Universität Duisburg-Essen

Fachbereich Mathematik

Multilevel Approach for Bermudan Option Pricing

Dissertation zur Erlangung des Doktorgrades Dr.rer.nat. der Fakultät für Mathematik

Betreuer:	Prof. Dr. D. Belomestny
Vorgelegt von:	Fabian Dickmann aus Bottrop
Eingereicht am:	17. Dezember 2014
Mündliche Prüfung:	24. Juni 2015
Erstgutachter:	Prof. Dr. Denis Belomestny
Zweitgutachter:	PD Dr. John Schoenmakers
Zweitgutachter:	Prof. Dr. Mikhail Urusov

Contents

Chapter 1

Introduction

The pricing of American and Bermudan style financial options has been a challenge for decades. Put simply, the holder of such an option has the right to exercise it once within a finite time horizon T. The expression "to exercise" means that the holder can get the payoff $g_t(X_t)$ immediately, where g_t is the "payoff function" and X_t is a given stochastic process modeling the "underlying asset". The latter could be a stock price, a commodity price, an exchange rate, etc. In the American case, the time of exercise can be chosen deliberately, whereas in the Bermudan case only a finite set of exercise dates is available.

The opportunity to freely choose when to exercise the option is what makes this kind of problem much harder than the pricing of European options. The latter provide the payoff only at the end of the time horizon, which is called "maturity" time and there is no mathematical problem to determine the optimal strategy when to exercise. In the American or Bermudan case, the best strategy that tells the holder when to exercise in order to maximize the expectation of the payoff is a stopping rule in the mathematical sense. It is characterized as the solution of an optimal stopping problem, which is a maximization problem over the set of all stopping times.

Solving such problems numerically is straightforward in low dimensions. There are a lot of fast and accurate methods to do so, both stochastic and deterministic ones. The binomial tree algorithm of Cox, Ross and Rubinstein [26] belongs to the class of deterministic algorithms and is widely used in practice. Other deterministic approaches make use of partial differential equations. In their books, the authors Bensoussan and Lions [13] and Jaillet, Lamberton and Lapeyre [50] discuss such approaches that are based on variational inequalities.

However when facing problems in higher dimension, namely the valuation of options depending on many assets, these methods become practically impossible. The computational complexity will explode as the number of assets increases, which is known as the "curse of dimensionality". One way out is the use of Monte-Carlo simulations. The advantage of the latter mainly consist of the fact that the complexity of a simple Monte Carlo estimator is of order ε^{-2} irrespectively of the dimension, where ε denotes the desired standard deviation of the estimator. This easy consequence of the central limit theorem motivates the use of Monte Carlo simulation for high dimensional problems in general. In many examples, the challenge consists of evaluating the value of an option at $t = 0$ only. It is intuitively clear that a deterministic approach cannot be competitive if it calculates the value function for all times within a horizon and for all asset values in a domain.

Of course, this simple consideration does not tell us how to calculate the price or how to draw Monte Carlo samples from an estimator of it. There are a lot of ideas about how to use stochastic algorithms to solve optimal stopping problems with different advantages and drawbacks. The methods presented in this work will use one of two procedures to do so. Either they will approximate the optimal stopping rule or they will try to approximate the Snell envelope of the payoff. Those approaches will produce low biased or high biased estimators respectively, so buyers and sellers will both be satisfied. The buyer of an option is typically interested in knowing a lower bound for the option price whereas the seller is interested in higher bounds, as both want to have information about their maximal expected loss when signing a contract.

In Chapter 2, the so-called fast approximation methods are used to estimate lower bounds. They construct stopping times recursively and work backwards in time. Afterwards, those stopping times have to be tested via Monte Carlo simulation. Chapter 3 is about the multilevel technique that is used to reduce the variance of these Monte Carlo simulations. Usually, variance reduction techniques are known for reducing the variance of a Monte Carlo simulation by a constant factor, e.g. control variates, importance sampling, stratified sampling or antithetic sampling (systematic sampling). In contrast, the multilevel technique will lead to a lower order of complexity of the simulation. The complexity will be reduced from ε^{-3} to $\varepsilon^{-2.5}$ in the usual case of a mesh approximation and a good-natured problem, i.e. we have a gain of $\varepsilon^{-0.5}$. Here, ε denotes the desired precision of the calculation measured in terms of the root-mean-squared error. Interestingly enough, depending on the fast approximation method in use the gain of complexity can be up to ε^{-1} or the multilevel technique can even become completely useless in some special cases. The qualitative results about the order of complexity should be complemented by some quantitative results in Section 3.2. The efficiency in case of a fixed number of levels is examined there and it turns out that the multilevel version of the fast approximation methods is not only recommended for very precise calculations because of the better order, but also leads to much better results in simple settings. Furthermore, it turns out that the quantity of the gain can be improved significantly by kind of a conditional nesting procedure called "Nested Conditional Monte Carlo", which will be discussed in Chapter 4.

Chapter 5 is about higher biased estimators, namely convex optimization methods and in particular the well known Andersen Broadie approach that belongs to the class of nested dual methods. The convergence behaviour of nested dual methods will be examined in detail and the complexity will turn out to be of order ε^{-4} in general, but can be reduced to $\varepsilon^{-3-\delta}$ under several assumptions with arbitrarily small $\delta > 0$ in the good-natured cases. The multilevel approach in Chapter 6 will reduce this complexity to ε^{-2} without those assumptions. Thus, there are two big advantages at the same time. In contrast to the multilevel approach for fast approximation methods, the multilevel approach will only be worthwhile here for very precise calculations since the standard dual methods also work quite well without any enhancement.

To be mathematically precise, suppose a stochastic process X_t, $t \in [0,T]$ defined on $(\Omega, \mathcal{F}, \mathrm{P})$ that takes values in $\mathbb{R}^D$, which will be used as the "state space" in the following. It models the price of a couple of financial assets and

$(\mathcal{F}_t)_{0\leq t\leq T}$ is a filtration that X_t is adapted to. This means the random variable X_t is $\mathcal{F}_t$-measurable and $\mathcal{F}_s \subset \mathcal{F}_t \subset \mathcal{F}$ for all $0 \leq s \leq t \leq T$. The usual benchmark examples are about assets modeled by geometric Brownian motion under the risk-neutral measure, i.e.

$$dX_t^d = (r-\delta)X_t^d dt + \sigma X_t^d dW_t^d, \quad i = 1, \ldots, D, \tag{1.1}$$

where r denotes the "interest rate", σ the "volatility", δ the "dividend yield" and x_0 the "spot price", which is the inital value of the process. In this case, it is possible to use the Brownian filtration $\mathcal{F}^W$ generated by the D-dimensional Brownian motion W. Unless stated differently, a Markov process with transition density will be assumed in the following.

An "American option" is a financial derivative that gives the holder the right to receive the (discounted) "payoff" $g_t(X_t)$ once in $[0,T]$, where $g_t : \mathbb{R}^D \to \mathbb{R}$. This will be called "exercising" the option and the holder can freely choose when to do that. Thus, the value of the option depends on the behaviour of the owner. In order to model the latter, let us recall the definition of an $\mathcal{F}$-stopping time. It is a mapping $\Omega \to [0,T]$, such that

$$\{\tau \leq t\} \in \mathcal{F}_t \tag{1.2}$$

is fulfilled for all $t \in [0,T]$. A well known result states that the fair price of the option is now given by

$$V_0^{AM} = \sup_{\tau\in\mathcal{T}} \mathrm{E}\left[g_\tau(X_\tau)|X_0 = x_0\right], \tag{1.3}$$

which is quite an intuitive representation. Here, $\mathcal{T}$ is the set of all $\mathcal{F}$-stopping times taking values in $[0,T]$. Using that representation, it seems that in order to evaluate the true value it is necessary to find the optimal exercise policy first. Since the set $[0,T]$ in not countable, the set of all stopping times in question seems to be unmanageable. Thus, several types of discretization will be necessary for a numerical approach.

As explained before, a so-called "Bermudan option" is an option that can only be exercised at a finite set of "exercise dates" $0 = t_0 < t_1 < \ldots < t_{\mathcal{J}} = T$. This work is about that case exclusively. A Bermudan option with a large number of exercise dates $\mathcal{J}$ can be seen as a natural approximation of an American option. Furthermore, Bermudan options are also interesting by themselves, since they are popular among traders. For the sake of notation, we will write X_j instead of X_{t_j}, V_j instead of V_{t_j}, and so on except for Section 1.2. Analogously to the American case, the fair price of a Bermudan option is given by

$$V_j(x) = \sup_{\tau\in\mathcal{T}_j} E\left[g_\tau(X_\tau)|X_j = x\right], \tag{1.4}$$

where

$$\mathcal{T}_j = \left\{\tau\ \mathcal{F}\text{-stopping time } \middle| \tau(\omega) \in \{j, \ldots, \mathcal{J}\} \text{ for all } \omega \in \Omega\right\} \tag{1.5}$$

is the set of stopping times that has to be searched for the supremum in (1.4).

Definition 1. *The optimization problem* (1.4) *is called the "primal representation" of the optimal stopping problem. The solution* $V_j(\cdot)$ *will be called the "true value function" and we also define the "true value process"* $V_j = V_j(X_j)$ *with a small abuse of notation.*

The stopping time that realizes the supremum in (1.4) for V_j will be called "optimal stopping time" τ_j^*. It will be unique almost surely under the assumptions imposed here, see (1.16). We also write $\tau^* := \tau_0^*$, which will fulfill $\tau_0^* = \tau_1^* > 0$ in non-degenerate examples. For some exotic options, discarding the right to exercise might be optimal due to a negative payoff. Since a non-negative payoff will also be assumed in the following, we can state that $\tau_{\mathcal{J}}^* = \mathcal{J}$. The family of optimal stopping times $\tau_0^*, \ldots, \tau_{\mathcal{J}}^*$ is a "consistent family" in the sense of the following definition.

Definition 2. *A family of $\mathcal{F}$-stopping times $\tau_0, \ldots, \tau_{\mathcal{J}}$ is called consistent if*

$$j \leq \tau_j \leq \mathcal{J}, \quad \textit{and} \quad \tau_j > j \quad \Longrightarrow \quad \tau_j = \tau_{j+1}. \tag{1.6}$$

for all $j = 0, \ldots, \mathcal{J}$.

For example a familiy of stopping times

$$\tau_j = \min\left\{ i \in \{j, \ldots, \mathcal{J}\} \,\Big|\, (j, X_j) \in \mathcal{S} \right\} \tag{1.7}$$

will be consistent, where $\mathcal{S} \subset \mathbb{N} \times \mathbb{R}^D$ is some (deterministic) set.

In the following, the discounted payoff will be given by

$$g_j(x) = e^{-rt_j} \widetilde{g}(x), \tag{1.8}$$

where the undiscounted payoff $\widetilde{g} : \mathbb{R}^D \to \mathbb{R}^{\geq 0}$ will be one of the following functions. In case of one asset, the payoff function $\widetilde{g}(x) := (x - \varkappa)^+$ defines a "call option" and $\varkappa$ denotes the "strike price". Furthermore, $\widetilde{g}(x) := (\varkappa - x)^+$ defines the "put option". The call option is the one-dimensional special case of the "max-call option" with payoff

$$\widetilde{g}(x) = \left(\max(x^1, \ldots, x^D) - \varkappa\right)^+. \tag{1.9}$$

The practical interpretation is easy. A max-call option gives the holder the right to buy the currently most expensive asset for a fixed price of $\varkappa$ that has been preassigned at time t_0. Analogously, the "min-put option" is defined by the payoff

$$\widetilde{g}(x) = \left(\varkappa - \min\left(x^1, \ldots, x^D\right)\right)^+ \tag{1.10}$$

and gives the holder the right to sell the lowest of all assets. These two payoffs are common examples for options which are path-independent. In general, the payoff may depend on all components of $(X_0, \ldots, X_j)$ up to the current time t_j. The most important of these are arithmetic and geometric Asian options. This kind of dependence also occurs in case of the LIBOR model that is used as an example in Section 6.5. Unless stated differently, the payoff is assumed to be path-independent and non-negative.

1.1 Dynamic Programming Principle

As mentioned before, we restrict ourselves to the Bermudan case, i.e. there is a finite set of exercise dates $0 = t_0 < \ldots < t_{\mathcal{J}} = T$. It is well known that the problem

$$V_j(x) = \sup_{\tau \in \mathcal{T}_j} \mathrm{E}\left[g_\tau(X_\tau) | X_j = x\right] \tag{1.11}$$

belongs to the class of optimization problems that can be solved via Bellman's principle of optimality [1]. In case of optimal stopping, this means that the problem should be solved backwards in time, iteratively from $t_{\mathcal{J}}$ to t_0. Since we assumed the payoff to be path-independent and the underlying to have the Markov property, we can infer the following dynamic recursion in this form [2].

We reformulate (1.11) and obtain

$$\begin{aligned} V_j(x) &= \max\Big(g_t(x), \sup_{\tau \in \mathcal{T}_{j+1}} \mathrm{E}\left[g_\tau(X_\tau)|X_j = x\right]\Big) && (1.12)\\ &= \max\Big(g_t(x), \mathrm{E}[V_{j+1}(X_{j+1})|X_j = x]\Big) && (1.13)\end{aligned}$$

because $\mathcal{T}_{j+1} \setminus \mathcal{T}_j$ is the set of all stopping times that equal j with probability one. This motivates the following definition, which assumes that the payoff is not path-dependent.

Definition 3. *For each time step t_j, $0 \leq j < \mathcal{J} - 1$, there is a "continuation value function" $C_j : \mathbb{R}^D \to \mathbb{R}$ defined via*

$$C_j(x) = \mathrm{E}\left[V_{j+1}(X_{j+1})|X_j = x\right] = \sup_{\tau \in \mathcal{T}_{j+1}} \mathrm{E}\left[V_\tau(X_\tau)\Big|X_j = x\right]. \qquad (1.14)$$

It represents the value of the option if the holder decides not to exercise it at time step t_j.

With this definition at hand, we have the following easy expression

$$V_j(x) = \max\left(g_j(x), C_j(x)\right), j = 0, \ldots, \mathcal{J} - 1. \qquad (1.15)$$

It is clear that holding the option at the last exercise date is useless if the payoff is non-negative, so we have $V_{\mathcal{J}}(x) = g_{\mathcal{J}}(x)$ and we define $C_{\mathcal{J}} \equiv -\infty$, which is consistent with (1.15). The optimal stopping time can now be written as

$$\tau_j^* = \min\left\{i \in \{j, \ldots, \mathcal{J}\} : g_i(X_i) \geq C_i(X_i)\right\}. \qquad (1.16)$$

So to say, the optimal stopping time is "induced by" the continuation value functions, which motivates the following definitions.

Definition 4. *The "continuation region" $\mathcal{C}$ and the "exercise region" $\mathcal{E}$ are given by*

$$\begin{aligned} \mathcal{E} &= \left\{(j, x) : g_j(X_j) \geq C_j(X_j)\right\}, && (1.17)\\ \mathcal{C} &= \left\{(j, x) : g_j(X_j) < C_j(X_j)\right\}, && (1.18)\end{aligned}$$

which leads to the easy expression $\tau_j^* = \min\{i \in \{j, \ldots, \mathcal{J}\}|(i, X_i) \in \mathcal{E}\}$.

Determining these regions is not easier than solving the original stopping problem itself. To characterize the solution theoretically, we have the following definition.

[1] Richard Bellman, 1957: "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."

[2] For a path-dependent payoff or an underlying that does not have the Markov property, an enlargement of the state space will be necessary.

Definition 5. *The "dynamic recursion" that theoretically provides a solution to the primal problem is given by*

$$C_{\mathcal{J}}(x) = -\infty, \tag{1.19}$$

$$C_j(x) = \mathrm{E}\left[\max\left(g_{j+1}(X_{j+1}), C_{j+1}\left(X_{j+1}\right)\right) | X_j = x\right], \tag{1.20}$$

which is just a reformulation of (1.15) *in terms of continuation values.*

Equation (1.20) is very unlikely to be solvable analytically. Even in case of $j = \mathcal{J} - 1$, the expectation will have to be evaluated numerically with respect to the distribution of the underlying process X. The key question is how to find a function that approximates (1.20) as exact and efficient as possible. It is also possible to formulate a dynamic recursion in terms of the true value function instead of the continuation values. But unfortunately, the maximum-function will lead to a kink in the solution, so the approximation would be more difficult. Of course, this difference will vanish in the American case because of the smooth-fit property. The latter is a typical property of American stopping problems and can be states as

$$\frac{\partial V}{\partial x}|_{\partial\mathcal{C}} = \frac{\partial g}{\partial x}|_{\partial\mathcal{C}} \tag{1.21}$$

under the assumption that X is a diffusion and the continuation value has some Lipschitz property, see Peskir and Shiryaev [62].

1.2 Continuation Regions of the American Max-Call Option

This is the only section about the continuous case, i.e. the American stopping problem as given in (1.3). We want to emphasize some basic properties of the continuation region and the exercise region of the American max-call option, which is the common benchmark example. It is useful to have a good perception of the shape of these regions, e.g. for finding suitable basis functions for regression methods in Section 2.1 or for the convex optimization technique in Section 5.4. Of course, their shape will be very similar in the Bermundan case. We mainly follow the description and notation from Broadie and Detemple [17].

Since the max-call payoff

$$g_t(x) = e^{-rt}\left(\max(x_1, \ldots, x_D) - \varkappa\right)^+ \tag{1.22}$$

is not path-dependent, we have again the simple representation

$$\tau^* = \inf\{t \in [0, T] : (t, X_t) \in \mathcal{E}\} \tag{1.23}$$

provided that X_t is a Markov process. The exercise region is a subset of $[0, T] \times \mathbb{R}^D$ and we also define the t-sections of these regions simply by

$$\mathcal{E}(t) = \{x \in \mathbb{R}^D : (t, x) \in \mathcal{E}\}, \quad \mathcal{C}(t) = \{x \in \mathbb{R}^D : (t, x) \in \mathcal{C}\}. \tag{1.24}$$

In case of one single asset, the continuation region is a convex set, as shown in Figure 1.1. Additionally, the t-sections of the exercise region are also convex

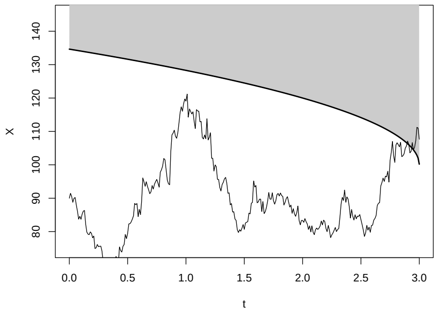

Figure 1.1: The exercise region of an American call option with strike price 100.

because they are one-dimensional intervals $[b(t), \infty[$. Here, $b : \mathbb{R} \to \mathbb{R}$ is a decreasing function, see Peskir and Shiryaev [62]. In the general case of arbitrarily many assets, we consider the "exercise boundary" $\partial\mathcal{E}(t)$ that is defined to be the boundary of the t-section of the exercise region in $\mathbb{R}^D$.

For more than one asset, there are many counterintuitive things to discover. Firstly, the t-sections of the exercise region $\mathcal{E}$ are no longer connected sets. Secondly, it is no longer clear that exercising becomes optimal as $(X_t^1, \ldots, X_t^D)$ grows in all components. The following three propositions are taken from Broadie and Detemple [17] and are valid for an American max-call option on two i.i.d. assets of geometric Brownian motion.

Proposition 6. *If $X_t^1 = X_t^2 > 0$ and $t < T$ then $(t, X_t) \notin \mathcal{E}$. That is, prior to maturity exercise it is not optimal when the prices of the underlying assets are equal.*

As both assets increase equally, the continuation value obviously grows faster than the payoff. It is clear that not only the line where $X_t^1 = X_t^2$ will then belong to the continuation region, but also the vicinity of it. In fact, there is even kind of a divergence of the exercise regions, as stated in the following proposition.

Proposition 7. *Fix $t < T$. There exists λ_1 and λ_2 with $\lambda_2 < 1 < \lambda_1$ such that*

$$\mathcal{E}(t) \cap R(\lambda_1, \lambda_2) = \emptyset, \tag{1.25}$$

where

$$R(\lambda_1, \lambda_2) = \{(X_t^1, X_t^2) \in \mathbb{R}_+^2 : \lambda_2 X_t^1 < X_t^2 < \lambda_1 X_t^1\} \tag{1.26}$$

By symmetry, one can guess that there must be subsets of $\mathcal{E}(t)$ called "exercise subregions", namely $\mathcal{E}_1(t) \subset G_1(t) \cap \mathcal{E}(t)$ and $\mathcal{E}_2(t) \subset G_2(t) \cap \mathcal{E}(t)$. Here we used

$$G_i(t) = \{(t, x) | x_i = \max(x_1, x_2)\}. \tag{1.27}$$

These subregions are connected and convex, as the next proposition shows.

Proposition 8. *Let $(t, X_t), (t, \widetilde{X}_t) \in \mathcal{E}_i(t)$. Then it also holds $(t, X_t(\lambda)) \in \mathcal{E}_i(t)$ for all $0 \leq \lambda \leq 1$, where $X_t(\lambda) = \lambda X_t + (1 - \lambda)\widetilde{X}_t$.*

In summary, these results give us a rough idea about the shape of the exercise region as illustrated in Figure 1.2.

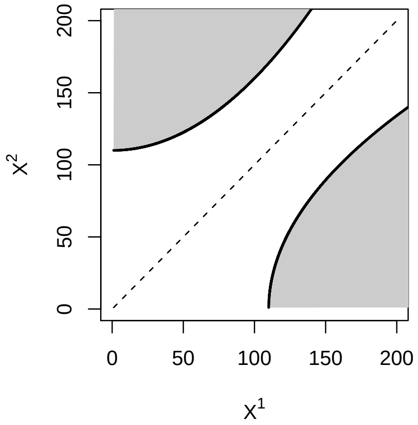

Figure 1.2: The t-section (grey) of the exercise region of an American max-call option on two i.i.d. assets of geometric Brownian motion at $0 \leq t < T$.

1.3 Quasi-Control Variates

The usage of control variates is a well known variance reduction technique for Monte Carlo estimators. Roughly speaking, instead of estimating the expectation of a random variable via some simple Monte Carlo estimator, one will try to estimate the expectation of the difference to another random variable that the expectation is known of. The latter will be called “control variate”, “control variable” or simply “control”.

This procedure will reduce (or sometimes increase) the variance of the Monte Carlo estimator by a constant factor. In other words, their use is equal to the use of a higher number of samples for the Monte Carlo estimator. However, findind efficient controls is often difficult as their expectation has to be known. In order to enlarge the class of variates that can be used, we consider quasi-control variates. This is particularly important with respect to the next chapters, as the use of quasi-controls can be seen as the precursor to the multilevel technique.

Definition 9. *If there is a variate Y that can easily be simulated alongside to X, then the simple Monte Carlo estimator of* $\mathrm{E}[X]$

$$\bar{X} = \frac{1}{N} \sum_{i=1}^{N} X^i \tag{1.28}$$

can be enhanced by using Y as a control variate (CV), so we define a new “controlled estimator”

$$\widehat{X} = \frac{1}{N} \sum_{i=1}^{N} \left(X^i - b(Y^i - E[Y]) \right) \tag{1.29}$$

that depends on a constant $b \in \mathbb{R}$.

It is clear that the variance of the controlled estimator will only be small if X and Y are highly (positively or negatively) correlated and b is chosen carefully. When gathering statistics about real objects, e.g. the opinion of people or the size of animals, Y could be any other dimension of those objects, e.g. the profession of those people of the weight of those animals. In case of a Bermudan option it could be the payoff at maturity or the value of the assets at a fixed time. Another good idea would be the value of the corresponding European option (or a similar option) at the exercise date. Such a control is known as "martingale control" since it corresponds to a control variate of the form $X = M_\tau$ where M is a martingale and τ a stopping time.

A simple calculation taken from Glasserman [40] shows that the variance of the estimator becomes minimal with $b^* = \frac{\operatorname{Cov}[X,Y]}{\operatorname{Var}[X]}$.

Proposition 10. *Using the control variate Y with $b^* = \frac{\operatorname{Cov}[X,Y]}{\operatorname{Var}[X]}$ provides a reduction of variance by a factor of*

$$\operatorname{Var}[\widehat{X}]/\operatorname{Var}[\bar{X}] = 1 - \rho_{XY}^2, \tag{1.30}$$

where ρ_{XY} is the correlation between X and Y. In particular if the computational complexity of a sample from Y is the same as of Y and X simultaneously, then we obtain a speedup by a factor of $\frac{1}{1-\rho_{XY}^2}$.

The slope of the function $\frac{1}{1-\rho_{XY}^2}$ tends to infinity for $\rho_{XY} \to 1$. Consequently, using a control variate will be very fruitful if ρ_{XY} is very near to one and it will be useless in case of $\rho_{XY} \ll 1$. Sadly, in many cases it won't be possible to determine b^* analytically, so one should estimated it via

$$\widehat{b}_n = \frac{\sum_{i=1}^n (X^i - \bar{X})(Y^i - \bar{Y})}{\sum_{i=1}^n (X^i - \bar{X})^2}. \tag{1.31}$$

in a pilot simulation. While n increases, $\widehat{b}_n$ is converging to b^* almost surely.

Remarkably, it is also a fruitful idea to estimate b^* during the Monte Carlo simulation itself instead of running a pilot simulation. In other words, the constant b used in (1.29) is estimated from the same replications as $\widehat{X}$. In this case, there will be a bias of $\mathrm{E}[-\widehat{b}_n(\bar{Y} - \mathrm{E}[Y])]$.

Proposition 11. *The sequence of estimators*

$$\widetilde{X}_N = \frac{1}{N}\sum_{i=1}^N \left(X^i - \widehat{b}_N(Y^i - E[Y])\right) \tag{1.32}$$

with

$$\widehat{b}_N = \frac{\sum_{i=1}^N (X^i - \bar{X})(Y^i - \bar{Y})}{\sum_{i=1}^N (X^i - \bar{X})^2}.$$

is asymptotically unbiased, i.e. $\mathrm{E}[\widetilde{X}_N] \to \mathrm{E}[X]$ *as $N \to \infty$, but each $\widetilde{X}_N$ is biased.*

Remark 12. *Without strict assumptions on the distributions of X and Y, finding bounds for this bias is quite difficult. Glasserman [40] states that "... the bias is typically $O(1/n)$, whereas the standard error is $O(1/\sqrt{n})$", so this aspect can be neglected even for relatively small amounts of samples.*

Lavenberg, Moeller and Welch [55] and Nelson [60] construct confidence intervals in the case that (X, Y) has a multivariate normal distribution. However, this will not be fulfilled if Y is the payoff of a financial option and X some related variate. Nelson [60] proposes the "batching" method to make the problem attackable for a broader class of variables (X, Y) with different distributions. He merges samples to batches, so one has n/k samples of batches of size k, instead of n samples. Following the central limit theorem, their distribution will be nearly normal and thus can be examined further. Nelson [60] and Bauer, Venkatraman and Wilson [5] also analyze the case that the covariance of the samples and the controls are analytically known. Unfortunately, they find that "...it generally produces estimators inferior to the usual method ...", as Glasserman [40] pointed out.

Summing up, Proposition 11 describes the most efficient method to use controls and thus will be used throughout this work. Indeed, no bias from estimating b^* becomes noticeable in the numerical examples presented in the following, so apparently the knowledge of the quantity $\mathrm{E}[X]$ is enough information to make things work out. Equation (1.31) is simply the result of the least squares problem

$$(\alpha, \beta) = \arg \min_{(a,b)\in\mathbb{R}^2} \sum_{i=1}^{N} \left(aY^i + b - X^i\right)^2 .$$

After solving this problem, the estimator can be written as $\widetilde{X} = \alpha + \beta\,\mathrm{E}[Y]$.

Unfortunately, it is very difficult in many cases to find really good control variates, especially because Proposition 10 tells us that the correlation must be very high to achieve good results. Ironically, the expectation of Y is unlikely to be easy to calculate analytically if Y is highly correlated with X and the expectation of X is the quantity in question. This motivates the idea of quasi-control variates.

Definition 13. *The "quasi-controlled" Monte Carlo estimator is given by*

$$\widehat{X}_{N,M} = \frac{1}{N} \sum_{i=1}^{N} \left(X^i - b(Y^i - \bar{Y})\right) \tag{1.33}$$

where $\bar{Y} = \frac{1}{M}\sum_{i=1}^{M} Y^i$ *has been calculated in a different, independent simulation.*

The variance of $\widehat{X}_{N,M}$ is given by

$$\mathrm{Var}[\widehat{X}_{N,M}] = \frac{1}{N}\,\mathrm{Var}[X](1-\rho_{XY}^2) + \frac{1}{M}\,\mathrm{Var}[Y]. \tag{1.34}$$

In order to analyse the efficiency of the quasi-controlled estimator, we compare this variance to the complexity given by

$$\mathrm{comp}(\widehat{X}_{N,M}) = N \times \mathrm{comp}(X) + M \times \mathrm{comp}(Y),$$

where the complexity of Y within the simulation of X is assumed to be negligable. Minimizing (1.34) subject to

$$\mathrm{comp}(X) \times N + \mathrm{comp}(Y) \times M = C$$

with the help of Lagrange multipliers leads to the following solution.

Proposition 14. *To minimize the variance of $\widehat{X}_{N,M}$ given the complexity, it is optimal to choose M and N such that*

$$N \sim \sqrt{\mathrm{Var}[X](1-\rho_{XY}^2)/\operatorname{comp}(X)}, \quad M \sim \sqrt{\mathrm{Var}[Y]/\operatorname{comp}(Y)}. \tag{1.35}$$

A dynamic algorithm could estimate ρ_{XY}, $\mathrm{Var}[X]$, and $\mathrm{Var}[Y]$ during the simulation and decide corresponding to (1.35) whether to add more samples from (X, Y) or from Y. Which means increasing N or M respectively. Furthermore, the algorithm should be aware of those cases that the quasi-controlled estimator is inferior to the simple Monte Carlo estimator. These kind of optimisations are analyzed in Emsermann and Simon [34]. They provide examples to queuing theory, stochastic linear programs and SPDE's. Thereby, they present a precursor of the multilevel technique.

Chapter 2

Lower Bounds via Fast Approximation Methods

Many numerical methods that solve optimal stopping problems rely on the same principle. They estimate continuation values, i.e. they provide functions $\widehat{C}_0, \ldots, \widehat{C}_{\mathcal{J}} : \mathbb{R}^D \to \mathbb{R}$ that are approximations of the true continuation value functions $C_0, \ldots, C_{\mathcal{J}}$. It is clear that the conditions

$$\widehat{C}_0, \ldots, \widehat{C}_{\mathcal{J}-1} \geq 0, \quad \widehat{C}_{\mathcal{J}} \equiv -\infty \tag{2.1}$$

will be optimal for a non-degenerate problem and a non-negative payoff.

There are many ideas about how to construct such estimators. One example are optimization methods that consider a set of estimators (e.g. linear combinations of some basis functions) and try to find the best one of those by simply testing the implied stopping rules, see (2.4). This would lead to simple discretized optimization problem.

However, most of the well-known methods belong to the class that will be called “fast approximation methods” in the following. These are easy and efficient methods defined in the next section that rely on the idea to approximate the dynamic recursion from Definition 5. Given an estimator $\widehat{C}_j$ for some time step t_j, they offer a possibility to construct another estimator $\widehat{C}_{j-1}$ based on $\widehat{C}_j$. Hence, these methods can proceed backwards in time from $t_{\mathcal{J}-1}$ to t_0 and we have

$$\begin{aligned} \widehat{C}_{\mathcal{J}}(x) &\equiv 0 \\ \widehat{C}_j(x) &\approx \mathrm{E}\left[\max\Big(g_{j+1}\left(X_{j+1}\right), \widehat{C}_{j+1}\left(X_{j+1}\right)\Big) \,\Big|\, X_j = x\right] \end{aligned} \tag{2.2}$$

where “≈” just indicates that some estimator was used.

The main problem about this procedure it that the errors are building up from step to step, so especially for Bermudan options with few exercise dates (that are no approximations to American options) these methods will be efficient. Broadie and Glasserman [19] analyse this problem of adding approximation errors in case of a mesh algorithm.

If all estimators have been constructed like in (2.2), one could use

$$\widetilde{V}_0(x_0) = \max\Big(g_0(x_0), \widehat{C}_0(x_0)\Big) \tag{2.3}$$

as an estimator for the true price $V_0(x_0)$. Depending on the method that is used, this estimator could be low or high biased. Every estimate of the continuation values implies a suboptimal stopping rule given by

$$\widehat{\tau} = \min\left\{ j \in \{0, \ldots, \mathcal{J}\} \middle| g_j(X_j) \geq \widehat{C}_j(X_j) \right\}, \tag{2.4}$$

compare (1.16). Since any stopping rule cannot be better than the optimal stopping rule τ^*, one can obtain a low biased estimator for the true price via

$$\widehat{V} = E\left[g_{\widehat{\tau}}(X_{\widehat{\tau}}) \middle| X_0 = x_0 \right]. \tag{2.5}$$

This estimator can be evaluated via Monte Carlo simulation.

To ensure the success of such methods and to make their analysis feasible, we have to impose several assumptions. Two assumptions will be defined in the next subsection, namely assumptions (AC) and (AQ) that are demanded from the algorithms and will be checked for each of the methods presented here.

Additionally, we have to consider an assumption that makes note of some property of the optimal stopping problem rather than the algorithms, the so-called "boundary assumption" taken from Belomestny [6]. It describes the behaviour of the solution of the optimal stopping problem in the vicinity of the exercise boundary.

Assumption (AB). *There exists constants $A > 0, \delta_0 > 0$ and $\alpha > 0$ such that*

$$P\left(|C_j(X_j) - g_j(X_j)| \leq \delta\right) \leq A\delta^\alpha \tag{2.6}$$

for all $j = 0, \ldots, \mathcal{J}$, and all $\delta < \delta_0$.

It turns out that this assumption is the key property of the optimal stopping problem to make the convergence analysis of the following methods possible. In the usual case that $C_j - g_j$ has a non-vanishing derivative in the vicinity of the exercise boundary, we have $\alpha = 1$. This will be called the "usual case" in Chapter 3 and Chapter 5. In fact, α can take all values from 1 to ∞ as Example 15 and Example 16 taken from Belomestny [6] will show.

