(\frac{v_n+v_m}{2}\right)}_{\geqslant \beta} \\
&\leqslant 4F(v_n) + 4F(v_m) - 8\beta \xrightarrow{m,n\to\infty} 0.
\end{aligned}$$

Step 3: Existence of a minimizer and stability estimate.

Fix any minimizing sequence $(v_n)_{n\in\mathbb{N}}$ and define $u := \lim_{n\to\infty} v_n$. Note that

$$\begin{aligned}|F(v_n) - F(u)| &= \left|\frac{1}{2}\left(\|v_n\|^2 - \|u\|^2\right) - L(v_n - u)\right| \\ &\leqslant \left|\frac{1}{2}A(v_n - u, v_n + u)\right| + |L(v_n - u)| \\ &\leqslant \frac{1}{2}C\,\|v_n - u\|_V\,\|v_n + u\|_V + \Lambda\,\|v_n - u\|_V\,.\end{aligned}$$

By equivalence of norms, we have $\|v_n - u\|_V \to 0$ while $\|v_n + u\|_V$ is bounded. Hence, $F(u) = \lim_{n\to\infty} F(v_n) = \beta$.

As a minimizer, u solves the variational problem. Hence, we have

$$\alpha\,\|u\|_V^2 \leqslant A(u,u) = L(u) \leqslant \Lambda\,\|u\|_V\,,$$

or $\|u\|_V \leqslant \frac{\Lambda}{\alpha}$.

Step 4: Uniqueness.

Assume that there are two solutions $u_1, u_2 \in V$. Then for any $v \in V$ we have

$$A(u_1, v) = L(v) = A(u_2, v).$$

This implies that $A(u_1 - u_2, v) = 0$ for any $v \in V$. In particular, for $v = u_1 - u_2$ we get $A(u_1 - u_2, u_1 - u_2) = 0$, and by ellipticity $\|u_1 - u_2\|_V^2 = 0$. Hence, $u_1 = u_2$. □

Example 6.13. We show that (6.20) together with (6.21) satisfies the conditions of Theorem 6.12 provided that

$$\|a\|_\infty < \infty, \quad \|r\|_\infty < \infty, \quad \|f\|_\infty < \infty, \quad \inf_{x\in[0,1]} a(x) > 0.$$

(The latter condition corresponds to ellipticity and is the important condition. All norms are considered on $(0,1)$.) Recall that $\|v\|_V^2 = \|v\|_{L^2}^2 + \|v'\|_{L^2}^2$ in this example. Before verifying the assumptions, note that $\|v\|_V^2 \leqslant 2\,\|v'\|_{L^2}^2$, since (by Jensen's inequality)

$$\begin{aligned}\int_0^1 v(x)^2 dx = \int_0^1 \left(\underbrace{v(0)}_{=0} + \int_0^x v'(y)dy\right)^2 dx &\leqslant \int_0^1 x \int_0^x \big(v'(y)\big)^2\,dydx \\ &= \int_0^1 \big(v'(y)\big)^2 \int_y^1 x\,dx\,dy \leqslant \frac{1}{2}\int_0^1 \big(v'(y)\big)^2\,dy,\end{aligned}$$

or, more compactly,

$$\|v\|_{L^2}^2 \leqslant \frac{1}{2}\,\|v'\|_{L^2}^2\,. \tag{6.26}$$

Note that (6.26) is a special case of the *Pincaré–Friedrichs inequality*.

We now verify the four assumptions of the theorem. Symmetry of A is clear. For ellipticity, note that

$$A(v,v) = \int_0^1 a(x)\big(v'(x)\big)^2\,dx + \int_0^1 r(x)v(x)^2 dx \geqslant (\inf a)\,\|v'\|_{L^2}^2 \geqslant \frac{1}{2}(\inf a)\,\|v\|_V^2\,.$$

For continuity of A note that

$$\begin{aligned}|A(v,w)| &\leqslant \int_0^1 |a(x)|\,|v'(x)w'(x)|\,dx + \int_0^1 |r(x)|\,|u(x)v(x)| \\ &\leqslant \|a\|_\infty\,\|v'w'\|_{L^1} + \|r\|_\infty\,\|vw\|_{L^1} \\ &\leqslant \|a\|_\infty\,\|v'\|_{L^2}\,\|w'\|_{L^2} + \|r\|_\infty\,\|v\|_{L^2}\,\|w\|_{L^2} \\ &\leqslant (\|a\|_\infty + \|r\|_\infty)\,\|v\|_V\,\|w\|_V\,.\end{aligned}$$

Continuity of L follows in the same way.

Note that we therefore obtain existence and uniqueness of solutions without any smoothness assumptions on the coefficients.

6.3.3 Error estimates

In general, as made precise by the following theorem, the error in the finite element method depends on

- the smoothness of the true solution;
- the approximation quality of $V_h \subset V$.

Theorem 6.14. *Assume the conditions of Theorem 6.12. Moreover, let $\pi : V \to V_h$ be a projection. Then for the solution u and the corresponding finite element approximation u_h of the variational problem based on $V_h \subset V$, we have*

$$\|u - u_h\|_V = \|u - u_h\|_{H^1} \leqslant \sqrt{\frac{C}{\alpha}} \|u - \pi u\|_V$$

Proof. By definition, we have

$$\begin{aligned} \forall v \in V : \ A(u, v) &= L(v), \\ \forall v \in V_h : \ A(u_h, v) &= L(v). \end{aligned}$$

Hence, the error $e := u - u_h$ is orthogonal to V_h in terms of the energy norm, as

$$\forall v \in V_h : \ A(u - u_h, v) = 0.$$

We have

$$\begin{aligned} A(e, e) &= A(e, u - \pi u) + A(e, \underbrace{\pi u - u_h}_{\in V_h}) \\ &= A(e, u - \pi u) \\ &\leqslant \|e\| \, \|u - \pi u\| . \end{aligned}$$

Dividing by $\|e\|$, squaring and using the continuity of A we get

$$\|e\|_V^2 \leqslant \frac{1}{\alpha} \|e\|^2 \leqslant \frac{1}{\alpha} A(u - \pi u, u - \pi u) \leqslant \frac{C}{\alpha} \|u - \pi u\|_V^2 . \qquad \square$$

Error estimates therefore require two pieces of information: how smooth is the true solution, and how well can such a function be approximated by functions in V_h. We come back to our one-dimensional example. We first discuss the latter question.

Lemma 6.15. *For V and V_h defined in (6.19) and (6.23), respectively, let $\pi : V \to V_h$ be defined by $\pi v(x) := \sum_{i=1}^N v(x_i)\phi_i(x)$, based on the basis ϕ_i of tent-functions. Let $v \in V$ be such that v'' exists in a weak sense and is square-integrable. Then we have*

$$\|v - \pi v\|_V \leqslant Ch, \quad h := \max_{i=0,\dots,N} |x_{i+1} - x_i| .$$

Proof. We first compute $\|(v - \pi v)'\|_{L^2}^2$. For $x \in (x_i, x_{i+1})$ we have

$$(\pi v)'(x) = \frac{v(x_{i+1}) - v(x_i)}{x_{i+1} - x_i} = v'(\xi)$$

for some $\xi \in (x_i, x_{i+1})$. (Note that ξ does not depend on x.) Hence,

$$v'(x) - (\pi v)'(x) = v'(x) - v'(\xi) = \int_\xi^x v''(s) ds.$$

Using Jensen's inequality, we further obtain

$$\begin{aligned}\int_{x_i}^{x_{i+1}} \left|v'(x) - (\pi v)'(x)\right|^2 dx &= \int_{x_i}^{x_{i+1}} \left|\int_\xi^x v''(s)ds\right|^2 dx \\ &\leqslant \int_{x_i}^{x_{i+1}} |x-\xi| \int_\xi^x (v''(s))^2 ds dx \\ &\leqslant h(x_{i+1}-x_i) \int_{x_i}^{x_{i+1}} (v''(s))^2 ds \\ &\leqslant h^2 \int_{x_i}^{x_{i+1}} (v''(s))^2 ds.\end{aligned}$$

Adding up the integrals, we obtain

$$\left\|v' - (\pi v)'\right\|_{L^2}^2 = \sum_{i=0}^{N} \int_{x_i}^{x_{i+1}} \left|v'(x) - (\pi v)'(x)\right|^2 dx \leqslant h^2 \int_0^1 (v''(s))^2 ds.$$

Similarly, for $x \in (x_i, x_{i+1})$ we get

$$v(x) - \pi v(x) = \left[v(x_i) - \pi v(x_i)\right] + \int_{x_i}^x \left[v'(s) - (\pi v)'(s)\right] ds = \int_{x_i}^x \left[v'(s) - (\pi v)'(s)\right] ds.$$

Hence,

$$\begin{aligned}\int_{x_i}^{x_{i+1}} |v(x) - \pi v(x)|^2 dx &= \int_{x_i}^{x_{i+1}} \left|\int_{x_i}^x \left[v'(s) - (\pi v)'(s)\right] ds\right|^2 dx \\ &\leqslant h \int_{x_i}^{x_{i+1}} \int_{x_i}^x \left[v'(s) - (\pi v)'(s)\right]^2 ds dx \\ &\leqslant h \underbrace{\int_{x_i}^{x_{i+1}} dx}_{\leqslant h} \underbrace{\int_{x_i}^{x_{i+1}} \left[v'(s) - (\pi v)'(s)\right]^2 ds}_{\leqslant h^2 \int_{x_i}^{x_{i+1}} (v''(s))^2 ds} \\ &\leqslant h^4 \int_{x_i}^{x_{i+1}} (v''(s))^2 ds.\end{aligned}$$

Summing up, we get

$$\|v - \pi v\|_{L^2}^2 \leqslant h^4 \int_0^1 (v''(s))^2 ds. \qquad \square$$

Regarding regularity of the solution, note that smoothness of the solution to (6.18) follows from the "elliptic regularity theorem". More precisely, if (in addition to the conditions in Example 6.13) a is continuously differentiable and f is square integrable, then we indeed have $u'' \in L^2$. In fact, we can bound $\|u\|_{H^2} \leqslant C \|f\|_{L^2}$.

Theorem 6.14 together with Lemma 6.15 give us the error estimate

$$\|u - u_h\|_{H^1} \leqslant \text{const} \left\|u''\right\|_{L^2} h \tag{6.27}$$

provided that u is twice weakly differentiable. By definition of the $\|\cdot\|_{H^1}$ this means that both $u - u_h$ and $u' - u_h'$ have an error of order h in the L^2 sense. However, in many situations we might only be interested in estimates for the L^2-error of the solution u, u_h rather then its derivative. The following result shows that the error estimate can be improved in this case.

Lemma 6.16 (Aubin–Nitsche duality)**.** *Under the conditions of Lemma 6.15 we have* $\|u - u_h\|_{L^2} \leqslant C \|u\|_{H^2} h^2$.

Proof. The constant C can change from occurrence to occurrence in this proof. With $e := u - u_h \in V = H_0^1((0,1))$, we consider the *dual problem* of finding $\phi \in V$ s.t.

$$\forall v \in V: \; A(\phi, v) = \langle e\,, v\rangle_{L^2}\,,$$

i.e., the original problem with f replaced by e. From our previous results, we know that there is a unique solution $\phi \in V$, and since $e \in H_0^1 \subset L^2$, we also have elliptic regularity.

Recall that Céa's lemma implies that e is orthogonal to V_h w.r.t. the energy norm. Hence, using the particular test function $v = e \in V$, we have

$$\|e\|_{L^2}^2 = \langle e\,, e\rangle_{L^2} = A(\phi, e) = A(e, \phi - \pi\phi) \leqslant \|e\|\,\|\phi - \pi\phi\|\,.$$

Lemma 6.15, we have

$$\|\phi - \pi\phi\| \leqslant Ch\,\|\phi\|_{H^2} \leqslant Ch\,\|e\|_{L^2}\,.$$

Inserting into the above inequality, dividing by $\|e\|_{L^2}$, and using equivalence of $\|\cdot\|$ and $\|\cdot\|_{H^1}$ as well as (6.27), we obtain

$$\|e\|_{L^2} \leqslant Ch\,\|e\| \leqslant Ch\,\|e\|_{H^1} \leqslant Ch^2\,\|u\|_{H^2}\,. \qquad \square$$

6.3.4 FEM for parabolic equations

For simplicity, we only consider the one-dimensional heat equation with vanishing Dirichlet boundary condition, i.e,

$$\partial_t u(t,x) = \Delta u(t,x) + f(t,x), \quad 0 < x < 1, \quad 0 < t \leqslant T, \tag{6.28a}$$

$$u(0,x) = u_0(x), \quad 0 \leqslant x \leqslant 1, \quad u(t,0) = u(t,1) = 0, \quad 0 \leqslant t \leqslant T, \tag{6.28b}$$

where $\Delta = \partial_x^2$ only acts on the space variable x.

We will not develop a proper solution theory for (6.28). Rather we assume that there is a unique solution in the following sense: Consider $t \mapsto u(t,\cdot)$. Then the classical derivative $t \mapsto \partial_t u(t,\cdot)$ exists and $u \in L^2\left([0,T]; H_0^1(G)\right)$ with $\partial_t u \in L^2\left([0,T]; H^{-1}(G)\right)$ with, in our case, $G = (0,1)$, and satisfies:

$$\forall 0 < t \leqslant T,\, \forall v \in H_0^1(G): \; \langle \partial_t u(t,\cdot)\,, v\rangle_{H^{-1}(G);H_0^1(G)} + A(u, v; t) = L(v; t), \quad u(0,\cdot) = u_0. \tag{6.29}$$

Remark 6.17. For the heat equation (6.28), the bi-linear form $A(w,v) = \langle w'\,, v'\rangle_{L^2}$ is independent of t, and we chose $L \equiv 0$. In addition, it is natural to assume that $\partial_t u(t,\cdot) \in H^{-1}(G)$, since, by the equation, $\partial_t u(t,\cdot) = \Delta u(t,\cdot)$, i.e., taking one time derivative corresponds to taking two space derivatives. In this context, $H^{-1}(G)$ is, in fact, the dual sppace of $H_0^1(G)$, and $\langle\cdot\,,\cdot\rangle_{H^{-1}(G);H_0^1(G)}$ denotes the duality bracket between $H^{-1}(G)$ and $H_0^1(G)$. For our purposes, we will act as if this bracket coincides with the $L^2(G)$-inner product $\langle\cdot\,,\cdot\rangle_{L^2}$, although $H^{-1}(G) \supsetneq L^2(G)$. For instance, as $H_0^1((0,1)) \subset C([0,1])$, the point-evaluation functional, i.e., the delta distribution $\delta_x \in H^{-1}((0,1))\backslash L^2((0,1))$, $0 < x < 1$.

An important property of the heat equation is *energy dissipation*.

Theorem 6.18 (Energy dissipation)**.** *There is a constant K s.t. the solution u of the heat equation (6.28) satisfies*

$$\|u(t,\cdot)\|_{L^2}^2 \leqslant e^{-Kt}\,\|u_0\|_{L^2}^2 + \frac{1}{K}\int_0^t e^{-K(t-s)}\,\|f(s,\cdot)\|_{L^2}^2\,ds.$$

Proof for the case $f \equiv 0$. We test $\partial_t u(t,\cdot)$ against the test function $u(t,\cdot)$ using that $A(u(t,\cdot), u(t,\cdot)) = \|\partial_x u(t,\cdot)\|_{L^2}^2 \geqslant 2\,\|u(t,\cdot)\|_{L^2}^2$ and obtain

$$\frac{1}{2}\frac{d}{dt}\,\|u(t,\cdot)\|_{L^2}^2 + 2\,\|u(t,\cdot)\|_{L^2}^2 = \langle \partial_t u(t,\cdot)\,, u(t,\cdot)\rangle_{L^2} + 2\,\|u(t,\cdot)\|_{L^2}^2 \leqslant 0.$$

This implies that

$$\frac{d}{dt}\left(e^{4t}\,\|u(t,\cdot)\|_{L^2}^2\right) \leqslant 0. \qquad \square$$

We now add time-discretization in the form of the Euler scheme. For simplicity, we only consider *uniform grids*, i.e., space grids $x_j := jh$, $0 \leqslant j \leqslant N+1$, which are used to define V_h as in (6.23), $h := 1/(N+1)$. We also consider a uniform time grid $t^m := m\Delta t$, $0 \leqslant m \leqslant M$, with $\Delta t := T/M$. For arbitrary functions $w = w(t,x)$ of (t,x) we introduce the notation

$$w^m := w(t^m, \cdot), \quad m = 0, \dots, M. \tag{6.30}$$

We obtain the *forward* and *backward* Euler methods by replacing the time-derivative in (6.29) by a forward and backward difference quotient, respectively, and restricting the test functions to $V_h \subset V$.

Definition 6.19. Consider a sequence $u_h^m \in V_h$, $m = 0, \dots, M$, such that u_h^0 is obtained as L^2-projection of u_0 on V_h, i.e., $\forall v \in V_h : \langle u_h^0 - u_0 \, , v \rangle_{L^2} = 0$. $(u_h^m)_{m=0}^M$ is the *forward Euler approximation* of the heat equation (6.28) iff

$$\forall v \in V_h : \left\langle \frac{u_h^{m+1} - u_h^m}{\Delta t} \, , v \right\rangle_{L^2} + A(u_h^m, v) = \langle f^m \, , v \rangle_{L^2} \, , \quad m = 0, \dots, M-1.$$

$(u_h^m)_{m=0}^M$ is the *backward Euler approximation* of the heat equation (6.28) iff

$$\forall v \in V_h : \left\langle \frac{u_h^{m+1} - u_h^m}{\Delta t} \, , v \right\rangle_{L^2} + A(u_h^{m+1}, v) = \langle f^{m+1} \, , v \rangle_{L^2} \, , \quad m = 0, \dots, M-1.$$

These equations can be reformulated as

$$\forall v \in V_h : \langle u_h^{m+1} \, , v \rangle_{L^2} = \langle u_h^m \, , v \rangle_{L^2} - \Delta t A(u_h^m, v) + \Delta t \langle f^m \, , v \rangle_{L^2} \, , \quad m = 0, \dots, M-1, \tag{6.31}$$

for the forward Euler method and

$$\forall v \in V_h : \langle u_h^{m+1} \, , v \rangle_{L^2} + \Delta t A(u_h^{m+1}, v) = \langle u_h^m \, , v \rangle_{L^2} + \Delta t \langle f^{m+1} \, , v \rangle_{L^2} \, , \quad m = 0, \dots, M-1, \tag{6.32}$$

for the backward Euler method. We also define the general θ-scheme in a similar way as for finite difference methods. Let us introduce another short-hand notation: for a function $w = w(t,x)$ in (t,x) and $0 \leqslant \theta \leqslant 1$, define

$$w^{m+\theta} := \theta w(t^{m+1}, \cdot) + (1-\theta) w(t^m, \cdot), \quad m = 0, \dots, M-1, \tag{6.33}$$

such that w^m and w^{m+1} in the sense of $w^{m+\theta}$ for $\theta = 0$ or $\theta = 1$, respectively, coincide with the original definitions in (6.30).

Definition 6.20. Consider a sequence $u_h^m \in V_h$, $m = 0, \dots, M$, such that u_h^0 is obtained as L^2-projection of u_0 on V_h, i.e., $\forall v \in V_h : \langle u_h^0 - u_0 \, , v \rangle_{L^2} = 0$, and fix $0 \leqslant \theta \leqslant 1$. Then $(u_h^m)_{m=0}^M$ is the solution of the *θ-scheme* iff

$$\left\langle \frac{u_h^{m+1} - u_h^m}{\Delta t} \, , v \right\rangle_{L^2} + A(u_h^{m+\theta}, v) = \langle f^{m+\theta} \, , v \rangle_{L^2} \, , \quad m = 0, \dots, M-1.$$

We note that the θ-scheme with $\theta = 0$ coincides with the forward Euler scheme and the θ-scheme with $\theta = 1$ coincides with the backward Euler scheme.

We next study *stability* of the Euler schemes. As in the case of finite difference scheme, it turns out that the backward Euler scheme is stable regardless of the choices of h and Δt, whereas the forward Euler scheme is only stable provided that Δt is sufficiently small for given h. More generally, we have

Theorem 6.21. *For $1/2 \leqslant \theta \leqslant 1$ the θ–scheme is unconditionally stable, i.e., its solution $(u_h^m)_{m=0}^M$ satisfies*

$$\max_{1 \leqslant m \leqslant M} \|u_h^m\|_{L^2}^2 \leqslant \left\|u_h^0\right\|_{L^2}^2 + \Delta t \sum_{m=0}^{M-1} \left\|f^{m+\theta}\right\|_{L^2}^2 .$$

For $0 \leqslant \theta < 1/2$ the θ-scheme is stable provided that for some $0 < \epsilon < 1$ we have $\Delta t \leqslant \frac{h^2}{6(1-2\theta)}(1-\epsilon)$. In this case, its solution $(u_h^m)_{m=0}^M$ satisfies

$$\max_{1\leqslant m\leqslant M} \|u_h^m\|_{L^2}^2 \leqslant \left\|u_h^0\right\|_{L^2}^2 + c_\epsilon \Delta t \sum_{m=0}^{M-1} \left\|f^{m+\theta}\right\|_{L^2}^2, \quad c_\epsilon := \frac{1}{4\epsilon^2} + \Delta t(1-2\theta)(1+1/\epsilon).$$

Proof of the case $1/2 \leqslant \theta \leqslant 1$. We test the scheme given in Definition 6.20 with $v := u_h^{m+\theta} \in V_h$, and obtain

$$\left\langle \frac{u_h^{m+1} - u_h^m}{\Delta t}, u_h^{m+\theta} \right\rangle_{L^2} + \left\|\nabla u_h^{m+\theta}\right\|_{L^2}^2 = \left\langle f^{m+\theta}, u_h^{m+\theta} \right\rangle_{L^2}. \tag{6.34}$$

Note that

$$u_h^{m+\theta} = \theta u_h^{m+1} + (1-\theta)u^m = (\theta - 1/2)(u_h^{m+1} - u_h^m) + \frac{u_h^{m+1} + u_h^m}{2} = \Delta t(\theta - 1/2)\frac{u_h^{m+1} - u_h^m}{\Delta t} + \frac{u_h^{m+1} + u_h^m}{2}.$$

Plugging this into the left-most term in (6.34), we obtain

$$\Delta t(\theta - 1/2)\left\|\frac{u_h^{m+1} - u_h^m}{\Delta t}\right\|_{L^2}^2 + \frac{1}{2\Delta t}\left(\left\|u_h^{m+1}\right\|_{L^2}^2 - \|u_h^m\|_{L^2}^2\right) + \left\|\nabla u_h^{m+\theta}\right\|_{L^2}^2 = \left\langle f^{m+\theta}, u_h^{m+\theta}\right\rangle_{L^2}.$$

As $\theta \geqslant 1/2$, the first term above is non-negative and can therefore be omitted for a lower bound

$$\frac{\left\|u_h^{m+1}\right\|_{L^2}^2 - \|u_h^m\|_{L^2}^2}{2\Delta t} + \left\|\nabla u_h^{m+\theta}\right\|_{L^2}^2 \leqslant \left\|f^{m+\theta}\right\|_{L^2} \left\|u_h^{m+\theta}\right\|_{L^2}.$$

The Poincaré–Friedrichs inequality (6.26) as well as the trivial bound $ab \leqslant (a^2 + b^2)/2$ imply that

$$\frac{\left\|u_h^{m+1}\right\|_{L^2}^2 - \|u_h^m\|_{L^2}^2}{2\Delta t} + 2\left\|u_h^{m+\theta}\right\|_{L^2}^2 \leqslant \frac{1}{2}\left\|f^{m+\theta}\right\|_{L^2}^2 + \frac{1}{2}\left\|u_h^{m+\theta}\right\|_{L^2}^2,$$

which, in turn, gives

$$\frac{\left\|u_h^{m+1}\right\|_{L^2}^2 - \|u_h^m\|_{L^2}^2}{2\Delta t} \leqslant \frac{1}{2}\left\|f^{m+\theta}\right\|_{L^2}^2. \tag{6.35}$$

□

We next come to the actual convergence result. We will restrict ourselves to the backward Euler scheme and the error in the L^2-sense, i.e., we formulate the analogue of Lemma 6.16.

Theorem 6.22. *Consider the solution u of the heat equation (6.28) and its backward Euler approximation $(u_h^m)_{m=0}^M$. In addition to our standing assumptions, we assume that for all $t \in [0,T]$ we have*

$$\sup_{t\in[0,T]} \left\|\partial_t^2 u(t,\cdot)\right\|_{L^2} < \infty, \quad \sup_{t\in[0,T]} \|\partial_t u(t,\cdot)\|_{H^2} < \infty.$$

Then there is a constant $C > 0$ such that

$$\max_{m=0,\dots,M} \|u^m - u_h^m\|_{L^2} \leqslant C\left(\Delta t + h^2\right).$$

Proof. Let $\mathcal{P}_h : V \to V_h$ denote the orthogonal projection w.r.t. the energy norm $\|v\| = \sqrt{A(v,v)}$, and denote $e_h^m := u^m - u_h^m \in V$. We consider the error decomposition

$$e_h^m = \eta^m + \xi^m, \quad \eta^m := u^m - \mathcal{P}_h u^m \in V, \quad \xi^m := \mathcal{P}_h u^m - u_h^m \in V_h. \tag{6.36}$$

By definition, $\forall v \in V_h : A(u^m - \mathcal{P}_h u^m, v) = 0$, i.e., *Galerkin orthogonality* holds for $\mathcal{P}_h u^m$. Recall that Galerkin orthogonality was the basis of the whole error analysis for the elliptic case. Hence, we

obtain the error results of Lemma 6.15 and Lemma 6.16 for u replaced by u^m and u_h replaced by $\mathcal{P}_h u^m$. Hence, we have

$$\|\eta^m\|_{L^2} \leqslant \text{const}\, \|u^m\|_{H^2}\, h^2, \quad m = 0, \dots, M. \tag{6.37}$$

(Regarding $m = 0$, we refer to the proof of Lemma 6.15.) A similar calculation shows that the same estimate holds for the difference quotient, i.e.,

$$\left\| \frac{\eta^{m+1} - \eta^m}{\Delta t} \right\|_{L^2} \leqslant \text{const} \left\| \frac{u^{m+1} - u^m}{\Delta t} \right\|_{H^2} h^2, \quad m = 0, \dots, M-1. \tag{6.38}$$

We next come to the estimation of ξ^m, starting with $m = 0$. As e_h^0 is L^2-orthogonal to V_h, we have

$$\left\langle \xi^0 \,, v \right\rangle_{L^2} = \left\langle e_h^0 - \eta^0 \,, v \right\rangle_{L^2} = - \left\langle \eta^0 \,, v \right\rangle_{L^2}, \quad v \in V_h.$$

Choosing $v = \xi^0 \in V_h$ and applying Cauchy–Schwarz, we obtain

$$\left\| \xi^0 \right\|_{L^2} \leqslant \left\| \eta^0 \right\|_{L^2} \leqslant \text{const}\, \|u_0\|_{H^2}\, h^2. \tag{6.39}$$

Moving on to $1 \leqslant m \leqslant M$, an elementary computation gives that

$$\forall v \in V_h : \left\langle \frac{\xi^{m+1} - \xi^m}{\Delta t}, v \right\rangle_{L^2} + A(\xi^{m+1}, v) = \left\langle \frac{u^{m+1} - u^m}{\Delta t} - \partial_t u^{m+1} - \frac{\eta^{m+1} - \eta^m}{\Delta t}, v \right\rangle_{L^2},$$

i.e., ξ^m is itself the solution of a backward Euler discretization of a heat equation with right hand side given by

$$\tilde{f}^{m+1} := \frac{u^{m+1} - u^m}{\Delta t} - \partial_t u^{m+1} - \frac{\eta^{m+1} - \eta^m}{\Delta t}.$$

By Theorem 6.21, we have

$$\max_{1 \leqslant m \leqslant M} \|\xi^m\|_{L^2}^2 \leqslant \left\|\xi^0\right\|_{L^2}^2 + \Delta t \sum_{m=0}^{M-1} \left\| \tilde{f}^{m+1} \right\|_{L^2}^2, \tag{6.40}$$

and we are left to estimate

$$\left\| \tilde{f}^{m+1} \right\|_{L^2} \leqslant \left\| \frac{u^{m+1} - u^m}{\Delta t} - \partial_t u^{m+1} \right\|_{L^2} + \left\| \frac{\eta^{m+1} - \eta^m}{\Delta t} \right\|_{L^2} =: I + II.$$

Regarding the estimation of I, Taylor's formula implies that

$$\frac{u^{m+1}(x) - u^m(x)}{\Delta t} - \partial_t u^{m+1}(x) = -\frac{1}{\Delta t} \int_{t^m}^{t^{m+1}} (t - t^m) \partial_t^2 u(t, x) dx,$$

further giving us

$$I^2 \leqslant \frac{1}{\Delta t} \int_0^1 \int_{t^m}^{t^{m+1}} \underbrace{(t - t^m)^2}_{\leqslant \Delta t^2} \left| \partial_t^2 u(t, x) \right|^2 dt\, dx \leqslant \Delta t \int_{t^m}^{t^{m+1}} \left\| \partial_t^2 u(t, \cdot) \right\|_{L^2}^2 dt.$$

Regarding the estimation of II, a similar calculation based on (6.38) shows that

$$\begin{aligned} II = \left\| \frac{\eta^{m+1} - \eta^m}{\Delta t} \right\|_{L^2} &\leqslant \text{const} \left\| \frac{u^{m+1} - u^m}{\Delta t} \right\|_{H^2} h^2 \\ &= \text{const} \left\| \frac{1}{\Delta t} \int_{t^m}^{t^{m+1}} \partial_t u(t, \cdot) dt \right\|_{H^2} h^2 \\ &\leqslant \text{const} \left(\int_{t^m}^{t^{m+1}} \|\partial_t u(t, \cdot)\|_{H^2}^2 \, dt \right)^{1/2} \frac{h^2}{\sqrt{\Delta t}}. \end{aligned}$$

Combining the estimates for I and II, we obtain

$$\left\|\widetilde{f}^{m+1}\right\|_{L^2}^2 \leqslant \text{const}\left(\Delta t\int_{t^m}^{t^{m+1}}\left\|\partial_t^2 u(t,\cdot)\right\|_{L^2}^2\,dt + \frac{h^4}{\Delta t}\int_{t^m}^{t^{m+1}}\|\partial_t u(t,\cdot)\|_{H^2}^2\,dt\right),$$

implying that

$$\begin{aligned}\Delta t\sum_{m=0}^{M-1}\left\|\widetilde{f}^{m+1}\right\|_{L^2}^2 &\leqslant \text{const}\left(\Delta t^2\int_0^T\left\|\partial_t^2 u(t,\cdot)\right\|_{L^2}^2\,dt + h^4\int_0^T\|\partial_t u(t,\cdot)\|_{H^2}^2\,dt\right)\\ &\leqslant \text{const}(\Delta t^2 + h^4).\end{aligned}$$

Plugging the estimate into (6.40) and combining it with (6.39), we obtain

$$\max_{m=0,\dots,M}\|\xi^m\|_{L^2} \leqslant \text{const}(\Delta t + h^2),$$

which finishes the proof. □

Remark 6.23. For the Crank–Nicolson scheme, i.e., the θ-scheme with $\theta = 1/2$, one can prove (under similar conditions) an L^2-error of order $\Delta t^2 + h^2$.