Example 15. *(Bermudan Power-Put Option) Consider a two-period Bermudan option, i.e. $\mathcal{J} = 1$ with payoff*

$$g_0(x) = g_1(x) = \left(\varkappa^{1/\alpha} - x^{1/\alpha}\right)^+$$

where $\alpha > 0$ is chosen arbitrarily. Given a Black-Scholes model with volatility σ and interest rate $r = 0$, it is possible to show that the continuation value at $t = t_0$ is given by

$$C_0(x) = \varkappa^{1/\alpha}\Phi(-d_2) - x^{1/\alpha}e^{(t_1-t_0)(\alpha^{-1}-1)(\sigma^2/2\alpha)}\Phi(-d_1), \tag{2.7}$$

where

$$d_1 = \frac{\log(x/\varkappa) + (\frac{1}{\alpha} - \frac{1}{2})\sigma^2(t_1 - t_0)}{\sigma\sqrt{t_1 - t_0}}, \tag{2.8}$$

$d_2 = d_1 - \sigma\sqrt{t_1 - t_0}/\alpha$ and Φ is the cumulative normal distribution function. For $x \to 0$ we have

$$|C_0(x) - g_0(x)| \asymp x^{1/\alpha} \tag{2.9}$$

and $C_0(x) > g_0(x)$ for all $x > 0$ if $\alpha \geq 1$. So we have

$$P\left(0 < |C_0(X_0) - g_0(X_0)| \leq \delta\right) \lesssim \delta^{\alpha}, \quad \delta \to 0, \quad \alpha \geq 1 \tag{2.10}$$

But $1 \leq \alpha < \infty$ was chosen arbitrarily.

The next example shows that the case $\alpha = \infty$ may also occur, which is important to note because the fast approximation methods will work particularly well in this case.

Example 16. *Suppose a two-period Bermudan option with payoff g_1 such that $C_0(x) = \mathrm{E}[g_1(X_1)|X_0 = x]$ is positive and monotonically increasing in x, e.g. a simple call option. By setting*

$$g_0 := \begin{cases} C_0(x_0) + \delta_0, & x < x_0, \\ C_1(x_0) - \delta_0, & x \geq x_0 \end{cases} \tag{2.11}$$

one achieves that

$$P_0(0 \leq |C_0(X(0)) - g_0(X(0))| \leq \delta_0) = 0, \tag{2.12}$$

without the exercise region or the continuation region to be degenerate. So to say, assumption (AB) holds with $\alpha = \infty$ here.

2.1 Fast Approximation Methods

We want to define the so-called “fast approximation methods” to be a class of algorithms that fulfill certain assumptions given below. In the succeeding subsections, four examples of such methods are presented, namely “global regression”, “local regression”, the “mesh method” and the “nearest-neighbours technique”. These are popular in practice and have different advantages and drawbacks. Furthermore, the approach of Kohler [52] that uses neural networks to construct continuation estimates is worth mentioning. It can also be seen as a fast approximation method, but will not be discussed here.

In our understanding, the “fast approximation methods” are methods that work by approximating the dynamic recursion (2.2), fulfill assumptions (AC) and (AM) and consist of two steps.

Firstly, the “training step” consists of generating a number of k trajectories, that are i.i.d. copies of the process $X.$ denoted by

$$Z_{\cdot}^{(1)}, \ldots, Z_{\cdot}^{(k)}, \tag{2.13}$$

that will be called “training paths”. The fast approximation methods will use them as information to construct estimators $C_0^k, \ldots, C_{\mathcal{J}}^k$ of the true continuation value functions $C_0, \ldots, C_{\mathcal{J}}$. The number k indicates the quality of the estimation, i.e. the difference between the true continuation value C_j and C_j^k is supposed to decrease, as k increases. Convergence is assumed, as explained in assumption (AQ).

Secondly, a “testing step” is performed. With the estimators form the training step at hand, we have the low biased estimator

$$V^k = \mathrm{E}\left[g_{\tau_k}\left(X_{\tau_k}\right)\middle|X_0 = x_0\right], \tag{2.14}$$

for the true price V_0, where

$$\tau_k = \min\left\{i \in \{0,\dots,\mathcal{J}\}\,\middle|\,C_i^k\left(X_i\right) \geq g_i\left(X_i\right)\right\} \tag{2.15}$$

is the stopping time associated with the continuation estimate. It is defined as a minimum of an $(\Omega, \mathcal{F}_{\mathcal{J}}, \mathrm{P})$-random set, but is an $\mathcal{F}$-stopping time. It is important to note that so far, the stochasticity of the training paths is omitted.

To replace the expectation with its Monte Carlo counterpart, let us generate another set of trajectories, called “testing paths” and the associated stopping times for these paths, i.e.

$$(X_\cdot^{(1)}, \tau_k^{(1)}), \dots, (X_\cdot^{(n)}, \tau_j^{(n)}) \tag{2.16}$$

are i.i.d. copies of the vector process $(X_\cdot, \tau^k)$ (of course independent from $Z_\cdot^{(1)}, \dots, Z_\cdot^{(n)}$). So we finally estimate V_0 via

$$V^{k,n} = \frac{1}{n}\sum_{r=1}^{n} g_{\tau_k^{(r)}}\left(X_{\tau_k^{(r)}}^{(r)}\right), \tag{2.17}$$

where

$$\tau_k^{(r)} = \min\left\{j \in \{0,\dots,\mathcal{J}\}\middle| g_j\left(X_j^{(r)}\right) \geq C_j^k\left(X_j^{(r)}\right)\right\} \tag{2.18}$$

and so we have an unbiased estimator for the lower bound

$$\mathrm{E}\left[g_{\tau_k}\left(X_{\tau_k}\right)\right] \leq V_0. \tag{2.19}$$

Remark 17. *It is also possible to reuse the training paths in* (2.17)*. However, the estimator is no longer low biased then. The continuation estimate will be too well adapted for the set of training paths, so that the estimator* (2.17) *will use information about the future with respect to each single path. This effect is called “overfitting” in statistics. It turns out that in many cases, this effect is much stronger than the suboptimality of the stopping rule, also compare Theorem 19 for the mesh case.*

The first assumption that is imposed here just makes note of the fact that the estimation of the continuation values is based on a number of k training paths.

Assumption (AC). *For any $k \in \mathbb{N}$, the estimates $C_0^k(x), \dots, C_{\mathcal{J}-1}^k(x)$ are defined on some filtered probability space $(\Omega^k, \mathcal{F}^k, P^k)$ independent of $(\Omega, \mathcal{F}, \mathrm{P})$. In fact, $\left(Z_\cdot^{(1)}, \dots, Z_\cdot^{(k)}\right)$ is a random variable under the product measure P^k that is determined via*

$$P^k(A_1 \times \ldots \times A_k) = P(A_1) \times \ldots \times P(A_k)$$

for all $A_i, \dots, A_k \in \mathcal{F}$. The complexity of the estimate $V^{k,n}$ is then is given by

$$\mathscr{C}^{k,n} = k^{1+\kappa_1} + n \times k^{\kappa_2} \tag{2.20}$$

with some constants $\kappa_1, \kappa_2 > 0$.

The first summand in (2.20) is due to the training step, the second one due to the testing step. It will turn out that in all cases discussed here, the complexity can indeed be written like that. In particular, this assumption emphasizes that the continuation estimates $C_1^k(x), \dots, C_{\mathcal{J}}^k(x)$ are stochastic rather than deterministic quantities for each x. To be precise, we want to call

$$V^k = \mathrm{E}\left[V^{n,k} \middle| \sigma\left(Z_\cdot^1, \dots, Z_\cdot^k\right)\right] \tag{2.21}$$

the "testing value", which is now a random variable with respect to the σ-algebra generated by the training paths. The "expected testing value" is denoted simply be $\mathrm{E}[V^k]$ then. Of course, a fast approximation method should provide an expected testing value that is converging to the true value as $k \to \infty$. Therefore, we take the following theorem from Belomestny [6].

Theorem 18. *If there is a sequence of positive real numbers γ_k with $\gamma_k \to 0$ as $k \to \infty$ such that*

$$\mathrm{P}^k\left(\sup_z \left|C_j^k(z) - C_j(z)\right| > \eta\sqrt{\gamma_k}\right) < B_1 e^{-B_2\eta}, \quad \eta > 0 \tag{2.22}$$

for some constants $B_1 > 0$ and $B_2 > 0$ and if boundary assumption (AB) is fulfilled with some α, then we have that

$$0 \le V_0 - \mathrm{E}[V^k] \le B\gamma_k^{(1+\alpha)/2} \tag{2.23}$$

holds with B depending only on α, B_1 and B_2.

In other words, Theorem 18 tells us that the convergence of the expected testing value based on the continuation estimates C_j^k is in some sense faster than the convergence of those continuation estimates themselves: Just compare $\gamma_k^{(1+\alpha)/2}$ to $\gamma_k^{1/2}$ if $\alpha > 0$. This kind of robustness is characteristic for optimal stopping problems and can be seen as motivation for the testing step and the technique of policy iteration, see Remark 35.

Furthermore, the convergence of the continuation estimates is not only slower when considering the order. It is very easy to construct an estimator $\widehat{C}_j$ that implies a stopping rule so that the expected testing value is acceptable, for example very few percent below the true value in Benchmark Example 48. Even a primitive guess, let's say $\widehat{C}_j \equiv c$ for all $j = 0, \dots, \mathcal{J}$ will do a rather good job with a suitable constant $c > 0$. Hence, both for rather rough estimates and for very precise calculations, the testing step is worthwile.

To ensure the convergence behaviour as in Theorem 18, we demand from a fast approximation method to fulfill the following second assumption.

Assumption (AQ). *Let the assumptions of Theorem 18 be fulfilled with $\gamma_k = k^{-\mu}$, $k \in \mathbb{N}$ such that it holds for the bias of the method*

$$V_0 - \mathrm{E}[V^k] \in O\left(k^{-\frac{\mu(1+\alpha)}{2}}\right). \tag{2.24}$$

In the convenient case of $\alpha = \infty$, a stronger statement than Theorem 18 is possible. In Belomestny [6] it is shown that under conditions that are even milder than (2.22), the bias of the testing value $\mathrm{E}[V^k]$ decreases exponentially in the number of training paths k then. Unfortunately, the problems occuring in practice are unlikey to belong to this class of problems.

2.1.1 Stochastic Mesh Method

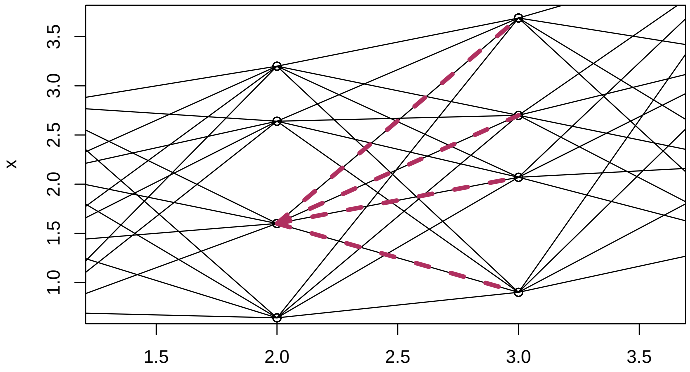

Figure 2.1: Using a stochastic mesh means comparing each node to all nodes of the succeeding time step.

This section is in particular about the approach described by Broadie and Glasserman [19]. Essentially, the idea behind the so called "stochastic mesh" is to fix nodes throughout the state space $\mathbb{R}^D$ for all time steps. Then, all nodes at a time step t_j are compared to to all nodes at t_{j+1} as indicated in Figure 2.1.

Thus, the complexity will depend quadratically on the number of nodes. The term "comparing" means that the dynamic recursion (2.2) is performed with some weighted average of the nodes in the succeeding time step

$$C_j^k(x) = \sum_{i=0}^{k} w_j^{(i)}(x) \max\left(g_{j+1}\left(N_{j+1}^{(i)}\right), C_{j+1}^k\left(N_{j+1}^{(i)}\right)\right), \quad j = 0, \ldots, \mathcal{J} - 1, \tag{2.25}$$

where $N_j^{(1)}, \ldots, N_j^{(k)}$ are the nodes at time t_j and $w_j^{(i)} : \mathbb{R}^D \to \mathbb{R}$ are the so-called "weight functions".

To implement a recursion, we now work backwards through the mesh by "glueing" the values

$$\zeta_j^{(i)} = \max\left(g_j\left(N_j^{(i)}\right), C_j^k\left(N_j^{(i)}\right)\right), \tag{2.26}$$

(where the index k is omitted) to the nodes $N_j^{(i)}$ at time t_j and then move to the previous time step with

$$C_j^k(x) = \sum_{i=0}^{k} w_j^{(i)}(x)\zeta_{j+1}^{(i)}, \quad j = 0, \ldots, \mathcal{J} - 1 \tag{2.27}$$

and so on. Once all the values $\zeta_j^{(i)}$, $i = 0, \ldots, k$, $j = 1, \ldots, \mathcal{J}$ have been estimated, we can proceed to the testing step by inserting the testing paths into (2.27) at the position of x. In general, the weights have to be calculated newly for every such x, but depending on the method there might be some possibilities to save computational time.

There are many ideas about how to choose nodes and weights. Glasserman [40] considers a relatively broad class of algorithms by demanding the following three assumptions. They describe a procedure with Markovian character, where the nodes are perceived as random variables. This is in the sense of assumption (AC).

(M1) The distribution of the random vectors $\mathbf{N}_0, \ldots, \mathbf{N}_{j-1}$ and $\mathbf{N}_{j+1}, \ldots, \mathbf{N}_{\mathcal{J}}$ are independent given $\mathbf{N}_j$ for all $j = 1, \ldots, \mathcal{J} - 1$

(M2) The weight $w_j^{(i)}(x)$ only depends on x and the vectors $\mathbf{N}_j$ and $\mathbf{N}_{j+1}$. This includes the case that it is a function of x and $N_{j+1}^{(i)}$ only.

(M3) We demand for each $h = 1, \ldots, k$ and $j = 0, \ldots, \mathcal{J} - 1$ that

$$\frac{1}{k}\sum_{i=1}^{k} \mathrm{E}\left[w_j^{(i)}\left(N_j^{(h)}\right) V_{j+1}\left(N_{j+1}^{(i)}\right)\middle|\mathbf{N}_j\right] = C_j\left(N_j^{(h)}\right), \tag{2.28}$$

where C_j is the true continuation value function and V_j the true value as defined in (1.4). In particular this means that at each node the estimator would be unbiased if the estimation in the succeeding time step $j+1$ was equal to the true value.

Theorem 19. *Under assumptions (M1)-(M3) the estimator*

$$\widetilde{V}_0 = \max\left(C_0^k\left(x_0\right), g_0\left(x_0\right)\right) \tag{2.29}$$

for V_0 is high biased.

In other words, the values $\zeta_j^{(i)}$ are too well adapted to the set of just these training paths. As explained before, this problem is just the motivation to perform the testing step.

We will use the most natural idea to generate nodes, namely the "independent-path construction", i.e. we simply use the training paths $Z_.^{(1)}, \ldots, Z_.^{(k)}$, which were i.i.d. trajectories of the underlying process X and use them ase nodes, i.e.

$$N_j^{(i)} = Z_j^{(i)}, \quad j = 0, \ldots, \mathcal{J}, \quad i = 1, \ldots, k. \tag{2.30}$$

Actually, this procedure is fulfilling assumption (M1) when the training paths are independent copies of a Markov chain, e.g. when the assets are modeled via geometric Brownian motion.

The independent-path construction can be seen as stratified sampling (see Glasserman [40]) from the density

$$g(j,x) = \frac{1}{k}\sum_{i=1}^{k} p_{j-1,j}\left(Z_{j-1}^{(i)}, x\right), \quad j = 1, \ldots, \mathcal{J}, \tag{2.31}$$

called the "average density" by Broadie and Glasserman [19]. Here, p is the transition density of the underlying process X. First of all, the independent-path construction is a heuristic choice because the distribution coincides with the marginal distribution of X_t at each time step. Secondly, as analysed in Broadie Glasserman [19], the exponential growth of variance that takes place within the mesh as $\mathcal{J} \to \infty$, can be avoided by using this density.

Choosing the weights $w_j^{(i)}(\cdot)$ is a science for itself. There are a lot of ideas, some of which have strong advantages compared to others as explained in the following.

Weights from Likelihood Ratios

In this section, we assume the exact transition density from the process X to be given analytically, i.e. we assume that we can evaluate a function $p : \mathbb{N} \times \mathbb{N} \times \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ such that for all $A \in \mathcal{B}(\mathbb{R}^D)$ from the Borel σ-algebra we have

$$\mathrm{P}(X_j \in A | X_i = x) = \int_A p_{ij}(x,y)dy. \tag{2.32}$$

Of course, this strong assumption will be fulfilled by geometric Brownian motion or CIR processes, but for many models this is not the case. We choose weights according to the likelihood ratio via

$$w_j^{(i)}(x) = \frac{p_{j,j+1}\left(x, Z_{j+1}^{(i)}\right)}{p_{0,j+1}\left(x_0, Z_{j+1}^{(i)}\right)}. \tag{2.33}$$

The complexity of this estimator can already be quite high, e.g. in case of geometric Brownian motion the exponential function has to be evaluated a lot of times. These weights fulfill (M2) and also (M3), since

$$\frac{1}{k}\sum_{i=1}^{k} w_j^{(i)}(x) V_{j+1}\left(N_{j+1}^{(i)}\right) \to \mathrm{E}\left[w_j^{(1)}(x) V_{j+1}\left(N_{j+1}^{(1)}\right)\right], \tag{2.34}$$

as $k \to \infty$ and by inserting our weights we have

$$\begin{aligned}
&\mathrm{E}\left[w_j^{(1)}(x) V_{j+1}\left(N_{j+1}^{(1)}\right)\right], \\
&= \int_{\mathbb{R}^D} \frac{p_{j,j+1}(x,y)}{p_{0,j+1}(x_0,y)} V_{j+1}(y) p_{0,j+1}(x_0,y)dy \\
&= C_j(x).
\end{aligned}$$

If the independent-path construction is not used, one has to insert another density into the denominator of (2.33) as well as into the expectation and obtains the same result. A first idea to improve these weights is to ensure that

$\sum_{i=1}^{k} w_j^{(i)}(x) = 1$, so to say the sum of the "outgoing weights" is normalized to one. This can easily be achieved by defining

$$\widetilde{w}_j^{(i)}(x) := \frac{w_j^{(i)}(x)}{\sum_{l=0}^{k} w_j^{(l)}(x)}, \tag{2.35}$$

so we just divide by the sum of all outgoing weights from a node located at x. This simple procedure does not at all increase complexity, but leads to a reasonable improvement.

However, a much better idea is to ensure that

$$\sum_{l=1}^{k} w_j^{(i)}\left(Z_j^{(l)}\right) = 1, \quad i = 1, \dots, k, \tag{2.36}$$

by dividing through the sum of the "incoming weights" at each node, so we define

$$\overline{w}_j^{(i)}(x) := \frac{p_{j,j+1}\left(x, Z_{j+1}^{(i)}\right)}{\frac{1}{k}\sum_{l=0}^{k} p_{j,j+1}\left(Z_j^l, Z_{j+1}^{(i)}\right)}. \tag{2.37}$$

Here, the denominator is quite costly to evaluate as it has to be calculated separately for every i when evaluating (2.27). Fortunately, it is at least possible to store its values and reuse it within the testing step. It should be emphasized that (2.37) is only easy possible in case of the independent-path construction (that is used in the numerical examples following below), whereas (2.33) is a convenient choice that can be applied without further restrictions.

Furthermore, we want to use a control variate to improve the estimator as explained in Section 1.3. It is called "inner control variate" in contrast to "outer control variates", because it is used within the mesh rather than to reduce the variance of the testing step. Applying such a control will have the same or nearly the same impact as a higher number of mesh nodes k, see Proposition 11.

Finally, the estimator we will use for the numerical examples in this work is given by the following proposition.

Proposition 20. *For each evaluation of $C_j^k(x)$, i.e. for each fixed pair x and j, define*

$$C_j^k(x) = \frac{\frac{1}{k}\sum_{i=1}^{k}\zeta_{j+1}^{(i)}\overline{w}_j^{(i)}(x) - \beta\left(\frac{1}{k}\sum_{i=1}^{k} v_j^{(i)}\overline{w}_j^{(i)}(x) - v_j\frac{1}{k}\sum_{i=1}^{k}\overline{w}_j^{(i)}(x)\right)}{\frac{1}{k}\sum_{i=1}^{k}\overline{w}_j^{(i)}(x)} \tag{2.38}$$

where $v_j^{(i)}$ denotes the control and v_j its expectation[1]*. To determine β, we have to solve the optimization problem*

$$\min_{\alpha,\beta\in\mathbb{R}} \frac{1}{k}\sum_{i=1}^{k}\overline{w}_j^{(i)}(Z_j^{(i)})\left[\zeta_{j+1}^{(i)} - \left(\alpha + \beta v^{(i)}\right)\right]^2 \tag{2.39}$$

and its solution leads to the identity $C_j^k(x) = \alpha + \beta v$.

[1] One has to be careful here, since the expectation has to be calculated with respect to the mesh nodes.

Hence, we have to look for a control $v^{(i)}$ that is correlated to $\zeta_{j+1}^{(i)}$ as strongly as possible. Broadie and Glasserman [19] recommend three kinds of inner controls in case of a max-call option that only depend on $Z_{j+1}^{(i)}$. Among them, the expectation of the highest asset at the next time step and the payoff of the corresponding European option on the highest asset. The latter will be used in the following chapters, i.e.

$$v_j^{(i)}(x) = \exp(-rt_{j+1}) \max_{d=1,\dots,D} (Z_{j+1}^{d,(i)} - \varkappa)^+ \tag{2.40}$$

such that $v_j(x) = \mathcal{E}\left(x, t_j, t_{j+1}\right)$, where

$$\mathcal{E}(x,t,T) := E\left[e^{-rT} \max_{d=1,\dots,D} \left(X_T^d - \varkappa\right)^+ \Big| X_t = x\right]. \tag{2.41}$$

To numerically calculate the estimation of the control (2.41), we recall the following theorem from Belomestny, Bender and Schoenmakers [8].

Theorem 21. *The value of a European max-call option with strike $\varkappa$ and maturity time T on D independent assets modeled via geometric Brownian motion with the same interest rate r, dividend yield δ and volatility σ is given by*

$$\sum_{l=1}^{D} X_0^l \frac{e^{-\delta T}}{\sqrt{2\pi}} \int_{(-\infty, d_-^l]} \exp\left[-\frac{1}{2}z^2\right] \prod_{l' \neq l} \Phi\left(\frac{\ln \frac{X_0^l}{X_0^{l'}}}{\sigma\sqrt{T}} - z + \sigma\sqrt{T}\right) dz \tag{2.42}$$

$$-\varkappa e^{-rT} + \varkappa e^{-rT} \prod_{l=1}^{D} (1 - \Phi(d_-^l)),$$

where

$$d_-^l = \frac{\ln \frac{X_0^l}{\varkappa} + (r - \delta - \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}, \quad d_+^l = d_-^l + \sigma\sqrt{T}. \tag{2.43}$$

and $\Phi : \mathbb{R} \to \mathbb{R}$ denotes the cumulative distribution function of the standard normal distribution.

Since this formula only includes a one-dimensional integral, it is easily approximated numerically. This integration has to be very precise for large values of k, since otherwise the advantage is spoiled. For $D = 1$ we have the Black-Scholes formula, see Remark 50.

Clearly, assumption (AC) is fulfilled with $\kappa_1 = \kappa_2 = 1$, since every node has to be compared to all succeeding nodes so that we have a complexity of $k \times k + n \times k$. To check assumption (AQ), we refer to the work of Avramidis and Matzinger [4] that is exclusively about the convergence of the mesh estimator. Indeed, the numerical examples in Section 3.1.1 and and Section 3.2.1 will show that the mesh estimator provides $\mu = 1$, when using nodes and weights as desribed before.

Nadaraya-Watson Estimator

In contrast to the weights from likelihood ratios, the weights considered in this section can be used without the transition density of the underlying process.

Estimator (2.27) is called a Nadaraya-Watson estimator if the weights are given by

$$w_j^{(i)}(x) = \frac{K\left(\frac{x-Z_j^{(i)}}{\delta}\right)}{\sum_{i=0}^{k} K\left(\frac{x-Z_j^{(i)}}{\delta}\right)} \tag{2.44}$$

with a kernel function $K : \mathbb{R}^D \to \mathbb{R}$ and bandwidth δ.

Choosing the kernel function $K(x) = 1_{\{|x|\leq 1\}}$ is very similar to the nearest neighbors technique presented in Section 2.1.4. One advantage of this similar approach is that sorting is not necessary. A clear drawback lies in the problem that depending on x, the number of neighbours can vary a lot. One also has to cope with the unlikely case that the denominator in (2.44) can equal zero when there are no points in the δ-neighbourhood of x. This method is used in Section 3.1.2, where the whole estimator is set to 0 when the neighbourhood is empty.

Weights from Optimization

Broadie, Glasserman and Ha [20] use an optimization approach to calibrate the weights. They try to balance a couple of heuristics at the same time. For example good weights are likely to provide exact results for problems that the analytical solution is known of. This could simply be the demand that "weights correctly 'price' the underlying asset itself at node i", see [20]. This means one would require $w_j^{(i)}\left(Z_j^{(r)}\right)$ to fulfill

$$Z_j^{(r)} = \frac{1}{k}\sum_{i=1}^{k} w_j^{(i)}\left(Z_j^{(r)}\right) Z_{j+1}^{(i)}, \quad j = 0, \ldots, \mathcal{J}-1. \tag{2.45}$$

At the same time one could try to maximize entropy given by

$$L_0 = -\sum_{i=1}^{N} w_j^{(i)} \log(w_j^{(i)}). \tag{2.46}$$

This would result in the most uniform choice of weights that fulfills (2.45). This approach is very variable as it could be used if the transition density is unknown or in case of a very inaccurate, fast calculation with small k. In the latter case the nearest neighbours technique will be almost useless.

2.1.2 Global Regression

This idea was mainly developed by Tsitsiklis and Van Roy [71] [72] and J. Carriére [23]. The continuation value function $C_j : \mathbb{R}^D \to \mathbb{R}$ will by approximated by regression, usually least squares regression, with some basis functions $\psi^1, \ldots, \psi^m : \mathbb{R}^D \to \mathbb{R}$. We fix a time step t_j and drop the dependence of ψ on j for the sake of notation. Of course, one can use different sets of functions for each t_j. Now, we try to find coefficients $\beta^1, \ldots, \beta^m \in \mathbb{R}$ such that

$$\sum_{h=1}^{m} \beta^h \psi^h(x) \approx \mathrm{E}\left[\max\left(g_{j+1}(X_{j+1}), C_{j+1}(X_{j+1})\right)|X_j = x\right]. \tag{2.47}$$

The drawback of this procedure is that the results strongly depend on the choice of basis functions. Due to the robustness of stopping rules as explained in Section 2.1, it is easy to find basis functions that provide a relatively small bias and the testing step will run very fast when compared to other methods like local regression or the mesh method. However, achieving a really small bias (let's say less than 0.1% of the true value in one of the multidimensional benchmark examples in this work) will be very difficult as a priori knowledge about the true value function must be considered to find really good such functions. Using the method of least squares, the approach reads

$$\min_{\beta\in\mathbb{R}^m} \mathrm{E}\left[\left(\sum_{h=1}^{m}\beta^h\psi^h(X_j) - \max\left(g_{j+1}\left(X_{j+1}\right), C_{j+1}^k\left(X_{j+1}\right)\right)\right)^2 \middle| X_0 = x\right]. \tag{2.48}$$

for each time step. We use again the training paths $Z^{(1)}_., \dots, Z^{(k)}_.$ to approximate this optimization problem and obtain

$$\widehat{\beta} = \operatorname*{arginf}_{\beta\in\mathbb{R}^m} \sum_{i=1}^{k}\left[\zeta_{j+1}^{(i)} - \beta^1\psi^1\left(Z_j^{(i)}\right) - \ldots - \beta^m\psi^m\left(Z_j^{(i)}\right)\right]^2 \tag{2.49}$$

as an estimator for β, where $\zeta_{j+1}^{(i)}$ has been calculated one time step before (just like in the previous subsection). Now we estimate the vector β via $\widehat{\beta} = \widehat{B}^{-1}\widehat{b}$, where

$$\widehat{B}_{pq} = \frac{1}{k}\sum_{i=1}^{k}\psi^p\left(Z_j^{(i)}\right)\psi^q\left(Z_j^{(i)}\right) \tag{2.50}$$

$$\widehat{b}_p = \frac{1}{k}\sum_{i=1}^{k}\psi^p\left(Z_j^{(i)}\right)\zeta_{j+1}^{(i)}, \tag{2.51}$$

for all $p, q = 1, \dots, m$. Of course, $\widehat{B}$ and $\widehat{b}$ are Monte Carlo counterparts of some variables B and p. For B given by

$$B_{pq} = \mathrm{E}\left[\psi^p\left(X_j\right)\psi^q\left(X_j\right)\right] \tag{2.52}$$

it may be possible to calculate the entries analytically.

Remark 22. *For some basis functions, the entries of the matrix B are available without simulation, i.e.* $\mathrm{E}[\psi_p(X_j)\psi_q(X_j)]$ *is analytically known for all $p, q = 1, \dots, m$. This case, that no Monte Carlo simulation is needed, is called "quasi regression" by some authors. The Hermite polynomials together with underlyings of Brownian Motion are a typical example. The Hermite polynomials are defined as follows, thereby the letter "e" denotes that they are a rescaled version of what are called the "physicists" Hermite polynomials.*

Definition 23. *The "Hermite polynomials" are given by*

$$H_{e_n(x)} = \sum_{i=0}^{\lfloor n/2\rfloor}\frac{(-1)^i n! x^{n-2i}}{(n-2i)!i!2^i}, \quad n = 0, 1, \dots \tag{2.53}$$

and fulfill the orthogonality relation

$$\mathrm{E}\left[H_{e_i}(W_1)H_{e_j}(W_1)\right]dx = \begin{cases} 0, & i \neq j \\ i!, & i = j \end{cases}, \tag{2.54}$$

where W_t is a standard Brownian motion.

Finally we use

$$C_j^k(x) := \sum_{h=1}^{m} \widehat{\beta}^h \psi^h(x) \tag{2.55}$$

as an estimator for C_j. We then glue the values $\zeta_j^{(i)} = C_j^k(Z_j^{(i)})$ to all the training paths $i = 1, \ldots, m$ at time step j and use those in the next time step according to the dynamic recursion (2.2).

We can fulfill the conditions from the beginning of Chapter 2 and have a small improvement by using

$$C_j^k(x) := \left(\sum_{h=1}^{m} \widehat{\beta}^h \psi^h(x)\right)^+ \tag{2.56}$$

instead of (2.55), because the continuation value cannot by negative by definition.

A possibility to improve this method of Tsitsiklis and Van Roy [72] is the use of some "interleaving estimator". The idea is to generate a subsample $\widetilde{Z}_j^{(i)}$ from the underlying process starting from $Z_j^{(i)}$ at t_j and evaluate the stopping time

$$\widetilde{\tau}^{(i)} = \min\left\{l \in \{j, j+1, \ldots, \mathcal{J}\} : g_l\left(\widetilde{Z}_l^{(i)}\right) \geq C_l^k\left(\widetilde{Z}_l^{(i)}\right)\right\} \tag{2.57}$$

in order to use $\zeta_j^{(i)} = g_{\widetilde{\tau}^{(i)}}(\widetilde{Z}_{\widetilde{\tau}^{(i)}}^{(i)})$. Longstaff and Schwartz [57] use the training paths themselves in place of $\widetilde{Z}_j^{(i)}$. Actually, both the algorithm of Tsitsiklis and Van Roy and the method of Longstaff and Schwartz can be seen as special cases of a "generalized look-ahead" algorithm with parameter $w = 0$ and $w = \mathcal{J} - j - 1$ respectively, see Egloff [32].

The theoretical assumptions to be asked from the basis functions have been studied intensively in the literature. Usually, the "quality" of the basis functions is measured via the Vapnik-Chervonenkis dimension, see the follwing two definitions taken from Zanger [77].

Definition 24. *The class $\mathcal{G}$ is said to shatter a subset $\{a_1, \ldots, a_n\} \subseteq \Sigma$ if there exists $\mathbf{r} = (r_1, \ldots, r_n) \in \mathbb{R}^n$ such that for every $\mathbf{b} = (b_1, \ldots, b_n) \in \{0,1\}^n$, there is a function $g \in \mathcal{G}$ such that for each i, $g(a_i) > r_i$ if $b_1 = 1$, and $g(a_i) \leq r_i$ if $b_i = 0$.*

Definition 25. *The "Vapnik-Chervonenkis dimension" also-called the "VC-dimension" or "pseudo-dimension"of a class of functions $\mathcal{G}$ is defined as*

$$vc(\mathcal{G}) = \sup\left\{|(a_1, \ldots, a_n\}|\middle|\{a_1, \ldots, a_n\} \textit{ is a subset of } \Sigma \textit{ shattered by } \mathcal{G}\right\},$$

where the notation $|\cdot|$ denotes the cardinality of a set.

The idea behind the VC-dimension originates from statistical learning theory and gives information about the possibility to use a set of functions to approximate a given target function. In case of linear regression, this set of functions will be the set of all linear combinations of the basis functions, which is a finite dimensional vector space. It is well known that for a ν-dimensional vector space $\mathcal{H}$, the Vapnik-Chervonenkis dimension fulfills $vc(\mathcal{H}) \leq \nu + 1$. Polynomial regression uses $\mathcal{H} = \mathcal{R}_D(l)$, where $\mathcal{R}_D(l)$ is the set of all D-dimensional polynomials with order less or equal to l and so $m = \dim(\mathcal{R}_D(l)) = \frac{(l+D)!}{l!D!}$. Clément, Lamberton and Protter [25] analyze the behaviour of the error of the Longstaff and Schwartz method if the basis functions stay fixed and the number of training paths k increases. Tsitsiklis and Van Roy [72] do the same study for their own method.

However, the more interesting questions arises when the number of basis functions m and training paths k grow at the same time. Actually, there are only two types of error. Firstly, the "sample error" that results from the fact that Monte Carlo estimations rather than exact values are used within calculations. Secondly, the "approximation error" that will exist since the linear combinations of basis functions will never fit perfectly for a generic function. The first ones to investigate such questions were Glasserman and Yu [42]. In order to study a worst-case scenario, they consider a single-period setting. Even in this case, the number of training paths k must grow exponentially in the number of basis functions, if one combines Hermite polynomials with Brownian motion. In a more realistic setting considering geometric Brownian motion, the number of training paths is even of order $\exp(m^2)$. Or vice versa, as the authors point out the "...number of polynomials $m(k)$ for which accurate estimation is possible from k paths is ... $O(\sqrt{\log k})$."

Fortunately this result has been improved, in particular by Zanger [77] and Stentoft [69]. In his work, Stentoft [69] uses Legendre polynomials for the Longstaff-Schwartz algorithm and chooses $m \sim k^{1/3}$. He assumes some smoothness conditions for the continuation values and a compact support of the underlying process. However, the strongest result about this issue is from Zanger [77]. With a small abuse of notation, he uses the expression $\mathcal{A}(E[\ldots])$ to denote that an algorithm was used to estimate the expectation of a quantity and $\tau^{\mathcal{A}}$ denotes the associated estimated stopping time. The following result from his article [76] is about the L^2-error, where $L^2(\rho_j)$ denotes the space of square integrable random variables with respect to the probability measure induced by the marginal distribution of X_j.

Lemma 26. *For each $j = 0, \ldots, \mathcal{J} - 1$.*

$$\begin{aligned}
&\left\|\mathcal{A}\left(\mathrm{E}\left[(g_{\tau^{\mathcal{A}}_{j+1}}(X_{\tau^{\mathcal{A}}_{j+1}}))\middle|X_j\right]\right) - \mathrm{E}\left[g_{\tau_{j+1}}\left(X_{\tau_{j+1}}\right)|X_j\right]\right\|_{L^2(\rho_j)} \\
\leq\; & 2\sum_{i=j}^{\mathcal{J}-1}\left\|\mathcal{A}\left(\mathrm{E}\left[g_{\tau^{\mathcal{A}}_{i+1}}\left(X_{\tau^{\mathcal{A}}_{i+1}}\right)\middle|X_i\right]\right) - \mathrm{E}\left[g_{\tau^{\mathcal{A}}_{j+1}}\left(X_{\tau^{\mathcal{A}}_{j+1}}\right)\middle|X_j\right]\right\|_{L^2(\rho_i)}.
\end{aligned}$$

Moreover

$$\begin{aligned}
&\left\|\mathrm{E}\left[g_{\tau^{\mathcal{A}}_{j+1}}\left(X_{\tau^{\mathcal{A}}_{j+1}}\right)\middle|X_j\right] - \mathrm{E}\left[g_{\tau_{j+1}}\left(X_{\tau_{j+1}}\right)|X_j\right]\right\|_{L^2(\rho_j)} \\
\leq\; & 2\sum_{i=j+1}^{\mathcal{J}-1}\left\|\mathcal{A}\Big(\mathrm{E}\Big[g_{\tau^{\mathcal{A}}_{i+1}}\Big(X_{\tau^{\mathcal{A}}_{i+1}})\Big)\Big|X_i\Big]\Big) - \mathrm{E}\left[g_{\tau^{\mathcal{A}}_{i+1}}(X_{\tau^{\mathcal{A}}_{i+1}})|X_i\right]\right\|_{L^2(\rho_i)}
\end{aligned}$$

for all $j = 0, \ldots, \mathcal{J} - 2$.

This lemma gives the possibility to control the sample errors that add up from one time step to the next one and gives rise to the next result, which is Zanger's central theorem about what he calls the "Longstaff-Schwartz" algorithm. The latter includes some truncation procedure and generates a new set of training paths for each regression at each time step. In his notation, $\mathcal{H}_{k,j}$ will be the set of all linear combinations of the m basis functions used at time step j.

Theorem 27. *For each* $j = 1, \ldots, \mathcal{J} - 1$, *suppose* $\mathcal{H}_{k,j}$ *to be an arbitrary subset of* $L^2(\rho_j)$, *that satisfies* $vc(\mathcal{H}_{k,j}) \leq \nu < \infty$ *for some* $\nu \geq 1$. *If the bound*

$$L = \max\left\{1, \|g_1(X_1)\|_{L^\infty(\rho_1)}, \ldots, \|g_{\mathcal{J}}(X_{\mathcal{J}})\|_{L^\infty(\rho_{\mathcal{J}})}\right\} \tag{2.58}$$

exists, then

$$\begin{aligned} \mathrm{E}\left[\left\|\mathcal{A}\left(\mathrm{E}\left[g_{\tau^{\mathcal{A}}_{j+1}}\left(X_{\tau^{\mathcal{A}}_{j+1}}\right)\middle|X_j\right]\right)(X_{jk}) - \mathrm{E}\left[g_{\tau_{j+1}}\left(X_{\tau_{j+1}}\right)|X_j\right]\right\|_{L^2(\rho_j)}\right] \\ \leq 5^{(\mathcal{J}-j)}\Big(\tfrac{2c_1}{\sqrt{k}} + \tfrac{\nu^{1/2}\left(2c_2+2c_3\log^{1/2}(k)\right)}{\sqrt{k}} \\ + \max\nolimits_{i=j,\ldots,\mathcal{J}-1}\left(\inf\nolimits_{f\in\mathcal{H}_{k,i}} \|f - \mathrm{E}\left[g_{\tau_{i+1}}\left(X_{\tau_{i+1}}\right)|X_i\right]\|_{L^2(\rho_i)}\right)\Big) \end{aligned} \tag{2.59}$$

and in particular for the initial time

$$\begin{aligned} \mathrm{E}\left[\left|\mathcal{A}\left(\mathrm{E}\left[g_{\tau^{\mathcal{A}}_0}(X_{\tau^{\mathcal{A}}_0})\right]\right)(X_k) - \mathrm{E}\left[g_{\tau_0}(X_{\tau_0})\right]\right|\right] \\ \leq 5^{\mathcal{J}}\Big(\tfrac{2c_1}{\sqrt{k}} + \tfrac{\nu^{1/2}\left(2c_2+2c_3\log^{1/2}(k)\right)}{\sqrt{k}} \\ + \max\nolimits_{j=1,\ldots,\mathcal{J}-1}\left(\inf\nolimits_{f\in\mathcal{H}_{k,j}} \|f - \mathrm{E}\left[g_{\tau_{j+1}}(X_{\tau_{j+1}})|X_j\right]\|_{L^2(\rho_j)}\right)\Big) \end{aligned} \tag{2.60}$$

This is a strong result, since it provides the order of the sample error via the first two summands in (2.59) and (2.60). The third summand about the approximation error stays unresolved and strongly depends on the exact choice of basis functions.

Given a finite dimensional vector space of dimension $\dim(\mathcal{H}_{k,j}) = \nu - 1$, the sample error in (2.59) and (2.60) converges to zero, as long as $\nu(k) = o(k\log^{-1}(k))$. Zanger points out that Glasserman and Yu [42] only obtain $\nu(k) = O(\log(k))$, so their rate is "significantly less favorable". Furthermore, Zanger states that this result is also stronger than that from Stentoft [69], who provides the rate $\nu(k) \in O(k^{1/3})$ in his Theorem 1. However, Stentoft has no strict assumption on the boundedness of the payoff, so the two results are difficult to compare.

To analyze the case of polynomial regression, Zanger asks the continuation values to fulfill $C_j \in C^n(Q_d(\sigma))$, where $Q_d(\sigma)$ is some cube with length 2σ. This assumption is not too strong. Gerhold [38] even shows that the continuation values will belong to $C^\infty(Q_d(\sigma))$ in the Black-Scholes setting. As a result, the total error of polynomial regression converges to zero at a rate of $O(k^{-\frac{n}{2n+D}})$ as long as the order of polynomials fulfills $l(k) = k^{\frac{1}{D+2n}}$. This is the optimal choice of l and k following Theorem 27, since it leads to the same order both for the sample error and the approximation error in (2.60). Thus, the dimension of the space of polynomials is of order $m = \dim\left(\mathcal{R}_d\left(l(k)\right)\right) \in O\left(k^{\frac{D}{D+2n}}\right)$.

To make the complexity analysis feasible, let us note the following simplified assumption. The complexity will be $m \times m \times k + k \times m \in O(m^2 k)$ within the training step, since the matrix $\widehat{B}$ in the linear equation system must be established involving all training paths and reading the new values $\zeta_j^{(1)}, \dots, \zeta_j^{(k)}$ from the estimation is less complex than that. In the testing step, the complexity will simply be of order $n \times m$, because one only has to evaluate a polynomial.

Assumption . *With a suitable sequence of basis functions $\psi_1, \psi_2, \dots$, it is possible to fulfill assumption (AQ) with $\gamma_k = k^{-\mu}$, as long as*

$$m \in O(k^\rho) \tag{2.61}$$

holds asymptotically with some $\rho > 0$, so assumption (AC) is fulfilled with $\kappa_1 = 2\rho$ and $\kappa_2 = \rho$.

In case of polynomial regression, we thus have $\rho = \frac{D}{D+2n}$ and $\mu = \frac{n}{2n+D}$ which leads to a complexity of

$$k^{2\rho+1} + nk^\rho = k^{\frac{3D+2n}{D+2n}} + nk^{\frac{D}{D+2n}}$$

In practice, simply adding basis functions that could be useful will be fruitful at the beginning. Especially the payoff function or value of a corresponding European option should belong to the set of basis functions. However, adding more such functions because of heuristic arguments will lead to a linear equation system that is bad conditioned or even singular. This is called the "problem of multicollinearity" in econometrics. Bouchard and Warin [14] describe ideas how to avoid this problem. One approach consists of separating the state space into a number of subregions. Afterwards, on each of these subregions one will use a set of basis functions that vanish on all other subregions. This will lead to a sparse equation system that is easy to solve, e.g. with the Cholesky decomposition technique. Another idea makes use of basis functions that are orthogonal, whereas orthogonality is defined with respect to the transition density of the underlying process. Another possibility is regularization. For example, the Tichonov regularization technique just consists of using $\widehat{B} + \lambda \mathrm{Id}$ instead of $\widehat{B}$, where Id is the $m \times m$ identity matrix, see Hofmann [48]. Nonetheless, it is still difficult to guarantee convergence because finding a sequence of functions with increasing Vapnik-Chervonenkis dimension that also leads to reasonable results in practice at the same time is very complicated. Hence, local regression might be preferred in theory to ensure convergence without such problems.

Proposition 28. *Global regression is a special case of a mesh method with suitable choice of weights, since*

$$C_j^k(x) = \sum_{h=1}^m \widehat{\beta}^h \psi^h(x) = \boldsymbol{\psi}(x)^T \widehat{B}^{-1} \widehat{b} = \frac{1}{k} \sum_{i=1}^k \underbrace{\left(\psi(x) B^{-1} \psi(Z_j^{(i)})\right)}_{=:w_j^{(i)}(x)} \zeta_{j+1}^{(i)}. \tag{2.62}$$

2.1.3 Local Regression

The main idea behind local regression is very similar to that behind global regression. The only difference is that for each evaluation of $C_j^k : \mathbb{R}^D \to \mathbb{R}$ at a point x, regression is performed separately. In doing so, the training paths are assigned different weights depending on their distance to x at time t_j. This sort of localization of course leads to a tremendous increase of complexity that is mainly due to setting up the matrix in the equation system (2.51) again and again.

Fix some time step t_j. For each x, we want to use the training paths $Z_\cdot^{(1)}, \ldots, Z_\cdot^{(k)}$ to find basis functions $\psi^1, \ldots, \psi^m$ such that for every $y \in \mathbb{R}^D$

$$\sum_{h=1}^{m} \beta^m \phi^h(y) \approx \mathrm{E}\left[\max\left(g_{j+1}(X_{j+1}), C_{j+1}(X_{j+1})\right) \middle| X_j = x + y\right] \tag{2.63}$$

holds especially for $y \approx 0$ and then use

$$C_j^k(x) := \sum_{h=1}^{m} \beta^m \phi^h(0) \tag{2.64}$$

as an estimator in (2.2).

A convenient choice for the basis functions are again polynomials, so we define the "local polynomial regression estimator" as follows. For some kernel function $K : \mathbb{R}^D \to \mathbb{R}_+$ and bandwidth $\delta > 0$, denote by q the solution of

$$\arg \min_{p \in \mathcal{R}_D(l)} \sum_{i=1}^{k} \left[\zeta_{j+1}^{(i)} - p_{z,k}\left(Z_j^{(i)} - x\right)\right]^2 K\left(\frac{Z_j^{(i)} - x}{\delta}\right), \tag{2.65}$$

where $\mathcal{R}_D(l)$ is again the space of all polynomials of order l defined on $\mathbb{R}^D$. The local polynomial estimator of order l for $C_j(x)$ is then defined as $C_j^k(x) = q(0)$ if the problem (2.65) is solvable. Use $C_j^k(x) = 0.001$ otherwise [2]. Hence, the complexity of evaluating $C_j^k(z)$ is of order k as $k \to \infty$, so we note $\kappa_1 = \kappa_2 = 1$. This calculation is different from the previous subsection, since an increase of the number of basis functions is not assumed here.

However, as the number of training paths increases and the weights become more and more localized, the local estimator converges to a Nadaraya Watson estimator similar to the nearest neighbours technique in Section 2.1.1, irrespectively of the choice of basis functions. In particular from a theoretical point of view, this is a big advantage of this method as convergence can be assumed in each case.

Theorem 29. *For given x and time step j, the solution of* (2.65) *can be calculated with the help of* $(\Gamma_{u,v})_{|u|,|v| \le l}$ *and* $(S_u)_{|u| \le l}$ *given by*

$$S = \frac{1}{k\delta^d} \sum_{i=1}^{k} \zeta_{j+1}^{(i)} \left(\frac{Z_j^{(i)} - x}{\delta}\right)^u K\left(\frac{Z_j^{(i)} - x}{\delta}\right),$$

[2] The reason for this choice is as follows. In such a case that x is far away from all the training paths, it is likely to be very far in-the-money or out-of-the-money. Thus, it is optimal to exercise immediately if there is some payoff.

$$\Gamma = \frac{1}{k\delta^d}\sum_{i=1}^{k}\left(\frac{Z_j^{(i)}-x}{\delta}\right)^{u+v} K\left(\frac{Z_j^{(i)}-x}{\delta}\right).$$

If Γ *is positive definite, then there is a unique polynomial solving* (2.65) *with coefficients* $\Gamma^{-1}S$.

Corollary 30. *The polynomial regression estimator is a special kind of a mesh method. When we use the coefficients* $\Gamma^{-1}S$ *in* (2.65)*, we have*

$$\begin{aligned} C_j^k(x) &= \mathcal{M}^T(0)\Gamma^{-1}S \\ &= \frac{1}{kh^d}\sum_{i=1}^{k}\zeta_{j+1}^{(i)}K\left(\frac{Z_j^{(i)}-x}{h}\right) \\ &\quad \times\mathcal{M}^T(0)\Gamma^{-1}\mathcal{M}\left(\frac{Z_j^{(i)}-x}{h}\right), \end{aligned} \tag{2.66}$$

where $\mathcal{M}(x) = (\mathcal{M}(x)_u)_{|u|\le l}$ *is the vector of monomials.*

Furthermore, we take from Belomestny [6] that under some mild regularity assumptions on the law of X, there is some result about how to choose the bandwidth δ.

Theorem 31. *If the continuation value functions* C_j, $j = 0,\ldots,\mathcal{J}$ *belong to the Hölder class* $\Sigma(\beta, H, \mathbb{R}^d)$*, then the local polynomial regression estimator fulfills assumption (AQ) with* $\gamma_k = k^{-2\beta/(2\beta+D)}\log^{-1}(k)$ *provided that* $\delta = k^{-1/(2l+D)}$.

2.1.4 Nearest-Neighbours Technique

The easiest way to construct an estimator for C_j at a point x is simply averaging the payoffs of all training paths in the vicinity. Since no a-priori information about the underlying or the payoff is used, we can expect this approach to be relatively inefficient.

Choose a number of neighbours $0 \le \eta \ll k$ to compare each path to. Then the estimator for $C_j(x)$ is given by

$$C_j^k(x) = \frac{1}{\eta}\sum_{i=0}^{\eta}\zeta_{j+1}^{(\iota(x,i))}, \tag{2.67}$$

where $\iota(x,\cdot) : \{1,\ldots,k\} \to \{1,\ldots,k\}$ defines an order such that

$$\|Z_j^{\iota(x,1)} - x\|_\infty \le \|Z_j^{\iota(x,2)} - x\|_\infty \le \ldots \le \|Z_j^{\iota(x,k)} - x\|_\infty.$$

Here, we chose the maximum norm in order to reduce complexity. Of course, other equivalent norms can be used as well. Using the quick sort algorithm would lead to a complexity of $O(\log(k)k)$ for the sort belonging to each evaluation of $C_j^k(x)$. Since it is not necessary to calculate the order of all elements, but only of the first η ones, a partial sort is sufficient here. Partial sorts based on quicksorts will simply throw away elements that are too small which is signalized by comparison to the according pivot element. In such a way, a complexity of $O(k + \eta \log(\eta))$ is realized, so we have a total complexity of

$$k + \eta \log(\eta) + \eta \in O(k) \tag{2.68}$$

for each evaluation of $C_j^k(\cdot)$ because a reasonable choice of η will fulfill $\eta \in O(k)$. Asymptotically, we now have $\kappa_1 = \kappa_2 = 1$.

The nearest-neighbours technique shares some advantages with local regression. Firstly, neither transition densities nor basis functions are needed. Secondly, convergence as $k \to 0$ can be assumed. Furthermore, an advantage if compared to the "false nearest-neighbours technique" from Section 2.1.1 is that the neighbours considered come from a wider neighbourhood if there are only few trainingpaths in the vicinity of x. In particular, the set of neighbours cannot be empty. As this method is very similar to local regression, it won't be competitive either unless the number of training paths k is very large. The complexity needed to provide good results will explode as the number of exercise dates increases, see Agarwal and Juneja [1]. Alternatively, a good inner control must be found to improve (2.67).

2.2 Complexity Analysis

We want to measure the performance of a fast approximation method by comparing the root-mean-squared error of its result to its computational complexity. Of course, there are also other possibilities to measure the performance of an estimator. For example, another good idea is to consider the sum of the bias and a multiple of the standard deviation, so to say the lower bound of some confidence interval.

Let us recall the definition of the complexity from assumption (AC)

$$\mathscr{C}^{n,k} = k^{1+\kappa_1} + n \times k^{\kappa_2}, \tag{2.69}$$

which will be a good model for the computational time. The root-mean-squared error is defined as

$$\sqrt{\mathrm{E}\left[V^{n,k} - V_0\right]^2}. \tag{2.70}$$

How much complexity is necessary to ensure that the root-mean-squared error is less than a given number ε, namely $\mathrm{E}\left[V^{n,k} - V_0\right]^2 \le \varepsilon^2$? Therefore, let us define the minimal complexity as

$$\mathscr{C}(\varepsilon) := \min_{n,k \in \mathbb{R}^+} \left\{ \mathscr{C}^{n,k} \middle| \mathrm{E}\left[V^{n,k} - V_0\right]^2 \le \varepsilon^2 \right\}, \tag{2.71}$$

where integers are treated like reals. It is not difficult to solve this minimization problem and we have the following solution.

Theorem 32. *For a problem fulfilling (AB) with $\alpha > 0$, the minimal complexity of a fast approximation method that achieves a root-mean-squared error of ε is given asymptotically by*

$$\mathscr{C}(\varepsilon) \in O\left(\varepsilon^{-2\cdot\max\left(\frac{\kappa_1+1}{\mu(1+\alpha)},\, 1+\frac{\kappa_2}{\mu(1+\alpha)}\right)}\right), \tag{2.72}$$

which is realized via the choice

$$k^* = \varepsilon^{-\frac{2}{\mu(1+\alpha)}}, \quad n^* = \varepsilon^{-2}. \tag{2.73}$$

This result includes the following fact about the variance that is caused by the randomness of the training paths.

Theorem 33. *The variance of the testing value conditional the training paths fulfills*

$$\operatorname{Var}\left[\mathrm{E}\left[V^{n,k}\middle|\sigma\left(Z_\cdot^{(1)},\dots,Z_\cdot^{(k)}\right)\right]\right] \lesssim C\gamma_k^{2(1+\alpha)},$$

i.e. the variance of the bias is decreasing at a higher order than the squared bias, compare Theorem 18.

The order in formula (2.72) is the optimal statement possible and we have a suitable corollary.

Corollary 34. *In case of $\kappa_1 = \kappa_2 := \kappa$ and $\alpha = 1$, the complexity of $V^{k,n}$ is given by*

$$\mathscr{C}(\varepsilon) \in O\left(\varepsilon^{-2\max\left(\frac{1}{\mu},1+\frac{1}{2\mu}\right)}\right). \tag{2.74}$$

Thus, the complexity is always larger than ε^{-3}, because $\mu \leq 1$ for all fast approximation methods.

This is the main result of this chapter that will be compared to the multilevel version of the fast approximation methods in Section 3. At the end of this section, let us make note of another interesting possibility to improve lower bounds that is quite generic. It is called “policy iteration” and does not belong to the class of fast approximation methods.

Remark 35. *According to Kolodko and Schoenmakers [54], every consistent family (see Definition 2) of stopping times $\tau_0,\dots,\tau_{\mathcal{J}}$ induces an estimator for the option price via*

$$\widehat{V}_j = \max_{j\leq i\leq\min\{j+\kappa,\mathcal{J}\}} \mathrm{E}\left[g_{\tau_i}(X_{\tau_i})|X_j\right], \tag{2.75}$$

where $\kappa \in \{1,\dots,\mathcal{J}\}$ is called the “window parameter”. Hopefully, the stopping times based on this estimator

$$\widehat{\tau}_j = \min\left\{i \in \{j,\dots,\mathcal{J}\}\middle|\widehat{V}_i \leq g_i\left(X_i\right)\right\}, \quad j = 0,\dots,\mathcal{J} \tag{2.76}$$

will be a better approximation of the optimal stopping time than the family τ. It is possible to use this technique to define a sequence of (families of) stopping times $\tau^0, \tau^1, \dots$. For example in case of $\kappa = 1$, such a sequence is given via the simple [3] *iteration rule*

$$\tau_j^k = \min\left\{i \in \{j,\dots,\mathcal{J}\}\,\middle|\,\mathrm{E}\left[g_{\tau_{i+1}^{k-1}}\left(X_{\tau_{i+1}^{k-1}}\right)\middle|X_i\right] \leq g_i\left(X_i\right)\right\}, \tag{2.77}$$

[3] This easy case with $\kappa = 1$ was already discussed by several authors before: Howard, Irle and Puterman.

since

$$\mathrm{E}[g_{\tau_{i+1}}(X_{\tau_{i+1}})|X_i] < \mathrm{E}[g_{\tau_i}(X_{\tau_i})|X_i] \Rightarrow \tau_i = i \Rightarrow \mathrm{E}[g_{\tau_i}(X_{\tau_i})|X_i] = g_{\tau_i}(X_{\tau_i})$$

for a consist family such that $\widehat{V}_j = \mathrm{E}[g_{\tau^{k-1}_{i+1}}(X_{\tau^{k-1}_{i+1}})\Big|X_i]$ *for* $j = 0, \ldots, \mathcal{J}-1$*. A starting point* τ^0 *could be given by* $\tau^0_j \equiv j$ *for all* $j = 0, \ldots, \mathcal{J}$*. This sequence of stopping times* $\tau^0, \tau^1, \ldots$ *converges to the true value and even*

$$\tau^m_i = \tau^*_i, \ m \geq \mathcal{J} - i \tag{2.78}$$

holds. In other words, the number of time steps j *that* τ^k_j *coincides with* τ^*_j *increases with every iteration at least by one (starting from the last time step).*

However, since the expectations in (2.77) *have to be estimated, the complexity is very high and* (2.78) *is merely a theoretical result. Using simple Monte Carlo estimations to approximate* (2.77) *would lead to a system of nested subsimulations with exponential complexity. Therefore, it is much more efficient to start with a stopping time* τ^0 *from one of the fast approximation methods from Section 2.1 and use a one step or two step iteration to improve it. Kolodko and Schoenmakers state that this will be optimal in usual cases.*

2.3 Proofs

Let us first note the following helpful theorem that allows us to find a bound for the difference of the expectations when following two different stopping rules expressed in terms of the implied exercise instructions.

Theorem 36. *Let* Y *be a process adapted to the filtration* $\mathcal{F}$ *and let* $\tau^1_0, \ldots, \tau^1_{\mathcal{J}}$ *and* $\tau^2_0, \ldots, \tau^2_{\mathcal{J}}$ *be two consistent families of stopping times. Then*

$$\begin{aligned} & \mathrm{E}_{\mathcal{F}_j}\left[Y_{\tau^1_j} - Y_{\tau^2_j}\right] \\ = \ & \mathrm{E}_{\mathcal{F}_j}\left\{\sum_{l=j}^{\mathcal{J}-1}\left(Y_l - \mathrm{E}_{\mathcal{F}_l}\left[Y_{\tau^1_{l+1}}\right]\right)\left(1_{\{\tau^1_l=l,\tau^2_l>l\}} - 1_{\{\tau^1_l>l,\tau^2_l=l\}}\right)1_{\{\tau^2_l>l\}}\right\} \end{aligned}$$

for any $j = 0, \ldots, \mathcal{J}-1$.

Proof. We have

$$\begin{aligned} Y_{\tau^1_j} - Y_{\tau^2_j} &= \left[Y_j - Y_{\tau^2_j}\right]1_{\{\tau^1_j=j,\tau^2_j>j\}} + \left[Y_{\tau^1_j} - Y_j\right]1_{\{\tau^1_j>j,\tau^2_j=j\}} \\ &\quad + \left[Y_{\tau^1_j} - Y_{\tau^2_j}\right]1_{\{\tau^1_j>j,\tau^2_j>j\}} \\ &= \left[Y_j - Y_{\tau^1_{j+1}}\right]1_{\{\tau^1_j=j,\tau^2_j>j\}} + \left[Y_{\tau^1_{j+1}} - Y_j\right]1_{\{\tau^1_j>j,\tau^2_j=j\}} \\ &\quad + \left[Y_{\tau^1_{j+1}} - Y_{\tau^2_{j+1}}\right]1_{\{\tau^1_j=j,\tau^2_j>j\}} + \left[Y_{\tau^1_{j+1}} - Y_{\tau^2_{j+1}}\right]1_{\{\tau^1_j>j,\tau^2_j>j\}}. \end{aligned}$$

Therefore it holds for $\Delta_j := \mathrm{E}_{\mathcal{F}_j}\left[Y_{\tau^1_j} - Y_{\tau^2_j}\right]$

$$\Delta_j = \left[Y_j - \mathrm{E}_{\mathcal{F}_j}\left[Y_{\tau^1_{j+1}}\right]\right]\left(1_{\{\tau^1_j=j,\tau^2_j>j\}} - 1_{\{\tau^1_j>j,\tau^2_j=j\}}\right) + \mathrm{E}_{\mathcal{F}_j}\left\{\Delta_{j+1}1_{\{\tau^2_j>j\}}\right\}$$

with $\Delta_{\mathcal{J}} = 0$ and

$$\Delta_j = \mathrm{E}_{\mathcal{F}_j}\left\{\sum_{l=j}^{\mathcal{J}-1}\left(Y_l - \mathrm{E}_{\mathcal{F}_l}\left[Y_{\tau^1_{l+1}}\right]\right)\left(1_{\{\tau^1_l = l, \tau^2_l > l\}} - 1_{\{\tau^1_l > l, \tau^2_l = l\}}\right)1_{\{\tau^2_l > l\}}\right\}.$$

□

Furthermore, we need a result about the probability that the stopping rule obtained from a fast approximation method gives us the wrong advice whether to exercise.

Theorem 37. *The probability of the event that a continuation estimate C^k_j gives us the wrong advice whether to stop at time j*

$$\begin{aligned}\mathcal{E}_{k,j} &= \{g_j(X_j) > C_j(X_j),\, g_j(X_j) \le C^k_j(X_j)\}\\ &\quad \cup \{g_j(X_j) \le C_j(X_j),\, g_j(X_j) > C^k_j(X_j)\},\end{aligned}$$

fulfills the asymptotic relation

$$P(\mathcal{E}_{k,l}) \lesssim \gamma_k^{\alpha/2}.$$

Proof. We define

$$\mathcal{A}_{k,j,0} = \left\{0 < |g_j(X_j) - C_j(X_j)| \le \gamma_k^{1/2}\right\},$$

$$\mathcal{A}_{k,j,i} = \left\{2^{i-1}\gamma_k^{1/2} < |g_j(X_j) - C_j(X_j)| \le 2^i\gamma_k^{1/2}\right\}$$

for $j = 0, \ldots, \mathcal{J} - 1$ and $i > 0$. Note that

$$\mathrm{P}(\mathcal{E}_{k,s}) = \sum_{i=0}^{\infty}\mathrm{P}(\mathcal{E}_{k,s} \cap \mathcal{A}_{k,s,i}) \tag{2.79}$$

and

$$\mathrm{P}(\mathcal{E}_{k,s} \cap \mathcal{A}_{k,s,i}) \quad \le \quad \mathrm{P}\left(|g_s(X_s) - C_s(X_s)| \le \gamma_{k_{l-1}}^{1/2}\right) \le A\gamma_{k_{l-1}}^{\alpha/2}$$

if $i = 0$ and

$$\begin{aligned}\mathrm{P}(\mathcal{E}_{k,s} \cap \mathcal{A}_{k,s,i}) &\le \mathrm{E}\left[1_{\left\{|g_s(X_s) - C_s(X_s)| \le 2^i\gamma_k^{1/2}\right\}}\mathrm{P}^k\left(\left|C^k_s(X_s) - C_s(X_s)\right| > 2^{i-1}\gamma_k^{1/2}\right)\right]\\ &\le A\gamma_k^{\alpha/2}2^{i\alpha}B_1\exp(-B_2 2^{i-1})\end{aligned}$$

for $i > 0$. Since the exponential decrease of $\exp(-B_2 2^{i-1})$ is stronger than the growth of $2^{i\alpha}$, the sum in (2.79) is finite. □

Proof of Theorem 18

Taking into account that

$$C_l(X_l) = \mathrm{E}_{\mathcal{F}_l}\left[g_{\tau_{l+1}}(X_{\tau^*_{l+1}})\right] \le g_l(X_l)$$

on $\{\tau_l^* = l\}$ and

$$C_l(X_l) < g_l(X_l)$$

on $\{\tau_l^* > l\}$, we get from Theorem 36 for $R = V^{n,k} - V_0$ that

$$\begin{aligned} |R| &= \left|\mathrm{E}\left[g_{\tau_k^*}(X_{\tau_k^*}) - g_{\tau_k}(X_{\tau_k})\right]\right| \\ &\leq \mathrm{E}\left[\sum_{l=0}^{\mathcal{J}-1} |C_l(X_l) - g_l(X_l)| \left(1_{\{\tau_{k,l}^*=l, \tau_{k,l}>l\}} + 1_{\{\tau_{k,l}^*>l, \tau_{k,l}=l\}}\right)\right]. \end{aligned}$$

Using the same notation as in Theorem 37, we see that it holds

$$\begin{aligned} |R| &\leq \mathrm{E}\left[\sum_{l=0}^{\mathcal{J}-1} |C_l(X_l) - g_l(X_l)|\, 1_{\{\mathcal{E}_{k,l}\}}\right] \\ &= \mathrm{E}\left[\sum_{i=0}^{\infty}\sum_{l=0}^{\mathcal{J}-1} |C_l(X_l) - g_l(X_l)|\, 1_{\{\mathcal{E}_{k,l}\cap\mathcal{A}_{k,l,i}\}}\right] \\ &= \gamma_k^{1/2} \sum_{l=0}^{\mathcal{J}-1} \mathrm{P}\left(|g_l(X_l) - C_l(X_l)| \leq \gamma_k^{1/2}\right) \\ &\quad + \mathrm{E}\left[\sum_{i=1}^{\infty}\sum_{l=0}^{\mathcal{J}-1} |C_l(X_l) - g_l(X_l)|\, 1_{\{\mathcal{E}_{k,l}\cap\mathcal{A}_{k,l,i}\}}\right]. \end{aligned}$$

Using the fact that $|g_l(X_l) - C_l(X_l)| \leq |C_l(X_l) - C_l(X_l)|$ on $\mathcal{E}_{k,l}$, we now derive

$$\begin{aligned} |R| &\leq \gamma_k^{1/2} \sum_{l=0}^{\mathcal{J}-1} \mathrm{P}\left(|g_l(X_l) - C_l(X_l)| \leq \gamma_k^{1/2}\right) \\ &\quad + \sum_{i=1}^{\infty} 2^i \gamma_k^{1/2} \mathrm{E}\left[\sum_{l=0}^{\mathcal{J}-1} 1_{\left\{|g_j(X_j) - C_j(X_j)| \leq 2^i \gamma_k^{1/2}\right\}} \mathrm{P}^k\left(\left|C_l^k(X_l) - C_l(X_l)\right| > 2^{i-1}\gamma_k^{1/2}\right)\right] \\ &\leq \gamma_k^{1/2}\left(A\mathcal{J}\gamma_k^{\alpha/2} + A\mathcal{J}\gamma_k^{\alpha/2}\sum_{i=1}^{\infty} 2^i B_1 \exp(-B_2 2^{i-1})\right). \\ &\leq \gamma_k^{(1+\alpha)/2} c \end{aligned}$$

similar as in Theorem 37.