Chapter 7

Fourier methods for option pricing

In Chapters 2 and 5 we have studied probabilistic numerical methods for the valuation of derivatives, while in Chapters 3 and 6 we have discussed deterministic numerical schemes. This chapter continues the study of deterministic methods for option pricing by introducing Fourier transform methods.

Let us recall that we are interested in computing the expectation of $f(X)$, *i.e.* $\mathbb{E}[f(X)]$. Fourier and other transform methods for option pricing can be schematically represented as follows:

(7.1)
$$\mathbb{E}[f(X)] = \int f(x)\mathbb{P}_X(\mathrm{d}x) \longrightarrow \int T^{-1}\big(T(f)\big)(x)\mathbb{P}_X(\mathrm{d}x) \longrightarrow \mathbb{E}[f(X)] = \int T(f)(u)T(\mathbb{P}_X)(u)\mathrm{d}u,$$

where T is a suitable transformation and T^{-1} its inverse. The most popular choices for the transformation T are the Fourier and Laplace transforms (Carr and Madan [7], Raible [48]), the Hilbert transform (Feng and Linetsky [19]), and the cosine series expansion (Fang and Oosterlee [18]).

These methods should be used when the computation of the right hand integral in (7.1) is (much) simpler than the computation of the left hand one; in particular, when the functions $T(f)$ and $T(\mathbb{P}_X)$ are well-defined and known explicitly. Let us point out that there are many examples in mathematical finance where $\mathbb{P}_X$ is not known explicitly, while its transform $T(\mathbb{P}_X)$ is known; think, for example, of Lévy and affine processes where the probability density function is typically unknown while the characteristic function is provided by the Lévy–Khintchine formula, resp. the solution of the generalized Riccati ODEs. In the sequel, we will provide explicit conditions such that these quantities are well-defined and discuss when it is advantageous to use Fourier transform methods for option pricing.

7.1 The Fourier transform

We will first provide a short and self-contained introduction to the Fourier transform following Deitmar [12] and Rudin [50], before proceeding with applications in mathematical finance.

Let us denote by $L^1(\mathbb{R})$ the space of all functions $f : \mathbb{R} \to \mathbb{C}$ with finite L^1-norm, *i.e.* such that

$$\|f\|_{L^1} = \int_{\mathbb{R}} |f(x)|\mathrm{d}x < \infty,$$

while $L^1_{\mathrm{bc}}(\mathbb{R})$ denotes the space of functions in $L^1(\mathbb{R})$ which are continuous and bounded.

Definition 7.1. Let $f \in L^1(\mathbb{R})$, then define its *Fourier transform* by

(7.2)
$$\widehat{f}(u) = \int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}ux} f(x)\mathrm{d}x, \quad u \in \mathbb{R}.$$

We extend the definition to $u \in \mathbb{C}$ provided that the integral above exists.

Remark 7.2. The following simple estimate shows that the Fourier transform is bounded for every $f \in L^1(\mathbb{R})$:

$$|\widehat{f}(u)| \leqslant \int_{\mathbb{R}} |\mathrm{e}^{\mathrm{i}ux}||f(x)|\mathrm{d}x \leqslant \|f\|_{L^1} < \infty. \tag{7.3}$$

The next result shows that the Fourier transform of a function, apart form being bounded, is also continuous and vanishes at infinity. The latter is also known as the *Riemann–Lebesgue Lemma*. Let C_0 denote the space of continuous functions vanishing at infinity.

Theorem 7.3. *Let $f \in L^1(\mathbb{R})$, then $\widehat{f} \in C_0$.*

Proof. Let $u_n \to u$, then

$$\left|\widehat{f}(u_n) - \widehat{f}(u)\right| \leqslant \int_{\mathbb{R}} |\mathrm{e}^{\mathrm{i}u_n x} - \mathrm{e}^{\mathrm{i}ux}||f(x)|\mathrm{d}x \leqslant 2\|f\|_{L^1} < \infty,$$

and by the dominated convergence theorem we get that

$$\widehat{f}(u_n) \xrightarrow[n\to\infty]{} \widehat{f}(u),$$

hence $\widehat{f}$ is a continuous function. Moreover, using that $\mathrm{e}^{\mathrm{i}\pi} = -1$, we get that

$$\begin{aligned}\widehat{f}(u) &= \int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}ux} f(x)\mathrm{d}x = -\int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}ux+\mathrm{i}\pi} f(x)\mathrm{d}x = -\int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}u(x+\frac{\pi}{u})} f(x)\mathrm{d}x \\ &= -\int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}ux} f\left(x - \frac{\pi}{u}\right)\mathrm{d}x.\end{aligned}$$

Therefore, we have that

$$\widehat{f}(u) = \frac{1}{2}\int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}ux}\left(f(x) - f\left(x - \frac{\pi}{u}\right)\right)\mathrm{d}x$$

thus, writing $f_{\pi/u}(x) = f(x - \pi/u)$, we get that

$$\left|\widehat{f}(u)\right| \leqslant \frac{1}{2}\left\|f - f_{\pi/u}\right\|_{L^1} \xrightarrow[|u|\to\infty]{} 0,$$

since the mapping $u \mapsto f_{\pi/u}$ is uniformly continuous (*cf.* [50, Thm. 9.5]). □

Next, we shall derive several useful properties of the Fourier transform. More precisely, the Fourier transform converts multiplication by a character, *i.e.* by $\mathrm{e}^{\mathrm{i}ux}$, into translation and *vice versa*, while it converts convolutions into pointwise products. Moreover, the Fourier transform converts differentiation into multiplication by $-\mathrm{i}u$ and *vice versa*, a fact that is very usefull in the study of differential equations.

Theorem 7.4. *Let $f \in L^1(\mathbb{R})$.*

(i) If $g(x) = f(x)\mathrm{e}^{\mathrm{i}ax}$ for $a \in \mathbb{R}$, then $\widehat{g}(u) = \widehat{f}(u + a)$.

(ii) If $g(x) = f(x - a)$ for $a \in \mathbb{R}$, then $\widehat{g}(u) = \mathrm{e}^{\mathrm{i}ua}\widehat{f}(u)$.

(iii) If $g(x) = f(\frac{x}{\lambda})$ for $\lambda \in \mathbb{R}_+$, then $\widehat{g}(u) = \lambda\widehat{f}(\lambda u)$.

(iv) If $g(x) = \overline{f(-x)}$, then $\widehat{g}(u) = \overline{\widehat{f}(u)}$.

(v) If $g \in L^1(\mathbb{R})$ and $h = f * g$, then $\widehat{h}(u) = \widehat{f}(u)\widehat{g}(u)$.

(vi) *If* $g(x) = \mathrm{i}xf(x)$ *and* $g \in L^1(\mathbb{R})$, *then* $\widehat{f}$ *is continuously differentiable with*

$$\left(\widehat{f}\right)'(u) = \widehat{g}(u).$$

(vii) *Let* $f \in C^1(\mathbb{R})$ *and assume that* $f, f' \in L^1_{\mathrm{bc}}(\mathbb{R})$. *Then*

$$\widehat{f'}(u) = -\mathrm{i}u\widehat{f}(u)$$

and, in particular, $u\widehat{f}(u)$ *is bounded.*

(viii) *Let* $f \in C^2(\mathbb{R})$ *and assume that* $f, f', f'' \in L^1_{\mathrm{bc}}(\mathbb{R})$. *Then* $\widehat{f} \in L^1_{\mathrm{bc}}(\mathbb{R})$.

Proof. $(i), (ii), (iii)$ and (iv) follow directly from Definition 7.1.

(v) The operation $f * g$ is called *convolution* and is defined via

$$(f * g)(x) = \int_{\mathbb{R}} f(y)g(x-y)\mathrm{d}y.$$

Then, using Fubini's theorem we get that

$$\begin{aligned}
\|f * g\|_{L^1} &= \int_{\mathbb{R}} \Big| \int_{\mathbb{R}} f(y)g(x-y)\mathrm{d}y\Big|\mathrm{d}x \\
&\leqslant \int_{\mathbb{R}} \int_{\mathbb{R}} |f(y)g(x-y)|\mathrm{d}y\mathrm{d}x \\
&= \int_{\mathbb{R}} |f(y)|\mathrm{d}y \int_{\mathbb{R}} |g(x)|\mathrm{d}x = \|f\|_{L^1}\|g\|_{L^1} < \infty.
\end{aligned}$$

Moreover, using Fubini's theorem once again, as well as the translation invariance of the Lebesgue measure, we have that

$$\begin{aligned}
\widehat{(f * g)}(u) &= \int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}ux}(f * g)(x)\mathrm{d}x \\
&= \int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}ux} \int_{\mathbb{R}} f(y)g(x-y)\mathrm{d}y\mathrm{d}x \\
&= \int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}uy} f(y)\mathrm{d}y \int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}u(x-y)} g(x-y)\mathrm{d}x = \widehat{f}(u)\widehat{g}(u).
\end{aligned}$$

(vi) Observe that

$$\frac{\widehat{f}(v) - \widehat{f}(u)}{v-u} = \int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}ux} \frac{\mathrm{e}^{\mathrm{i}(v-u)x} - 1}{v-u} f(x)\mathrm{d}x. \tag{7.4}$$

Let $\varphi(x,u) = \frac{1}{u}(\mathrm{e}^{\mathrm{i}ux} - 1)$, then $|\varphi(x,u)| \leqslant |x|$ for all $u \neq 0$ and

$$\varphi(x,u) \to ix \quad \text{as} \quad u \to 0,$$

where the convergence is locally uniform in x. Therefore, using the dominated convergence theorem, as $v \to u$ we get from (7.4) that

$$\left(\widehat{f}\right)'(u) = \int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}ux}\mathrm{i}xf(x)\mathrm{d}x = \widehat{g}(u).$$

(vii) Since f is integrable, there exist sequences $A_n, B_n \to \infty$ such that $f(-A_n), f(B_n) \to 0$. Then, using integration by parts, we have that

$$\begin{aligned}
\widehat{f'}(u) &= \lim_{n\to\infty} \int_{-A_n}^{B_n} \mathrm{e}^{\mathrm{i}ux} f'(x)\mathrm{d}x \\
&= \lim_{n\to\infty} \left\{\mathrm{e}^{\mathrm{i}uB_n} f(B_n) - \mathrm{e}^{-\mathrm{i}uA_n} f(-A_n)\right\} - \lim_{n\to\infty} \int_{-A_n}^{B_n} \mathrm{i}u\mathrm{e}^{\mathrm{i}ux} f(x)\mathrm{d}x \\
&= -\mathrm{i}u\widehat{f}(u).
\end{aligned}$$

The boundedness of $u\widehat{f}(u)$ follows from the previous equality and Remark 7.2.

(*viii*) Since f, f', f'' are integrable and f is two times continuously differentiable, applying (*vii*) twice yields that $\widehat{f''}(u) = -u^2\widehat{f}(u)$. Moreover, from the latter result together with Remark 7.2 we get that $(1+u^2)\widehat{f}(u)$ is a bounded function. Hence, we arrive at the following:

$$\int_{\mathbb{R}} |\widehat{f}(u)|\mathrm{d}u \leqslant \text{const} \cdot \int_{\mathbb{R}} \frac{1}{1+u^2}\mathrm{d}u < \infty. \qquad \square$$

We have shown that certain operations on functions correspond nicely to operations on their Fourier transforms. This correspondence would be of great interest if there was a way to return from the transform to the function itself, in other words, if there was an *inversion formula.* The inversion formula obviously involves the inverse Fourier transform whose definition is provided below and shows that, apart from a constant and a sign change, the Fourier transform is inverse to itself.

Definition 7.5. Let $g \in L^1(\mathbb{R})$, then the *inverse Fourier transform* of g is defined as

$$\breve{g}(u) = \frac{1}{2\pi}\int_{\mathbb{R}} \mathrm{e}^{-\mathrm{i}ux} g(x)\mathrm{d}x, \quad u \in \mathbb{R}.$$

In order to prove the inversion formula, we need some auxiliary definitions and results. Define, for $\lambda \in \mathbb{R}_+$ and $x \in \mathbb{R}$

$$h_\lambda(x) = \frac{1}{2\pi}\int_{\mathbb{R}} \mathrm{e}^{-\lambda|u|}\mathrm{e}^{-\mathrm{i}ux}\mathrm{d}u, \tag{7.5}$$

and note that

$$0 < \mathrm{e}^{-\lambda|u|} \leqslant 1 \quad \text{and} \quad \mathrm{e}^{-\lambda|u|} \xrightarrow[\lambda\to 0]{} 1.$$

Lemma 7.6. *We have that*

$$h_\lambda(x) = \frac{\lambda}{\pi(x^2+\lambda^2)} \quad \textit{and} \quad \int_{\mathbb{R}} h_\lambda(x)\mathrm{d}x = 1. \tag{7.6}$$

Moreover, $h_\lambda(x) = \frac{1}{\lambda}h_1(\frac{x}{\lambda})$ *for every* $\lambda > 0$.