Proof of Theorem 19

We show that

$$\mathrm{E}\left[\zeta_j^{(h)}\right] \geq V_j\left(N_j^{(h)}\right), \quad h = 1, \ldots, k \tag{2.80}$$

by induction from $t_{\mathcal{J}}, \ldots, t_0$, see [40] and [19]. The start of induction is clear, since $\zeta_{\mathcal{J}}^{(h)} = g_{\mathcal{J}}\left(N^{(h)}\right)$ which is equal to the true value $V_{\mathcal{J}}\left(N_{\mathcal{J}}^{(h)}\right)$ and in particular fulfills (2.80). Now fix a time step t_j.

$$\mathrm{E}\left[\zeta_j^{(h)}|\mathbf{N}_j\right] = \mathrm{E}\left[\max\left\{g_j\left(N_j^{(h)}\right), \frac{1}{k}\sum_{i=1}^{k} w_j^{(i)}\left(N_j^{(h)}\right)\zeta_{j+1}^{(i)}\right\}\Big|\mathbf{N}_j\right]$$

$$\geq \max\left\{g_j\left(N_j^{(h)}\right), \frac{1}{k}\sum_{i=1}^{k} \mathrm{E}\left[w_j^{(i)}\left(N_j^{(h)}\right)\zeta_{j+1}^{(i)}\Big|\mathbf{N}_j\right]\right\} \tag{2.81}$$

The first step is due to Jensen's inequality and so we have due to (M2) that

$$\mathrm{E}\left[w_j^{(i)}\left(N_j^{(h)}\right)\zeta_{j+1}^{(i)}|\mathbf{N}_j, \mathbf{N}_{j+1}\right] = w_j^{(i)}\left(N_j^{(h)}\right)\mathrm{E}\left[\zeta_{j+1}^{(i)}|\mathbf{N}_j, \mathbf{N}_{j+1}\right]$$

$$= w_j^{(i)}\left(N_j^{(h)}\right)\mathrm{E}\left[\zeta_{j+1}^{(i)}|\mathbf{N}_{j+1}\right], \tag{2.82}$$

where the second step follows from (M1). Inserting this result into (2.82) we have

$$\mathrm{E}\left[\zeta_j^{(h)}|\mathbf{N}_j\right] \geq \max\left\{g_j\left(N_j^{(h)}\right), \frac{1}{k}\sum_{i=1}^{k}\mathrm{E}\left[w_j^{(i)}\left(N^{(h)}\right)\zeta_{j+1}^{(i)}\Big|\mathbf{N}_{j+1}\right]\right\}.$$

Applying the induction assumption (2.80) with $j+1$ leads to

$$\ldots \geq \max\left\{g_j\left(N_j^{(h)}\right), \frac{1}{k}\sum_{i=1}^{k}\mathrm{E}\left[w_j^{(i)}\left(N^{(h)}\right)V_{j+1}^{(i)}\Big|\mathbf{N}_{j+1}\right]\right\}$$

and finally because of (M3)

$$\ldots = \max\left\{g_j\left(N_j^{(h)}\right), \mathrm{E}\left[C_{j+1}(N_j^{(h)})\right]\right\} = V_j\left(N_j^{(h)}\right).$$

Proof of Theorem 32

Denote by $\mathcal{G} = \sigma(Z_\cdot^1, \ldots, Z_\cdot^k)$ the $\sigma-$algebra generated by the training paths. Using the bias-variance decomposition, we obtain that

$$\begin{aligned}
&\mathrm{E}\left[\left(V_0^{n,k} - V_0\right)^2\right] \\
=&\,\mathrm{E}\left[\left(V_0^{n,k} - \mathrm{E}\left[V_0^{n,k}\right] + \mathrm{E}\left[V_0^{n,k}\right] - V_0\right)^2\right] \\
=&\,\mathrm{E}\left[\left(V_0^{n,k} - \mathrm{E}\left[V_0^{n,k}\right]\right)^2 + \left(\mathrm{E}\left[V_0^{n,k}\right] - V_0\right)^2\right] \\
&+\underbrace{2\,\mathrm{E}\left[V_0^{n,k} - \mathrm{E}\left[V_0^{n,k}\right]\right]\left(\mathrm{E}\left[V_0^{n,k}\right] - V_0\right)}_{=0}
\end{aligned}$$

and repeating the procedure yields

$$\begin{aligned}
\ldots =& \mathrm{E}\left[\mathrm{E}\left[\left(V_0^{n,k}-\mathrm{E}\left[V_0^{n,k}\right]\right)^2\Big|\mathcal{G}\right]\right]+\left(\mathrm{E}\left[V_0^{n,k}\right]-V_0\right)^2\\
=& \mathrm{E}\left[\mathrm{E}\left[\left(V_0^{n,k}-\mathrm{E}\left[V_0^{n,k}\Big|\mathcal{G}\right]+\mathrm{E}\left[V_0^{n,k}\Big|\mathcal{G}\right]-\mathrm{E}\left[V_0^{n,k}\right]\right)^2\Big|\mathcal{G}\right]\right]+\left(\mathrm{E}\left[V_0^{n,k}\right]-V_0\right)^2\\
=& \mathrm{E}\left[\mathrm{E}\left[\left(V_0^{n,k}-\mathrm{E}\left[V_0^{n,k}\Big|\mathcal{G}\right]\right)^2+\left(\mathrm{E}\left[V_0^{n,k}\Big|\mathcal{G}\right]-\mathrm{E}\left[V_0^{n,k}\right]\right)^2\Big|\mathcal{G}\right]\right]\\
&+\underbrace{2\,\mathrm{E}\left[\mathrm{E}\left[V_0^{n,k}-\mathrm{E}\left[V_0^{n,k}\Big|\mathcal{G}\right]\Big|\mathcal{G}\right]\left(\mathrm{E}\left[V_0^{n,k}\Big|\mathcal{G}\right]-\mathrm{E}\left[V_0^{n,k}\right]\right)\right]}_{=0}\\
&+\left(\mathrm{E}\left[V_0^{n,k}\right]-V_0\right)^2\\
=& \mathrm{E}\left[\frac{1}{n}\operatorname{Var}\left[V_0^{1,k}|\mathcal{G}\right]\right]+\operatorname{Var}\left[\mathrm{E}\left[V_0^{n,k}|\mathcal{G}\right]\right]+Ck^{-\mu(1+\alpha)}\\
\lesssim& \frac{1}{n}\,\mathrm{E}\left[g_{\tau^*}(X_{\tau^*})^2\right]+Ck^{-\mu 2(1+\alpha)}+Ck^{-\mu(1+\alpha)}.
\end{aligned}$$

We can neglect the variance of the bias here following Theorem 33 and thus have

$$\mathrm{E}\left[\left(V_0^{k,n}-V_0\right)^2\right]\lesssim\frac{C}{n}+Ck^{-\mu(1+\alpha)},$$

where the term $Ck^{-\mu(1+\alpha)}$ originates from Assumption (AQ). To ensure that the bias is smaller than $\varepsilon/\sqrt{2}$, we have

$$k^{-\mu(1+\alpha)/2}\lesssim\varepsilon/\sqrt{2}\;\Rightarrow\;k=c\varepsilon^{-\frac{2}{\mu(1+\alpha)}}.$$

The variance will be less or equal to $\varepsilon^2/2$ if $n=c\varepsilon^{-2}$. Thus, the complexity will have the same order as

$$\varepsilon^{\frac{-2(\kappa_1+1)}{\mu(1+\alpha)}}+\varepsilon^{-2}\cdot k^{\kappa_2\frac{2}{\mu(1+\alpha)}}$$

according to assumption (AC).

Proof of Theorem 33

Let us define $C:=\mathrm{E}[V_0^{k,n}|\mathcal{G}]-V_0$, where $\mathcal{G}$ is again the σ-algebra generated by the training paths.

$$\begin{aligned}
|C| &= \left|\mathrm{E}\left[g_{\tau^*}(X_{\tau^*})-g_{\tau_k}(X_{\tau_k})|\mathcal{G}\right]\right|\\
&\leq \mathrm{E}\left[\sum_{l=0}^{\mathcal{J}-1}|C_l(X_l)-g_l(X_l)|\left(1_{\{\tau_l^*=l,\tau_{k,l}>l\}}+1_{\{\tau_l^*>l,\tau_{k,l}=l\}}\right)\Big|\mathcal{G}\right].
\end{aligned}$$

It also holds in the conditional case

$$\begin{aligned}
|C| &\leq \mathrm{E}\left[\sum_{l=0}^{\mathcal{J}-1} |C_l(X_l) - g_l(X_l)|\, 1_{\{\mathcal{E}_{k,l}\}} \Big| \mathcal{G}\right] \\
&= \gamma_k^{1/2} \sum_{l=0}^{\mathcal{J}-1} \mathrm{P}\left(|g_l(X_l) - C_l^*(X_l)| \leq \gamma_k^{1/2}\right) \\
&\quad + \mathrm{E}\left[\sum_{i=1}^{\infty} \sum_{l=0}^{\mathcal{J}-1} |C_l(X_l) - g_l(X_l)|\, 1_{\{\mathcal{E}_{k,l} \cap \mathcal{A}_{k,l,i}\}} \Big| \mathcal{G}\right].
\end{aligned}$$

Using the fact that $|g_l(X_l) - C_l(X_l)| \leq \left|C_l^k(X_l) - C_l(X_l)\right|$ on $\mathcal{E}_{k,l}$, we derive

$$\begin{aligned}
|C| &\leq A\mathcal{J}\gamma_k^{(1+\alpha)/2} \\
&\quad + \sum_{i=1}^{\infty} 2^i \gamma_k^{1/2} \mathrm{E}\left[\sum_{l=0}^{\mathcal{J}-1} 1_{\left\{|g_j(X_l) - C_l(X_l)| \leq 2^i \gamma_k^{1/2}\right\}} 1_{\left\{\left|C_l^k(X_l) - C_l(X_l)\right| > 2^{i-1}\gamma_k^{1/2}\right\}} \Big| \mathcal{G}\right] \\
&\leq A\mathcal{J}\gamma_k^{(1+\alpha)/2} \\
&\quad + A\gamma_k^{(1+\alpha)/2} \sum_{i=1}^{\infty} 2^{i(1+\alpha)} \sum_{l=0}^{\mathcal{J}-1} 1_{\left\{\left|C_l^k(X_l) - C_l(X_l)\right| > 2^{i-1}\gamma_k^{1/2}\right\}},
\end{aligned}$$

where we have made use of the assumptions (AQ) and (AM). Let us define

$$\iota(t) = \frac{\log\left(\frac{t}{A\mathcal{J}\gamma_k^{(1+\alpha)/2}} - 1\right)}{(1+\alpha)\log 2}, \tag{2.83}$$

so we have

$$\begin{aligned}
& F(t) := P(|C| < t) \\
\geq\; & \begin{cases} 0, & t \leq A\mathcal{J}\gamma_k^{(1+\alpha)/2} \\ \prod_{l=0}^{\mathcal{J}-1}\left(1 - P\left(\left|C_l^k(X_l) - C_l(X_l)\right| > 2^{\iota(t)-1}\gamma_k^{1/2}\right)\right), & t > A\mathcal{J}\gamma_k^{(1+\alpha)/2} \end{cases} \\
\geq\; & \begin{cases} 0, & t \leq A\mathcal{J}\gamma_k^{(1+\alpha)/2} \\ \left(1 - B_1 e^{-B_2 2^{\iota(t)-1}}\right)^{\mathcal{J}}, & t > A\mathcal{J}\gamma_k^{(1+\alpha)/2} \end{cases}.
\end{aligned}$$

Now, it follows that

$$\begin{aligned}
\mathrm{Var}\,[C] &= \mathrm{Var}\left[\mathrm{E}[V_0^{k,n}|\mathcal{G}]\right] \\
&= \mathrm{E}\left[\left(\mathrm{E}[V_0^{k,n}|\mathcal{G}] - \mathrm{E}[V_0^{k,n}]\right)^2\right] \\
&\leq \mathrm{E}\left[\left(\mathrm{E}[V_0^{k,n}|\mathcal{G}] - V_0\right)^2\right] \\
&= 2\int_0^{\infty} tP\left(\left(V_0 - E[V_0^{k,n}|\mathcal{G}]\right)^2 > t\right) dt \\
&= 2\int_0^{\infty} tP\left(|C| > \sqrt{t}\right) dt \\
&= 2\int_0^{\infty} t\left(1 - F\left(\sqrt{t}\right)\right) dt,
\end{aligned}$$

so when inserting the above result:

$$\begin{aligned}\operatorname{Var}[C] &\leq 2\int_0^{A^2\gamma_k^{(1+\alpha)}} tdt + 2\int_{A^2\gamma_k^{(1+\alpha)}}^{\infty} t\left(1-\left(1-B_1 e^{-B_2 2^{\iota(\sqrt{t})-1}}\right)^{\mathcal{J}}\right)dt \\ &\lesssim A^4\gamma_k^{2(1+\alpha)} + O(\gamma_k^{2(1+\alpha)})\end{aligned}$$

holds, which was to be shown. The last inequality follows because

$$2^{\iota(\sqrt{t})} = e^{\log\left(\frac{\sqrt{t}}{A\mathcal{J}\gamma_k^{(1+\alpha)/2}}-1\right)/(1+\alpha)} = \left(\frac{\sqrt{t}}{A\mathcal{J}\gamma_k^{(1+\alpha)/2}}-1\right)^{1/(1+\alpha)},$$

so $f(t,\gamma_k) = 1-\left(1-B_1 e^{-B_2 2^{\iota(\sqrt{t})-1}}\right)^{\mathcal{J}}$ is a function that fulfills asymptotically,

$$\begin{aligned} f(t,\gamma_k) &\asymp \mathcal{J}B_1 e^{-B_2 2^{\iota(\sqrt{t})-1}} \\ f(t,\gamma_k) &\in o(\gamma_k^{\beta}), \quad \forall\beta > 0\end{aligned}$$

as $\gamma_k \to 0$ and $t \to \infty$. Thus, $\int_0^\infty tf(t)dt$ will be finite and it holds

$$\int_{A^2\gamma_k^{(1+\alpha)}}^{\infty} tf(t)dt \lesssim \int_0^\infty t\mathcal{J}B_1 e^{-B_2 2^{\sqrt{\iota}(t)-1}}dt \lesssim \gamma_k^{2(1+\alpha)}. \tag{2.84}$$

Chapter 3

Multilevel for Fast Approximation Methods

Multilevel Monte Carlo methods were introduced into stochastics by Heinrich [47] and the path-breaking work of Giles [39]. The multilevel technique is an idea about how to reduce the complexity of Monte Carlo simulations and can be understood as an extension of the quasi-control variate approach, see Section 1.3.

Assume a stochastic algorithm that yields a random quantity A^k, where k is a parameter that has influence on the complexity and the accuracy of the algorithm, howsoever accuracy and complexity are measured. Instead of estimating its expectation via Monte Carlo simulation, we write at first

$$\mathrm{E}[A^k] = \mathrm{E}[A^K] + \mathrm{E}[A^k - A^K] \tag{3.1}$$

with $k \neq K$. Then, we replace the two expectations with their Monte Carlo estimators and run two independent simulations. More precisely, $\mathrm{E}[A^K]$ is estimated with N samples and $\mathrm{E}[A^k - A^K]$ with n samples. Hopefully, there is a tuple (k, K, n, N) such that the sum of these two estimators yields a better relation between complexity and accuracy than the standard Monte Carlo estimator. In other words, the same algorithm with lower accuracy is used as a quasi-control variate for the algorithm itself. This would correspond to a multilevel estimator with $L = 1$ higher levels.

One can extend this idea and use a sequence of random variables, where each element in the sequence serves as a quasi-control variate to its successor (except for the last one), so we fix a sequence of parameters $(k_0, \ldots, k_L)$ and write

$$\mathrm{E}[A^{k_L}] = \mathrm{E}[A^{k_0}] + \sum_{l=1}^{L} \mathrm{E}[A^{k_l} - A^{k_{l-1}}], \tag{3.2}$$

where L is the number of higher levels. The total variance that arises if each of the expectations in (3.2) is estimated independently with N_l Monte Carlo samples is nebulous and therefore has to be studied.

The reason for the gain of complexity will be twofold. Firstly, if A^{k_l} and $A^{k_{l-1}}$ are drawn simultaneously from a common probability space in a natural way, there will probably be a strong correlation between them. This will lead to a small variance for all the higher levels and we will need only few samples to estimate the summands in (3.2) accurately. So let us choose k_0 such that A^{k_0} is a fast and inaccurate result and estimate it with many samples. Together, both effects could give us an estimator that is fast and accurate at the same time. Secondly, it is typical for the multilevel approach that there might be problem-depending enhancements that achieve coupling effects between the minuend variable and subtrahend variable, which could lead to even stronger correlations. This will be the case for the two applications of the multilevel technique in the following chapters.

Recently, there has been a lot of research about the multilevel technique, i.e. it has been applied to quite a variety of problems. First of all, Giles used this idea to improve the efficiency of sampling from stochastic differential equations. Suppose a stochastic process $X_.$ that is the solution of an SDE with respect to Brownian motion. Trajectories of this process are simulated via some time discretization with k time steps and are denoted by $X_.^{k_L}$. We have a function f of X_T, so there is the usual European setting and $P_L = f(X_T^{k_L})$ plays the role of A^{k_L}. A quite general version of his main result reads as follows, where the accuracy is measured in terms of the mean-squared error.

Theorem 38. *Let P denote a random variable, and let P_l denote the corresponding level l numerical approximation. If there exist independent estimators Y_l based on N_l Monte Carlo samples, and positive constants $\alpha, \beta, \gamma, c_1, c_2, c_3$ such that $\alpha \geq \frac{1}{2}\min(\beta, \gamma)$ and*

1. $E[P_l - P] \leq c_1 2^{-\alpha l}$
2. $E[Y_l] = \begin{cases} E[P_0], & l = 0 \\ \mathrm{E}[P_l - P_{l-1}], & l > 0 \end{cases}$
3. $\mathrm{Var}[Y_l] \leq c_2 N_l^{-1} 2^{-\beta l}$
4. $\mathrm{E}[C_l] \leq c_3 N_l 2^{\gamma l}$,

(where C_l is the computational complexity of Y_l). Then there exists a positive constant c_4, such that for any $\epsilon < e^{-1}$ there are values L and N_l for which the multilevel estimator

$$Y = \sum_{l=0}^{L} Y_l,$$

has a mean-square-error with bound

$$\mathrm{E}\left[(Y - \mathrm{E}[P])^2\right] < \epsilon^2$$

with a computational complexity C with bound

$$\mathrm{E}[C] \leq \begin{cases} c_4 \epsilon^{-2}, & \beta > 1 \\ c_4 \epsilon^{-2} (\log \epsilon)^2, & \beta = 1 \\ c_4 \epsilon^{-2-(1-\beta)/\alpha}, & 0 < \beta < 1 \end{cases}$$

There are many related results. For example, Dereich and Li [30] reduce the complexity that originates from sampling paths of a Lévy process. Their setting is quite similar to that one of Giles. The levels are again based on different approximation schemes, but each of those schemes depends on two parameters. Firstly, a parameter h that is "a threshold for the size of the jumps being considered large and causing immediate updates". Secondly, a parameter ε that plays a role similar to the fineness of the time discretization, namely "the length of the regular update intervals" in their words.

The result of Giles strongly depends on the parameter β. As we will see below, this parameter typically results from what is called the "strong order of convergence" of the approximation technique. In many applications of the multilevel technique, the size of β is a decisive issue to prove. Belomestny, Nagapetyan and Shiryaev [9] try to broaden the class of problems that are attackable via the multilevel by considering weak approximation schemes without assuming strong convergence.

In this chapter, we want to use the multilevel technique to enhance what was called the testing step (3.3) of a fast approximation method in Section 2.1. Another application of the multilevel technique will be discussed in Chapter 6, where the complexity of nested dual methods is improved. Recall, for example the mesh method from Section 2.1.1, which provides an estimator

$$V^k(x_0) = \mathrm{E}\left[g_{\tau_k}(X_{\tau_k}) | X_0 = x_0\right], \tag{3.3}$$

where k controls the number of mesh paths and thus has influence on the accuracy and the complexity of the method.

The multilevel algorithm can be summarized as follows: Fix a number of levels L, an increasing sequence $k_0, \ldots, k_L$ and a sequence of stopping times $\tau_{k_0}, \ldots, \tau_{k_L}$ each of them based on k_l training paths with the help of some fast approximation method. Thus, τ_{k_L} plays the role of τ_k in (3.3). It is the best but also the most expensive of these stopping times measured in terms of computational complexity. It is clear that

$$\mathrm{E}\Big[g_{\tau_{k_L}}(X_{\tau_{k_L}})\Big] = \mathrm{E}\Big[g_{\tau_{k_0}}(X_{\tau_{k_0}})\Big] + \sum_{l=1}^{L} \mathrm{E}\Big[\underbrace{g_{\tau_{k_l}}(X_{\tau_{k_l}}) - g_{\tau_{k_{l-1}}}(X_{\tau_{k_{l-1}}})}_{=:\Delta_l}\Big], \tag{3.4}$$

where we defined the "increments" Δ_l. Now, fix a decreasing sequence $n_0, \ldots, n_L$ and replace expectations with their Monte Carlo estimators.

Definition 39. *We define the multilevel Monte Carlo estimator (MLMC) as*

$$V^{\mathbf{n},\mathbf{k}} = \frac{1}{n_0}\sum_{r=1}^{n_0} g_{\tau_{k_0}^{(r)}}\left(X^{(r)}_{\tau_{k_0}^{(r)}}\right) + \sum_{l=1}^{L} \underbrace{\frac{1}{n_l}\sum_{r=1}^{n_l}\left[g_{\tau_{k_l}^{(r)}}\left(X^{(r)}_{\tau_{k_l}^{(r)}}\right) - g_{\tau_{k_{l-1}}^{(r)}}\left(X^{(r)}_{\tau_{k_{l-1}}^{(r)}}\right)\right]}_{=:\bar{\Delta}_l^{n_l}} \tag{3.5}$$

based on n_l independent realizations $\left(X^{(r)}_\cdot, \tau_{k_l}^{(r)}\right)$ of $(X_\cdot, \tau_{k_l})$ within each level, where

$$\tau_k^{(r)} = \inf\left\{0 \le j \le \mathcal{J} : g_j\left(X_j^{(r)}\right) \ge C_j^k\left(X_j^{(r)}\right)\right\}, \quad k \in \mathbb{N},$$

if a fast approximation method is used.

It is eye-catching that the variance can significantly be reduced in case of a fast approximation method if the complexity in the lower levels is very low and the “difference” between the stopping times in the higher levels is small enough. This is very likely to be the case, because for very large k and k' the estimators C_j^k and $C_j^{k'}$ will hardly differ, as both of them are very near to the true continuation value function C_j. At the same time, the difference of complexities might be large. So to say, it is possible to exploit the robustness of fast approximation methods explained in Section 2.1.

Unfortunately, we have to be cautious, since this setting does not fit into the setting described by Giles. The reason is simply the stochastic behaviour of the training paths, see Theorem 33. The expectation of Δ_l is a random variable with respect to the σ-algebra generated by two sets of training paths: Those that were in use to generate τ_{k_l} and those for $\tau_{k_{l-1}}$. That issue will be further discussed in in Section 3.1, where the complexity of the multilevel estimator is analyzed in the general case of a fast approximation method, as defined in Section 2.1. The result will then be compared to the complexity of the standard Monte Carlo estimator calculated in Section 2.2. Afterwards, two numerical examples will illustrate the gain of efficiency in case of a mesh method and the loss of efficiency in case of local regression. The succeeding section is about an algorithm that determines the number of levels during the runtime, similar to Giles [39]. Afterwards, some practical issues will be discussed, which will be the motivation for Chapter 4.

Another interesting enhancement of the multilevel technique should be mentioned here. Rhee and Glynn [63] and similarly McLeish [59] introduce related ideas about how to produce unbiased estimators from a sequence of approximation schemes that are not unbiased. Colloquially, their method could be called a “multilevel estimator with randomly many levels”. The number of levels in use is determined via an independent, positive integer valued random variable N. Now define

$$\bar{Z}_n = \sum_{k=0}^{n\wedge N} \Delta_k / \mathrm{P}(N \geq k) \tag{3.6}$$

that obviously is an unbiased estimator for $\mathrm{E}[P_n]$, which can be shown using Walds's identity. The variable $\bar{Z}$ that $\bar{Z}_n$ is converging to almost surely is then given by

$$\bar{Z} = \sum_{k=0}^{N} \Delta_k / \mathrm{P}(N \geq k). \tag{3.7}$$

In their Theorem 1, Glynn and Rhee state that $\bar{Z}$ is indeed an unbiased estimator of $\mathrm{E}[P]$ under some “appropriate conditions”. To sample from an unbiased estimator of course may be a tremendous advantage in some practical situations. Though the variance, and thus the complexity given the desired accuracy may increase. McLeish [59] states that “...we have purchased unbiasedness of the estimator at a cost of increasing the MSE by a factor ...”, where the factor depends on the order of decay of the expectations of the levels.

3.1 Complexity Analysis

Let us consider the multilevel approach for a fast approximation method under assumptions (AB), (AC) and (AQ). It is clear that for the bias of the multilevel estimator it holds $\mathrm{E}[V^{\mathbf{k},\mathbf{n}}] = \gamma_{k_L}^{(\alpha+1)/2}$, i.e. the bias results from the finest one of L approximations, see (3.4). The complexity of $V^{\mathbf{k},\mathbf{n}}$ is asymptotically given by

$$\sum_{l=0}^{L}(k_l^{\kappa_1+1} + n_l \cdot k_l^{\kappa_2}) \tag{3.8}$$

up to a constant. For the bias-variance decomposition, we need to know the behaviour of the variances in each level. The following theorem provides some upper bound.

Theorem 40. *If the payoff and the underlying stochastic process X satisfy*

$$M_p := \mathrm{E}\Big[\big|\max_{l=0,\dots,\mathcal{J}} g_l(X_l)\big|^{2p}\Big] < \infty \tag{3.9}$$

for some $p \geq 1$, then we have for the increment Δ_l that

$$\mathrm{E}[|\Delta_l|^2] = \mathrm{E}\left[\left|g_{\tau_{k_l}}(X_{\tau_{k_l}}) - g_{\tau_{k_{l-1}}}(X_{\tau_{k_{l-1}}})\right|^2\right] \leq C M_p^{1/p} \gamma_{k_{l-1}}^{\alpha/(2q)}$$

for any $l = 1, \dots, L$. Here, we have $\alpha > 0$ from assumption (AB), some absolute constant $C > 0$ and q satisfying $1/p + 1/q = 1$.

Remark 41. *Let us check whether assumption* (3.9) *is fulfilled for the max-call option with strike $\varkappa$ and D assets of geometric Brownian motion. It holds*

$$\begin{aligned}
&\mathrm{E}\left[\left|\max_{l=0,\dots,\mathcal{J}} g_l(X_l)\right|^{2p}\right] \leq \mathrm{E}\left[\sum_{l=0}^{\mathcal{J}} |g_l(X_l)|^{2p}\right] \\
&\leq c\sum_{j=0}^{\mathcal{J}}\sum_{d=1}^{D}\int_0^\infty \max(x_0^d e^{\left(\mu^d-\frac{1}{2}(\sigma^d)^2\right)t_j+\sigma^d x} - \varkappa, 0)^{2p}\frac{1}{\sqrt{2\pi\sigma^d t}}\, e^{-\frac{x^2}{2}}dx \\
&\lesssim \int_0^\infty \exp(-\frac{x^2}{2} + 2p\max_d \sigma^d x)dx < \infty
\end{aligned}$$

for all p, so to say $q = 1$. Since the min-put option has a bounded payoff, it is the same situation.

Remark 42. *As mentioned before, the difference to the setting of Giles is randomness of the training paths. To obtain levels that are independent of each other, we would have to generate two new sets of training paths in each level. Then, the coarser and the finer approximation would be independent of the other levels. Unfortunately, the variance of the bias that arises in this manner would have a large influence of the total root-mean-squared error of the whole algorithm, especially in the lower levels. Thus, we have to reuse the finer approximation of one level as coarser approximation for the next one. This reuse of the old training paths will be called a small "trick" in the following, since it leads to a small dependence of the levels.*

In order to simplify the bias-variance decomposition for the whole multilevel estimator, let us fix some notation. Let $\mathcal{G}_l$ denote the $\sigma-$algebra generated by the k_l training paths and define $\Delta_0 = g_{\tau_0}(X_{\tau_0})$ and $\mathcal{G}_{-1} = \{\Omega, \emptyset\}$.

The variance decomposition formula yields

$$\begin{aligned}\operatorname{Var}\left[V^{\mathbf{n},\mathbf{k}}\right] &= \operatorname{Var}\left[\sum_{l=0}^{L} \bar{\Delta}_l^{n_l}\right] \\ &= \mathrm{E}\left[\operatorname{Var}\left[\sum_{l=0}^{L} \bar{\Delta}_l^{n_l} | \mathcal{G}_1 \wedge \ldots \wedge \mathcal{G}_L\right]\right] + \operatorname{Var}\left[\mathrm{E}\left[\sum_{l=0}^{L} \bar{\Delta}_l^{n_l} | \mathcal{G}_1 \wedge \ldots \wedge \mathcal{G}_L\right]\right].\end{aligned}$$

The first summand includes the variances within the levels. If Theorem 40 holds, then

$$\begin{aligned}&\mathrm{E}\left[\operatorname{Var}\left[\sum_{l=0}^{L} \bar{\Delta}_l^{n_l} | \mathcal{G}_1 \wedge \ldots \wedge \mathcal{G}_L\right]\right] \\ =\; &\mathrm{E}\left[\frac{1}{n_l}\sum_{l=0}^{L} \operatorname{Var}\left[\Delta_l | \mathcal{G}_l \wedge \mathcal{G}_{l-1}\right]\right] \\ =\; &\sum_{l=0}^{L} \mathrm{E}\left[\frac{1}{n_l} \mathrm{E}\left[|\Delta_l|^2 | \mathcal{G}_l \wedge \mathcal{G}_{l-1}\right]\right] \\ \leq\; &\sum_{l=0}^{L} \frac{1}{n_l} \mathrm{E}[|\Delta_l|^2] \lesssim \frac{\mathrm{E}\left[\left(g_{\tau_{k_0}}(X_{\tau_{k_0}})\right)^2\right]}{n_0} + \sum_{l=1}^{L} \frac{1}{n_l} \gamma_{k_{l-1}}^{\alpha/(2q)},\end{aligned}$$

The second summand includes the variance resulting from the randomness of the training paths. Assuming that Theorem 33 holds, we now have

$$\begin{aligned}&\operatorname{Var}\left[\mathrm{E}\left[\sum_{l=0}^{L} \bar{\Delta}_l^{n_l} | \mathcal{G}_0 \wedge \ldots \wedge \mathcal{G}_L\right]\right] \\ =\; &\operatorname{Var}\left[\sum_{l=0}^{L} \mathrm{E}\left[\Delta_l | \mathcal{G}_l \wedge \mathcal{G}_{l-1}\right]\right] \\ =\; &\operatorname{Var}\left[\mathrm{E}\left[g_{\tau_{k_0}}(X_{\tau_{k_0}}) | \mathcal{G}_0\right] + \sum_{l=1}^{L} \mathrm{E}\left[g_{\tau_{k_l}}(X_{\tau_{k_l}}) | \mathcal{G}_l\right] - \mathrm{E}\left[g_{\tau_{k_{l-1}}}(X_{\tau_{k_{l-1}}}) | \mathcal{G}_{l-1}\right]\right] \\ =\; &\operatorname{Var}\left[\mathrm{E}\left[g_{\tau_{k_L}}(X_{\tau_{k_L}}) | \mathcal{G}_L\right]\right] \lesssim \gamma_{k_L}^{2(1+\alpha)}.\end{aligned}$$

Only the variance of the bias of the highest level remains, all other terms cancel out each other in the telescopic sum due to the trick. If they stayed, the variance of the estimator would be higher. In particular, the multilevel approach will become useful if the approximations used in the lower levels are very coarse when compared to the higher levels. Thus, this additional variance would be very high if the trick was not used.

Remark 43. *Summing up, we can use the bound*

$$\operatorname{Var}\left[V^{\mathbf{n},\mathbf{k}}\right] \lesssim \frac{\mathrm{E}\left[\left(g_{\tau_{k_0}}(X_{\tau_{k_0}})\right)^2\right]}{n_0} + \sum_{l=1}^{L} \frac{\gamma_{k_{l-1}}^{\alpha/(2q)}}{n_l} + \gamma_{k_L}^{2(1+\alpha)}. \tag{3.10}$$

for the total variance of the multilevel algorithm.

This bound will be minimized in the following. As the numerical experiments will show, this is a tight asymptotic bound.

Definition 44. *We denote the minimal complexity of the multilevel estimator, see Definition 39, by*

$$\mathscr{C}_L(\varepsilon) = \min_{\mathbf{n},\mathbf{k}\in\mathbb{R}_+^L} \left\{ \sum_{l=0}^{L} k_l^{\varkappa_1+1} + n_l \cdot k_l^{\varkappa_2} \middle| \mathrm{E}\left[\left(V^{\mathbf{k},\mathbf{n}} - V_0\right)^2\right] \leq \varepsilon^2 \right\}.$$

and further $\mathscr{C}_{ML}(\varepsilon) = \min_{l\geq 0} \mathscr{C}_l(\varepsilon)$.