Exercise 7.1. Show that $\int_{\mathbb{R}} \frac{1}{x^2+\lambda^2}\mathrm{d}x = \frac{\pi}{\lambda}$ and prove Lemma 7.6.

Lemma 7.7. *Let* $f \in L^1_{\mathrm{bc}}(\mathbb{R})$*, then for every* $\lambda > 0$

$$(f * h_\lambda)(x) = \frac{1}{2\pi}\int_{\mathbb{R}} \mathrm{e}^{-\lambda|u|}\mathrm{e}^{-\mathrm{i}ux}\widehat{f}(u)\mathrm{d}u.$$

Proof. Using Fubini's theorem, we can compute directly

$$\begin{aligned}(f * h_\lambda)(x) &= \int_{\mathbb{R}} f(y)h_\lambda(x-y)\mathrm{d}y \\ &= \int_{\mathbb{R}} f(y)\frac{1}{2\pi}\int_{\mathbb{R}} \mathrm{e}^{-\lambda|u|}\mathrm{e}^{-\mathrm{i}u(x-y)}\mathrm{d}u\mathrm{d}y \\ &= \frac{1}{2\pi}\int_{\mathbb{R}} \mathrm{e}^{-\lambda|u|}\mathrm{e}^{-\mathrm{i}ux}\left(\int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}uy}f(y)\mathrm{d}y\right)\mathrm{d}u,\end{aligned}$$

which yields the result. $\square$

Lemma 7.8. *Let* $f \in L^1_{\mathrm{bc}}(\mathbb{R})$*, then for every* $x \in \mathbb{R}$ *we have*

$$\lim_{\lambda\to 0}(f * h_\lambda)(x) = f(x).$$

Proof. Since $\int_{\mathbb{R}} h_\lambda(x)\mathrm{d}x = 1$, using a change of variables we get that

$$\begin{aligned}(f * h_\lambda)(x) - f(x) &= \int_{\mathbb{R}} f(y)h_\lambda(x-y)\mathrm{d}y - \int_{\mathbb{R}} f(x)h_\lambda(y)\mathrm{d}y \\ &= \int_{\mathbb{R}} \left(f(x-y) - f(x)\right) h_\lambda(y)\mathrm{d}y \\ &= \int_{\mathbb{R}} \left(f(x-y) - f(x)\right) \frac{1}{\lambda} h_1\Big(\frac{y}{\lambda}\Big)\mathrm{d}y \\ &= \int_{\mathbb{R}} \left(f(x-\lambda y) - f(x)\right) h_1(y)\mathrm{d}y.\end{aligned}$$

Since $f \in L^1_{\mathrm{bc}}(\mathbb{R})$, there exists a $C > 0$ such that $|f(x)| \leqslant C$ for every $x \in \mathbb{R}$. Hence, the integrand is dominated by $2Ch_1(y)$. Moreover, as $\lambda \to 0$ then $f(x - \lambda y) \to f(x)$, thus by the dominated convergence theorem we get the result. □

Theorem 7.9 (Inversion Theorem)**.** *Let $f \in L^1_{\mathrm{bc}}(\mathbb{R})$ and assume that $\widehat{f} \in L^1(\mathbb{R})$. Then, for every $x \in \mathbb{R}$, we have*

$$f(x) = \frac{1}{2\pi}\int_{\mathbb{R}} \mathrm{e}^{-\mathrm{i}ux}\widehat{f}(u)\mathrm{d}u \quad \textit{i.e.} \quad f(x) = \check{\widehat{f}}(x). \tag{7.7}$$

Proof. We have shown, for $\lambda > 0$, that

$$(f * h_\lambda)(x) = \frac{1}{2\pi}\int_{\mathbb{R}} \mathrm{e}^{-\lambda|u|}\mathrm{e}^{-\mathrm{i}ux}\widehat{f}(u)\mathrm{d}u.$$

The left hand side tends to $f(x)$ as $\lambda \to 0$, *cf.* Lemma 7.8, while the integrand on the right hand side is dominated by $|\widehat{f}(u)|$. Hence, the claim follows from the dominated convergence theorem. □

7.2 The Fourier method for the computation of expectations and option prices

We will start by showing how to compute integrals of the form $I[f;X] = \mathbb{E}[f(X)]$ using the Fourier transform. Let $\mathbb{P}_X$ denote the law and M_X the (extended) moment generating function of the random variable X; that is

$$M_X(u) = \mathbb{E}\big[\mathrm{e}^{uX}\big]$$

for suitable $u \in \mathbb{C}$ such that the expectation is finite. We associate a *dampened* function f_R to any function f, defined via

$$f_R(x) = \mathrm{e}^{-Rx}f(x), \quad x \in \mathbb{R}, \tag{7.8}$$

for some $R \in \mathbb{R}$. Moreover, we define the following sets:

$$\mathcal{I} := \big\{R \in \mathbb{R} : f_R \in L^1_{\mathrm{bc}}(\mathbb{R}) \text{ and } \widehat{f_R} \in L^1(\mathbb{R})\big\} \quad \text{and} \quad \mathcal{J} := \big\{R \in \mathbb{R} : M_X(R) < \infty\big\}. \tag{7.9}$$

Theorem 7.10. *Assume that $\mathcal{R} := \mathcal{I} \cap \mathcal{J} \neq \varnothing$ and let $R \in \mathcal{R}$. Then, the expectation $I[f;X]$ is provided by*

$$I[f;X] = \frac{1}{2\pi}\int_{\mathbb{R}} M_X(R - \mathrm{i}u)\widehat{f}(u + \mathrm{i}R)\mathrm{d}u. \tag{7.10}$$

Proof. Using the definition of the dampened function (7.8) we have that

$$I[f;X] = \int_{\mathbb{R}} f(x)\mathbb{P}_X(\mathrm{d}x) = \int_{\mathbb{R}} \mathrm{e}^{Rx}f_R(x)\mathbb{P}_X(\mathrm{d}x). \tag{7.11}$$

Take an $R \in \mathcal{R} \neq \varnothing$, then we have that $f_R \in L^1_{\mathrm{bc}}(\mathbb{R})$ and its Fourier transform $\widehat{f_R}$ is well defined for every $u \in \mathbb{R}$ and is also continuous and bounded. Additionally, $\widehat{f_R} \in L^1_{\mathrm{bc}}(\mathbb{R})$. Therefore, using the Inversion Theorem, *cf.* Theorem 7.9, $\widehat{f_R}$ can be inverted and f_R can be represented, for *all* $x \in \mathbb{R}$, as

$$f_R(x) = \frac{1}{2\pi} \int_{\mathbb{R}} \mathrm{e}^{-\mathrm{i}xu} \widehat{f_R}(u) \mathrm{d}u. \tag{7.12}$$

Now, returning to the integration problem (7.11) we get that

$$\begin{aligned} I[f; X] &= \int_{\mathbb{R}} \mathrm{e}^{Rx} \left(\frac{1}{2\pi} \int_{\mathbb{R}} \mathrm{e}^{-\mathrm{i}xu} \widehat{f_R}(u) \mathrm{d}u \right) \mathbb{P}_X(\mathrm{d}x) \\ &= \frac{1}{2\pi} \int_{\mathbb{R}} \left(\int_{\mathbb{R}} \mathrm{e}^{(R-\mathrm{i}u)x} \mathbb{P}_X(\mathrm{d}x) \right) \widehat{f_R}(u) \mathrm{d}u \\ &= \frac{1}{2\pi} \int_{\mathbb{R}} M_X(R - \mathrm{i}u) \widehat{f}(u + \mathrm{i}R) \mathrm{d}u, \end{aligned} \tag{7.13}$$

where for the second equality we have applied Fubini's theorem, while for the last one we have

$$\widehat{f_R}(u) = \int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}ux} \mathrm{e}^{-Rx} f(x) \mathrm{d}x = \widehat{f}(u + \mathrm{i}R).$$

Finally, the application of Fubini's theorem is justified since

$$\begin{aligned} \int_{\mathbb{R}} \int_{\mathbb{R}} \mathrm{e}^{Rx} |\mathrm{e}^{-\mathrm{i}ux}| |\widehat{f_R}(u)| \mathrm{d}u \mathbb{P}_X(\mathrm{d}x) &\leqslant \int_{\mathbb{R}} \mathrm{e}^{Rx} \left(\int_{\mathbb{R}} |\widehat{f_R}(u)| \mathrm{d}u \right) \mathbb{P}_X(\mathrm{d}x) \\ &\leqslant \mathrm{const} \cdot M_X(R) < \infty, \end{aligned}$$

where we have used again that $\widehat{f_R} \in L^1(\mathbb{R})$ and $M_X(R)$ is finite for $R \in \mathcal{R}$. □

Remark 7.11 (Dual assumptions)**.** Assumption $\mathcal{R} = \mathcal{I} \cap \mathcal{J} \neq \varnothing$ implies in particular that the function f is continuous. However, dealing with discontinuous functions in this framework is also of significant interest; think, for example, of the function $f(x) = 1_{\{x \leqslant b\}}$ which corresponds to the payoff of a digital option in mathematical finance. In that case we can work with the 'dual' sets

$$\mathcal{I}' := \big\{ R \in \mathbb{R} : f_R \in L^1(\mathbb{R}) \big\} \quad \text{and} \quad \mathcal{J}' := \big\{ R \in \mathbb{R} : M_X(R) < \infty \text{ and } M_X(R - \mathrm{i}\cdot) \in L^1(\mathbb{R}) \big\}, \tag{7.14}$$

and assume that $R \in \mathcal{R}' := \mathcal{I}' \cap \mathcal{J}' \neq \varnothing$. This assumption yields that $\mathrm{e}^{R\cdot}\mathbb{P}_X$ possesses a continuous bounded Lebesgue density, say ρ; *cf.* Breiman [4, Theorem 8.39]. Then, we can identify ρ, instead of f_R, with the inverse of its Fourier transform, *i.e.* with the inverse of the characteristic function of the random variable X, and the proof goes through with the obvious modifications. This statement is almost identical to Theorem 3.2 in Raible [48]. Let us point out that there is an interesting trade-off of continuity between the function f and the distribution of X; or, equivalently, a trade-off of integrability between $\widehat{f}$ and M_X.

Remark 7.12 (Minimal assumptions)**.** The minimal assumptions for the existence of a Fourier integration formula are the following: $R \in \mathcal{I}_{\min} \cap \mathcal{J}_{\min} \neq \varnothing$, where

$$\mathcal{I}_{\min} := \big\{ R \in \mathbb{R} : f_R \in L^1(\mathbb{R}) \big\} \quad \text{and} \quad \mathcal{J}_{\min} := \big\{ R \in \mathbb{R} : M_X(R) < \infty \big\}, \tag{7.15}$$

and the formula exists as a pointwise limit; *cf.* Eberlein et al. [16, Theorem 2.7].

The prerequisites of Theorem 7.10 are quite easy to check in specific cases, apart from the condition $f_R \in L^1_{\mathrm{bc}}(\mathbb{R})$. In general, it is also an interesting question to know when the Fourier transform of an integrable function is integrable. This problem is well understood for smooth (C^2 or C^∞) functions, see *e.g.* Deitmar [12], but the functions we are dealing with are typically not

smooth. Hence, we will provide below an easy-to-check condition for a non-smooth function to have an integrable Fourier transform, which generalizes Theorem 7.4(viii).

Let us consider the Sobolev space $H^1(\mathbb{R})$, with

$$H^1(\mathbb{R}) = \Big\{ g \in L^2(\mathbb{R}) \;\Big|\; \partial g \text{ exists and } \partial g \in L^2(\mathbb{R}) \Big\},$$

where ∂g denotes the *weak* derivative of a function g; see *e.g.* Sauvigny [52]. Let $g \in H^1(\mathbb{R})$, then from Proposition 5.2.1 in Zimmer [61] we get that

$$\widehat{\partial g}(u) = -iu\widehat{g}(u) \tag{7.16}$$

and $\widehat{g}, \widehat{\partial g} \in L^2(\mathbb{R})$.

Lemma 7.13. *Let $g \in H^1(\mathbb{R})$, then $\widehat{g} \in L^1(\mathbb{R})$.*

Proof. Using the above results, we have that

$$\infty > \int_{\mathbb{R}} \Big(\big|\widehat{g}(u)\big|^2 + \big|\widehat{\partial g}(u)\big|^2 \Big) \mathrm{d}u = \int_{\mathbb{R}} \big|\widehat{g}(u)\big|^2 \big(1 + |u|^2\big) \mathrm{d}u. \tag{7.17}$$

Now, by the Hölder inequality, using that $(1+|u|)^2 \leqslant 3(1+|u|^2)$ and (7.17), we get

$$\begin{aligned} \int_{\mathbb{R}} \big|\widehat{g}(u)\big| \mathrm{d}u &= \int_{\mathbb{R}} \big|\widehat{g}(u)\big| \frac{1+|u|}{1+|u|} \mathrm{d}u \\ &\leqslant \left(\int_{\mathbb{R}} \big|\widehat{g}(u)\big|^2 (1+|u|)^2 \mathrm{d}u \right)^{\frac{1}{2}} \left(\int_{\mathbb{R}} \frac{1}{(1+|u|)^2} \mathrm{d}u \right)^{\frac{1}{2}} < \infty, \end{aligned}$$

and the result is proved. □

7.2.1 Applications in option pricing

Let us now turn our attention to the computation of option prices using Fourier methods. In the sequel we will work in the following framework: Let $S = (S_t)_{t \geqslant 0}$ denote the price of a financial asset which is modeled as an exponential semimartingale, *i.e.*

$$S_t = S_0 \mathrm{e}^{X_t}, \quad t \geqslant 0, \tag{7.18}$$

assuming that S is a martingale under some probability measure $\mathbb{P}$, while the interest rate equals zero for simplicity. We want to compute the price of a European option with payoff $F(S_T)$ and assume we can rewrite this in terms of the log-price X; then it follows:

$$F(S_T) = f(X_T + s) \quad \text{and} \quad \mathbb{E}[F(S_T)] = \mathbb{E}[f(X_T + s)] =: I[f; X], \tag{7.19}$$

where $s = \log S_0$ and $X = X_T + s$. We call f the *payoff function* and X the *payoff variable*. Then, using Theorem 7.10 we get immediately that the price of this option is provided by

$$\mathbb{E}[f(X_T + s)] = \frac{1}{2\pi} \int_{\mathbb{R}} S_0^{R - \mathrm{i}u} M_{X_T}(R - \mathrm{i}u) \widehat{f}(u + \mathrm{i}R) \mathrm{d}u, \tag{7.20}$$

for a suitable $R \in \mathcal{R}$. Indeed, in order to arrive at this formula from (7.10) it suffices to note that

$$M_X(u) = \mathbb{E}\big[\mathrm{e}^{u(X_T + s)}\big] = S_0^u \; M_{X_T}(u).$$

Remark 7.14. We consider a European 'vanilla' option for the sake of simplicity, *i.e.* an option that cannot be exercised early and depends only on the value of S at time T. We can also consider exotic, path-dependent options in the same framework, assuming that the payoff variable X takes values on a space of paths and its moment generating function is known; *cf.* [16, p. 5].

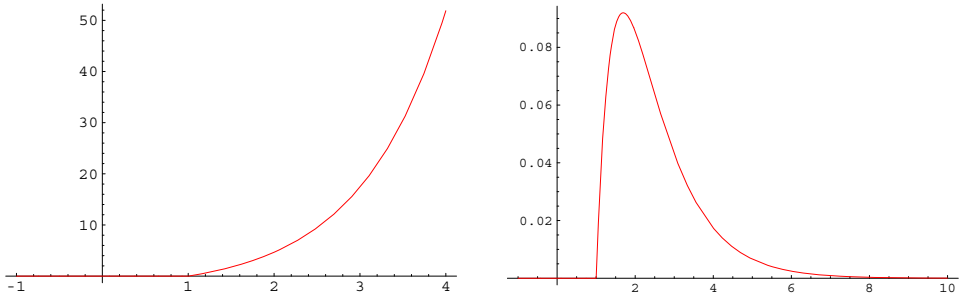


Figure 7.1: Call payoff function in log-price (left) and its dampened counterpart (right).

Remark 7.15. There are many different ways to derive the Fourier formula for option pricing in (7.20), see Schmelzle [53] for a comprehensive overview of the related literature.

Example 7.16 (Call option)**.** The payoff of the standard call option with strike $K \in \mathbb{R}_+$ is $f(x) = (\mathrm{e}^x - K)^+$. Let $z \in \mathbb{C}$ with $\Im z \in (1, \infty)$, then the Fourier transform of the payoff function of the call option is

$$\begin{aligned} \widehat{f}(z) &= \int_{\mathbb{R}} \mathrm{e}^{\mathrm{i}zx}(\mathrm{e}^x - K)^+ \mathrm{d}x = \int_{\ln K}^{\infty} \mathrm{e}^{(1+\mathrm{i}z)x}\mathrm{d}x - K\int_{\ln K}^{\infty} \mathrm{e}^{\mathrm{i}zx}\mathrm{d}x \\ &= -K^{1+\mathrm{i}z}\frac{1}{1+\mathrm{i}z} + K^{iz}\frac{K}{\mathrm{i}z} = \frac{K^{1+\mathrm{i}z}}{\mathrm{i}z(1+\mathrm{i}z)}. \end{aligned} \tag{7.21}$$

Now, regarding the dampened payoff function of the call option, we easily get for $R \in (1, \infty)$ that $f_R \in L^1_{\mathrm{bc}}(\mathbb{R}) \cap L^2(\mathbb{R})$. The weak derivative of f_R is

$$\partial f_R(x) = \begin{cases} 0, & \text{if } x < \ln K, \\ \mathrm{e}^{-Rx}(\mathrm{e}^x - R\mathrm{e}^x + RK), & \text{if } x > \ln K. \end{cases} \tag{7.22}$$

Again, we have that $\partial f_R \in L^2(\mathbb{R})$. Therefore, $f_R \in H^1(\mathbb{R})$ and using Lemma 7.13 we can conclude that $\widehat{f_R} \in L^1(\mathbb{R})$. Summarizing, the Fourier transform of the call payoff function is provided by (7.21) and $\mathcal{I} = (1, \infty)$.

Let us point out here that the call payoff function is neither integrable nor bounded as required in order to apply the Fourier transform method. The role of the dampening parameter R is exactly to make this function integrable and bounded; see Figure 7.1 for an illustration.

Exercise 7.2. Show that the Fourier transform of the put payoff function $f(x) = (K - \mathrm{e}^x)^+$ is also provided by (7.21) and that $\mathcal{I} = (-\infty, 0)$.

Exercise 7.3. Compute the Fourier transforms and the sets $\mathcal{I}$ or $\mathcal{I}'$ for the following payoff functions: (i) digital $1_{\{x \leqslant b\}}$, (ii) double digital $1_{\{a \leqslant x \leqslant b\}}$, (iii) asset-or-nothing digital $\mathrm{e}^x 1_{\{x \leqslant b\}}$, (iv) self-quanto call $\mathrm{e}^x(\mathrm{e}^x - K)^+$, and (v) power call $[(\mathrm{e}^x - K)^+]^2$.

Example 7.17. Fourier methods for option pricing are particularly well-suited for the class of models known in mathematical finance as *exponential Lévy models*, *i.e.* when the semimartingale X in (7.18) is actually a Lévy process; see Appendix B for a brief introduction to Lévy processes. This class of models is very broad and contains, among many others, the VG and CGMY processes (Madan and Seneta [40], Carr et al. [8]), the generalized hyperbolic and normal inverse Gaussian (NIG) distributions (Eberlein and Prause [15], Barndorff-Nielsen [3]), as well as the Meixner process (Schoutens [54]). The density function is typically not known in these models, however the characteristic function of Lévy processes admits an explicit representation via the Lévy-Khintchine formula in terms of the Lévy triplet; see Theorem B.5. Consider now a Lévy process $X = (X_t)_{t \geqslant 0}$ with triplet (b, c, ν) and assume that its moment generating function is finite for all $u \in [a, b]$. Then,

we can show, *cf.* Exercise 7.5, that the extended moment generating function is well-defined on the strip $\{u \in \mathbb{C} : a \leqslant \Re u \leqslant b\}$ and equals

$$M_{X_1}(u) = \exp\left\{bu + \frac{cu}{2} + \int_{\mathbb{R}} (\mathrm{e}^{ux} - 1 - ux)\,\nu(\mathrm{d}x)\right\}. \tag{7.23}$$

In other words, for exponential Lévy models we get that $\mathcal{J} = [a, b]$, and we can use Fourier methods for computing, for example, the price of a call option as long as $[a, b] \cap (1, +\infty) \neq \varnothing$.

Exercise 7.4. Prove that the set $[a, b]$ above contains $[0, 1]$. (*Hint:* S is a martingale.)

Exercise 7.5. Let ρ be a measure on the space $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$, define its characteristic function via $\widehat{\rho}(u) = \int \mathrm{e}^{iux}\rho(\mathrm{d}x)$ for $u \in \mathbb{R}$, and assume that $\int \mathrm{e}^{ux}\rho(\mathrm{d}x) < \infty$ for all $u \in [a, b]$. Show that the characteristic function $\widehat{\rho}$ has an extension that is continuous in $(-\infty, \infty) \times i[-b, -a]$ and is analytic in the interior of this strip, *i.e.* in $(-\infty, \infty) \times i(-b, -a)$.

Example 7.18. Fourier methods are also well-suited for *affine models*, see Appendix C for an introduction, since the moment generating is provided by the solution of a system of (generalized) Riccati ODEs, see (C.1) and (C.2). Many stochastic volatility models are of affine form, *i.e.* the log-price is given as the first component of an n-dimensional affine process, while the other components are interpreted as some sort of volatility, *e.g.* volatility, volatility of volatility and so on. The interpretation of the Heston model as an affine process is provided in Example C.6.

7.2.2 Computation of Greeks by Fourier methods

The structure of the asset price model as an exponential semimartingale (7.18), and the resulting structure of the option price function (7.20), allow us to easily derive general formulas for the sensitivities of the option price with respect to model parameters. In this subsection we will focus on the delta, the sensitivity of the option price with respect to the initial value, while sensitivities with respect to other parameters can be derived analogously.

The delta (Δ) of an option price is the partial derivative of the price with respect to the initial value S_0. Therefore, for a generic option with payoff function f and payoff variable X, we have that

$$\Delta_f(X; S_0) = \frac{\partial}{\partial S_0}\mathbb{E}[f(X_T + \log S_0)]. \tag{7.24}$$

The following theorem provides a formula for the computation of the delta based on Fourier transforms.

Theorem 7.19. *Assume that the asset price process is an exponential semimartingale as in* (7.18) *and the price of an option with payoff function f and payoff variable X is given by* (7.20)*. Moreover, assume that one of the following holds:*

(i) $|u||M_{X_T}(R - \mathrm{i}u)|$ *is integrable and* $\widehat{f}(\cdot + \mathrm{i}R)$ *is bounded;*

(ii) $|u||\widehat{f}(u + \mathrm{i}R)|$ *is integrable and* $M_{X_T}(R - \mathrm{i}\cdot)$ *is bounded.*

Then, the delta of the option price is provided by

$$\Delta_f(X; S_0) = \frac{1}{2\pi}\int_{\mathbb{R}} S_0^{R-1-\mathrm{i}u} M_{X_T}(R - \mathrm{i}u)\frac{\widehat{f}(u + \mathrm{i}R)}{(R - \mathrm{i}u)^{-1}}\mathrm{d}u. \tag{7.25}$$

Proof. Assuming we can exchange integration and differentiation, it follows easily that

$$\begin{aligned}
\Delta_f(X; S_0) &= \frac{\partial}{\partial S_0}\mathbb{E}[f(X_T + s)] \\
&= \frac{1}{2\pi}\int_{\mathbb{R}} \frac{\partial}{\partial S_0} S_0^{R-\mathrm{i}u} M_{X_T}(R - \mathrm{i}u)\widehat{f}(u + \mathrm{i}R)\mathrm{d}u \\
&= \frac{1}{2\pi}\int_{\mathbb{R}} S_0^{R-1-\mathrm{i}u} M_{X_T}(R - \mathrm{i}u)\frac{\widehat{f}(u + \mathrm{i}R)}{(R - \mathrm{i}u)^{-1}}\mathrm{d}u.
\end{aligned}$$

Now we have to justify this operation. Using Elstrodt [17, Satz IV.5.7] and the elementary inequality $|\mathrm{Im} f| + |\mathrm{Re} f| \leqslant 2|f|$, we get that we can differentiate under the integral sign if there exists an integrable function ℓ such that for all $u \in \mathbb{R}$ and all $S_0 > 0$

$$\Big|\frac{\partial}{\partial S_0} G(u, S_0)\Big| \leqslant \ell(u),$$

where

$$G(u, S_0) = S_0^{R-\mathrm{i}u} M_{X_T}(R - \mathrm{i}u)\widehat{f}(u + \mathrm{i}R).$$

Now we can estimate the partial derivative of the function G:

$$\begin{aligned}\Big|\frac{\partial}{\partial S_0} G(u, S_0)\Big| &= |\mathrm{e}^{(R-1-\mathrm{i}u)\log S_0}||R - \mathrm{i}u||M_{X_T}(R - \mathrm{i}u)\widehat{f}(u + \mathrm{i}R)| \\ &\leqslant c(1 + |u|)|M_{X_T}(R - \mathrm{i}u)||\widehat{f}(u + \mathrm{i}R)| =: \ell(u). \end{aligned} \tag{7.26}$$

Sufficient conditions for the function ℓ in (7.26) to be integrable are provided by conditions (i) and (ii) in the statement above. □

Remark 7.20. Let us point out that the first condition implies that the measure $\mathbb{P}_{X_T}$ has a density of class C^1; see Sato [51, Prop. 28.1]. Moreover, both conditions highlight once again the interplay between the properties of the measure and of the payoff function.

7.2.3 The multi-dimensional case

Let us now consider the multi-dimensional case, *i.e.* let $f : \mathbb{R}^d \to \mathbb{R}$ be a d-dimensional payoff function and X be a d-dimensional payoff variable. The Fourier transform of the function $f \in L^1(\mathbb{R}^d)$ is defined as $\widehat{f}(u) = \int_{\mathbb{R}^d} \mathrm{e}^{\mathrm{i}\langle u,x\rangle} f(x)\mathrm{d}x$ for $u \in \mathbb{R}^d$, the dampened function f_R is defined via $f_R(x) = \mathrm{e}^{\langle R,x\rangle} f(x)$ for all $x \in \mathbb{R}^d$ and some $R \in \mathbb{R}^d$, while the (extended) moment generating function of X is defined as $M_X(u) = \mathbb{E}[\mathrm{e}^{\langle u,X\rangle}]$ for suitable $u \in \mathbb{C}^d$. Consider also the following sets:

$$\begin{aligned}&\mathcal{I} := \big\{R \in \mathbb{R}^d : f_R \in L^1_{\mathrm{bc}}(\mathbb{R}^d) \text{ and } \widehat{f_R} \in L^1(\mathbb{R}^d)\big\} \quad \text{and} \quad \mathcal{I}' := \big\{R \in \mathbb{R} : f_R \in L^1(\mathbb{R}^d)\big\} \\ &\mathcal{J} := \big\{R \in \mathbb{R}^d : M_X(R) < \infty\big\} \quad \text{and} \quad \mathcal{J}' := \big\{R \in \mathbb{R}^d : M_X(R) < \infty \text{ and } M_X(R - \mathrm{i}\cdot) \in L^1(\mathbb{R}^d)\big\}.\end{aligned}$$

Then, a result analogous to Theorem 7.10 holds true.

Theorem 7.21. *Assume that either $\mathcal{R} = \mathcal{I} \cap \mathcal{J} \neq \varnothing$ or $\mathcal{R}' = \mathcal{I}' \cap \mathcal{J}' \neq \varnothing$ and let $R \in \mathcal{R} \cup \mathcal{R}'$. Then, the expectation $I[f; X] = \mathbb{E}[f(X)]$ is provided by*

$$I[f; X] = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} M_X(R - \mathrm{i}u)\widehat{f}(u + \mathrm{i}R)\mathrm{d}u. \tag{7.27}$$

Proof. See Eberlein et al. [16, Thm. 3.2] □

Now, let $S^1, \dots, S^d$ denote asset price processes that are modeled as exponential semimartingales, *i.e.*

$$S^i_t = S^i_0 \exp(X^i_t), \quad t \geqslant 0.$$

Assume that the processes $S^1, \dots, S^d$ are martingales with respect to a (common) probability measure $\mathbb{P}$, while the interest rate is zero for simplicity. Consider an option on the assets $S^1, \dots, S^d$ with payoff $F(S^1_T, \dots, S^d_T)$, that can be written in terms of the log-prices, *i.e.* $F(S^1_T, \dots, S^d_T) = f(X^1_T + s^1_0, \dots, X^d_T + s^d_0)$, where $s^i_0 = \log S^i_0$. Then, analogously to (7.20), the price of this option is provided by

$$\mathbb{E}[f(X_T + s)] = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \mathrm{e}^{\langle R - \mathrm{i}u, s_0\rangle} M_{X_T}(R - \mathrm{i}u)\widehat{f}(u + \mathrm{i}R)\mathrm{d}u, \tag{7.28}$$

for a suitable $R \in \mathbb{R}^d$ such that the assumption of Theorem 7.21 is satisfied. Here $s_0 = (s^1_0, \dots, s^d_0)$.