Theorem 45. *Let assumptions of Theorem 18 hold with* $\gamma_{k_l} = k_l^{-\mu}$, *for some* $\mu > 0$. *Then under the choice* $k_l^* = k_0 \cdot \theta^l$, $l = 0, 1, \dots, L$, *with* $\theta > 1$,

$$L = \left\lceil \frac{2}{\mu(1+\alpha)} \log_\theta \left(\varepsilon^{-1} \cdot k_0^{-\mu(1+\alpha)/2} \right) \right\rceil$$

and

$$n_l^* = c\varepsilon^{-2} \left(\sum_{i=1}^{L} \sqrt{k_i^{(\kappa_2 - \mu\alpha/(2q))}} \right) \cdot \sqrt{k_l^{(-\kappa_2 - \mu\alpha/(2q))}}$$

the complexity of the estimate (3.5) *is bounded, up to a constant, from above by*

$$\mathscr{C}_{ML}(\varepsilon) \lesssim \begin{cases} \varepsilon^{-2\cdot\max\left(\frac{\kappa_1+1}{\mu(1+\alpha)},1\right)}, & 2\cdot q\cdot\kappa_2 < \mu\alpha \\ \varepsilon^{-2\cdot\frac{\kappa_1+1}{\mu(1+\alpha)}}, & 2\cdot q\cdot\kappa_2 = \mu\alpha \textit{ and } \frac{\kappa_1+1}{\mu(1+\alpha)} > 1 \\ \varepsilon^{-2}\cdot(\log\varepsilon)^2, & 2\cdot q\cdot\kappa_2 = \mu\alpha \textit{ and } \frac{\kappa_1+1}{\mu(1+\alpha)} \leq 1 \\ \varepsilon^{-2\cdot\max\left(\frac{\kappa_1+1}{\mu(1+\alpha)},1+\frac{\kappa_2-\mu\alpha/(2q)}{\mu(1+\alpha)}\right)}, & 2\cdot q\cdot\kappa_2 > \mu\alpha \end{cases} \tag{3.11}$$

We want to compare this result to the complexity $\mathscr{C}(\varepsilon)$ of the standard MC estimator. Since we remember that (2.72) was optimal, we know that in case of $\kappa_1 = \kappa_2 := \bar\kappa$, the computational gain $\mathscr{C}(\varepsilon)/\mathscr{C}_{ML}(\varepsilon)$ is asymptotically bounded by

$$\mathscr{R}(\varepsilon) := \begin{cases} \varepsilon^{-2\cdot\min\left(\frac{\bar\kappa}{\mu(1+\alpha)},1-\frac{1}{\mu(1+\alpha)}\right)}, & 2\cdot q\cdot\bar\kappa < \mu\alpha \\ \varepsilon^{-2\cdot\left(1-\frac{1}{\mu(1+\alpha)}\right)}, & 2\cdot q\cdot\bar\kappa = \mu\alpha \text{ and } \frac{\bar\kappa+1}{\mu(1+\alpha)} > 1 \\ \varepsilon^{-2\cdot\frac{\bar\kappa}{\mu(1+\alpha)}}, & 2\cdot q\cdot\bar\kappa = \mu\alpha \text{ and } \frac{\bar\kappa+1}{\mu(1+\alpha)} \leq 1 \\ \varepsilon^{-2\cdot\min\left(1-\frac{1}{\mu(1+\alpha)},\frac{\mu\alpha/(2q)}{\mu(1+\alpha)}\right)}, & 2\cdot q\cdot\bar\kappa > \mu\alpha \end{cases} \tag{3.12}$$

up to a logarithmic factor. We have the following easy rule of thumb.

Proposition 46. *In case of* $\kappa_1 = \kappa_2$ *(for example local regression or mesh method) and a good-natured problem, i.e.* $\alpha = 1$, *using the multilevel technique is recommended if*

$$\mu > 1/(2q). \tag{3.13}$$

In other words, the multilevel extension will further improve good algorithms and worsen bad algorithms. The last line of (3.12) will be particularly important as it is relevant for the “usual mesh case” as below.

Remark 47. *Let us define the "usual mesh case" to be the situation that $\kappa_1 = \kappa_2 = \alpha = \mu = 1$ and the payoff is bounded, i.e. Theorem 40 holds for all $q > 1$. In this case, we have*

$$\mathscr{C}(\varepsilon) \in O(\varepsilon^{-3}), \quad \mathscr{C}_{ML}(\varepsilon) \in O(\varepsilon^{-2.5}), \tag{3.14}$$

so the complexity gain is of order $\varepsilon^{-0.5}$.

In the best case, the gain can be of order ε^{-1}, see the third line of (3.12). To validate Theorem 32 in Section 2.2 and Theorem 45, two numerical examples are presented in the following. We also want to show that our results are efficient, in particular that Proposition 46 holds. We choose the mesh method that fulfills $\mu = 1$ and local regression with $\mu \approx 1/6$. Indeed, the numerical results will show that the orders of complexity that were calculated in the above theorems are achieved in practice.

3.1.1 Mesh Method

We consider a well-known benchmark example from Glasserman [40] and Broadie and Glasserman [19] considering a max-call option. A desired accuracy ε will be fixed and k^* and n^* are chosen for the standard MC algorithm according to (2.73). Since we only know the optimal order of these parameters, we have to find suitable constants. The root-mean-squared error is then measured based on many cycles of training and testing. The numerical result show that indeed the desired accuracy is achieved for different given ε. Afterwards, the same procedure is applied for the multilevel algorithm, where $\mathbf{n}$, $\mathbf{k}$ and L are chosen according to Theorem 45.

Benchmark Example 48. *(Bermudan max-call option) Suppose D underlying assets $X_t = (X_t^1, \dots, X_t^D)$, modeled by geometric Brownian motion under the risk-neutral measure, i.e.,*

$$dX_t^i = (r - \delta)X_t^i dt + \sigma X_t^i dW_t^i, \quad i = 1, \dots, D \tag{3.15}$$

where $r = 0.05$ is the risk-free interest rate, $\delta = 0.1$ the dividend yield, $\sigma = 0.2$ the volatility, $X_0 = (90, \dots, 90)$ the spot price and $W_t = (W_t^1, \dots, W_t^D)$ is a vector of D independent standard Brownian motions. At one of the $\mathcal{J} + 1$ equally distributed exercise dates $0 = t_0, \dots, t_{\mathcal{J}} = T$, the holder of the option may receive the payoff

$$g_t(X_t) = e^{-rt}(\max(X_t^1, ..., X_t^D) - \varkappa)^+,$$

where $\varkappa = 100$ is the strike price and $T = 3$ the time horizon. This example fulfills assumption (AB) with $\alpha = 1$.

In this section, that example will be used with $D = 5$ assets and $\mathcal{J} = 3$ exercise dates. For the true value, we have the 95% confidence interval $[15.995, 16.016]$, see Broadie and Glasserman [58]. To generate trajectories, we use the exact solution of the SDE

$$X_j^{(i)} = X_{j-1}^{(i)} \exp\left(\left[r - \delta - \frac{1}{2}\sigma^2\right](t_j - t_{j-1}) + \sigma\sqrt{(t_j - t_{j-1})} \cdot \xi_j^i\right)$$

$j = 1, \ldots, \mathcal{J}$, where ξ_j^i, are i.i.d. standard normal random variables. The weights inside the mesh are generated as explained in Proposition 20. The transition density of geometric Brownian motion is given by

$$p_j(x, y) = \prod_{i=1}^{D} p_j(x_i, y_i), \quad x = (x_1, \ldots, x_D), \quad y = (y_1, \ldots, y_D),$$

where

$$\begin{aligned} p_j(x_i, y_i) &= \frac{x_i}{y_i \sigma \sqrt{2\pi(t_j - t_{j-1})}} \times \\ &\quad \times \exp\left(\frac{-\left(\log\left(\frac{y_i}{x_i}\right) - \left(r - \delta - \frac{1}{2}\sigma^2\right)(t_j - t_{j-1})\right)^2}{2\sigma^2(t_j - t_{j-1})} \right). \end{aligned}$$

As explained in Proposition 20, we use the European max-call payoff of the next time step as an inner control. Additionally, the European value is used as outer control (martingale control) in level 0, i.e. we use Proposition 11 with

$$X^i = g_{\tau_0}(X_{\tau_0}^{(i)}), \quad Y^i = \mathcal{E}(X_{\tau_0}^{(i)}, \tau_0, T) \tag{3.16}$$

and $\mathcal{E}$ is taken from (2.41).

The standard Monte Carlo estimator is tested seven times with

$$\varepsilon \in \{0.64, 0.32, 0.16, 0.08, 0.04, 0.02, 0.01\},$$

$$k^* = (\varepsilon/2.4)^{-1}, \quad n^* = (\varepsilon/2.4)^{-2},$$

according to Theorem 32. We estimate the root-mean square error of V^{n^*,k^*} with 100 repetitions of training and testing and denote the result as $\sqrt{\text{mse}}$. The plot of the estimated quotient $\sqrt{\text{mse}}/\varepsilon$ is shown in the left of Figure 3.1.

Similarly, we calculate the quotient $\sqrt{\text{mse}}/\varepsilon$ for the multilevel estimator $V^{\mathbf{n}^*,\mathbf{k}^*}$ where $\mathbf{n}^* = (n_1^*, \ldots, n_l^*)$ and $\mathbf{k}^* = (k_1^*, \ldots, k_l^*)$ are defined as

$$\varepsilon = (0.8, 0.4, 0.2, 0.1, 0.05, 0.025)$$

$$k_l^* = k_0 \cdot \theta^l, \quad n_l^* = \frac{1}{(\varepsilon/8)^2} \left(\sum_{i=1}^{L} \sqrt{(k_i^*)^{1/f2}} \right) \sqrt{(k_l^*)^{-3/2}}, \quad l = 0, \ldots, L$$

with $k_0 = 5$, $\theta = 2$ and

$$L = \left\lceil \log_\theta \left(\frac{8 \cdot k_0}{\varepsilon} \right) \right\rceil.$$

according to Theorem 45. Remark 41 motivates $q = 1$. Following the “trick” as explained in Section 3.1, we need to construct the sequence of estimates

$$C_0^{k_l}(\cdot), \ldots, C_{\mathcal{J}}^{k_l}(\cdot), \quad l = L, \ldots, 0.$$

Each of them uses k_l training paths and serves as coarser approximation in level l and finer approximation in level $l - 1$.

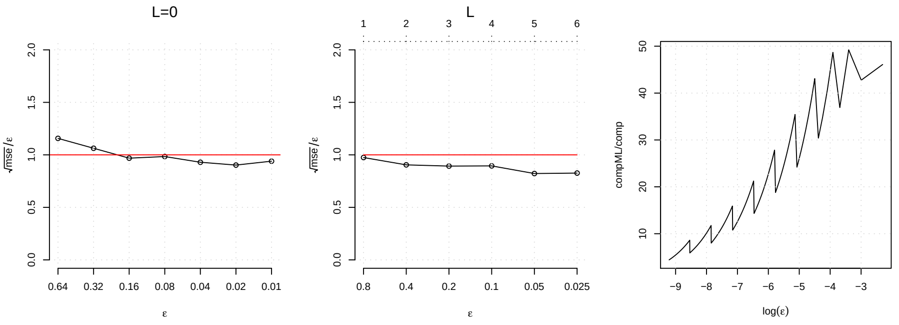

Figure 3.1: Root-mean-squared errors of the standard Monte Carlo estimator V^{n^*,k^*} (left) and the multilevel estimator $V^{\mathbf{n}^*,\mathbf{k}^*}$ (middle) measured in units of the expected error ε and the comparison of their complexities (right).

Remark 49. *Once the training step is completed for the best approximation in the highest level, i.e. the numbers $\zeta_j^{(i)}$ are glued to all the k_L paths at each time step, it is not necessary to accomplish the same procedure for $k_{L-1}, \dots, k_0$. It is better to use a subset of the training paths of the best approximation within the estimator* (2.38) *and also reuse the values $\zeta_j^{(i)}$ from that best approximation. This would reduce both the variance and the complexity of $V^{\mathbf{n},\mathbf{k}}$ and should be done in practice. However, it is not clear if the orders assumed in Theorem 38 would change, so in particular the results in Sections 3.2.1 and 3.2.1 are only valid without this reuse.*

Figure 3.1 suggests that the rates given in Theorem 18 and Theorem 45 do hold. In the two plots on the left, we see that the achieved accuracy is indeed a little below the given accuracy ε. For the multilevel case, the values of given ε are chosen in a way such that a new level has to be introduced for each of them. The small value of $\theta = 2$ allows us to obtain results for $L = 1, \dots, 8$. Of course, larger values of θ are recommended in practice for the multilevel technique to be efficient.

Let us consider the right hand side plot. It is clearly visible that introducing a new level at first increases the variance and then becomes fruitful as ε is further decreasing. This is why we recognize the plot as a so-called "sawtooth function". It also shows that the variance of the multilevel estimator becomes smaller compared to the variance of the standard MC estimator, as $\varepsilon \to 0$. However, the absolute value of the ratio is very high, because $\theta = 2$ is not at all optimal. In contrast ot the other two plots, this curve is based on the predicted behavior of the algorithms as in Theorem 45, instead of numerical simulation.

3.1.2 Local Regression

We want to perform the same test on the same example as in the previous section for the local regression technique in order to present a counterexample for the strength of the multilevel approach. The method used here is similar to the nearest-neighbours technique and can be seen as local polynomial regression

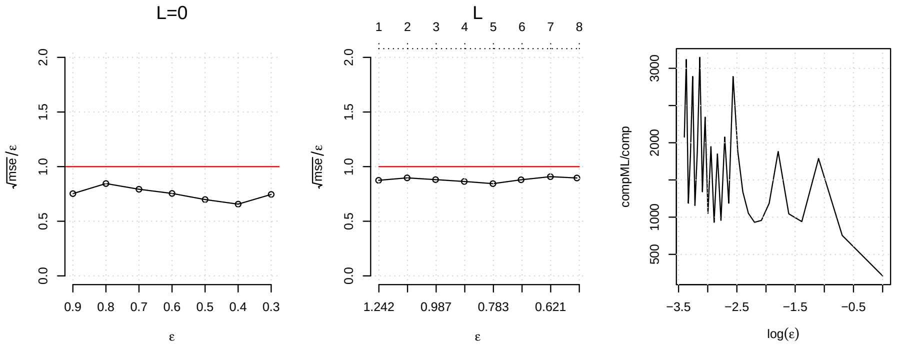

Figure 3.2: The same plots as in the mesh case in Figure 3.1.

with order zero, see Section 2.1.1. We want to use the simple kernel of the form

$$K(x) = 1_{\{|x|\le 1\}}, \quad x \in \mathbb{R}^D \tag{3.17}$$

(which corresponds to the Nadaraya-Watson type "false nearest neighbours technique") and have

$$C_j^k(x) = \sum_{i=1}^k w_j^{(i)}(x)\zeta_{j+1}^{(i)}, \quad j = 0, \ldots, \mathcal{J}-1 \tag{3.18}$$

where

$$w_j^{(i)}(x) = 1_{\left\{|x-Z_j^{(i)}|\le\delta_k\right\}} \Big/ \left(\sum_{l=1}^k 1_{\left\{|x-Z_j^{(l)}|\le\delta_k\right\}}\right) \tag{3.19}$$

with some bandwidth δ_k that decreases as the number of paths k grows. We find experimentally that a bias of order $k^{-1/6}$ can be achieved if $\delta_k = 100\cdot k^{-1/(D+2)}$, at least in this example. For any $\varepsilon \in \{0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3\}$ we set

$$k^* = (\varepsilon/1.2)^{-6}, \quad n^* = (\varepsilon/1.2)^{-2},$$

corresponding to the choice $\gamma_k = k^{-1/6}$ and estimate the root-mean squared error of the estimate V^{n^*,k^*} based on 100 repetitions of training and testing cycles, see left hand side of Figure 3.2. For the multilevel algorithm we take $\theta = 2$, $k_0 = 100$,

$$L = \left\lceil 6 \cdot \log_\theta\left(\frac{1}{(\varepsilon/3)\cdot k_0^{1/6}}\right)\right\rceil,$$

and

$$k_l^* = k_0\theta^l, \quad n_l^* = \frac{10}{(\varepsilon/3)^2}\left(\sum_{i=1}^L \sqrt{(k_i^*)^{11/12}}\right)\sqrt{(k_l^*)^{-13/12}}, \quad l = 0, \ldots, L.$$

The plot of the estimated quotient $\sqrt{\text{mse}}/\varepsilon$ is shown in the middle of Figure 3.2. Furthermore, one can see on the right hand side that the ratio of complexities $\mathscr{C}_{ML}(\varepsilon)/\mathscr{C}(\varepsilon)$ is not at all converging to zero. This is no surprise as $\mu = \frac{1}{6} \ngeq \frac{1}{2}$, see Proposition 46.

3.2 Practical Issues

The results so far are mostly of theoretical interest, as they are about the order of convergence and show that the multilevel algorithm is working. However, the numerical experiments from the previous section are hardly usable in practice. For example, the constants in Theorem 45 will not be known when attacking a new problem.

At first, we want to present an algorithm in Section 3.2.1 that determines the number of levels during the runtime. Furthermore, such a procedure could be improved by importance sampling, see Section 3.2.2. However, it is not at all clear that many levels will really be necessary. The choice $\theta = 2$ alone indicates the theoretical purpose. Much higher values of θ will be recommended instead of more levels to render the multilevel approach efficient. Therefore, we want to note some remarks about the gain of efficiency that can be obtained when using a finite number of levels in Section 3.2.3.

3.2.1 The Algorithm

In contrast to the examples that verify the complexity analysis, we now want to present an algorithm that can be used in practice. The algorithm below calculates the optimal "investment" of the computational complexity into the levels as well as the number of levels during the runtime. Roughly speaking, every time that a new level is introduced, the algorithm updates the number of testing paths in the different levels to optimize the variance with least computational complexity. Then, the error criterion (3.21) tells us whether to introduce a new level and so on.

Algorithm 1 Multilevel Algorithm

```
Set $L = 2$.
for For $l = 0, \dots, L$, do
  Generate $k_l$ training paths.
  Estimate continuation values $C_1^{k_l}, \dots, C_{\mathcal{J}}^{k_l}$.
  Generate $10^4$ testing paths and estimate the variance $\operatorname{Var} \Delta_l$.
  Calculate $n_l$, $l = 0, \dots, L$, according to

    $$n_l = \left\lceil 3 \cdot \varepsilon^{-2} \cdot \left( \sum_{i=1}^{L} \sqrt{k_i^{\kappa_2} \cdot \operatorname{Var} \Delta_i} \right) \cdot \sqrt{k_l^{-\kappa_2} \cdot \operatorname{Var} \Delta_l} \right\rceil \qquad (3.20)$$

  Estimate/update $\bar{\Delta}_0^{n_0}, \dots, \bar{\Delta}_L^{n_L}$.
  if

    $$\max(\bar{\Delta}_{L-1}^{n_{L-1}}/2, \bar{\Delta}_L^{n_L}) \leq \varepsilon/\sqrt{3}, \qquad (3.21)$$

  then
    Set $L = L + 1$.
  end if
end for
Return $\sum_{l=0}^{L} \bar{\Delta}_l^{n_l}$.
```

This method is similar to that of Giles [39]. However, there is still the difference caused by the randomness of the training paths. Despite our "trick" that only affects the variance of the whole algorithm, the values $\bar{\Delta}_l^{n_l}$ will still include variance due to the randomness of the training paths. To cope with this problem, we used the constant 3 instead of 2 for the criterion that determines whether to add another level. As before, we want to analyze the complexity of the multilevel estimator given the desired root-mean-squared error ε. While the exact cost of this multilevel algorithm is given by

$$\sum_{l=0}^{L} k_l^{\kappa_1+1} + n_l \cdot (k_l^{\kappa_2} + k_{l-1}^{\kappa_2}), \tag{3.22}$$

we have that the cost of the standard MC algorithm is of order

$$k_L^{\kappa_1+1} + 3 \cdot k_L^{\kappa_2} \cdot \varepsilon^{-2} \cdot \sum_{l=0}^{L} \operatorname{Var} \Delta_l. \tag{3.23}$$

Here, the variances of all levels are added to estimate the variance of a standard MC that only uses the finest approximation of the multilevel. These two complexities will be compared in the following two examples.

Mesh Method

The first numerical example is again about Benchmark Example 48 in the two-dimensional case with $\mathcal{J} = 9$. The true value is 8.08, see Glasserman [40]. We use the mesh method as before, but this time $b = 1.2$ is fixed for the control, see Remark 12. Thus, the estimator now reads

$$\begin{aligned} C_j^k(x) \quad = \quad & \sum_{i=1}^{k} w_j^{(i)}(x) \cdot \max\left(g_{j+1}\left(Z_{j+1}^{(i)}\right), C_{j+1}^k\left(Z_{j+1}^{(i)}\right)\right) \\ & - b \cdot \left(\exp\left(-r t_{j+1}\right) \max_{k=1,\ldots,d}\left(Z_{j+1}^{(i)} - \varkappa\right)^+ - \mathcal{E}(x, t_j, t_{j+1})\right), \end{aligned}$$

where $\mathcal{E}$ is the value of the European option as before, see (2.41). The control is also used as outer control, but this time for all the levels. Hence, we also have to change

$$\begin{aligned} \bar{\Delta}_l^{n_l} \quad = \quad & \frac{1}{n_l} \sum_{r=1}^{n_l} \left[g_{\tau_{k_l}^{(r)}}\left(X_{\tau_{k_l}^{(r)}}^{(r)}\right) - b \cdot \left(\exp\left(-r t_{\tau_{k_l}^{(r)}}\right) \max_{k=1,\ldots,d}\left(X_{\tau_{k_l}^{(r)}}^k - \varkappa\right)^+ - \mathcal{E}\left(x_0, 0, T\right)\right) \right. \\ & \left. - g_{\tau_{k_{l-1}}^{(r)}}\left(X_{\tau_{k_{l-1}}^{(r)}}^{(r)}\right) + b \cdot \left(\exp\left(-r t_{\tau_{k_{l-1}}^{(r)}}\right) \max_{k=1,\ldots,d}\left(X_{\tau_{k_{l-1}}^{(r)}}^k - \varkappa\right)^+ - \mathcal{E}\left(x_0, 0, T\right)\right) \right] \end{aligned}$$

when comparing to (3.5). We choose $k_l = 20 \times 2^l$ and generate paths from the exact solution of the SDE of geometric Brownian motion as in Section 3.1.1.

In the upper left corner of Figure 3.3, the logarithms of the "increments" Δ_l are plotted against the logarithms of the number of training paths. In the right upper corner, the same is done for the variances of Δ_l. Those data are mean values with respect to many cycles of training and testing, as the

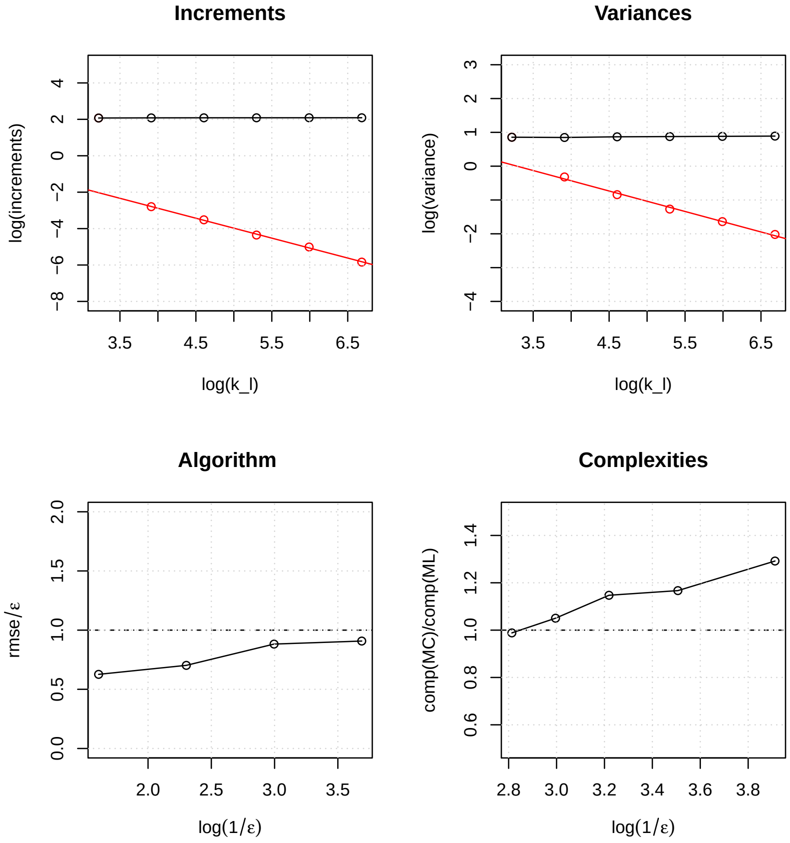

Figure 3.3: Mesh method: The red regression lines indicate an estimated decay rate of increments of -0.6054 and an estimated decay rate of variances of -1.0941.

variance of the training paths causes the increments and variances to be random variables. It turns out that the corresponding fitted regression lines are given by $-0.6054 \cdot l + 0.0596$ and $-1.0941 \cdot l - 2.9253$. Thus, they are in agreement with our theoretical analysis that provides the rates -1 and -0.5.

The algorithm from Section 3.2.1 is tested for

$$\varepsilon = 0.2, 0.1, 0.05, 0.025.$$

Indeed, the results stay below the desired accuracy, see lower left corner of Figure 3.3. Furthermore, as ε decreases the complexity of this algorithm becomes better and better better when compared to the complexity of the suitable standard Monte Carlo algorithm via (3.23). This is indicated in the right lower corner for

$$\varepsilon = 0.02, 0.03, 0.04, 0.05, 0.06.$$

This choice of ε is now different, because the corresponding average number of levels in use are

$$4.60, 4.33, 4.01, 4.05, 3.55,$$

so this is the interesting region, where the number of levels is exploding.

Global Regression

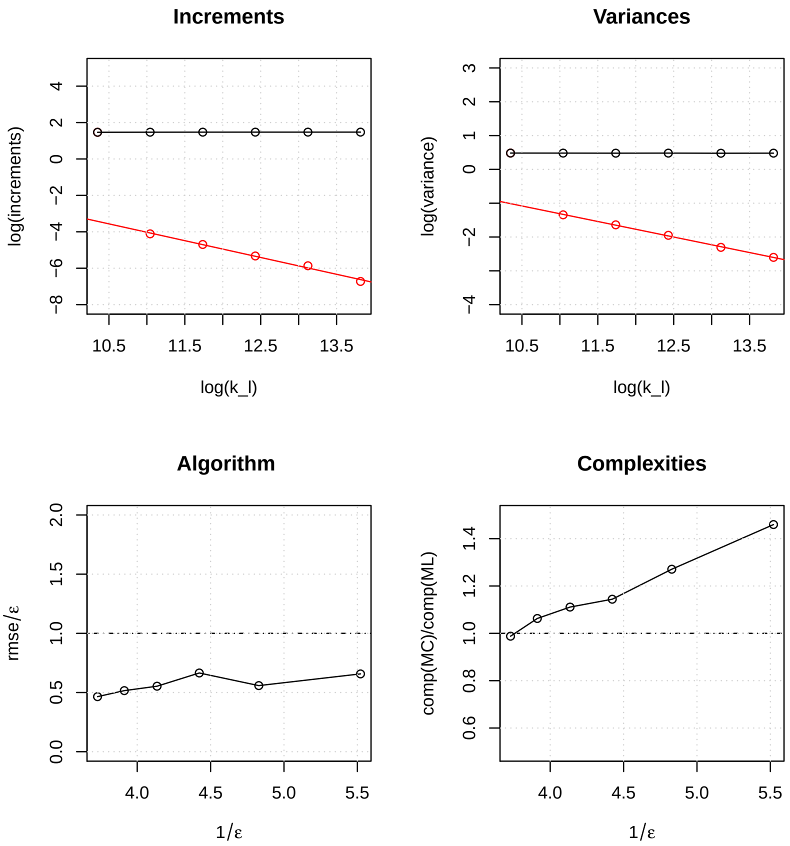

Figure 3.4: Global regression. The red regression lines indicate an estimated decay rate of increments of -0.4592 and an estimated decay rate of variances of -0.9230.

The second experiment is about the same example as before, but with $\mathcal{J} = 9$ and $D = 1$. In one dimension, the corresponding European option can be evaluated analytically by means of the well-known Black-Scholes formula.

Remark 50. *(Discounted Black-Scholes Formula) For an asset X modeled via*

$$dX_t = (r - \delta)X_t dt + \sigma X_t dW_t,$$

i.e. geometric Brownian motion with drift $r - \delta$ and volatility σ, the value of the European call option with spot price x at initial time t, strike κ and maturity T is given by

$$\mathcal{E}(x, t, T) = e^{-\delta T - t(r-\delta)} \left(x_0 \Phi(d_1) - \varkappa e^{(r-\delta)(T-t)} \Phi(d_2) \right)$$

in time-0 dollars, where

$$d_1 = \frac{\log(x_0/\varkappa) + (r - \delta + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}$$

and $d_2 = d_1 - \sigma\sqrt{T-t}$.

We perform regression on piecewise constant basis functions. Fix a number of basis functions m, set $\Delta = 100/m$ and define

$$\psi_i(x) = \begin{cases} 0, & x - 50 > (i-1)\Delta, \\ 1, & \text{otherwise}, \\ 0, & x - 50 \leq i\Delta \end{cases}$$

for all $i = 1, \dots, m$. Given a sequence of natural numbers $m(k)$, $k \in \mathbb{N}$, the continuation values estimates are

$$C_j^k(x) = \begin{cases} \mathcal{E}(x, j, \mathcal{J}), & x < 50, \\ \sum_{i=1}^{m(k)} \alpha_i \psi_i(x), & \text{otherwise} \ , \\ \mathcal{E}(x, j, j+1), & x > 150, \end{cases} \tag{3.24}$$

Truncation at $x = 150$ is allowed, as it is clear that this region will belong to the exercise region. Thus, it is optimal to exercise as soon as possible and the continuation value is equal to the corresponding European payoff with maturity at the next exercise date. Similarly, for $x < 50$ the option is very unlikely to provide any money again. It will be exercised at T if at all.

This truncation procedure is very important. Especially for very high values of x, there will only be very few training paths in the vicinity, so the sample error will be very high. This will have influence on the next regression in the previous time step and will thus spoil the results. Furthermore, we will randomize the initial value of the training paths which now start at $\tilde{x}_0 = x_0 \exp(0.4 \times \xi)$ to ensure a dense distribution in all areas of interest, where ξ is an independent standard normal variable. Thus, we can be sure that only the sample error and the approximation error and no other effects will be decisive. Zanger [77] also uses some similar truncation procedure, see Section 2.1.2. The same control $\mathcal{E}$ as before is used both as outer and inner control and is very strong in the one-dimensional case. The results provide lower bounds that differ less than 0.01% from the true value.

To test the multilevel, the number of training paths in each level is chosen to be

$$k_l = 31250 \times 2^l, \quad l = 1, \dots, L,$$

while the sequence $m(k)$ is given by $m(k) = \lceil 7 \cdot k^\rho \rceil$ with $\rho = 0.5$.

Figure 3.4 contains the same results as the plot before. Now, the red regression lines indicate $\mu = -0.923$ and the dacay of variance is approximately one half of this. This time, the algorithm is tested for

$$\varepsilon = 0.004, 0.008, 0.012, 0.016, 0.020, 0.024$$

for both lower plots. Whereas we used $\theta = 4$ for the right hand plot to make the multilevel efficient. The corresponding average number of levels in use are

$$5.00, 4.42, 4.1, 3.94, 3.67, 3.30.$$

3.2.2 Importance Sampling

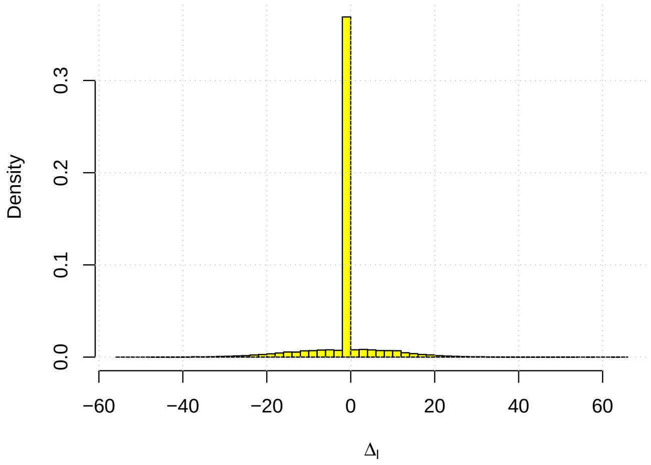

Figure 3.5: Histogram of the r.v. Δ_l based on 10000 realisations. An atom in 0 is clearly visible.

When looking at the the density of the random variable

$$\Delta_l = g_{\tau_{k_l}}(X_{\tau_{k_l}}) - g_{\tau_{k_{l-1}}}(X_{\tau_{k_{l-1}}}),$$

one will immediately notice that Δ_l vanishes for 80%-90% of the testing paths, as shown in the histogram in Figure 3.5. Thus, it is convenient to improve the efficiency of the multilevel algorithm by importance sampling, i.e. generating only testing paths that lead to $\Delta_l \neq 0$. Let us fix some $l > 0$ and define

$$\mathcal{I}_l := \{\tau_{k_{l-1}} \neq \tau_{k_l}\},$$

such that $\{\Delta_l \neq 0\} \subset \mathcal{I}_l$. These two sets will almost be equal in a non-degenerate setting. Generating only paths that cause different stopping times, which means sampling conditional $\mathcal{I}_l$, would mean a change of measure from P to Q via

$$\frac{d\mathrm{Q}}{d\mathrm{P}}(\omega) = \begin{cases} 1/\mathrm{P}(\mathcal{I}_l), & \omega \in \mathcal{I}_l, \\ 0, & \omega \in \Omega \setminus \mathcal{I}_l, \end{cases}.$$

For a set of testing paths $\widetilde{X}_\cdot^{(1)}, \ldots, \widetilde{X}_\cdot^{(r)}$ generated under Q , the unbiased Monte-Carlo estimator for $\mathrm{E}[\Delta_l] = \mathrm{E}_\mathrm{P}[\Delta_l]$ is

$$\widetilde{\Delta}_l^{n_l} = \frac{1}{n_l} \sum_{r=1}^{n_l} \mathrm{P}(\mathcal{I}_l) \left\{ g_{\tau_l^{(r)}}\left(\widetilde{X}_{\tau_l^{(r)}}^{(r)}\right) - g_{\tau_{l-1}^{(r)}}\left(\widetilde{X}_{\tau_{l-1}^{(r)}}^{(r)}\right) \right\} \tag{3.25}$$

Moreover, it holds

$$\mathrm{Var}_\mathrm{Q}[\Delta_l] = \mathrm{E}_\mathrm{Q}[\Delta_l^2] - \mathrm{E}_\mathrm{Q}^2[\Delta_l] = \mathrm{E}_\mathrm{P}\left[\Delta_l^2 \frac{d\mathrm{Q}}{d\mathrm{P}}\right] - \mathrm{E}_\mathrm{P}^2\left[\Delta_l \frac{d\mathrm{Q}}{d\mathrm{P}}\right]$$

$$
\begin{aligned}
&= \frac{1}{\mathrm{P}(\mathcal{I}_l)} \cdot \mathrm{E}_{\mathrm{P}}[\Delta_l^2] - \left(\frac{1}{\mathrm{P}(\mathcal{I}_l)} \cdot \mathrm{E}_{\mathrm{P}}[\Delta_l]\right)^2 \\
&= \frac{1}{\mathrm{P}(\mathcal{I}_l)} \left(\mathrm{Var}_{\mathrm{P}}[\Delta_l] + \mathrm{E}_{\mathrm{P}}^2[\Delta_l]\right) - \left(\frac{1}{\mathrm{P}(\mathcal{I}_l)} \mathrm{E}_{\mathrm{P}}[\Delta_l]\right)^2 \\
&= \frac{1}{\mathrm{P}(\mathcal{I}_l)} \mathrm{Var}_{\mathrm{P}}[\Delta_l] + \underbrace{\mathrm{E}_{\mathrm{P}}^2[\Delta_l] \left(\frac{1}{\mathrm{P}(\mathcal{I}_l)} - \frac{1}{\mathrm{P}(\mathcal{I}_l)^2}\right)}_{\leq 0} \qquad (3.26)
\end{aligned}
$$

and as a consequence

$$\mathrm{Var}_{\mathrm{Q}}[\Delta_l] \leq \frac{1}{\mathrm{P}(\mathcal{I}_l)} \mathrm{Var}_{\mathrm{P}}[\Delta_l].$$

The last inequality is quite tight, as $E_{\mathrm{P}}^2[\Delta_l]$ in (3.26) will be very small and will converge to zero at a higher order. As a result we have

$$\mathrm{Var}_{\mathrm{Q}}[\widetilde{\Delta}_l^n] \leq \mathrm{P}(\mathcal{I}_l)\, \mathrm{Var}[\bar{\Delta}_l^{n_l}], \qquad (3.27)$$

meaning that importance sampling reduces the variance by a factor of at least $\mathrm{P}(\mathcal{I}_l)$.