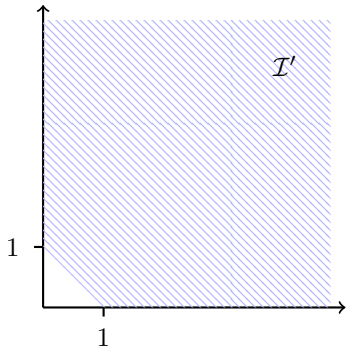


Figure 7.2: A graphical representation of the set $\mathcal{I}'$ for the call option on the minimum of two assets.

Remark 7.22 (Curse of dimensionality)**.** The numerical integration in (7.27) suffers obviously from the curse of dimensionality, and does not lead to a competitive numerical scheme in dimensions higher than two or three.

Example 7.23. The payoff function of a call option on the minimum of d assets is

$$f(x) = (\mathrm{e}^{x_1} \wedge \cdots \wedge \mathrm{e}^{x_d} - K)^+,$$

for $x \in \mathbb{R}^d$, where $a \wedge b = \min\{a, b\}$. The Fourier transform of this payoff function is

$$\widehat{f}(z) = -\frac{K^{1+\mathrm{i}\sum_{k=1}^d z_k}}{(-1)^d(1+\mathrm{i}\sum_{k=1}^d z_k)\prod_{k=1}^d(\mathrm{i}z_k)}, \tag{7.29}$$

for $z \in \mathbb{C}^d$ with $\Im z_k > 0$ for $1 \leqslant k \leqslant d$ and $\Im(\sum_{k=1}^d z_k) > 1$; for more details we refer to Eberlein et al. [16, Appendix A]. Then, we can easily deduce for the dampened payoff function that $f_R \in L^1_{\mathrm{bc}}(\mathbb{R}^d)$ for $R \in \{R \in \mathbb{R}^d : R_i > 0, 1 \leqslant i \leqslant d; \sum_{i=1}^d R_i > 1\} =: \mathcal{I}'$.

Exercise 7.6. The payoff function of the put option on the maximum of d assets is

$$f(x) = (K - \mathrm{e}^{x_1} \vee \cdots \vee \mathrm{e}^{x_d})^+,$$

for $x \in \mathbb{R}^d$, where $a \vee b = \max\{a, b\}$. Show that its Fourier transform equals

$$\widehat{f}(z) = \frac{K^{1+\mathrm{i}\sum_{k=1}^d z_k}}{(1+\mathrm{i}\sum_{k=1}^d z_k)\prod_{k=1}^d(\mathrm{i}z_k)} \tag{7.30}$$

and that $\mathcal{I}' = \{R \in \mathbb{R}^d : R_i < 0, 1 \leqslant k \leqslant d\}$.

Example 7.24. The payoff function of the basket put option on d assets has the form

$$f(x_1, \ldots, x_d) = \left(1 - \sum_{l=1}^d \mathrm{e}^{x_l}\right)^+, \tag{7.31}$$

for $x \in \mathbb{R}^d$. The Fourier transform of this payoff function has been derived by Hubalek and Kallsen [26], see also Hurd and Zhou [27], and we get that

$$\widehat{f}(z) = \frac{\prod_{l=1}^d \Gamma(\mathrm{i}z_l)}{\Gamma\big(\mathrm{i}\sum_{l=1}^d z_l + 2\big)}, \tag{7.32}$$

for $z \in \mathbb{C}^d$ with $\Im z < 0$, where Γ denotes the Gamma function. Then, we can easily deduce for the dampened payoff function that $f_R \in L^1_{\mathrm{bc}}(\mathbb{R}^d)$ for $R \in \{R \in \mathbb{R}^d : R_i < 0, 1 \leqslant i \leqslant d\} =: \mathcal{I}'$.

Exercise 7.7. Let $f : \mathbb{R}^d \to \mathbb{R}$, $f_i : \mathbb{R} \to \mathbb{R}$ for all $1 \leqslant i \leqslant d$, and define $f(x) = \prod_{i=1}^d f_i(x_i)$. Determine its Fourier transform and show that $\mathcal{I}' \supset \prod_{i=1}^d \mathcal{I}'_i$.

7.3 The fast Fourier transform (FFT)

In the previous section we derived concrete formulas for computing option prices in terms of an inverse Fourier transform. Now we would like to discuss how to implement these formulas on a computer. Recall that the price of an option with payoff $f(X_T + s)$, expressed below as a function of the log-initial value $s = \log S_0$, equals

$$\mathbb{O}_f(s) = \frac{1}{2\pi} \int_{\mathbb{R}} \mathrm{e}^{(R-\mathrm{i}u)s} M_{X_T}(R - \mathrm{i}u) \widehat{f}(u + \mathrm{i}R) \mathrm{d}u =: \frac{\mathrm{e}^{Rs}}{2\pi} \int_{\mathbb{R}} \mathrm{e}^{-\mathrm{i}us} \psi(u) \mathrm{d}u. \tag{7.33}$$

In order to implement this integration on a computer, we first have to truncate the indefinite integral in order to obtain an integral on a finite domain, and then to discretize the integral on the finite domain.

We are left with a problem of the form

$$\mathbb{O}_f(s) \approx \frac{\mathrm{e}^{Rs}}{2\pi} \int_a^b \mathrm{e}^{-\mathrm{i}us} \psi(u) du. \tag{7.34}$$

As a first step, let us apply the trapezoidal rule to the above integral, using a uniform grid $u_l := a + \eta l$ for a constant η and for $0 \leqslant l \leqslant N-1$, implying that $b - a = \eta(N-1)$. Therefore we approximate

$$\int_a^b \mathrm{e}^{-\mathrm{i}us} \psi(u) du \approx \eta \left(\frac{\mathrm{e}^{-\mathrm{i}as} \psi(a)}{2} + \sum_{l=1}^{N-2} \mathrm{e}^{-\mathrm{i}u_l s} \psi(u_l) + \frac{\mathrm{e}^{-\mathrm{i}bs} \psi(b)}{2} \right) =: \eta \sum_{l=0}^{N-1} \mathrm{e}^{-\mathrm{i}u_l s} \psi'(u_l),$$

where $\psi'(u) = \psi(u)$ for all $u \in (a, b)$ and $\psi'(u) = \psi(u)/2$ for $u \in \{a, b\}$. This approximation requires computational work proportional to N.

Now, assume that we do not only want to compute the price at one log-initial value s, but for a whole variety of log-initial values — or, equivalently, for a whole variety of strikes, as is the case in a typical calibration situation. We choose a uniform grid again in the log-S_0 domain, *i.e.* we set $s_j := -\beta + \lambda j$ where $\beta := \lambda N/2$. Thus, we want to compute the values

$$\sum_{l=0}^{N-1} \mathrm{e}^{-\mathrm{i}(a+\eta l)\lambda j} e^{\mathrm{i}\beta u_l} \psi(u_l) \eta, \quad j = 0, \ldots, N-1.$$

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

$$\Phi_j = \sum_{l=0}^{N-1} \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N} lj} \phi_l, \quad j = 0, \ldots, N-1, \tag{7.35}$$

where $\phi_l := \mathrm{e}^{\mathrm{i}\beta u_l} \psi(u_l)$. Indeed, the option price with log-initial value s_j is approximated by

$$\mathbb{O}_f(s_j) \approx \frac{\mathrm{e}^{\beta R - \lambda j (R - \mathrm{i}a)}}{2\pi} \eta \Phi_j.$$

We have used all these assumptions and notation, because the vector Φ defined in (7.35) is the *discrete Fourier transform* of the vector ϕ, and there is a very efficient numerical algorithm for computing discrete Fourier transforms. The computational cost of a usual implementation of (7.35) is proportional to N^2, but the so-called *fast Fourier transform* (FFT) reduces the work to $N \log_2(N)$.

Let $\omega_N := \mathrm{e}^{-2\pi\mathrm{i}/N}$ 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 (7.35) as $\Phi = T_N \phi$.

Lemma 7.25. *Let $\phi \in \mathbb{C}^{2N}$ and let $\Phi := T_{2N}\phi$ denote its discrete Fourier transform. Denote by $\phi' := (\phi_1, \phi_3, \ldots, \phi_{2N-1})$ and by $\phi'' := (\phi_2, \phi_4, \ldots, \phi_{2N})$, and furthermore denote by $\Phi' := (\Phi_1, \ldots, \Phi_N)$ and by $\Phi'' := (\Phi_{N+1}, \ldots, \Phi_{2N})$. Moreover, denote $D_N := \operatorname{diag}(\omega_{2N}^0, \ldots, \omega_{2N}^{N-1})$, $c := T_N\phi'$ and $d := D_N T_N \phi''$. Then*

$$\Phi' = c + d \quad \text{and} \quad \Phi'' = c - d.$$

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

Lemma 7.25 forms the basis of a classical *divide-and-conquer* algorithm, the celebrated FFT.

Algorithm 7.26 (FFT)**.** *Assume that $N = 2^J$, $J \geqslant 1$. Given $\phi \in \mathbb{C}^N$, apply the following recursive algorithm to compute its discrete Fourier transform $\Phi = T_N\phi$:*

1. *If $N = 2$ go to 2, otherwise: split ϕ into ϕ' and ϕ'' as in Lemma 7.25, apply the FFT to compute $c = T_{N/2}\phi'$, $d = D_{N/2}T_{N/2}\phi''$ and return $\Phi = (\Phi', \Phi'')$ given by $\Phi' = c + d$ and $\Phi'' = c - d$.*

2. *If $N = 2$ compute $\Phi = T_2\phi$ directly.*

It can be easily shown that the computational effort to compute the discrete Fourier transform using Algorithm 7.26 is, indeed, proportional to $N \log_2(N)$.

Lemma 7.27. *Let N be a power of 2. Let C denote the computational work of a floating point operation (addition, subtraction, multiplication). Then the computational work $\mathcal{W}(N)$ of the FFT can be bounded by*

$$\mathcal{W}(N) \leqslant C\left(\frac{3}{2}\log_2(N) + \frac{1}{2}\right)N.$$

Proof. From Lemma 7.25 we see that for an FFT in dimension N we need two FFTs in dimension $N/2$, one vector addition and one vector subtraction in dimension $N/2$ and one element-wise multiplication of two vectors in dimension $N/2$. In total, this means

$$\mathcal{W}(N) \leqslant 2\mathcal{W}(N/2) + 3/2CN, \quad \mathcal{W}(2) \leqslant 4C.$$

Let $w(N) := \mathcal{W}(N)/(CN)$, we get the recursion

$$w(N) \leqslant 2w(N/2) + 3/2N, \quad w(2) \leqslant 2,$$

which gives $w(N) \leqslant 1/2 + 3/2\log_2(N)$. □

Remark 7.28. In the same way, we can compute the inverse discrete Fourier transform.

Remark 7.29. Many variants of FFT exist. Although 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 its re-discovery by Cooley and Tukey [11]. 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 (for the fast evaluation of polynomials).

Remark 7.30. Although Carr and Madan [7] use FFT for evaluating the option price formula based on the Fourier transform, other authors like Kahl and Lord [28] advocate alternative specialized algorithms or classical quadrature because strike prices in practical calibration scenarios are usualy not uniformly arranged.

7.4 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 [18] have constructed a very fast method based on cosine expansions.

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., $\mathrm{supp}(q) \subset [0,\pi]$ with $q(\pi) = 0$. Now, *Pontryagin duality*, as a starting point see [58], 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.

Coming back to the concrete problem, 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)

$$q(x) = \sum_{k=0}^{\infty}{}' A_k\cos\left(k\pi\frac{x-a}{b-a}\right), \quad A_k := \frac{2}{b-a}\int_a^b q(x)\cos\left(k\pi\frac{x-a}{b-a}\right)dx. \tag{7.36}$$

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

(7.37)
$$A_k = \frac{2}{b-a}\Re\left(\phi_1\left(\frac{k\pi}{b-a}\right)\exp\left(-i\frac{ka\pi}{b-a}\right)\right) \approx \frac{2}{b-a}\Re\left(\phi\left(\frac{k\pi}{b-a}\right)\exp\left(-i\frac{ka\pi}{b-a}\right)\right) =: F_k.$$

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

$$q(x) \approx q_1(x) := \sum_{k=0}^{N-1}{}' F_k \cos\left(k\pi\frac{x-a}{b-a}\right). \tag{7.38}$$

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

Remark 7.31. There are three different approximation errors in (7.38). 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 (7.38) (where we drop the subscript T). Thus, we obtain the approximation

$$C(S_0,T) \approx C_1(S_0,T) := e^{-rT}\sum_{k=0}^{N-1}{}'\,\Re\left(\phi_T\left(\frac{k\pi}{b-a}\right)e^{-ik\pi\frac{a}{b-a}}\right)C_k, \tag{7.39}$$

where

$$C_k := \frac{2}{b-a}\int_a^b f(x)\cos\left(k\pi\frac{x-a}{b-a}\right)dx. \tag{7.40}$$

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 (7.39) 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 7.32. 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 [18] also analyse the error of the approximation (7.39), 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 (7.39), 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

$$a = c_1 - L\sqrt{c_2 + \sqrt{c_4}}, \quad b = c_1 + L\sqrt{c_2 + \sqrt{c_4}} \tag{7.41}$$

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 [47, Theorem V.7].

Theorem A.1. *Let Z be a d-dimensional càdlàg semimartingale with $Z_0 = 0$ and let $F : \mathbb{R}_{\geqslant 0} \times \mathbb{R}^n \to \mathbb{R}^{n\times d}$ be Lipschitz in the sense that for every $t \geqslant 0$ there is a constant K_t such that*

$$\forall x, y \in \mathbb{R}^n : \ |F(t,x) - F(t,y)| \leqslant K_t \, |x - y| \, .$$

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

$$\int_0^t H_{s-} \circ dZ_s := \int_0^t H_{s-} dZ_s + \frac{1}{2}[H,X]^c_t. \tag{A.1}$$

The advantage of the Stratonovich integral is that Ito's formula holds in a much simpler form: let $f : \mathbb{R}_{\geqslant 0} \times \mathbb{R}^n \to \mathbb{R}^n$ be C^1 in the first and C^2 in the second component. Then

$$\begin{aligned} f(t, Z_t) = f(0, Z_0) + \int_0^t \partial_t f(s, Z_{s-}) ds + \int_{0+}^t \nabla f(Z_{s-}) \cdot \circ dZ_s \\ + \sum_{0<s\leqslant t} \left(f(Z_s) - f(Z_{s-}) - \nabla f(Z_{s-}) \cdot \Delta Z_s \right). \end{aligned} \tag{A.2}$$

The following existence and uniqueness result for Stratonovich SDEs is a special case of Protter [47, Theorem V.22].