Proposition 51. *Asymptotically, we have* $\mathrm{P}(\mathcal{I}_l) \lesssim \gamma_{k_l}^{\alpha/2}$*, which implies* $P(\mathcal{I}) \in O(k_{l-1}^{-1/2})$ *in case of* $\mu = 1$ *and* $\alpha = 1$*.*

Inserting this result into Theorem 45 could even lead to a reduction of the order of complexity of $V^{\mathbf{k},\mathbf{n}}$. In the usual mesh case, see Remark 47, it could lead to $\mathscr{C}_{ML}(\varepsilon) \lesssim \varepsilon^{-2}$. Unfortunately, sampling directly from Q is not possible. Determining whether ω belongs to $\mathcal{I}_l$ requires generating a path and evaluating the corresponding stopping time. Hence, collecting a set of trajectories and pick the interesting ones would take $1/\mathrm{P}(\mathcal{I}_l)$ times longer and there is no benefit. Another problem is that the factor $\mathrm{P}(\mathcal{I}_l)$ is not known analytically, so samples drawn from $\Omega \setminus \mathcal{I}_l$ will be needed anyway to estimate it. Though this idea is practically unprofitable, it motivates the NCMC presented in Chapter 4.

3.2.3 Quantitative Gain

We will note that many levels will only be optimal for extremely precise calculations, since the introduction of a new level always adds new variance to the estimator. This will only become worthwhile, if a lot of testing paths are used in the new level. The right hand side in Figure 3.1 illustrates this effect. Corollary 52 says that using more than 2 oder 3 levels is unlikely to be advantageous with respect to the order of convergence, because the latter will already be very near to 2.5 then. The corollary can be inferred from Theorem 99 in Section 6.2 by inserting $\gamma = 1$ and $\beta = 1/2$. It uses Definition 44 and it is relevant in the particularly important case of the mesh estimator, see Remark 47.

Theorem 52. *In case of a finite number of levels* $L > 0$ *and* $\mu = \alpha = \kappa_1 = \kappa_2 = 1$*, the complexity of the multilevel estimator is given by*

$$\mathscr{C}_L(\varepsilon) = \varepsilon^{-\frac{5\times 2^L - 2}{2^{L+1}-1}} \qquad (3.28)$$

and the optimal choice of training and testing paths is given asymptotically up to a constant by

$$k_l = \varepsilon^{-\frac{0.5^l - 2}{0.5^L - 2}}, \quad n_l = \varepsilon^{\frac{2 - 2\times 0.5^{L+1} + 0.5^{l+1} - 0.5}{0.5^{L+1} - 1}}. \tag{3.29}$$

Table 3.1 shows the results from Theorem 52 in case of $L = 0, \ldots, 3$ and we see that in case of $L = 2$, we have

$$k_0 = \varepsilon^{-2/3}, \;\; k_1 = \varepsilon^{-1}, \;\; n_0 = \varepsilon^{-2}, \;\; n_1 = \varepsilon^{-5/3}. \tag{3.30}$$

and $\mathscr{C}_L(\varepsilon) \in O(\varepsilon^{-2.571})$. The difference to an order of 2.5 will already be very difficult to measure.

L	k_0	k_1	k_2	k_3	n_0	n_1	n_2	n_3	$-\log(\mathscr{C}_L(\varepsilon))$
0	-1				-2				3
1	-2/3	-1			-2	-5/3			$\frac{8}{3} = 2.667$
2	-4/7	-6/7	-1		-2	-12/7	-11/7		$\frac{18}{7} = 2.571$
3	-8/15	-4/5	-14/15	-1	-2	-26/15	-8/5	-23/15	$\frac{38}{15} = 2.533$

Table 3.1: The optimal asymptotic setting for the multilevel technique given in powers of ε. The next values of $-\log(\mathscr{C}_L(\varepsilon))$ are $2.516, 2.508, 2.504$.

Not only the order of complexity, but also the absolute gain of complexity can be very pleasant for small L. This is a big difference to the multilevel approach for nested dual methods in Chapter 6. The reason is just that increasing the number of subsimulations in case of a dual nested method as presented in Chapter 5 not only improves the bias but also reduces the variance at the same time if a good martingale is used. In case of fast approximation methods for lower bounds, improving the bias will not lead to variance reduction. Hence, the multilevel technique is a convenient tool to achieve the latter.

As stated before, Table 3.1 tells us to use less levels and higher θ. We want to analyze the absolute gain of efficiency by considering the variance given the desired bias for $L = 1$. The multilevel estimator $V^{\mathbf{k},\mathbf{n}}$ is compared to a standard Monte Carlo estimator $V^{K,N}$, where k_1, k_2, n_1, n_2, K, N are chosen such that both estimators will have approximately the same complexity and $k_L = K$, which ensures the same bias. A stochastic mesh for Benchmark Example 48 will illustrate this. The mesh method is used according to Table 3.2, so to say $\theta = 100$. In Figure 3.6, we see the clear improvement due to the multilevel technique and the reason is simple: The stopping rule hardly becomes better when changing from $k_0 = 100$ to $k_1 = 10000$, but the complexity changes tremendously.

By assuming that the variances in higher levels $l > 0$ are nearly zero, we obtain an upper bound for the gain of variance in such a setting. It is bounded from above by $(\frac{k_L}{k_0})^{\kappa_2}$. This would mean a maximal gain of variance by a factor of 100 in our example. Indeed, 2.56% is quite near to this optimal result.

Level		MC	ML	ML Δ_l
$l = 0$	training k_0	10,000	100	$\mathrm{Var}[\Delta_0] \approx 0.01658$
	testing n_0	10,000	500,000	
$l = 1$	training k_1		10,000	$\mathrm{Var}[\Delta_1] = 0.00356$
	testing n_1		5,000	

Table 3.2: Setting of the multilevel algorithm. About 50% of the complexity is invested in each of the two levels.

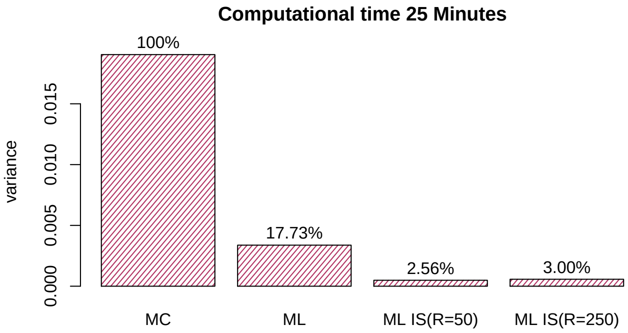

Figure 3.6: Comparing Monte Carlo to multilevel Monte Carlo according to Table 3.2. "ML IS" denotes importance sampling, but those results were based on NCMC with different parameters R, see Chapter 4.

3.3 Proofs

Proof of Theorem 40

We have

$$\begin{aligned}\mathrm{E}\left[g_{\tau_{k_l}}\left(X_{\tau_{k_l}}\right) - g_{\tau_{k_{l-1}}}\left(X_{\tau_{k_{l-1}}}\right)\right]^2 \quad \leq \quad & 2\,\mathrm{E}\left[g_{\tau^*}\left(X_{\tau^*}\right) - g_{\tau_{k_{l-1}}}\left(X_{\tau_{k_{l-1}}}\right)\right]^2 \\ & +2\,\mathrm{E}\left[g_{\tau_{k_l}}\left(X_{\tau_{k_l}}\right) - g_{\tau^*}\left(X_{\tau^*}\right)\right]^2.\end{aligned}$$

It follows from Lemma 36 that

$$\begin{aligned}&\mathrm{E}\left[g_{\tau^*}\left(X_{\tau^*}\right) - g_{\tau_{k_{l-1}}}\left(X_{\tau_{k_{l-1}}}\right)\right]^2 \leq \\ &\qquad\qquad \mathrm{E}\left\{\sum_{s=0}^{\mathcal{J}-1} 2^s \xi_s \left(1_{\{\tau_s^*=s,\tau_{k_{l-1},s}>s\}} + 1_{\{\tau_s^*>s,\tau_{k_{l-1},s}>s\}}\right)\right\}\end{aligned}$$

with $\xi_s = \mathrm{E}_{\mathcal{F}_s}\left[\left|g_s(X_s) - V_{s+1}(X_{s+1})\right|^2\right]$. We use the same notation as in the proof section of the previous chapter and with the Hölder inequality we get

$$\begin{aligned}
&\mathrm{E}\left[\xi_s\left(1_{\{\tau_s^*=s,\tau_{k_{l-1},s}>s\}} + 1_{\{\tau_s^*>s,\tau_{k_{l-1},s}>s\}}\right)\right] \\
&\qquad \leq 4M_p^{1/p}\left[\mathrm{P}\left(\tau_s^* = s, \tau_{k_{l-1},s} > s\right) + \mathrm{P}\left(\tau_s^* > s, \tau_{k_{l-1},s} = s\right)\right]^{1/q} \\
&\qquad\qquad = 4M_p^{1/p}\left[\mathrm{P}(\mathcal{E}_{k_{l-1},s})\right]^{1/q}
\end{aligned}$$

with $1/p + 1/q = 1$, since

$$\begin{aligned}
\mathrm{E}\left[|\xi_s|^p\right] &\leq \mathrm{E}\left[\left|g_s(X_s) - V_{s+1}(X_{s+1})\right|^{2p}\right] \\
&\leq 2^{2p-1}\mathrm{E}\left[\left|g_s(X_s)\right|^{2p}\right] + 2^{2p-1}\mathrm{E}\left[\left|V_{s+1}(X_{s+1})\right|^{2p}\right] \\
&\leq 2^{2p-1}\mathrm{E}\left[\left|g_s(X_s)\right|^{2p}\right] + 2^{2p-1}\mathrm{E}\left[\left|\max_{k=s+1,\ldots,\mathcal{J}} g_k(X_k)\right|^{2p}\right] \\
&\leq 2^{2p}M_p.
\end{aligned}$$

We now have

$$\mathrm{E}\left[g_{\tau^*}\left(X_{\tau^*}\right) - g_{\tau_{k_{l-1}}}\left(X_{\tau_{k_{l-1}}}\right)\right]^2 \leq C_p^{1/p}\mathrm{E}\left[\sum_{i=0}^{\infty}\sum_{s=0}^{\mathcal{J}-1} 2^s\left[\mathrm{P}(\mathcal{E}_{k_{l-1},s} \cap \mathcal{A}_{k_{l-1},s,i})\right]^{1/q}\right]$$

and because of the bounds for

$$\mathrm{P}(\mathcal{E}_{k_{l-1},s} \cap \mathcal{A}_{k_{l-1},s,i})$$

from the proof of Theorem 37, we have as a result

$$\mathrm{E}\left[g_{\tau^*}\left(X_{\tau^*}\right) - g_{\tau_{k_{l-1}}}\left(X_{\tau_{k_{l-1}}}\right)\right]^2 \leq C\gamma_{k_{l-1}}^{\alpha/(2q)}$$

for some $C > 0$.

Proof of Theorem 45

At first we want to determine L via

$$\gamma_{k_L}^{(1+\alpha)/2} = k_L^{-\mu(1+\alpha)/2} = \left(k_0 \cdot \theta^L\right)^{-\mu(1+\alpha)/2} \lesssim \varepsilon \tag{3.31}$$

to ensure that the bias is small enough, so we choose

$$L = \frac{2}{\mu(1+\alpha)}\log_\theta\left(\varepsilon^{-1} \cdot k_0^{-\mu(1+\alpha)/2}\right).$$

That also ensures that

$$\gamma_{k_L}^{2(1+\alpha)} = k_L^{-\mu 2(1+\alpha)} \lesssim \varepsilon$$

which is part of the variance we have to consider. To minimize the computational cost given the rest of ε^2, we solve the following optimization problem:

$$\sum_{l=0}^{L} k_l^{\kappa_1+1} + n_l \cdot k_l^{\kappa_2} \to \min \tag{3.32}$$

$$\sum_{l=1}^{L}\frac{\gamma_{k_{l-1}}^{\alpha/(2q)}}{n_l} \asymp k_0^{-\mu\alpha/2} \cdot \sum_{l=1}^{L}\frac{\theta^{-l\mu\alpha/(2q)}}{n_l} \asymp \varepsilon^2. \tag{3.33}$$

$$n_0 \asymp \varepsilon^{-2} \tag{3.34}$$

Now the Lagrange multiplier method with respect to n_l gives us

$$k_l^{\kappa_2} = -\lambda \frac{k_l^{-\mu\alpha/(2q)}}{n_l^2} \Rightarrow n_l = \sqrt{(-\lambda) \cdot k_l^{(-\kappa_2 - \mu\alpha/(2q))}}.$$

Now one can put the value of n_l in (3.33):

$$\sum_{l=1}^{L} \frac{\gamma_{k_{l-1}}^{\alpha/(2q)}}{n_l} \asymp \sum_{l=1}^{L} \frac{k_l^{-\mu\alpha/(2q)}}{\sqrt{(-\lambda) \cdot k_l^{(-\kappa_2-\mu\alpha/(2q))}}} \asymp \varepsilon^2 \ \Rightarrow$$

$$\sqrt{(-\lambda)} = \varepsilon^{-2} \cdot \sum_{l=1}^{L} \sqrt{k_l^{(\kappa_2-\mu\alpha/(2q))}} \ \Rightarrow$$

$$n_l = \varepsilon^{-2} \left(\sum_{i=1}^{L} \sqrt{k_i^{(\kappa_2-\mu\alpha/(2q))}} \right) \cdot \sqrt{k_l^{(-\kappa_2-\mu\alpha/(2q))}}.$$

Now we can rewrite (3.32) as

$$\sum_{l=0}^{L} k_l^{\kappa_1+1} + n_l \cdot k_l^{\kappa_2} \asymp k_L^{\kappa_1+1} + \varepsilon^{-2} \cdot \left(\sum_{l=1}^{L} \sqrt{k_l^{(\kappa_2-\mu\alpha/(2q))}} \right)^2 + \varepsilon^{-2} \cdot k_0^{\kappa_2},$$

so we will have three cases.

Case 1. $2 \cdot q \cdot \kappa_2 = \mu\alpha$.

$$\begin{aligned} k_L^{\kappa_1+1} + \sum_{l=0}^{L} n_l \cdot k_l^{\kappa_2} &\lesssim k_L^{\kappa_1+1} + \varepsilon^{-2} \cdot L^2 \\ &\lesssim \varepsilon^{-\frac{2\cdot(\kappa_1+1)}{\mu(1+\alpha)}} + \varepsilon^{-2} \cdot L^2 \end{aligned}$$

Case 2. $2 \cdot q \cdot \kappa_2 < \mu\alpha$.

$$\begin{aligned} k_L^{\kappa_1+1} + \sum_{l=0}^{L} n_l \cdot k_l^{\kappa_2} &\lesssim k_L^{\kappa_1+1} + \varepsilon^{-2} \\ &\lesssim \varepsilon^{-\frac{2\cdot(\kappa_1+1)}{\mu(1+\alpha)}} + \varepsilon^{-2} \end{aligned}$$

Case 3. $2 \cdot q \cdot \kappa_2 > \mu\alpha$.

$$\begin{aligned} k_L^{\kappa_1+1} + k_L^{\kappa_1+1} + \sum_{l=0}^{L} n_l \cdot k_l^{\kappa_2} &\lesssim k_L^{\kappa_1+1} + \varepsilon^{-2} \cdot k_L^{\kappa_2-\mu\alpha/(2q)} \\ &\geq \varepsilon^{-\frac{2\cdot(\kappa_1+1)}{\mu(1+\alpha)}} + \varepsilon^{-2-\frac{2\kappa_2-\mu\alpha/q}{\mu(1+\alpha)}} \end{aligned}$$

Proof of Proposition 51

We define

$$\begin{aligned} d_j(C_j^{k'}(X_j), C_j^k(X_j)) &= \{C_j^k(X_j) < g_j(X_j),\ C_j^{k'}(X_j) \geq g_j(X_j)\} \\ &\cup \ \{C_j^k(X_j) \geq g_j(X_j),\ C_j^{k'}(X_j) < g_j(X_j)\} \end{aligned}$$

to be the event that the stopping times give us different advices whether to stop at time step j. Since $d_j\left(C_j^{k'}(X_j), C_j^k(X_j)\right) \subset \mathcal{E}_{k,j} \cup \mathcal{E}_{k',j}$, see Theorem 37, it follows that

$$\begin{aligned}
P(\tau_{k'} \neq \tau_k) &= 1 - \prod_{j=1}^{\mathcal{J}} \left(1 - d_j(C_j^{k'}(X_j), C_j^k(X_j))\right) \leq 1 - \prod_{j=1}^{\mathcal{J}} (1 - P(\mathcal{E}_{k',j} \cup \mathcal{E}_{k,j})) \\
&\leq 1 - \left(1 - \sum_{j=1}^{\mathcal{J}} P(\mathcal{E}_{k,j}) + P(\mathcal{E}_{k',j})\right)^{\mathcal{J}} \lesssim \sum_{j=1}^{\mathcal{J}} P(\mathcal{E}_{k,j}) + P(\mathcal{E}_{k',j}) \\
&\lesssim \gamma_k^{\alpha/2}\mathcal{J} + \gamma_{k'}^{\alpha/2}\mathcal{J} \lesssim \gamma_{\min(k,k')}^{\alpha/2}\mathcal{J}.
\end{aligned}$$

Chapter 4

Nested Conditional Monte Carlo

In this chapter, we want to present a method for efficiently "comparing" two stopping times. Here, comparing means that we are interested in estimating

$$\Delta = \mathrm{E}[Y_{\tau^A}] - \mathrm{E}[Y_{\tau^B}],$$

where Y_j is a stochastic process in discrete time and τ^A and τ^B are two stopping times. This task is motivated, inter alia, by the multilevel technique in Chapter 3, where $Y_j = g_j(X_j)$ is the payoff of an underlying asset X_j and τ^A and τ^B are stopping times of different accuracy.

A simple Monte Carlo algorithm for this problem consists of simulating n trajectories of Y_j until both stopping times have occurred and then averaging over the resulting realizations of $Y_{\tau^A} - Y_{\tau^B}$, that is using

$$\frac{1}{n}\sum_{i=1}^{n} Y^{(i)}_{\tau^{A(i)}} - Y^{(i)}_{\tau^{B(i)}} \tag{4.1}$$

as an estimate for Δ. However, many of the samples of Y_j might contribute zero to the sum in this Monte Carlo estimator, especially in all the cases that τ^A and τ^B happen to be the same (see Section 3.2.2). Since it would be more efficient to generate more samples that induce different τ^A and τ^B, we write

$$\Delta = \mathrm{E}\left[\mathrm{E}\left[Y_{\tau^A} - Y_{\tau^B} | \mathcal{F}_{\tau^\wedge}\right]\right], \quad \text{where} \quad \tau^\wedge = \min(\tau^A, \tau^B), \tag{4.2}$$

where $\mathcal{F}_{\tau^\wedge}$ denotes the the $\sigma-$algebra generated by the information until the first of the stopping times. Now, simulate n trajectories of Y until the first stopping time occurs, i.e., until $\tau^\wedge$. Then simulate R conditionally independent copies of each of the n trajectories until the second stopping time $\tau^\vee = \max(\tau^A, \tau^B)$ occurs and estimate Δ by the mean of the $R \cdot n$ realizations of $Y_{\tau^A} - Y_{\tau^B}$. The resulting estimator can be interpreted as estimating first for each of the n initial trajectories the inner conditional expectation in (4.2) by the mean over the R replications of that trajectory, and then averaging over the n initial trajectories to estimate the outer expectation.

In the algorithm, we use the same number of replications on each path. Yet, as demonstrated in a different type of application in Broadie, Du and Moallemi

[18], numerical efficiency can be improved by allocating more replications to "critical" trajectories. An extension of the method which achieves this – while retaining unbiasedness – splits the trajectories at every time point between $\tau^\wedge$ and $\tau^\vee$. In this way, trajectories with a large value of $\tau^\vee - \tau^\wedge$ are automatically investigated more intensively. Alternatively – if one is not concerned about a small bias – combinations of the method with importance sampling might be fruitful. We leave these and further extensions and applications of the method to future research.

The idea of using subsimulations to estimate an inner conditional expectation relates the approach to the literature on Nested Simulation, namely Gordy and Juneja [45] and Broadie, Du, and Moallemi[18]. In the applications considered there, there is a non-linear dependence on the inner conditional expectation so that the inner simulations are indispensable – and are generally regarded as an unavoidable burden. From this perspective, it is interesting to see that in the numerical examples, where inner simulations are introduced deliberately as a variance reduction technique, the estimated optimal numbers of inner paths do not differ much from what is typically used in these applications, e.g. $R = 100$.

Variance reduction by deliberately inserting a conditional expectation is a common technique if these conditional expectations are available in closed form. This classical method is known as Rao-Blackwellization or Conditional Monte Carlo, see Boyle, Broadie and Glasserman [15] and Asmussen and Binswanger [3]. It is usually not applicable in our setting since closed-form expressions for expectations of stopped processes are rare or at least very costly to evaluate. Since the method mimics Conditional Monte Carlo by Nested Simulation, we refer to it as "Nested Conditional Monte Carlo".

In a sense, the algorithm closely resembles the splitting algorithms for rare event simulation studied, e.g., in Villén-Altamirano and Villén-Altamiranoe [74] and Glasserman, Heidelberger and Shahabuddin [41]. In the applications considered in this literature (e.g. barrier option pricing or estimating the probability of large losses), the rare event typically consists of Y taking exceptionally large or small values. Thus, the trigger events for replicating a trajectory are chosen as the hitting times of some threshold value of Y. In this way, computational effort can be allocated efficiently to the regions where it is needed the most.

A particular advantage of splitting methods such as ours is that they do not change the expectation of the estimator. This will not be the case for related algorithms such as importance sampling or particle methods, see the survey of Carmona, Del Moral, Hu and Oudjane [22]. Unbiasedness is of particular importance in option pricing applications, as one needs unbiased estimators for lower and upper bounds to construct confidence intervals.

4.1 The Algorithm

Let Y_j, $j = \{0, 1, \ldots, \mathcal{J}\}$ be a square-integrable, real-valued stochastic process adapted to a complete filtered probability space $(\Omega, \mathcal{F}, P)$. We want to estimate

$$\Delta = \mathrm{E}[Y_{\tau^A} - Y_{\tau^B}],$$

where τ^A and τ^B are stopping times on $(\Omega, \mathcal{F}, P)$. Therefore, we define the stopping times $\tau^\wedge = \min(\tau^A, \tau^B)$ and $\tau^\vee = \max(\tau^A, \tau^B)$ and the random variable

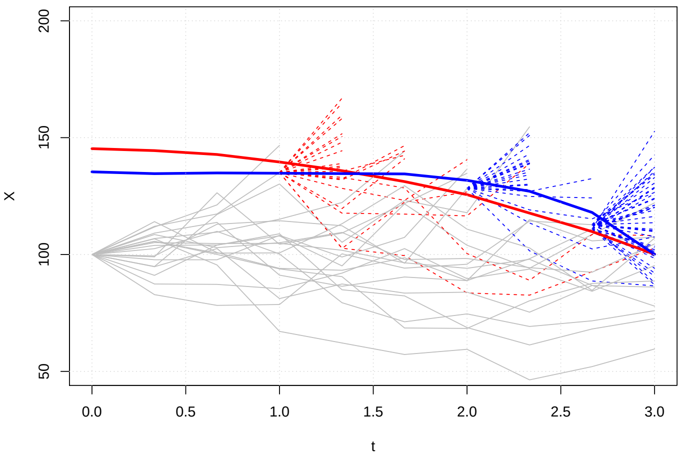

Figure 4.1: Simulated trajectories are stopped and replicated as soon as they enter the symmetric difference of the two exercise regions.

$S = \text{sign}(\tau^A - \tau^B)$ and note that

$$\tau^A - \tau^B = S(\tau^\vee - \tau^\wedge), \quad \text{that} \quad Y_{\tau^A} - Y_{\tau^B} = S(Y_{\tau^\vee} - Y_{\tau^\wedge})$$

and that S is observable at time $\tau^\wedge$. We assume that (conditionally) independent copies of random variables are available as needed on our probability space and propose the following two-stage simulation algorithm which is determined by two integer-valued, positive integer parameters n and R:

A1. Simulate independent copies $Y_0^{(i)}, \ldots, Y_{\tau^{\wedge,(i)}}^{(i)}$ of $Y_0, \ldots, Y_{\tau^\wedge}$ for $i = 1, \ldots n$. Denote by $\mathcal{F}^{\tau^{\wedge,(i)}}$ the information generated along the i^{th} trajectory and by $S^{(i)}$ the associated copy of S.

A2. Conditionally on $\mathcal{F}^{\tau^{\wedge,(i)}}$ simulate for each i with $S^{(i)} \neq 0$ and for $r = 1, \ldots, R$ copies $Y_{\tau^{\wedge,(i)}+1}^{(i,r)}, \ldots, Y_{\tau^{\vee,(i,r)}}^{(i,r)}$ of $Y_{\tau^\wedge+1}, \ldots, Y_{\tau^\vee}$ which are independent across the i and conditionally independent across the r. If $S^{(i)} = 0$ and thus $\tau^{\vee,(i,r)} = \tau^{\wedge,(i)}$ set $Y_{\tau^{\vee,(i,r)}}^{(i,r)} = Y_{\tau^{\wedge,(i)}}^{(i)}$.

A3. Estimate Δ by

$$\Delta^{(n,R)} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{R} \sum_{r=1}^{R} S^{(i)} \left(Y_{\tau^{\vee,(i,r)}}^{(i,r)} - Y_{\tau^{\wedge,(i)}}^{(i)} \right) \tag{4.3}$$

Figure 4.1 illustrates the simulation procedure for an example where the two stopping times are given in the form of exercise boundaries, the blue and red curves in the figure. Whenever the process Y crosses one of the two boundaries, one of the stopping times occurs. Note that it can also happen that both boundaries are crossed simultaneously. Again, we will have another zero in the

Monte Carlo estimator. From the small amounts of red and blue in the picture, we observe that the subsampling in Step A2 only has to be carried out rarely.

The following proposition shows that $\Delta^{(n,R)}$ is unbiased and gives an expression for its variance.

Proposition 53. *We have* $\mathrm{E}[\Delta^{(n,R)}] = \Delta$ *and*

$$\mathrm{Var}[\Delta^{(n,R)}] = \frac{v_1}{n} + \frac{v_2}{Rn}$$

where

$$v_1 = \mathrm{Var}\left[\mathrm{E}\left[Y_{\tau^A} - Y_{\tau^B} | \mathcal{F}_{\tau^\wedge}\right]\right] \quad \text{and} \quad v_2 = \mathrm{E}\left[\mathrm{Var}\left[Y_{\tau^A} - Y_{\tau^B} | \mathcal{F}_{\tau^\wedge}\right]\right].$$

The basic motivation for the algorithm is as follows: If τ^A and τ^B are not far apart, Step A2 of the algorithm is much cheaper computationally than Step A1. If they happen to coincide, Step A2 is for free. Therefore, large values of R are comparatively cheap. Moreover, if circumstances are favorable, namely, if the bulk of the variance in $Y_{\tau^A} - Y_{\tau^B}$ actually comes from what happens between τ^A and τ^B, i.e., if $v_1 \ll v_2$, the estimator will behave like an estimator with $R \cdot n$ rather than n samples.

We close this section by pointing out that $\Delta^{(n,R)}$ can be understood as an interpolation between two well-known Monte Carlo algorithms: $R = 1$ corresponds to a simple Monte Carlo estimator and $R = \infty$ corresponds to Conditional Monte Carlo. For $R = 1$, the algorithm collapses to a simple Monte Carlo estimator $\Delta^{MC} = \Delta^{(n,1)}$ of $Y_{\tau^A} - Y_{\tau^B}$ along n sample trajectories of $Y_0, \ldots, Y_{\tau^\vee}$. Moreover, we have

$$\mathrm{Var}[\Delta^{MC}] = \frac{\mathrm{Var}[Y_{\tau^A} - Y_{\tau^B}]}{n} = \frac{v_1 + v_2}{n}.$$

The final equality is the well-known conditional variance decomposition formula. The inner sum in the estimator $\Delta^{(n,R)}$,

$$\frac{1}{R}\sum_{r=1}^{R} S^{(i)}\left(Y^{(i,r)}_{\tau^\vee,(i,r)} - Y^{(i)}_{\tau^\wedge,(i)}\right),$$

can be interpreted as a Monte Carlo estimator for

$$D^{(i)} = \mathrm{E}\left[S^{(i)}\left(Y^{(i,1)}_{\tau^\vee,(i,1)} - Y^{(i)}_{\tau^\wedge,(i)}\right)\middle|\mathcal{F}_{\tau^\wedge,(i)}\right].$$

which is exact in the limit $R \to \infty$. The limiting Monte Carlo estimator

$$\Delta^{CMC} = \frac{1}{n}\sum_{i=1}^{n} D^{(i)}$$

is the so-called Conditional Monte Carlo (CMC) estimator for Δ. In the applications we consider, the conditional expectation in $D^{(i)}$ typically cannot be computed explicitly and thus the estimator $\Delta^{CMC} = \Delta^{(n,\infty)}$ is purely of theoretical interest. The variance of Δ^{CMC} is given by

$$\mathrm{Var}\left[\Delta^{CMC}\right] = \frac{v_1}{n},$$

thus reducing the variance by a factor $v_1/(v_1 + v_2)$ compared to Δ^{MC}. By employing nested simulation ($R > 1$) to approximate the conditional expectations, we construct implementable estimators which achieve at least part of this variance reduction.

4.2 Calibrating the Algorithm

In the previous section, we saw that the variance of $\Delta^{(n,R)}$ is of course smaller than the variance of the simple Monte Carlo estimator $\Delta^{(n,1)}$. However, the use of R replications leads to higher computational costs. These additional costs depend on the complexity of simulating $Y_{\tau^\wedge}, \dots, Y_{\tau^\vee}$, which includes simulating Y as well as evaluating a stopping time τ^A or τ^B whichever has not been stopped first.

The main issue of this section is the optimal choice of R, which will be denoted by R^*. This includes the question whether it is beneficial to use the nested Conditional Monte Carlo technique instead of the simple Monte Carlo estimator ($R^* > 1$?).

It turns out that the answer depends on two natural questions considering the problem:

1. Would the theoretical Conditional Monte Carlo estimator Δ^{CMC} lead to a substantial variance reduction?

2. Is generating samples of $Y_{\tau^\vee} - Y_{\tau^\wedge}$ conditionally on $Y_{\tau^\wedge}$ (on average) cheaper than sampling copies of $Y_{\tau^\wedge}$?

The computational cost of implementing the estimator for fixed n and R is again random as it was before in the setting of Giles in Theorem 38. Denote by ρ_1 the expected computational cost of simulating a realization of $Y_{\tau^\wedge}$ in step A1 of the algorithm. ρ_1 takes into account the expected length $\mathrm{E}[\tau^\wedge]$ of a path and the costs of evaluating both stopping times along that path. Denote by ρ_2 the expected computational cost of simulating a realization of $Y_{\tau^\vee} - Y_{\tau^\wedge}$ for given $Y_{\tau^\wedge}$ in step A2 of the algorithm. ρ_2 takes into account the expected length $\mathrm{E}[\tau^\vee - \tau^\wedge]$ of such a path and the costs of evaluating either of the two stopping times along that path. The expected computational cost of implementing the estimator with parameters n and R is thus given by

$$c(n, R) = n\rho_1 + nR\rho_2.$$

For simplicity, we assume that v_1, v_2, ρ_1 and ρ_2 are strictly positive.

The next proposition characterizes how to optimally choose n and R for a given computational budget C. In particular, we derive an expression for the optimizer R^* which is independent of the overall budget, showing that in relative terms the optimal allocation of computational costs between Steps A1 and A2 is independent of C. For this reason, the question of calibrating the algorithm is basically reduced to finding a good choice of R. For the moment, we ignore the integer-constraints on n and R. We do however take into account that in order to obtain an implementable algorithm we must have $R \geq 1$. By identifying the situations where $R^* > 1$, we identify those cases where the Nested Conditional Monte Carlo algorithm is more efficient than simple Monte Carlo.

Proposition 54. *For any $C > 0$, the solution (n^*, R^*) to*

$$\min_{n,R} \operatorname{Var}\left[\Delta^{(n,R)}\right] \quad \text{s.t.} \quad c(n,R) \leq C, \ R \geq 1, \ n \geq 0$$

is given as follows: If

$$\frac{\rho_1}{\rho_2}\frac{v_2}{v_1} > 1 \tag{4.4}$$

then

$$R^* = \sqrt{\frac{\rho_1}{\rho_2}\frac{v_2}{v_1}} \quad \textit{and} \ n^* = \frac{C}{\rho_1 + \rho_2 R^*}.$$

If condition (4.4) *is violated, the optimal choice is*

$$R^* = 1 \quad \textit{and} \ n^* = \frac{C}{\rho_1 + \rho_2}.$$

From condition (4.4) we can characterize the cases where the algorithm with $R > 1$ is preferable to simple Monte Carlo as follows: (4.4) is fulfilled if the cost of a single sample is smaller in Step A2 than in Step A1, $\rho_2 < \rho_1$, and if a perfect CMC estimator would reduce the variance by at least a factor 2, $v_1 < v_2$. If either of these conditions fails, (4.4) can only hold if the other condition is satisfied sufficiently strongly. If n and R are such that the budget constraint $c(n, R) = C$ holds with equality then the variance of $\Delta^{(n,R)}$ can be written as

$$\mathrm{Var}\left[\Delta^{(n,R)}\right] = \frac{V(R)}{C} \text{ where } V(R) = (\rho_1 + \rho_2 R)\left(v_1 + \frac{v_2}{R}\right).$$

Therefore, in order to compare the resulting variance across different values of R it suffices to compare the values $V(R)$. The next proposition quantifies the gain from using the algorithm with R^* subsamples rather than a simple Monte Carlo estimator:

Proposition 55. *If condition* (4.4) *holds, the relative gain (variance reduction) from using Nested Conditional Monte Carlo with R^* subsamples instead of simple Monte Carlo is given by*

$$\gamma^* = \frac{V(R^*)}{V(1)} = \frac{\left(\sqrt{\frac{v_1}{v_2}} + \sqrt{\frac{\rho_2}{\rho_1}}\right)^2}{(1 + \frac{v_1}{v_2})(1 + \frac{\rho_2}{\rho_1})}.$$

Moreover,

$$\max\left(\frac{\rho_2}{\rho_1 + \rho_2}, \frac{v_1}{v_1 + v_2}\right) \leq \gamma^* \leq 4 \max\left(\frac{\rho_2}{\rho_1 + \rho_2}, \frac{v_1}{v_1 + v_2}\right).$$

The lower bound on γ^* shows that the variance parameters v_i and the cost parameters ρ_i independently place a bound on the variance reduction we can hope to achieve: We can reduce the variance at most by a factor $v_1/(v_1 + v_2)$, no matter how small ρ_2 is compared to ρ_1. The intuitive reason for this is that $\Delta^{(N,R)}$ can never beat the theoretical CMC estimator Δ^{CMC}. Likewise, no matter how small the CMC-variance v_1 is compared to v_2, we can never gain more than the speed-up from concentrating our simulations on the interval from $\tau^\wedge$ and $\tau^\vee$ instead of the whole interval from 0 to $\tau^\vee$. This speed-up is captured by the ratio between ρ_2 and $\rho_1 + \rho_2$. Since our upper bound on γ^* is four times the lower bound, we see that the lower bound is never too far off. To sum up, we can expect drastic variance reductions if (and only if) $v_1 \ll v_2$ and $\rho_2 \ll \rho_1$.

In practical implementations, we will not be able to work with exactly R^* subsamples for at least two reasons: Since we will not know the parameters v_1, v_2, ρ_1 and ρ_2, these have to be estimated in pilot simulations. Moreover, R has to be set to an integer value. Thus, it is important to make sure that the performance of the algorithm is not too sensitive to the choice of R. The next proposition shows that this is indeed the case, giving an upper bound on the loss in variance reduction if we can only guarantee that R lies in an interval around R^*.

Proposition 56. *Suppose that $R^* > 1$ and $\alpha^{-1}R^* \leq R \leq \alpha R^*$ for some $\alpha > 1$. Then we have the following bound on the loss in variance reduction:*

$$\frac{V(R)}{V(R^*)} \leq \frac{1}{2} + \frac{\alpha + \alpha^{-1}}{4}, \quad \textit{and thus} \quad \frac{V(R)}{V(1)} \leq \left(\frac{1}{2} + \frac{\alpha + \alpha^{-1}}{4}\right) \gamma^*.$$

This bound is fairly tight for realistic values of α. For $\alpha = 1.2$, implying that R is misspecified by about 20%, we are still within 1% of the optimal variance reduction. For $\alpha = 2$, almost 90 % of the optimal variance reduction are achieved. We thus conclude that even a crude attempt at optimizing the number of subsamples R should lead to near-optimal results.

A key observation in the proof of Proposition 56 is the identity $V(\alpha R^*) = V(\alpha^{-1}R^*)$. The next corollary collects some of its practical implications for the choice of R: If R^* is significantly larger than 1, then there is a wide interval of values for R which give an improvement over simple Monte Carlo: Any value of R which is smaller than the square of the optimum R^* is better than $R = 1$. Moreover, given a fixed computational budget it is always better to overestimate R^* by a fixed amount, than to underestimate it by the same amount. Finally, rounding R^* to the nearest integer can never produce an algorithm which is worse than simple Monte Carlo.

Corollary 57. *Suppose condition* (4.4) *holds, i.e., $R^* > 1$. Then the following assertion are true:*

(i) For every R with $1 < R < {R^*}^2$ we have an improvement over simple Monte Carlo, $V(R) < V(1)$.

(ii) Let $r > 0$ be such that $R^* - r \geq 1$. Then $V(R^* + r) < V(R^* - r)$.

(iii) Let $R^\#$ be the integer nearest to R^*. If $R^\# > 1$, then $V(R^\#) < V(1)$.

4.3 Numerical Experiments

In the following sections we want to present some numerical examples illustrating the benefit provided by the Nested CMC in practice. We restrict ourselves to examples related to Bermudan option pricing, but many other fields of application are also conceivable. An example is credit risk modelling, where events of default and distress are often modelled by stopping times. Another potential application can be found in revenue management, see Talluri and van Ryzin [70]. One of the classical problems arising there is the decision when to start the end-of-season sales. In the easiest setting, two possible prices for a product are given and the demand function depending on time and prices is supposed to be known from marketing research. In order to find the optimal stopping time when to change prices Gallego and van Ryzin developed a heuristic approach that can be found in Feng and Gallego [35]. These calculations are normally less complex than the problem of Bermudan option pricing considered in the following. On the other hand, the decisions taken in revenue management have to be very precise, since millions of dollars or euros could be lost due to suboptimal strategies. This is why stopping rules arising in this field have to be tested very accurately, which could be achieved by the NCMC and/or the multilevel technique.

4.3.1 Parameter Uncertainty

In this example we have to compare two stopping times, because we are interested in the sensitivity of the option pricing problem to parameter misspecifications. More precisely, we use a wrong value for the volatility σ within the training step of a fast approximation method and measure the impact of this error to the estimated payoff.

We use again Benchmark Example 48 with $\mathcal{J} = 10$ and $D = 2$ assets. Denote by τ^σ a stopping time calculated via global regression, namely the Tsitsiklis-Van Roy method from Section 2.1.2, using $k = 100,000$ training paths which have the correct volatility σ:

$$Z_j^{d,(i)} = Z_{j-1}^{d,(i)} \exp\left(\left(r - \delta - \frac{\sigma}{2}\right)(t_j - t_{j-1}) + \sigma\left(W_j^{d,(i)} - W_{j-1}^{d,(i)}\right)\right).$$

Here, $W_j^{(i)}, i = 1, \ldots, k$ are trajectories of D-dimensional Brownian motion.

In contrast, the stopping rule $\tau^{\widehat{\sigma}}$ is calculated via the same method using training paths with volatility $\widehat{\sigma} \neq \sigma$:

$$\widetilde{Z}_j^{d,(i)} = \widetilde{Z}_{j-1}^{d,(i)} \exp\left(\left(r - \delta - \frac{\widetilde{\sigma}}{2}\right)(t_j - t_{j-1}) + \widetilde{\sigma}(W_j^{d,(i)} - W_{j-1}^{d,(i)})\right).$$

To decrease variance the same set of Brownian paths is used again.

Having calculated the two stopping times we now want to estimate the costs that arise from exercising the option based on such a misspecified calculation, which means to estimate

$$\Delta(\widehat{\sigma}) = \mathrm{E}[Y_{\tau^\sigma} - Y_{\tau^{\widehat{\sigma}}}], \tag{4.5}$$

where $Y_t = g_t(X_t)$. The expectation in (4.5) is, of course, taken with respect to the correct model with volatility σ. Table 4.1 reports estimates of the parameters ρ_i and v_i for different values of the misspecified volatility $\widehat{\sigma}$.

The first thing to observe from the table is that in all four cases Nested CMC leads to a substantial variance reduction, varying between a factor of about 60 and about 20, with the largest gains if σ and $\widehat{\sigma}$ are most similar. The ratio between v_1 and v_2 is fairly constant and (much) smaller than the ratio between ρ_2 and ρ_1 which is thus decisive for the achieved variance reduction. We also report the probability that the two stopping times differ – so that the subsimulations actually have to be carried out – and find that it lies between 2% and 8%. These numbers are one key reason for the small values of ρ_2 and the high optimal numbers of subsamples R^*, between 103 and 272. The units in which we report the ρ_i are irrelevant, only the ratios matter.

4.3.2 Improved Quasi-Control Variates

In this section and the next, we turn to the more classical problem of calculating $\mathrm{E}[Y_{\tau^A}]$ for a given stopping time τ^A. To reduce the variance, the technique

$\widehat{\sigma}-\sigma$	0.005	0.01	0.015	0.02
$\Delta(\widehat{\sigma})$	0.011	0.026	0.043	0.066
$\mathrm{E}\left[Y_{\tau^\sigma}\right]$	8.042	8.042	8.042	8.042
$\mathrm{E}\left[Y_{\tau^{\widehat{\sigma}}}\right]$	8.031	8.016	7.999	7.976
$P(\tau^\sigma \neq \tau^{\widehat{\sigma}})$	0.022	0.043	0.062	0.081
ρ_1	7.975	7.974	7.972	7.972
ρ_2	0.053	0.104	0.154	0.199
v_1	0.008	0.020	0.037	0.061
v_2	4.023	8.016	12.053	16.066
R^*	271.8	176.0	129.4	103.3
γ^*	0.016	0.026	0.037	0.047
speed-up	62.5	38.5	27.0	21.3

Table 4.1: Estimated simulation parameters for different values of $\widehat{\sigma}$. The speed-up $1/\gamma^*$ from using Nested CMC is given in the last row.

of quasi-control variates from Section 1.3 is applied, which is in the sense of Emsermann and Simon [34]. Therefore we introduce a second stopping time τ^B and write

$$\mathrm{E}[Y_{\tau^A}] = \mathrm{E}[Y_{\tau^B}] + \mathrm{E}[Y_{\tau^A} - Y_{\tau^B}]. \tag{4.6}$$

In a classical control variate approach, one would choose τ^B such that the first expected value on the right hand side can be calculated explicitly and would then estimate only the second one by Monte Carlo, for example $\tau^B = \mathcal{J}$, which is the European price, see Theorem 21. Of course this choice would lead to a situation that the two stopping times are equal in many cases, since it is a characteristic property of max-call examples like ours that many trajectories are stopped quite late. However that does not mean that it is a very fruitful control, because even in case of $\tau^\vee - \tau^\wedge = 1$ the correlation may be quite small due to fluctuations of the process that take place within one time step.

A superior choice for the control is $\mathrm{E}[Y_\mathcal{J}|\mathcal{F}_{\tau^A}]$. This can be understood as a conditional Monte Carlo estimator with $R = \infty$ and the same stopping time $\tau^B = \mathcal{J}$, but European prices are not always available in closed form. This is why we want to estimate both expectations in (4.6) by Monte Carlo simulations, which is a rewarding procedure if Y_{τ^B} is significantly cheaper to simulate than Y_{τ^A}. In that case, the first summand could be estimated with many (cheap) simulations of Y_{τ^B}. For the second summand, only a small number of (expensive) paths would be necessary due to the variance reduction effect.

We retain Benchmark Example 48 including the GBM asset X_j and increase the dimension to $D = 3$ with $\mathcal{J} = 10$ time steps. Again, $Y_j = g_j(X_j)$ and we choose τ^A as an approximate optimal stopping time calculated by the mesh method with 2500 training paths, see Proposition 20. As τ^B we choose a stopping time obtained by global regression with $100,000$ training paths, namely by the Tsitsiklis-Van Roy approach, see Section 2.1.2. In Table 4.2 we state estimates of the expected values of $\mu^A = \mathrm{E}[Y_{\tau^A}]$, the variance $v^A = \mathrm{Var}\left[Y_{\tau^A}\right]$

μ^A	v^A	ρ^A	μ^B	v^B	ρ^B	v_1	ρ_1	v_2	ρ_2
11.276	182	37.92	11.224	206	0.0124	0.044	36.23	19.536	1.728

Table 4.2: Estimated simulation parameters.

and the cost ρ^A for generating a sample of Y_{τ^A}, as well as the corresponding quantities for τ^B. As expected, v^A and v^B are similar, but ρ^A is by a factor three thousand larger than ρ^B. Note also that μ^A is considerably larger than μ^B. Since both estimates have a downward bias, this reflects the greater accuracy of the mesh method. In particular, μ^A lies within the 95% confidence interval $[11.265, 11.308]$ for $\mathrm{E}\,[Y_{\tau^*}]$ from Andersen and Broadie [2].

Denote by $\rho(R)$ and $v(R)$ the computational costs and the variance per testing path when estimating $E[Y_{\tau^A} - Y_{\tau^B}]$ by Nested CMC with R replications, such that

$$\rho(R) = \rho_1 + R\,\rho_2 \quad \text{and} \quad v(R) = v_1 + \frac{v_2}{R},$$

where the ρ_i and v_i are defined exactly as in Section 4.1. Let n^B be the number of paths used to estimate μ^B and let n be, as before, the number of paths used in the estimation of $\mu^A - \mu^B$. For a given computational budget C and fixed R, the optimal choice of n^B and n is given as the solution of

$$\min_{n^B,\, n} \; \frac{v^B}{n^B} + \frac{v(R)}{n} \quad \text{s.t.} \;\; \rho^B\, n^B + \rho(R) n \leq C.$$

It follows from Proposition 54 that the optimal ratio between n and n^B is given by

$$\frac{n^B}{n} = \sqrt{\frac{v^B}{v(R)}\,\frac{\rho(R)}{\rho^B}} \tag{4.7}$$

regardless of the size of the computational budget C. Finally, observe that the optimal value R^* of R is the same as in Section 4.2 regardless of how we allocate computational effort between the estimations of μ^B and $\mu^A - \mu^B$. From the values of the v_i and ρ_i we note that $R^* = 97.037$ and the estimated gain γ^* in the simulation of $\mu^A - \mu^B$ is given by $\gamma^* = 0.067$, corresponding to a speed-up of almost fifteen times in this part of the estimation.

Table 4.3 compares the performance of three Monte Carlo estimators for μ^A which have (approximately) the same computational costs and with parameters guided by the above considerations. The first line gives the variance of a direct Monte Carlo estimator of μ^A with $3,150$ sample paths. The second line shows the variance of a simple quasi-control variate estimator ($R = 1$) with $n^B = 536,178$ paths in the estimation of μ^B and $n = 2,989$ paths in the estimation of $\mu^A - \mu^B$. The third line shows the variance of a quasi-control variate estimator with $R = 100$ replications in each of the $n = 468$ paths in the estimation of $\mu^A - \mu^B$ and $n^B = 1,784,813$ paths in the estimation of μ^B. We thus see an improvement of more than a factor 100, which comes in equal parts from the quasi-control variate and from including nested simulations.

In the present example, the ratio between v_1 and v_2 is far more favorable than the ratio between ρ_1 and ρ_2, implying that the latter ratio governs the variance

Method	Variance	Running time
Simple Monte Carlo	60.4×10^{-3}	$125s$
Quasi-Control Variate	6.77×10^{-3}	$121s$
Quasi-Control Variate with Nested CMC	0.514×10^{-3}	$105s$

Table 4.3: Comparison of the three methods with similar running times. This table reports averages over hundred runs of the simulation implemented in C++ on a standard system with a 2.6 GHz AMD processor.

reduction we achieve. This is due to the relatively high value of ρ_1 which arises since in about 60% of cases it is the cheap Tsitsiklis-Van Roy stopping time which stops first. One can construct an even more efficient quasi-control variate by modifying the Tsitsiklis-Van Roy stopping time to be slightly biased towards late stopping, thus increasing the variance v_1 but decreasing ρ_2. This can be achieved, e.g., by adding a small constant to the estimated continuation values.

4.3.3 An Improved Multilevel Algorithm

In Section 3.2.2, we noticed that importance sampling could lead to a better order of convergence of the multilevel approach, if it was possible to apply it in practice. For our nesting technique that tries to imitate this kind of importance sampling, we want to pursue this idea under some mild assumptions.

If the two stopping times τ^A and τ^B are converging to the optimal stopping time τ^* of a stopping problem, it is clear that $\rho_1 \to \mathrm{E}[\tau^*]$, which can be assumed to be non-zero. Simultaneously, $\tau^\vee - \tau^\wedge$ will in each case be less than $\mathcal{J}$ and will be zero if $\tau^A = \tau^B$. Let us define $p = P(\tau^A \neq \tau^B)$. Because of $\rho_2 \leq \mathcal{J} p$, we now have $\rho_2/\rho_1 \lesssim p$.

For the ratio of variances v_1/v_2, things are more difficult. In case of a good-natured stopping problem and under some assumptions on the payoff, the underlying asset and the two stopping times, we could postulate that

$$\inf_{x \in \mathbb{R}^D, j \in \{0,\ldots,\mathcal{J}-1\}} h(x,j)$$

exists and is different from zero, where

$$h(x,j) = \left\{ \mathrm{Var}\left[Y_{\tau^A} - Y_{\tau^B} \middle| \tau^\wedge = j \neq \tau^\vee \wedge X_j = x \right] \right\} .$$

In other words, h gives us the variance that arises when one of the stopping times stops at x at time j and the other stopping time is still running for at least one additional time step. The quantity v_2 will then be bounded from below by the product of p and a constant, so $v_2 \gtrsim p$. To analyze the behaviour of v_1, let $\mathcal{H}$ denote the σ-algebra generated by the event $\{\tau^A \neq \tau^B\}$ and use the formula of total variance with respect to it.

$$\begin{aligned} v_1 &= \mathrm{Var}\left[\mathrm{E}\left[\mathrm{E}\left[Y_{\tau^A} - Y_{\tau^B} \middle| \mathcal{F}_{\tau^\wedge}\right] \mathcal{H}\right]\right] + \mathrm{E}\left[\mathrm{Var}\left[\mathrm{E}\left[Y_{\tau^A} - Y_{\tau^B} \middle| \mathcal{F}_{\tau^\wedge}\right] \middle| \mathcal{H}\right]\right] \\ &= \mathrm{Var}\left[\mathrm{E}\left[Y_{\tau^A} - Y_{\tau^B} \middle| \mathcal{H}\right]\right] + p\, \mathrm{Var}\left[\mathrm{E}\left[Y_{\tau^A} - Y_{\tau^B} \middle| \mathcal{F}_{\tau^\wedge}\right] \middle| \tau^A \neq \tau^B\right] \end{aligned}$$

It is clear that

$$\mathrm{Var}\left[\mathrm{E}\left[Y_{\tau^A} - Y_{\tau^B}\middle|\mathcal{H}\right]\right] = p(1-p)\left(\frac{\mathrm{E}[Y_{\tau^A} - Y_{\tau^B}]}{p}\right)^2$$

so we have

$$\frac{v_1}{v_2} \lesssim \frac{1-p}{p^2}(\mathrm{E}[Y_{\tau^A} - Y_{\tau^B}])^2 + \mathrm{Var}\left[\mathrm{E}\left[Y_{\tau^A} - Y_{\tau^B}\middle|\mathcal{F}_{\tau^\wedge}\right]\middle|\tau^A \neq \tau^B\right] \tag{4.8}$$

Inserting notation and the results from Chapter 3, we have for two stopping times based on k_l and k_{l-1} training paths that

$$\begin{aligned}
& \frac{1-p}{p^2}\left(\mathrm{E}[Y_{\tau^A} - Y_{\tau^B}]\right)^2 \\
\simeq\; & \left(\frac{1}{k_{l-1}^{-\mu\alpha/2}}\right)^2 (\mathrm{E}[g_{\tau_{k_l}}] - \mathrm{E}[g_{\tau_{k_{l-1}}}])^2 \\
=\; & k_{l-1}^{\mu\alpha}\left(k_l^{-\mu(1+\alpha)/2} - k_{l-1}^{-\mu(1+\alpha)/2}\right)^2 \\
=\; & k_0\theta^{(l-1)\mu\alpha}k_0\left(\theta^{-l\mu(1+\alpha)/2} - \theta^{-(l-1)\mu(1+\alpha)/2}\right)^2 \\
=\; & k_0\theta^{(l-1)\mu\alpha}k_0\left((\theta^{-l\mu(1+\alpha)/2})(1-\theta^{\mu(1+\alpha)/2})\right)^2 \\
\simeq\; & k_0\theta^{-\mu\alpha}\theta^{l\mu\alpha}k_0\left(\theta^{-l\mu(1+\alpha)/2}\right)^2 \\
=\; & (k_0^2\theta^{-\mu\alpha})\theta^{-l\mu} \lesssim \theta^{-l\mu},
\end{aligned}$$

where we assumed Theorem 37 to be efficient and $k_l = k_0 \times \theta^l$ like in the multilevel setting in Theorem 45. Hence, the first summand in (4.8) converges to zero, as $l \to \infty$. The second term turns out to be hard to analyze. Though one might argue that

$$\mathrm{Var}\left[\mathrm{E}\left[Y_{\tau^A} - Y_{\tau^B}\middle|\mathcal{F}_{\tau^\wedge}\right]\middle|\tau^A \neq \tau^B\right]$$

also converges to zero, the order will be very difficult to calculate. Detailed knowledge about the approximation procedure in the vicinity of the exercise boundary will be necessary. However, we now have from Proposition 55 that

$$\gamma^* = \frac{\left(\sqrt{\frac{v_1}{v_2}} + \sqrt{\frac{\rho_2}{\rho_1}}\right)^2}{(1+\frac{v_1}{v_2})(1+\frac{\rho_2}{\rho_1})} \to 0 \tag{4.9}$$

and a reduction of order could be possible, which is made note of in Chapter 8.

For a numerical example, we look again at Benchmark Example 48 with $D = 5$ assets and $\mathcal{J} = 10$ time steps. We work with two levels, $L = 2$, and $(k_0, k_1, k_2) = (100, 1000, 10000)$ training paths. The three stopping times τ_{k_l} are calculated again by means of the mesh method as in Proposition 20 with European control variate. Besides the number of training paths k_l, we also increase the approximation quality of the numerical integration in the European control variate across levels, choosing precision parameters $(u_0, u_1, u_2) = (0.5, 0.05, 0.005)$. In light of (3.4), we can apply our Nested CMC twice, and obtain two sets of parameters ρ_l and v_l which are summarized in Table 4.4.

Level l	1	2
$\mathrm{E}\left[Y_{\tau(k_l)} - Y_{\tau(k_{l-1})}\right]$	0.886	0.026
ρ_1	9.8	111.3
ρ_2	1.4	1.8
v_1	2.292	0.037
v_2	55.429	14.485
γ^*	0.28	0.031
R^*	13.018	158.707

Table 4.4: Simulation parameters at the two levels.

Method				Variance
	n			
Simple Monte Carlo	1790			0.13
	n_0^*	n_1^*	n_2^*	
Multilevel	38760	5550	880	0.033
Multilevel with Nested CMC	86780	2650	100	0.0067

Table 4.5: Overall expected variances of the three methods with identical expected computational costs.

We thus see, that Nested CMC leads to drastic speed-up of about a factor 32 at the high-precision level $i = 2$, and to a still decent one of about 3.5 at the intermediate level $i = 1$. At the "base level", i.e., the calculation of $\mathrm{E}\left[Y_{\tau_{k_0}}\right] = 15.698$ we have a variance of $\mathrm{Var}\left[Y_{\tau k_0}\right] = 251.3$ and a cost per sample which we normalize to 1. Following (3.4), the expected value we are calculating is thus

$$\mathrm{E}\left[Y_{\tau_{k_L}}\right] = 15.698 + 0.886 + 0.026 = 16.610$$

which is well within the confidence interval $[16.60, 16.66]$ for $\mathrm{E}\left[Y_{\tau^*}\right]$ from Andersen Broadie [2] for this example. To determine the optimal number of testing paths for each level, we use Proposition 14: For a fixed computational budget, the number of paths n_i^* in the estimation of each summand should be proportional to the square root of variance divided by the square root of the computational cost per sample..

Table 4.5 compares Multilevel Monte Carlo with and without nesting for a fixed expected computational budget of 200000 time units. As suggested by Table 4.4, we use $R = 13$ and $R = 159$ replications in the Nested CMC algorithms at the two levels. We also present results for a simple Monte Carlo estimator of the same expected value, $\mathrm{E}[Y_{\tau_{k_2}}]$, under the same budget. Simple Monte Carlo has a cost per sample of 112.9 and $\mathrm{Var}[Y_{\tau_{k_2}}] = 234.1$. There is a variance reduction by a factor 19.6 between simple Monte Carlo and Multilevel Monte Carlo with Nested CMC, the larger part of which (a factor 5) comes from incorporating the nested simulations.

4.4 Proofs

Proof of Proposition 53

To see the unbiasedness, note that

$$\begin{aligned} E[\Delta^{(n,R)}] &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{R}\sum_{r=1}^{R}\mathrm{E}\left[S^{(i)}(Y^{(i,r)}_{\tau^{\vee},(i,r)} - Y^{(i)}_{\tau^{\wedge},(i)})\right] \\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{R}\sum_{r=1}^{R}\mathrm{E}\left[Y_{\tau^A} - Y_{\tau^B}\right] = \Delta, \end{aligned}$$

where the second equality simply used that the term inside the expectation is an independent copy of $Y_{\tau^A} - Y_{\tau^B}$. For the variance, note first that the outer sum over i is a sum of independent, identically distributed random variables and thus

$$\mathrm{Var}\left[\Delta^{(n,R)}\right] = \frac{1}{n}\mathrm{Var}\left[\frac{1}{R}\sum_{r=1}^{R} S^{(1)}(Y^{(1,r)}_{\tau^{\vee},(1,r)} - Y^{(1)}_{\tau^{\wedge},(1)})\right].$$

Applying the conditional variance decomposition formula yields

$$\mathrm{Var}\left[\Delta^{(n,R)}\right] = \frac{v_1}{n} + \frac{v_2}{Rn}$$

with

$$v_1 = \mathrm{Var}\left[\mathrm{E}\left[\frac{1}{R}\sum_{r=1}^{R} S^{(1)}(Y^{(1,r)}_{\tau^{\vee},(1,r)} - Y^{(1)}_{\tau^{\wedge},(1)})\right| \mathcal{F}^{\tau^{\wedge},(1)}\right]\right]$$

and

$$v_2 = R\cdot\mathrm{E}\left[\mathrm{Var}\left[\frac{1}{R}\sum_{r=1}^{R} S^{(1)}(Y^{(1,r)}_{\tau^{\vee},(1,r)} - Y^{(1)}_{\tau^{\wedge},(1)})\right| \mathcal{F}^{\tau^{\wedge},(1)}\right]\right]$$

and it remains to see that these values of v_1 and v_2 coincide with those in the proposition. Note that the summands are independent and identically distributed conditionally on $\mathcal{F}^{\tau^{\wedge},(1)}$. For v_1 this implies that

$$\mathrm{E}\left[S^{(1)}(Y^{(1,r)}_{\tau^{\vee},(1,r)} - Y^{(1)}_{\tau^{\wedge},(1)})\right| \mathcal{F}^{\tau^{\wedge},(1)}\right]$$

does not depend on r and thus

$$v_1 = \mathrm{Var}\left[\mathrm{E}\left[S^{(1)}(Y^{(1,1)}_{\tau^{\vee},(1,1)} - Y^{(1)}_{\tau^{\wedge},(1)})\right| \mathcal{F}^{\tau^{\wedge},(1)}\right]\right].$$

A similar argument for v_2 now yields

$$v_2 = \mathrm{E}\left[\mathrm{Var}\left[S^{(1)}(Y^{(1,1)}_{\tau^{\vee},(1,1)} - Y^{(1)}_{\tau^{\wedge},(1)})\right| \mathcal{F}^{\tau^{\wedge},(1)}\right]\right].$$

Noting that $S^{(1)}(Y^{(1,1)}_{\tau^{\vee},(1,1)} - Y^{(1)}_{\tau^{\wedge},(1)})$ and $\mathcal{F}^{\tau^{\wedge},(1)}$ are copies of $Y_{\tau^A} - Y_{\tau^B}$ and $\mathcal{F}_{\tau^{\wedge}}$ allows to conclude the proof. Finally, let us emphasize that the above argument does take into account the fact that on some trajectories, those where τ^A and τ^B coincide, $Y_{\tau^A} - Y_{\tau^B}$ is $\mathcal{F}_{\tau^{\wedge}}$ measurable.

Proof of Proposition 54

Since the objective function decreases in both R and n, it is clear that the budget constraint holds with equality at the optimum, $c(n, R) = C$. Solving the constraint for n and substituting the result into the objective yields

$$\min_R \frac{1}{C}\left(v_1(\rho_1 + \rho_2 R) + v_2 \frac{\rho_1 + \rho_2 R}{R}\right) \quad \text{s.t.} \quad R \geq 1.$$

Using that the minimization problem is invariant to monotone transformations, we can write this as

$$\min_R \frac{v_1\rho_2}{v_2\rho_1} R + \frac{1}{R} \quad \text{s.t.} \quad R \geq 1.$$

Clearly, the solution to this convex minimization problem is $R^* = \max(1, R')$ where R' is the solution of the associated unconstrained minimization problem which is given by $R' = \sqrt{\frac{v_2\rho_1}{v_1\rho_2}}$.

Proof of Proposition 55

The formula for γ^* follows with a few algebraic manipulations after substituting $R^* = \sqrt{\frac{v_2\rho_1}{v_1\rho_2}}$ into V. We turn to the lower bound. By symmetry, it suffices to prove that for all positive real numbers a and b with $ab \leq 1$

$$\frac{(a+b)^2}{(1+a^2)(1+b^2)} \geq \frac{a^2}{1+a^2}.$$

To see this, note that we can bound the numerator as follows

$$(a+b)^2 \geq a^2 + ab \geq a^2 + a^2b^2 = a^2(1+b^2).$$

For the upper bound it suffices to observe that

$$\begin{aligned} \frac{(a+b)^2}{(1+a^2)(1+b^2)} &\leq \frac{2a^2}{(1+a^2)(1+b^2)} + \frac{2b^2}{(1+a^2)(1+b^2)} \\ &\leq \frac{2a^2}{1+a^2} + \frac{2b^2}{1+b^2} \\ &\leq 4\max\left(\frac{a^2}{1+a^2}, \frac{b^2}{1+b^2}\right), \end{aligned}$$

where we used in the first step that $(a+b)^2 \leq 2(a^2+b^2)$.

Proof of Proposition 56

Observe first that we can write

$$V(R) = \rho_1 v_1 + \rho_2 v_2 + \rho_2 v_1 \left(R + \frac{R^{*2}}{R}\right)$$

and

$$V(\alpha R^*) = \rho_1 v_1 + \rho_2 v_2 + \rho_2 v_1 (\alpha + \alpha^{-1}) R^*.$$

Therefore we have $V(\alpha R^*) = V(\alpha^{-1}R^*)$ and by convexity $V(R) \leq V(\alpha R^*)$ for all R in the interval. It thus suffices to prove the upper bound for

$$\frac{V(\alpha R^*)}{V(R^*)} = \frac{\Gamma + \alpha + \alpha^{-1}}{\Gamma + 2} \quad \text{where} \quad \Gamma = \frac{\rho_1 v_1 + \rho_2 v_2}{\rho_2 v_1 R^*}.$$

Since $\alpha + \alpha^{-1} \geq 2$, we can bound this expression from above by replacing Γ with a smaller number. In particular, $\Gamma \geq 2$ yields the desired inequality

$$\frac{V(\alpha R^*)}{V(R^*)} \leq \frac{2 + \alpha + \alpha^{-1}}{4}.$$

To see that we indeed have $\Gamma \geq 2$, note that by inserting the expression for R^* we can write

$$\Gamma = \sqrt{\frac{\rho_2 v_2}{\rho_1 v_1}} + \sqrt{\frac{\rho_1 v_1}{\rho_2 v_2}}.$$

$\Gamma \geq 2$ now follows from the fact that $x + x^{-1} \geq 2$ for all $x \geq 0$.

Proof of Corollary 57

In the proof of Proposition 56, we saw that $V(\alpha R^*) = V(\alpha^{-1}R^*)$. For $\alpha = R^*$ this gives $V({R^*}^2) = V(1)$. Thus, (i) follows from the convexity of V. The argument for (ii) is similar. For (iii) note first that if $R^* < 1.5$ we have $R^\# = 1$ and nothing is to prove. By (i) it thus suffices to show $R^\# < {R^*}^2$ for $R^* \geq 1.5$. To see this, note that $R^\# \leq R^* + \frac{1}{2} < {R^*}^2$ where the last inequality holds for all $R^* > \frac{1+\sqrt{3}}{2} \approx 1.37$.

Chapter 5

Upper Bounds via Dual Methods

The dual approach provides a possibility to simulate upper bounds for the fair price of an American oder Bermudan option via Monte Carlo simulation. It was introduced by Haugh and Kogan [46] and also independently by Rogers [64]. The work of Haugh and Kogan is based on earlier results of Davis and Karatzas [29]. In particular, upper bounds are useful to assess the quality of lower bounds. Together, lower and upper bounds provide a confidence interval for the true option price and a small distance between those two bounds will then guarantee a small bias for both of them.

Essentially, the dual approach says that each martingale provides an upper bound for the price of a Bermudan option when inserted into an expression that can be evaluated via Monte Carlo. This expression reads

$$V_0 \leq \mathrm{E}\left[\max_j(g_j(X_j) - M_j)\right], \tag{5.1}$$

where M is a martingale with initial value 0 and the maximum has to be evaluated with respect to all exercise dates, both in the American and the Bermudan case. In other words, the dual approach requires a suitable martingale instead of a profitable stopping time. This hunt for a good martingale has led to a variety of ideas and results.