Theorem A.2. *Assume that $F : \mathbb{R}_{\geqslant 0} \times \mathbb{R}^{n\times d} \to \mathbb{R}^n$ 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

$$V_0(x) := V(x) - \frac{1}{2}\sum_{i=1}^d DV_i(x)V_i(x), \tag{A.4}$$

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 [25]. 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 [30, 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 $2\kappa\theta \geqslant \xi^2$, 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 $k : [0,T] \times \mathbb{R}^n \to \mathbb{R}_{\geqslant 0}$, consider the Cauchy problem

$$\begin{cases} \dfrac{\partial}{\partial t} u(t,x) + Lu(t,x) + g(x) = k(t,x)u(t,x), \quad (t,x) \in [0,T) \times \mathbb{R}^n, \\ u(T,x) = f(x), \quad x \in \mathbb{R}^n. \end{cases} \tag{A.11}$$

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

$$u(t,x) = E\left[f(X_T) \exp\left(-\int_t^T k(s, X_s) ds \right) + \int_t^T g(s, X_s) \exp\left(-\int_t^s k(v, X_v) dv \right) ds \,\middle|\, X_t = x \right]. \tag{A.12}$$

Similar *stochastic representations* exist for the corresponding Dirichlet and Neumann problems.

A.3 The first variation

Let X_t^x, $x \in \mathbb{R}^n$, $t \geqslant 0$, 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 $t \geqslant 0$, the map $x \mapsto X_t^x(\omega)$ is a homeomorphism of $\mathbb{R}^n \to \mathbb{R}^n$ – see [47, 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 $V, V_1, \dots, V_d$ 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 $V, V_1, \dots, V_d$ 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 [47, 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 [44]. 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 $V, V_1, \dots, V_d : \mathbb{R}^n \to \mathbb{R}^n$ 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

$$V_{[I]} = \begin{cases} \left[V_{i_1}, V_{[(i_2,\dots,i_{k+1})]}\right], & i_1 \neq 0, \\ \left[V, V_{[(i_2,\dots,i_{k+1})]}\right], & i_1 = 0. \end{cases}$$

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

$$C_K(x, \eta) \coloneqq \sum_{k=1}^{K} \sum_{I \in \{0,\dots,d\}^k,\, i_k \neq 0} (V_{[I]}(x) \cdot \eta)^2,\ C_K(x) \coloneqq \inf_{|\eta|=1} C_K(x, \eta),\ C_K \coloneqq \inf_{x \in \mathbb{R}^n} C_K(x).$$

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

$$\sum_{k,j=1}^{n} a_{k,j}(x) \eta^k \eta^j \geqslant C \left|\eta\right|^2$$

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

$$\sum_{i=1}^{d} (V_i(x) \cdot \eta)^2 \geqslant C \left|\eta\right|^2,$$

which is satisfied iff $C \coloneqq C_1 > 0$.

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 [33].

Appendix B

Lévy processes

TO BE REVISED.

We cite a few facts about Lévy processes. For more information about Lévy processes and their stochastic analysis we refer to Cont and Tankov [9] and Protter [47].

Definition B.1. A stochastic process $(X_t)_{t\geqslant 0}$ 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

$$\psi(u) = -iu \cdot \alpha + \frac{1}{2}\Sigma u \cdot u - \int_{\mathbb{R}^d} \left(\exp(iu \cdot x) - 1 - iu \cdot x\mathbf{1}_{|x|\leqslant 1}\right) \nu(dx).$$

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

$$\text{(B.1)} \quad Lf(x) = \nabla f(x) \cdot \alpha + \frac{1}{2} \sum_{j,k=1}^{d} \Sigma_{j,k} \frac{\partial^2}{\partial x^j \partial x^k} f(x) + \\ + \int_{\mathbb{R}^d} \left(f(x+y) - f(x) - \nabla f(x) \cdot y\mathbf{1}_{|y|\leqslant 1}\right) \nu(dy).$$

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

$$\text{(B.2)} \quad dX_t = \sigma(X_{s-})dZ_s.$$

Then, given some boundedness and regularity conditions on $f : \mathbb{R}^n \to \mathbb{R}$, $u(t,x) := E\left[f(X_T)|X_0 = x\right]$ satisfies the PIDE

$$\text{(B.3)} \quad \frac{\partial}{\partial t} u(t,x) = Au(t,x) + \\ + \int_{\mathbb{R}^d} \left(u(t, x+\sigma(x)z) - u(t,x) - (\sigma(x)z) \cdot \nabla u(t,x)\mathbf{1}_{|\sigma(x)z|\leqslant 1}\right) \nu(dz),$$

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

Affine processes are very popular in mathematical finance because they can combine realistic features, such as stochastic volatility and jumps, with efficient computations using Fourier methods, see Section 7. The authorative reference on affine processes is Duffie, Filipovic and Schachermayer [14], while Filipović [20, Chapter 10] offers a very nice introduction to affine diffusion processes.

Definition C.1. A stochastically continuous, time-homogeneous Markov process $X = (X_t)_{t\geqslant 0}$ with state space $D = \mathbb{R}^m_{\geqslant 0} \times \mathbb{R}^n$ is called *affine* iff the logarithm of the characteristic function is affine in the initial state $X_0 = x$. More precisely, if there exist functions $\phi(t,u)$ taking values in $\mathbb{C}$ and $\psi(t,u)$ taking values in $\mathbb{C}^d$, $d = m+n$, such that

$$E_x[e^{u\cdot X_t}] = \exp\big(\phi(t,u) + x\cdot\psi(t,u)\big)$$

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 $x, y \in \mathbb{C}^d$.

The popularity of affine processes for numerical applications is due to the fact that there exist tractable equations for the characteristic exponent, i.e. for the functions ϕ and ψ. In particular, they are solutions to the *generalized Riccati equations*

$$\frac{\partial}{\partial t}\phi(t,u) = F(\psi(t,u)), \quad \phi(0,u) = 0, \tag{C.1a}$$

$$\frac{\partial}{\partial t}\psi(t,u) = R(\psi(t,u)), \quad \psi(0,u) = u, \tag{C.1b}$$

where the right hand side is given by

$$F(u) = \frac{1}{2}(au)\cdot u + b\cdot u - c + \int_D \big(e^{\xi\cdot u} - 1 - h_F(\xi)\cdot u\big)\, m(d\xi), \tag{C.2a}$$

$$R_i(u) = \frac{1}{2}(\alpha^i u)\cdot u + \beta^i\cdot u - \gamma_i + \int_D \big(e^{\xi\cdot u} - 1 - h^i_R(\xi)\cdot u\big)\, \mu_i(d\xi), \tag{C.2b}$$

with $F : \mathbb{C}^d \to \mathbb{C}$ and $R = (R_1,\dots,R_d) : \mathbb{C}^d \to \mathbb{C}^d$, for all $i = 1,\dots,d$.

The parameters $(a, \alpha^i, b, \beta^i, c, \gamma^i, m, \mu^i)_{1\leqslant i\leqslant d}$ should satisfy the *admissibility conditions*: Let $I = \{1,\dots,m\}$, $J = \{m+1,\dots,d\}$ 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, $b, \beta^i \in \mathbb{R}^d$, $c, \gamma_i \geqslant 0$, 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 $l \in I\setminus\{i\}$;
- $b \in D$, $\beta^i_k \geqslant 0$ 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 $j \in J$;
- $\int_{D\setminus\{0\}} \min\left(|x_{I\setminus\{i\}}| + |x_{J\cup\{i\}}|^2, 1\right) \mu_i(dx) < \infty$ for $i \in I$.

Moreover, h_F and h_R^i are truncation functions as in the Lévy–Khintchine formula.

Remark C.2. The admissibility conditions ensure that the affine process is well-defined, in the sense that the process exists and does not leave the state space $D = \mathbb{R}^m_{\geqslant 0} \times \mathbb{R}^n$.

Example C.3. In order to gain an intuitive understanding of the admissibility conditions and the interplay with the geometry of the state space, let us look at a simple example, a 1D affine diffusion process. This is given by the SDE

$$dX_t = (b + \beta X_t)dt + \sqrt{a + \alpha X_t}dW_t, \quad X_0 = x \in D, \tag{C.3}$$

where W is a standard Brownian motion. If the state space is $D = \mathbb{R}$, then we can directly deduce that the process will be well-defined in the sense above if

$$b \in \mathbb{R}, \quad \beta \in \mathbb{R}, \quad a \in \mathbb{R}_{\geqslant 0} \quad \text{and} \quad \alpha = 0.$$

On the other hand, if the state space is $D = \mathbb{R}_{\geqslant 0}$ we can equally easily deduce that the process will be well-defined if

$$b \in \mathbb{R}_{\geqslant 0}, \quad \beta \in \mathbb{R}, \quad a = 0 \quad \text{and} \quad \alpha \in \mathbb{R}_{\geqslant 0}.$$

Now one can check that these conditions coincide with the admissibility conditions for the respective states spaces. Let us also mention that in the first case the process is an Ornstein–Uhlenbeck (OU) process, while in the second case it is a square root diffusion (also know as CIR process).

The infinitesimal generator L of an affine process X admits also an explicit expression in terms of the admissible parameters. Indeed, it has the form

$$\begin{aligned} Lf(x) = \frac{1}{2}\sum_{k,l=1}^{d}\left(a_{kl} + \sum_{i=1}^{m}\alpha_{kl}^i x_i\right)\frac{\partial^2}{\partial x_k \partial x_l}f(x) + \left(b + \sum_{i=1}^{d}\beta^i x_i\right)\cdot\nabla f(x) + \\ + c + \sum_{i=1}^{d}\gamma^i x_i + \int_{D\setminus\{0\}}\left(f(x+\xi) - f(x) - h_F(\xi)\cdot\nabla f(x)\right)m(d\xi) + \\ + \sum_{i=1}^{m} x_i \int_{D\setminus\{0\}}\left(f(x+\xi) - f(x) - h_R^i(\xi)\cdot\nabla f(x)\right)\mu^i(d\xi). \end{aligned} \tag{C.4}$$

Conversely, given admissible parameters there exists an affine process with generator (C.4).

Remark C.4. Another characterization of affine processes as semimartingales can be given in terms of the (local) semimartingale characteristics, see Kallsen [29].

Example C.5. The previous example on affine diffusion processes yields immediately that the Brownian motion is an affine process. More generally, every Lévy process X with characteristic exponent κ is an affine process and the functions ϕ and ψ are provided by

$$\phi(t,u) = t\kappa(u) \quad \text{and} \quad \psi(t,u) = u.$$

This follows directly from the Lévy–Khintchine formula.

Example C.6. Consider the Heston stochastic volatility model presented in Example A.3 under a risk neutral measure, i.e. we set $\mu = r$ in (A.6a). We change variables to the log-price $X := \log S$, hence the dynamics are provided by the SDE

$$\begin{aligned} dX_t &= \left(r - \frac{1}{2}V_t\right)dt + \sqrt{V_t}dB_t^1, \quad X_0 = x \in \mathbb{R} \\ dV_t &= \kappa(\theta - V_t)dt + \eta\sqrt{V_t}\left(\rho dB_t^1 + \sqrt{1-\rho^2}dB_t^2\right), \quad V_0 = v \in \mathbb{R}_{\geqslant 0}. \end{aligned}$$

We can deduce that (V, X) is an affine process on $\mathbb{R}_{\geqslant 0} \times \mathbb{R}$ by calculating its infinitesimal generator and comparing it with (C.4). Another way is to use the characterization of affine diffusion processes in [20]. In both cases, we have the following: there are no jumps, hence the integral terms vanish. The drift term is clearly affine in (v, x) and the parameters are provided by

$$b = \begin{pmatrix} \kappa\theta \\ r \end{pmatrix}, \quad \beta^1 = \begin{pmatrix} -\kappa \\ -\frac{1}{2} \end{pmatrix} \quad \text{and} \quad \beta^2 = 0.$$

Similarly, the diffusion term is affine of the form $a(v, x) = a + \alpha^1 \cdot v + \alpha^2 \cdot x$ since the volatilities are linear in the square root of the state, with

$$a = 0, \quad \alpha^1 = \begin{pmatrix} \eta^2 & \eta\rho \\ \eta\rho & 1 \end{pmatrix} \quad \text{and} \quad \alpha^2 = 0.$$

The right hand sides of the Riccati equations (C.1) are provided by replacing the parameters above to (C.2), and we get that

$$\begin{aligned} F(u_1, u_2) &= \kappa\theta u_1 + r u_2 \\ R(u_1, u_2) &= -\kappa u_1 - \frac{1}{2}u_2 + \frac{1}{2}u_2^2 + \frac{1}{2}\eta^2 u_1^2 + \eta\rho u_1 u_2. \end{aligned} \tag{C.5}$$

Therefore, the characteristic function of the log-spot price X_t is provided by

$$E_x[\exp(uX_t)] = \exp\big(\phi(t, 0, u) + x\psi(t, 0, u)\big).$$

Several generalizations of the Heston model like the *Bates model*, a stochastic volatility model with jumps, are affine processes, too.

Example C.7. Next, we consider a fairly general jump-diffusion model. Let Z be a pure-jump semi-martingale with state-dependent intensity $\lambda(x)$ and jump measure ν. Consider the SDE

$$dX_t = b(X_t)dt + \sigma(X_t)dB_t + dZ_t,$$

with $b : \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \to \mathbb{R}^{d\times d}$, both smooth enough. Thus, the generator of the Markov process X is given by

$$\begin{aligned} Lf(x) = b(x) \cdot \nabla f(x) + \frac{1}{2}\operatorname{trace}\big(\sigma(x)\sigma(x)^T Hf(x)\big) + \\ + \lambda(x)\int_{\mathbb{R}^n} \left(f(x+\xi) - f(x) - h_F(\xi)\nabla f(x)\right)\nu(d\xi), \end{aligned}$$

where Hf denotes the Hessian matrix of f. Comparing this generator with the generic generator of an affine process given in (C.4), 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 $\sigma(x)\sigma(x)^\mathsf{T}$ is an affine function in x. In other words, the relationship between the coefficients $(b, \sigma, \lambda, \nu)$ of the jump-diffusion process X and the corresponding admissible parameters is

$$\begin{aligned} b(x) &= b + \sum_{i=1}^d \beta^i x_i \\ \sigma(x)\sigma(x)^\mathsf{T} &= a + \sum_{i=1}^d \alpha^i x_i \\ \lambda(x) &= l + \sum_{i=1}^d \lambda^i x_i, \end{aligned}$$

where $l, \lambda^1, \ldots, \lambda^n \in \mathbb{R}_{\geqslant 0}$ and

$$m(d\xi) = l\nu(d\xi) \quad \text{and} \quad \mu^i(d\xi) = \lambda^i\nu(d\xi), 1 \leqslant i \leqslant d.$$

Appendix D

Weak derivatives and Sobolev spaces

Let $G \subset \mathbb{R}^d$ be open and let

$$C_0^\infty(G) \coloneqq \{\, f \in C^\infty(G) \mid \operatorname{supp} f \subset G \text{ and bounded} \,\},$$

where $\operatorname{supp} f$ denotes the largest closed set (in $\mathbb{R}^d$) outside which f vanishes. *Integration by parts* implies for any $u \in C^k(G)$ and any *test function* $v \in C_0^\infty(G)$ and any multi-index $\alpha \in \mathbb{N}^d$, $|\alpha| \coloneqq \sum_{i=1}^d \alpha_i \leqslant k$, we have

$$\int_G D^\alpha u(x) v(x)\, dx = (-1)^{|\alpha|} \int_G u(x) D^\alpha v(x)\, dx,$$

where

$$D^\alpha f(x) \coloneqq \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}} f(x).$$

Hence, integration by parts allows us to *shift* derivatives from u to v, provided that both u and v are sufficiently regular. The idea of a *weak derivative* of u is now to *define* a derivative of a function u lacking classical differentiability by shifting the derivatives on smooth test functions. More precisely, let u be locally integrable, i.e., $u \in L^1_{\mathrm{loc}}(G)$ – as usual, in this section we omit the sigma algebra $\mathcal{B}(G)$ as well as the Lebesgue measure dx from the notation – which means that the restriction of u to any compact set $K \subset G$ is integrable.

Definition D.1. Let $u \in L^1_{\mathrm{loc}}(G)$, $\alpha \in \mathbb{N}^d$. If there is a function $w_\alpha \in L^1_{\mathrm{loc}}(G)$ such that

$$\forall v \in C_0^\infty(G): \quad \int_G w_\alpha(x) v(x)\, dx = (-1)^{|\alpha|} \int_G u(x) D^\alpha v(x)\, dx,$$

then u is called *weakly differentiable* (at order α), and w_α is called its weak derivative, symbolically $D^\alpha u \coloneqq w_\alpha$.

We give two examples, whose (elementary) proofs are left to the reader.

Example D.2. Let $G = \mathbb{R}$ and $u(x) \coloneqq (1 - |x|)^+$. Then u is once weakly differentiable, with weak derivative

$$u'(x) = \begin{cases} 0, & |x| > 1, \\ 1, & -1 < x < 0, \\ -1, & 0 < x < 1. \end{cases}$$

However, u is *not* twice weakly differentiable.

Example D.2 shows a case of an absolutely continuous function u, and verifies that it is weakly differentiable. We will see in Example D.3 that weakly differentiable functions may fail to be absolutely continuous, or even continuous.

Example D.3. Consider $u : \mathbb{R}^d \to \mathbb{R}$, $x \mapsto |x|^{-\alpha}$, $\alpha > 0$. Note that this function is discontinuous at 0. If $\alpha + 1 < d$, then u is weakly differentiable with weak derivatives

$$\frac{\partial}{\partial x_i} u(x) = -\alpha \frac{x_i}{\|x\|^{\alpha+2}} = \left(-\alpha \frac{x_i}{|x|}\right) |x|^{-(\alpha+1)}, \quad i = 1, \dots, d.$$

As a hint for the proof, recall that $\left(-\alpha \frac{x_i}{|x|}\right)$ is bounded, and that the d-dimensional volume element in spherical coordinates $\sim r^{d-1} dr$.

We now define *Sobolev spaces*, the natural spaces for solving partial differential equations, in essence the "convenient" counterparts of the classical C^k spaces.

Definition D.4. For $1 \leqslant p < \infty$ and $k \in \mathbb{N}$ we define

$$W_p^k(G) := \{\, u \in L^p(G) \mid \forall\, |\alpha| \leqslant k : \; D^\alpha u \in L^p(G) \,\},$$
$$W_\infty^k(G) := \{\, u \in L^\infty(G) \mid \forall\, |\alpha| \leqslant k : \; D^\alpha u \in L^\infty(G) \,\},$$

which we endow with the norms

$$\|u\|_{W_p^k} := \left(\sum_{|\alpha| \leqslant k} \|D^\alpha u\|_{L^p}^p \right)^{1/p}, \quad \|u\|_{W_\infty^k} := \max_{|\alpha| \leqslant k} \|D^\alpha u\|_{L^\infty}.$$

Note that for $1 \leqslant p \leqslant \infty$ $K_p^k(G)$ is a Banach space. In the case $p = 2$ the Sobolev space is even a Hilbert space. Due to its special importance, a special notation is usually used.

Definition D.5. We set $H^k(G) := W_2^k(G)$, which is endowed with the inner product

$$\langle u, v \rangle_{H^k} := \sum_{|\alpha| \leqslant k} \langle D^\alpha u, D^\alpha v \rangle_{L^2}, \quad u, v \in H^k(G).$$

We further define $H_0^k(G)$ as the *closure* of $C_0^\infty(G) \subset H^k(G)$ w.r.t. the topology of $H^k(G)$. It is a Hilbert space with $\langle \cdot, \cdot \rangle_{H^k}$.

Note that intuitively $u \in H_0^k(G)$ if $u \in H^k(G)$ and u vanishes on the boundary ∂G. Recall, however, from Example D.3 that $u \in H^k(G)$ does not, in general, imply that u is continuous. Hence, the notion of "evaluating u at ∂G" may not be well defined. We note that there are ways – involving the *trace* – of defining such point evaluations in the context of $H_0^k(G)$-spaces rigorously.

Remark D.6. The above paragraph touches on the question when a Sobolev space "contains" only continuous functions. More precisely, there is a big literature devoted to the question when a Sobolev space $W_p^k(G)$ can be continuously embedded into spaces such as $L^q(G)$, $W_q^m(G)$, $C^{n,\alpha}(G)$ – the latter understood in the sense of C^n-functions with α-Hölder nth derivatives.

Remark D.7. Recall that the Fourier transform of the derivative of a function u can be expressed by multiplying the Fourier transform of u by a polynomial in the Fourier variable ξ. Hence, the above Sobolev norms can be expressed in terms of integrals of $(1 + |\xi|^2)^{k/2} \widehat{u}(\xi)$, where $\widehat{u}$ denotes the Fourier transform of u. Clearly, this definitions can be extended to non-integer $k \in \mathbb{R}$, leading to defintions for spaces $W_p^k(G)$, $k \in \mathbb{R}$. In addition, Sobolev spaces with negative index $k < 0$ often allow an interpretation as *dual spaces* of $W_p^{|k|}(G)$.

Bibliography

[1] S. Asmussen and J. Rosiński. Approximations of small jumps of Lévy processes with a view towards simulation. *J. Appl. Probab.*, 38(2):482–493, 2001.

[2] V. Bally and D. 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. doi: 10.1007/BF01303802.

[3] O. E. Barndorff-Nielsen. Processes of normal inverse Gaussian type. *Finance Stoch.*, 2:41–68, 1998.

[4] L. Breiman. *Probability.* Addison-Wesley Publishing Company, 1968.

[5] M. Broadie and P. Glasserman. Pricing American-style securities using simulation. *J. Econom. Dynam. Control*, 21(8-9):1323–1352, 1997. Computational financial modelling.

[6] R. 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.

[7] P. Carr and D. Madan. Option valuation using the Fast Fourier Transform. *Journal of Computational Finance*, 2(4):61–73, 1999.

[8] P. Carr, H. Geman, D. B. Madan, and M. Yor. The fine structure of asset returns: An empirical investigation. *Journal of Business*, 75(2):305–332, 2002.

[9] R. Cont and P. Tankov. *Financial Modelling with Jump Processes.* Chapman & Hall/CRC, 2004.

[10] R. Cont and E. Voltchkova. A finite difference scheme for option pricing in jump diffusion and exponential Lévy models. *SIAM J. Numer. Anal.*, 43:1596–1626, 2005.

[11] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comp.*, 19:297–301, 1965.

[12] A. Deitmar. *A First Course in Harmonic Analysis.* Springer, 2nd edition, 2005.

[13] L. Devroye. *Nonuniform Random Variate Generation.* Springer, 1986. Available online from http://cg.scs.carleton.ca/~luc/rnbookindex.html.

[14] D. Duffie, D. Filipović, and W. Schachermayer. Affine processes and applications in finance. *Ann. Appl. Probab.*, 13(3):984–1053, 2003.

[15] E. Eberlein and K. Prause. The generalized hyperbolic model: financial derivatives and risk measures. In H. Geman, D. Madan, S. Pliska, and T. Vorst, editors, *Mathematical Finance – Bachelier Congress 2000*, pages 245–267. Springer, 2002.

[16] E. Eberlein, K. Glau, and A. Papapantoleon. Analysis of Fourier transform valuation formulas and applications. *Appl. Math. Finance*, 17:211–240, 2010.

[17] J. Elstrodt. *Maß- und Integrationstheorie.* Springer, 2nd edition, 1999.

[18] 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.

[19] L. Feng and V. Linetsky. Pricing discretely monitored barrier options and defaultable bonds in Lévy process models: a fast Hilbert transform approach. *Math. Finance*, 18:337–384, 2008.

[20] D. Filipović. *Term-structure models*. Springer, 2009. A graduate course.

[21] M. B. Giles. Multilevel Monte Carlo path simulation. *Oper. Res.*, 56(3):607–617, 2008.

[22] M. B. Giles, D. J. Higham, and X. Mao. Analysing multi-level Monte Carlo for options with non-globally Lipschitz payoff. *Finance Stoch.*, 13(3):403–413, 2009.

[23] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, 2004.

[24] D. Grebenkov, D. Belyaev, and P. W. Jones. A multiscale guide to brownian motion. *Journal of Physics A*, 49(4):043001, 2016.

[25] S. 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.

[26] F. Hubalek and J. Kallsen. Variance-optimal hedging and Markowitz-efficient portfolios for multivariate processes with stationary independent increments with and without constraints. Working paper, TU München, 2005.

[27] T. R. Hurd and Z. Zhou. A Fourier transform method for spread option pricing. *SIAM J. Financial Math.*, 1:142–157, 2010.

[28] C. Kahl and R. Lord. Fourier inversion methods in finance. Preprint, `rogerlord.com`, 2010.

[29] J. Kallsen. A didactic note on affine stochastic volatility models. In *From stochastic calculus to mathematical finance*, pages 343–368. Springer, Berlin, 2006.

[30] I. Karatzas and S. E. Shreve. *Brownian Motion and Stochastic Calculus*. Springer, 1991.

[31] P. E. Kloeden and E. Platen. *Numerical solution of stochastic differential equations*, volume 23 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin, 1992. ISBN 3-540-54062-8.

[32] D. E. Knuth. *The Art of Computer Programming. Vol. 2*. Addison-Wesley, second edition, 1981.

[33] S. Kusuoka and D. Stroock. Applications of the Malliavin calculus, part II. *J. Fac. Sci. Univ. Tokyo*, 32:1–76, 1985.

[34] P. L'Ecuyer. Uniform random number generation. *Ann. Oper. Res.*, 53:77–120, 1994.

[35] P. L'Ecuyer. Quasi-Monte Carlo methods with applications in finance. *Finance Stoch.*, 13: 307–349, 2009.

[36] P. L'Ecuyer, B. Oreshkin, and R. Simard. Random numbers for prallel computers: Requirements and methods. Preprint, 2014.

[37] P. Lévy. *Processus stochastiques et mouvement brownien*. Suivi d'une note de M. Loève. Deuxième édition revue et augmentée. Gauthier-Villars & Cie, Paris, 1965.

[38] A. L. Lewis. A simple option formula for general jump-diffusion and other exponential Lévy processes. Working paper, 2001.

[39] F. Longstaff and E. Schwartz. Valuing American options by simulation: a simple least-squares approach. *Review of Financial Studies*, 14:113–147, 2001.

[40] D. B. Madan and E. Seneta. The variance gamma (VG) model for share market returns. *J. Business*, 63:511–524, 1990.

[41] G. Marsaglia and W. W. Tsang. The Ziggurat method for generating random variables. *Journal of Statistical Software*, 5(8), 2000.

[42] A.-M. Matache, P.-A. Nitsche, and C. Schwab. Wavelet Galerkin pricing of American options on Lévy driven assets. *Quant. Finance*, 5:403–424, 2005.

[43] N. Metropolis. The beginning of the Monte Carlo method. *Los Alamos Sci.*, 15, Special Issue: 125–130, 1987.

[44] D. Nualart. *The Malliavin calculus and related topics. 2nd ed.* Probability and Its Applications. Berlin: Springer. xiv, 382 p., 2006.

[45] NVIDIA. CUDA. http://www.nvidia.com/object/cuda_home_new.html. Accessed on August 24, 2014.

[46] B. Øksendal. *Stochastic Differential Equations.* Springer, 6th edition, 2003.

[47] P. E. Protter. *Stochastic Integration and Differential Equations.* Springer, 2nd edition, 2005.

[48] S. Raible. *Lévy processes in finance: Theory, numerics, and empirical facts.* PhD thesis, Univ. Freiburg, 2000.

[49] B. D. Ripley. *Stochastic Simulation.* John Wiley & Sons, 1987.

[50] W. Rudin. *Real and Complex Analysis.* McGraw-Hill, 3rd edition, 1987.

[51] K. Sato. *Lévy Processes and Infinitely Divisible Distributions.* Cambridge University Press, 1999.

[52] F. Sauvigny. *Partial Differential Equations 2.* Springer, 2006.

[53] M. Schmelzle. Option pricing formulae using Fourier transform: Theory and application. Preprint, `http://pfadintegral.com`, 2010.

[54] W. Schoutens. The Meixner process: Theory and applications in finance. In O. E. Barndorff-Nielsen, editor, *Mini-proceedings of the 2nd MaPhySto Conference on Lévy Processes*, pages 237–241, 2002.

[55] R. Seydel. *Tools for Computational Finance.* Springer, 4th edition, 2009.

[56] D. Talay and L. Tubaro. Expansion of the global error for numerical schemes solving stochastic differential equations. *Stochastic Anal. Appl.*, 8(4):483–509 (1991), 1990. ISSN 0736-2994.

[57] D. B. Thomas, W. Luk, P. H. W. Leong, and J. D. Villasenor. Gaussian random number generators. *ACM Comput. Surv.*, 39(4), 2007.

[58] Wikipedia. Harmonic analysis — wikipedia, the free encyclopedia, 2010. [Online; accessed 9-July-2010].

[59] Wikipedia. Linear congruential generator — wikipedia, the free encyclopedia, 2010. [Online; accessed 22-March-2010].

[60] P. Wilmott. *Paul Wilmott on Quantitative Finance. 3 Vols.* John Wiley & Sons, 2nd edition, 2006.

[61] R. J. Zimmer. *Essential Results of Functional Analysis.* University of Chicago Press, 1990.