The first observation is that the Doob martingale part M_j^*, see Theorem 63, of the true value process V_j will not only lead to the best upper bound, but will also lead to equality in (5.1). Furthermore, the equation will hold almost surely, i.e.

$$V_0 = \max_j \left(g_j(X_j) - M_j^*\right) \quad \text{a.s.}. \tag{5.2}$$

Surprisingly, the Doob martingale part of the true value process is not the only martingale fulfilling (5.2). Martingales fulfilling this equation will be called “surely optimal martingales”. Thus, if it was possible to draw samples from such a surely optimal martingale, only one single trajectory would be necessary to obtain the exact result V_0.

Since all those surely optimal martingales can in general not be assumed to be given analytically, the big question arising is how to construct martingales similar to M_j^*. For example, it is convenient to use an approximation of the continuation value provided by a method mentioned in Chapter 2. There are two ways to use such continuation estimates to create martingales via nested simulation. Those will be presented in Subsection 5.2.1 and Subsection 5.2.2. The former will use one-step subsimulations to estimate conditional expectations that can be used to extract a process that has the martingale property. The latter uses multi-step subsimulations to determine a lower bound via what is called the "testing step" in Chapter 2. Of course, this leads to a tremendous increase of complexity, especially in case of options with many exercise dates.

Another approach was introduced by Belomestny, Bender and Schoenmakers [8]. They use a regression estimator for the Doob martingale part of a given approximation and thus do not need nested simulations to construct a martingale.

Desai, Farias and Moallemi [31] also developed a non-nested method. They fix a finite set of martingales and use optimization techniques to find the best of all linear combinations of them to minimize the bias. This approached will be summarized at the beginning of Section 5.4.

Finding a good martingale is considered to be "more art than science" by Rogers [64]. He uses the value of the European counterpart of the Bermudan option in question together with some heuristic arguments and Proposition 70. His results are surprisingly good, especially in the one-dimensional case. However, in higher dimensions or when considering options that are more complicated, such an approach will be difficult.

Another idea, similar to the dual approach was developed by Jamshidian [51]. He uses the multiplicative Doob decomposition, see Theorem 67, to formulate a "multiplicative dual approach". In contrast, the dual approach (5.1) will then be called "additive dual approach". Unfortunately, his approach is clearly inferior to the additive dual because the variance of the resulting Monte Carlo estimator is higher. An interplay of the two dual approaches is discussed by Chen and Glasserman [24]. They show that "... any multiplicative dual can be improved by an additive dual and vice versa."

This chapter is organized as follows. In the next section, the most important results and definitions concerning the dual approach are given. In Subsection 5.2, general assumptions will be imposed to analyze what will be called "nested methods" in the following. This includes the well-known method of Andersen and Broadie. Based on these assumptions, the complexity of such nested methods can be calculated, see Section 5.2.3. Section 5.3 is about the generalization of the dual approach for the BSDE setting. Finally in Section 5.4, martingales are found by convex optimization algorithms and the sieves method.

In the succeeding chapter, the multilevel technique will be used to improve the efficiency of nested methods. The resulting complexity will then be compared to Subsection 5.2.3.

5.1 Dual Formulation

We recall the definition of the optimal stopping problem

$$V_j(x) := \max_{\tau \in \mathcal{T}_j} \mathrm{E}\left[g_\tau\left(X_\tau\right) \middle| X_j = x \right], \quad j = 0, \ldots, \mathcal{J}, \tag{5.3}$$

from Chapter 1, which will be called "primal representation" in the following. It is clear that this true value process V has the supermartingale property

$$\mathrm{E}[V_{j+1}|\mathcal{F}_j] \leq V_j \quad \text{a.s.} \quad j = 0, \dots, \mathcal{J}-1,$$

since

$$\begin{aligned}\mathrm{E}[V_{j+1}|\mathcal{F}_j] &= \sup_{\tau \in \mathcal{T}_{j+1}} \mathrm{E}[g_\tau(X_\tau)|\mathcal{F}_j] \\ &\leq \sup_{\tau \in \mathcal{T}_j} \mathrm{E}[g_\tau(X_\tau)|\mathcal{F}_j] = V_j. \quad \text{a.s.}\end{aligned}$$

Additional to the primal representation (5.3), we can now characterize V as the "Snell envelope" of the payoff process. In the following, we will fix the notation

$$Z_j = g_j(X_j), \quad j = 0, \dots, \mathcal{J}. \tag{5.4}$$

Definition 58. *We say that a process Y "dominates" the process $\widetilde{Y}$, if $Y_j \geq \widetilde{Y}_j$ almost surely for all $j = 0, \dots, \mathcal{J}$.*

Definition 59. *A "supersolution" is an $\mathcal{F}$-supermartingale that dominates the payoff process Z_j.*

Definition 60. *We define the "Snell envelope" Y of the payoff process Z to be the smallest supersolution. More precisely,*

$$Y_j \leq \widetilde{Y}_j, \quad a.s., \quad j = 0, \dots, \mathcal{J},$$

must hold for all other supersolutions $\widetilde{Y}$.

In line with denomination (5.4), we define

$$\mathcal{Z}(M) = \max_{j=0,\dots,\mathcal{J}} (Z_j - M_j), \quad \text{and} \quad Y(M) = \mathrm{E}[\mathcal{Z}(M)]$$

for $M \in \mathfrak{M}_0$, where $\mathfrak{M}$ is the set of all $\mathcal{F}$-martingales and $\mathfrak{M}_0$ the set of all $\mathcal{F}$-martingales with initial value zero. Since $\mathcal{Z}(M)$ depends on the whole trajectory of $(Z, M)_j$ we have that it is a random variable with respect to the probability space $(\Omega, \mathcal{F}_\mathcal{J}, P)$. The two main results of Rogers [64] and Haugh and Kogan [46] are summarized in the next two Theorems.

Theorem 61. *For each $M \in \mathfrak{M}_0$ it holds $Y(M) \geq V_0$.*

It suggests itself that this theorem can be used to derive upper bounds for the option price. It is just sufficient to pick a martingale M and estimate $Y(M) = \mathrm{E}[\mathcal{Z}(M)]$ via Monte Carlo simulation to obtain an unbiased estimate of an upper bound. For such an estimator

$$Y^N := \frac{1}{N} \sum_{n=1}^{N} \max_{j=0,\dots,\mathcal{J}} (Z_j^{(n)} - M_j^{(n)}). \tag{5.5}$$

it is necessary to generate N i.i.d. samples $\left(Z_\cdot^{(n)}, M_\cdot^{(n)}\right)$, $n = 1, \dots, N$, from the vector process $(Z_\cdot, M_\cdot)$.

Theorem 62. *If the martingale part M^* of the true value process V is used in Theorem 61, then the upper bound becomes minimal and and it even holds that*

$$V_0 = \mathcal{Z}(M^*) = \max_{j=0,\dots,\mathcal{J}} \left(g_j(X_j) - M_j^* \right) \quad a.s., \tag{5.6}$$

which is called the "surely optimal" property of M^* that in particular leads to

$$\operatorname{Var}[\mathcal{Z}(M^*)] = 0. \tag{5.7}$$

It is important to point out that this statement is not formulated in terms of expectations like Theorem 61 and uses the following definition, see for example Föllmer and Schied [37].

Theorem 63. *(Doob Decomposition) Suppose that Y is an $\mathcal{F}-$adapted stochastic process that fulfills $\mathrm{E}[|Y_j|] < \infty$ for all $j = 0, \dots, \mathcal{J}$. Then there exists a martingale M starting at $M_0 = 0$ and an integrable, predictable process A starting at $A_0 = 0$ such that*

$$Y_j = Y_0 + M_j - A_j, \quad j = 0, \dots, \mathcal{J}. \tag{5.8}$$

This decomposition is a.s. unique and the process A will be non-decreasing if Y is a supermartingale.

Since the true value process V is a supermartingale we have that

$$V_j = V_0 + M_j^* - A_j, \quad j = 0, \dots, \mathcal{J}, \tag{5.9}$$

with a non-decreasing process A, where

$$A_j := \sum_{n=1}^{j} V_{n-1} - \mathrm{E}\left[V_n | \mathcal{F}_{n-1}\right], \quad M_j^* = \sum_{n=1}^{j} V_n - \mathrm{E}\left[V_n | \mathcal{F}_{n-1}\right] \tag{5.10}$$

for all $j = 1, \dots, \mathcal{J}$. Obviously, M^* fulfills the martingale property and A is a non-decreasing process. Approximating the true value process V is just the problem in optimal stopping theory. Since it must be known to construct a martingale according to (5.10), Theorem 62 seems to be almost useless. However, it could give us a hint about which martingale to use for the Monte Carlo simulation (5.5). Every information about the true value process also provides an information about the optimal martingale via (5.10). This could be information about its asymptotic behaviour, its boundedness, its smoothness, or the relation to the payoff due to the existence of the exercise region, see Section 1.2.

With these two results at hand, it is possible to reformulate the pricing of a Bermudan option as the following minimization problem, which will be called "dual representation" in the following.

Proposition 64. *The "dual representation" is given by the minimization problem*

$$V_0 = \inf_{M \in \mathfrak{M}_0} \mathrm{E}\left[\max_{j=0,\dots,\mathcal{J}} (Z_j - M_j) \right] \tag{5.11}$$

and has the same solution as the primal problem.

We say that a martingale M "is surely optimal" if it fulfills (5.6) and thus is a solution to (5.11). Against intuition, there are infinitely many martingales substantial different from each other that possess this surely optimal property. The exact characterization of those is given below and was taken from Schoenmakers, Zhang and Huang [66].

Theorem 65. *A martingale $M \in \mathfrak{M}_0$ has the surely optimal property if and only if there exists a sequence of $\mathcal{F}$-adapted random variables $(\zeta_i)_{0\leq i\leq \mathcal{J}}$ fulfilling* $\mathrm{E}[\zeta_i|\mathcal{F}_{i-1}] = 1$ *and $\zeta_i \geq 0$ for all $0 < i \leq \mathcal{J}$ such that*

$$M_j = M_j^* - A_j + \sum_{l=1}^{j} \zeta_j (A_l - A_{l-1}), \quad j = 0, \dots, \mathcal{J},$$

where M^* and A result from the decomposition (5.10).

Theorem 65 includes the statement that all surely optimal martingales are somehow related to M^*. Apparently, there is no loophole to find a good martingale without approximating M^* at the same time.

Furthermore, it is also clear that there are not only infinitely many martingales with the surely optimal property, but also infinitely many other ones that realize the infimum in (5.11) without being surely optimal. On the other hand, martingales that lead to zero variance without minimizing (5.11) are hard to imagine. In general, one will expect the martingales to become "better" as the variance of $\mathcal{Z}(M)$ decreases. And indeed, there is another result from Schoenmakers, Zhang and Huang [66] about that issue that uses the following definition.

$$\theta_i(M) = \max_{i\leq j\leq \mathcal{J}} (Z_j - M_j + M_i). \tag{5.12}$$

Theorem 66. *Let $i \in \{0, \dots, \mathcal{J}\}$. If* $\mathrm{Var}[\theta_i(M^k)|\mathcal{F}_i] \xrightarrow{P} 0$ *and if in addition the sequence $M^1, M^2, \dots$ is uniformly integrable, then it holds that*

$$\mathrm{E}[\theta_i(M^k)|\mathcal{F}_i] \xrightarrow{L^1} V_i$$

as $k \to \infty$.

Since $\theta_0(M^k) = Y(M^k)$, we know now that in particular $Y(M^k)$ will converge to the true value, as long as the variances $\mathrm{Var}[\mathcal{Z}(M^k)]$ converge to zero. This statement allows us to look for good martingales only by considering their variances, as it does not imply the convergence of the martingales themselves. This relation will be exploited in Section 5.4 and by Belomestny [7]. This variance-minimizing effect is actually the big advantage of the (additive) dual approach when compared to the primal approach. Even when considering an approximation of M^* that is very time-consuming to evaluate, it might be an efficient idea to use (5.11). If it is really close to M^*, only very few paths will have to be simulated, see also Remark 92.

There is another dual approach, called "the multiplicative dual" introduced by Farshid Jamshidian [51]. The following Proposition summarizes his result and uses the multiplicative Doob decomposition, see for example Jacod and Shiryaev [49].

Theorem 67. *(Multiplicative Doob Decomposition) Let Y be an $\mathcal{F}-$adapted supermartingale starting at $Y_0 = 1$ taking values in $(0,\infty)$. Then there is an a.s. unique decomposition*

$$Y_j = B_j L_j, \quad j = 0, \ldots, \mathcal{J}$$

in which B is a positive local martingale with $B_0 = 1$ and L is a nonincreasing, predictable process with $L_0 = 1$.

Proposition 68. *Let $\mathfrak{B}_0$ be the class of all positive $\mathcal{F}$-adapted martingales with initial value 1. Then we have*

$$\begin{aligned} V_j &= \inf_{B\in\mathfrak{B}_0} \mathrm{E}^B \left[\max_{i\geq j} \frac{Z_i}{B_i} \Big| \mathcal{F}_j \right] B_j \qquad (5.13) \\ &:= \inf_{B\in\mathfrak{B}_0} \mathrm{E} \left[\max_{i\geq j} \frac{Z_i}{B_i} B_{\mathcal{J}} \Big| \mathcal{F}_j \right] B_j \end{aligned}$$

and in particular

$$V_0(x) = \inf_{B\in\mathfrak{B}_0} \mathrm{E} \left[\max_{j=0,\ldots,\mathcal{J}} \frac{g_j(X_j)}{B_j} B_{\mathcal{J}} \Big| X_0 = x \right]. \qquad (5.14)$$

It is impossible to find a martingale B that has the surely optimal property here. Analoguously to the additive dual, the infimum in equation (5.14) is attained when inserting the martingale B^*, which is the multiplicative martingale part of the true value process V, see Theorem 67. But even then, the variance is not zero. Chen and Glasserman [24] point out that

$$V_0 = \max_j (Z_j - M_j^*) = \max_j \frac{Z_j}{B_j^*}, \quad \text{a.s.} \qquad (5.15)$$

holds, where the additive dual can be found in the middle. But the expression on the right hand side is not the multiplicative dual. The factor $B_{\mathcal{J}}^*$ is missing. It would lead to a positive variance that could only be avoided via a change of measure, which in general won't be feasible analytically.

It is not only true that approximations of the true value V can lead to good dual upper bounds, but also vice versa. Because of the representation of the true value process V as Snell envelope of the payoff process, it is clear that every supersolution W dominates V, see Definition 60. On the one hand, we can define the martingale part of a supersolution W via

$$M_j^W = \sum_{i=1}^{j} W_i - \mathrm{E}[W_i|\mathcal{F}_{i-1}], \quad j = 0, \ldots, \mathcal{J} \qquad (5.16)$$

and the multiplicative martingale part via

$$B_j^W = \prod_{i=1}^{j} \frac{W_i}{\mathrm{E}[W_i|\mathcal{F}_{i-1}]}, \quad j = 0, \ldots, \mathcal{J} \qquad (5.17)$$

if W is assumed to be strictly positive almost surely.[1] On the other hand it is possible to derive supersolutions from (multiplicative) martingale parts, via

$$W_j^M = \mathrm{E} \left[\max_{j\leq i\leq \mathcal{J}} (Z_i - M_i - M_0) \Big| \mathcal{F}_j \right] + M_j, \quad j = 0, \ldots, \mathcal{J} \qquad (5.18)$$

[1] In usual cases like a max-call or a min-put option, there are supersolutions that are not strictly positive. Chen and Glasserman [24] point out that one could avoid this technical problem by setting $\widetilde{g}_j(x) := g_j(x) + \varepsilon$ with an extra ε, which can be arbitrarily small.

and

$$W_j^B = \mathrm{E}^B \left[\max_{j \le i \le \mathcal{J}} \frac{Z_i}{B_i} \middle| \mathcal{F}_j \right] B_j, \quad j = 0, \ldots, \mathcal{J}. \tag{5.19}$$

These four formulas could be used to construct a sequence of supersolutions $V^1, V^2, \ldots$ by applying (5.16) or (5.17) and then (5.18) or (5.19) alternately. It is not clear that such a sequence would be decreasing and approach the true value V. However, this is the case as the next theorem from Chen and Glasserman [24] says.

Theorem 69. *Suppose that a supersolution W has the additive and multiplicative Doob decompositions*

$$W = M + A, \quad \textit{or} \quad W = BL.$$

Let W^M and W^B be as in (5.18) *and* (5.19)*. Then for $j = 0, \ldots, \mathcal{J}$*

$$V_j \le W_j^M \le W_j \tag{5.20}$$

and

$$V_j \le W_j^B \le W_j. \tag{5.21}$$

Furthermore, Chen and Glasserman prove that the weak inequalities in (5.20) and (5.21) are strict, as long as the supersolution W_j^M is different from the true value. This means that the iteration $V^1, V^2, \ldots$ cannot stop at a suboptimal point. This procedure can be seen as an analogon of policy iteration used for lower bounds, see Remark 35.

It is possible to improve the efficiency of the Monte Carlo simulation (5.11). Therefore, let us note an easy proposition about the behaviour of the optimal martingale with respect to the exercise region.

Proposition 70. *If $Z_i \le V_i$ for $l \le i < k$, i.e. the path doesn't enter the exercise region, then*

$$M_k^* = M_l^* - V_l + V_k. \tag{5.22}$$

In particular, if $Z_i \le V_i$ for $0 \le i < k$, then $M_k^* = V_k - V_0$.

Theorem 71. *Let us define the set of all optimal exercise dates with respect to the adapted family of optimal stopping times, see Definition 2, by*

$$\mathfrak{O} = \left\{ j \in \{0, \ldots, \mathcal{J}\} \middle| \tau_j^* = j \right\}, \tag{5.23}$$

which is a random set on $(\Omega, \mathcal{F}_{\mathcal{J}}, P)$. For every superset $\mathfrak{N} \supseteq \mathfrak{O}$, we now have that

$$Y_{\mathfrak{N}}(M) = \mathrm{E} \left[\max_{j \in \mathfrak{N}} (Z_j - M_j) \right] \tag{5.24}$$

is also an upper bound of V_0. For M^* it even holds that

$$\max_{j \in \mathfrak{N}} \left(Z_j - M_j^* \right) = \max_{j = 0, \ldots, \mathcal{J}} \left(Z_j - M_j^* \right) = V_0. \tag{5.25}$$

We recall that $\tau^*_{\mathcal{J}} = \mathcal{J}$ by definition. To find a random set $\mathfrak{N}$ that does surely include all time steps where $\tau^*_j = j$, it suffices to know a deterministic lower bound for the continuation value C_j. For example, the European counterpart of the option can be used here. In case of a Bermudan max-call on D assets of GBM, we know that

$$\max_{d=1,\ldots D} \mathcal{E}(X^d_j, t_j, t_{\mathcal{J}}) \le C_j(X_j), \tag{5.26}$$

where $\mathcal{E}$ denotes the Black-Scholes formula of the one-dimensional European call option, see Remark 50. We can choose

$$\mathfrak{N} = \{\mathcal{J}\} \cup \left\{ j \in \{0, \ldots, \mathcal{J}-1\} \middle| g_j(X_j) \ge \max_{d=1,\ldots,D} \mathcal{E}\left(X^d_j, t_j, t_{\mathcal{J}}\right) \right\}$$

and also some European exchange option might be inserted here. If the European price is not available analytically, it is still an improvement to know that

$$\mathfrak{N} = \{\mathcal{J}\} \cup \left\{ j \in \{0, \ldots, \mathcal{J}-1\} \middle| g_j(X_j) > 0 \right\}$$

is a superset of $\mathfrak{D}$. This may still lead to a significant speed up for options that are far out-of-the-money.

5.2 Nested Methods

The term "nested methods" will be used for a class of algorithms that try to approximate the Doob martingale part of the true value process V via subsimulations. In most cases, they will construct a martingale that will be close to the martingale part of an approximation of the true value process. A quite general setting will be described and two examples for such algorithms will be presented in the following two subsections. Actually, this is motivated by the work of Andersen and Broadie [2], who developed a very well-known approach that will be analyzed in Section 5.2.2.

The use of subsimulations motivates the introduction of an enlarged probability space. Furthermore, we will suppose that those algorithms can generate martingales with varying precision, so a sequence of martingales will be examined. Therefore, let us define a probability space $(\Omega, (\mathcal{F}'_j)_{j\ge 0}, \mathrm{P})$, where $\mathcal{F}_j \subset \mathcal{F}'_j$ for each j. In other words, $\mathcal{F}'$ is a finer filtration than $\mathcal{F}$ and includes more information, namely the information generated by the subsimulations. We assume $(M^k)_{k\in\mathbb{N}}$ to be a sequence of $\mathcal{F}'$-martingales with initial value 0, which converges in some sense to a martingale M adapted to $\mathcal{F}$. The latter will be called "target martingale" in the following and the kind of convergence of M^k will be made precise below.

We are now interested in the convergence behavior of $Y(M^k)$ to $Y(M)$ as $k \to \infty$. Therefore, some assumptions about the algorithm in use are necessary. They are called (AC), (AR), (AR'), (AL) and (AQ). Afterwards, we will give some results about those assumptions, especially how to satisfy them. The main results are given in Theorem 75 and Theorem 76 and with their help, the complexity analysis becomes feasible.

(AC) The numerical complexity of obtaining a single realization of M^k_j is of order $O(k)$ for each $j = 1, \ldots, \mathcal{J}$.

Note that a re-parametrization could lead to another complexity in (AC). However, it fixes the use of the term "complexity" and is thus necessarry. The next two assumptions (AR) and (AR') make note of the rate of convergence of the sequence of martingales and also the kind of convergence that may be assumed. In particular, the second part of assumption (AR') is stronger than (AR).

(AR) There exists an $\mathcal{F}$-adapted martingale M such that

$$\mathrm{E}\left[\max_{j=0,\dots,\mathcal{J}}(M_j^k - M_j)^2\right] \leq Bk^{-\beta}, \quad k \in \mathbb{N},$$

for some $\beta > 0$ and $B > 0$.

(AR') There exists an $\mathcal{F}$-adapted martingale M such that

$$\max_{j=0,\dots,\mathcal{J}} \left|\mathrm{E}_{\mathcal{F}_\mathcal{J}}\left[M_j^k - M_j\right]\right| \leq Ak^{-\alpha}, \quad \mathrm{E}_{\mathcal{F}_\mathcal{J}}\left[\max_{j=0,\dots,\mathcal{J}}(M_j^k - M_j)^2\right] \leq Bk^{-\beta},$$

$\mathcal{F}_\mathcal{J}$-almost surely, for all $k \in \mathbb{N}$, some $\beta > 0$, $\alpha \geq \beta/2$, $A > 0$ and $B > 0$.

The last two assumptions make use of the following definition.

Definition 72. *Denote by $\mathcal{Q}$ and $\mathcal{Q}^k$ the $(\Omega, \mathcal{F}_\mathcal{J}, \mathrm{P})$-random sets*

$$\mathcal{Q} \quad := \{j : Z_j - M_j = \mathcal{Z}(M)\}, \quad \mathcal{Q}^k \quad := \left\{j : Z_j - M_j^k = \mathcal{Z}(M^k)\right\}, \quad k \in \mathbb{N},$$

and the $\mathcal{F}_\mathcal{J}$-measurable random variable

$$\Lambda := \min_{j \notin \mathcal{Q}} \left(\mathcal{Z}(M) - Z_j + M_j\right), \tag{5.27}$$

with $\Lambda := +\infty$ in case of $\mathcal{Q} = \{0, ..., \mathcal{J}\}$.

(AL) Λ satisfies $\mathrm{E}\left[\Lambda^{-a}\right] < \infty$, for some $a > 0$.

(AQ) $\#\mathcal{Q} = 1$ a.s.

Let us check how to satisfy (AQ) and (AL). One can state that (AQ) will be satisfied quite obviously in non-degenerate examples. Especially if there is a density for M_j, $j = 1, \dots, \mathcal{J}$, it won't be violated. Assumption (AL) is more difficult.

Corollary 73. *Assumption (AL) is fulfilled for all $0 \leq a < q + 1$ when $\Lambda > 0$ has a density g in a neighborhood of zero such that $g(z) = O(|z|^q)$ for some $q \geq 0$ as $|z| \to 0$.*

The reason for this corollary is simply that the definition of the Landau symbol provides the existence of a number ϵ such that

$$\mathrm{E}[\Lambda^{-a}] \leq \int_0^\epsilon z^{q-a} dz + \int_\epsilon^\infty z^{-a} g(z) dz \leq c + \epsilon^{-a} \int_\epsilon^\infty g(z) dz < \infty$$

where the first integral is finite since $q - a > -1$. We can deduce another good result about how to fulfill assumption (AL).

Corollary 74. *If the boundary condition (AB) from Chapter 2*

$$P(0 < C_j(X_j) - g_j(X_j) \leq \delta) \leq B\delta^{\alpha}, \quad \delta > 0$$

is fulfilled for all $j = 0, \ldots, \mathcal{J}$ and $M = M^*$, then assumption (AL) is fulfilled for all $a < \alpha$.

In Chapter 2, we showed that for good-natured problems we can assume that $\alpha = 1$. Because of Corollary 74, we can thus assume that (AL) will be fulfilled with $0 < a < 1$ when the target martingale is the Doob martingale part M^* of V.

Depending on the different assumptions spelled out above, we now have the following central theorem.

Theorem 75. *Under assumption (AR) alone it holds that*

$$|\operatorname{E}[\mathcal{Z}(M^k) - \mathcal{Z}(M)]| \leq Ck^{-\beta/2}, \quad \operatorname{E}\left[\left(\mathcal{Z}(M^k) - \mathcal{Z}(M)\right)^2\right] \leq Bk^{-\beta} \tag{5.28}$$

with some constants $C, B > 0$.

One should emphasize that Theorem 75 only uses the mild assumption (AR) which will be fulfilled quite easily, see Corollary 83 and Corollary 85. The first inequality in (5.28) provides the order of the bias and is an easy consequence of Jensen's inequality. In case of a nested method as in Section 5.2.1 this would lead to a bias of order $k^{1/2}$, which is not a tight bound, as we will see later in Table 5.1. Improving this inequality is the main purpose of the next theorem and of the other assumptions from the beginning of this section.

Theorem 76. *If assumptions (AR'), (AL) and (AQ) are satisfied, then*

$$|\operatorname{E}[\mathcal{Z}(M^k) - \mathcal{Z}(M)]| \leq Ck^{-\gamma}, \quad \operatorname{E}\left[\left(\mathcal{Z}(M^k) - \mathcal{Z}(M)\right)^2\right] \leq Bk^{-\beta} \tag{5.29}$$

with $\gamma = \min\{\alpha, \beta \min\{1, (a+1)/2\}\}$ and some constants $C, B > 0$.

Remark 77. *Theorem 76 proves that the bias will be of order $O(k^{-1+\delta})$ for arbitrary small $\delta > 0$, if (AL) is fulfilled with $0 < a < 1$ and $\beta = 1$. This leads to the expectation that the Andersen Broadie estimator from Subsection 5.2.2 (and also the method from Subsection 5.2.1) will have a bias of order $O(1/k)$ in practice for a usual, good-natured example. This will be verified with the example in Subsection 5.2.2 and was also observed before in Kolodko and Schoenmakers [53].*

To show that these results are really efficient, we look at the following example. It shows that without assumptions (AL) and (AQ) to be fulfilled, it can really be the case that Theorem 75 gives the strongest statement possible. Since it is based on the use of subsamples, it is strongly related to the applications in the following sections.

Example 78. *Consider the simple situation where $\mathcal{J} = 1$, $Z_0 = Z_1 = 0$, $Y_0^* = 0$ and $M_0^* = M_1^* = 0$. Define a target martingale via $M_0 = 0$, $M_1 = Y_1 - \operatorname{E} Y_1 = \xi - \operatorname{E}[\xi]$ with a r.v. ξ given by*

$$\xi = \begin{cases} 3b/2 & \textit{with probability } 1/4 \\ b & \textit{with probability } 1/2 \\ b/2 & \textit{with probability } 1/4 \end{cases}$$

for some $b > 0$ and the approximation by

$$M_0^k = 0, \quad M_1^k = \xi - \frac{1}{k}\sum_{l=1}^{k} \xi^{(l)}, \tag{5.30}$$

where $\xi^{(l)}$ are i.i.d. copies of ξ. For the target martingale we thus have $M_1 = \xi - \mathrm{E}[\xi]$.

This example fulfills Theorem 75, since for the approximation (5.30), we have

$$\begin{aligned} \mathcal{Z}(M^k) &= \max_j (Z_j - M_j^k) \\ &= \max\left(0, -M_1^k\right) = \left(\frac{1}{k}\sum_{l=1}^{k}\xi^{(l)} - \xi\right)^+ . \end{aligned}$$

The first inequality in Theorem 75 cannot hold with $\beta/2$ better than 1/2, because

$$\mathrm{E}[\mathcal{Z}(M^k) - \mathcal{Z}(M)] = \mathrm{E}\left[\left(\frac{1}{k}\sum_{l=1}^{k}\xi^{(l)} - \xi\right)^+ - (\mathrm{E}\,\xi - \xi)^+\right]$$

$$\geq \frac{1}{2}\,\mathrm{E}\left[\left(\frac{1}{k}\sum_{l=1}^{k}\xi^{(l)} - b\right)^+\right] = \frac{b/\sqrt{8}}{2\sqrt{k}}\,\mathrm{E}\left[\left(\frac{1}{\sqrt{k}}\sum_{l=1}^{k}\frac{\xi^{(l)} - b}{b/\sqrt{8}}\right)^+\right] \asymp \frac{b}{8\sqrt{\pi k}}$$

as $k \to \infty$. Note that $\mathrm{E}[\xi] = b$, $\mathrm{Var}[\xi] = b^2/8$ and $\mathrm{E}[\eta^+] = \frac{1}{\sqrt{2\pi}}$ for a standard normal random variable η. Considering the second inequality in Theorem 75, we notice that for this example it holds

$$\mathrm{E}\left[\left(\mathcal{Z}(M^k) - \mathcal{Z}(M)\right)^2\right] \geq \left(\mathrm{E}\left[\left(\mathcal{Z}(M^k) - \mathcal{Z}(M)\right)\right]\right)^2$$

$$\geq c\,\mathrm{Var}\left[\left(\frac{1}{\sqrt{k}}\sum_{l=1}^{k}\frac{\xi^{(l)} - b}{b/\sqrt{8}}\right)^+\right] \in O(1/k),$$

so the second inequality cannot hold with β better than 1 either. Simultaneously, assumption (AR) is fulfilled with $\beta = 1$ in this example, because of

$$\mathrm{E}\left[\max_{j=0,1}\left(M_j^k - M_j\right)^2\right] = \mathrm{E}\left[\left(M_1^k - M_1\right)^2\right] = \mathrm{E}\left[\left(\frac{1}{k}\sum_{l=1}^{k}\xi^{(l)} - \mathrm{E}[\xi]\right)^2\right]$$

and the central limit theorem. Therefore, this example shows that Theorem 75 is indeed the strongest statement possible in the sense that under assumption (AR) only, the case $\gamma = \beta/2$ can occur.

Remark 79. *In Example 78, the assumptions (AQ) and (AL) for all $a > 0$ are violated.*

The purpose of Remark 79 is just to show that Example 78 illustrates the efficiency of Theorem 75 and Theorem 76.

The sequence of approximative martingales $(M^k)_{k\in\mathbb{N}}$ will induce a sequence of upper bounds $(Y(M^k))_{k\in\mathbb{N}}$. The latter ones can be approximated via a Monte Carlo estimator like in (5.5)

$$Y^{N,k} := \frac{1}{N}\sum_{n=1}^{N} \max_{j=0,\dots,\mathcal{J}} (Z_j^{(n)} - M_j^{k,(n)}). \tag{5.31}$$

that is now also depending on k. So we have an unbiased estimator of an upper bound. To see the high bias, i.e. $Y(M^k) \geq V_0$ for all k, we have to consider some simple technical reasons. We know from the previous section that every $\mathcal{F}$-martingale $M \in \mathfrak{M}_0$ induces an upper bound via $Y(M)$. However, M^k will be an $\mathcal{F}'$-martingale.

Remark 80. *We define*

$$\widetilde{M}_j^k := \mathrm{E}_{\mathcal{F}_\mathcal{J}}[M_j^k], \quad j = 0, \dots, \mathcal{J}, \tag{5.32}$$

and expect $\widetilde{M}^k$ to be an $\mathcal{F}$-martingale (one could expect the target martingale M to coincide with $\widetilde{M}$). We then have that

$$\begin{aligned} Y(M^k) &= \mathrm{E}[\mathcal{Z}(M^k)] \\ &= \mathrm{E}\,\mathrm{E}_{\mathcal{F}_\mathcal{J}}[\max_j(Z_j - M_j^k)] \\ &\geq \mathrm{E}[\max_j(Z_j - \mathrm{E}_{\mathcal{F}_\mathcal{J}}[M_j^k])] \\ &= \mathrm{E}[\mathcal{Z}(\widetilde{M}^k)] = Y(\widetilde{M}^k) \end{aligned}$$

holds, since Jensen's inequality can be applied with respect to the $\sigma-$algebra $\mathcal{F}_\mathcal{J}$ and obtain

$$Y(M^k) \geq Y(\widetilde{M}^k) \geq V_0. \tag{5.33}$$

However, without assuming that $\widetilde{M}^k$ is an $\mathcal{F}$-martingale, it is still possible to show that at least $Y(M^k) \geq V_0$. This is just a generalization of Theorem 61, which uses that every $\mathcal{F}$-stopping time is an $\mathcal{F}'$-stopping time.

5.2.1 Martingales from Continuation Estimates

The easiest method to generate martingales via nesting is using some approximation

$$\widehat{V}_j : \mathbb{R}^D \to \mathbb{R}, \quad j = 0, \dots \mathcal{J} \tag{5.34}$$

of the true value function and approximate the martingale part of it. For example, one could use the estimate $\widehat{C}_j(x) := C_j^{k'}(x)$ from one of the fast approximation methods in Section 2.1, where the parameter k' remains fixed in the following. A suitable estimate of the true value process is then given by

$$\widehat{V}_j(x) = \max\left(\widehat{C}_j(x), g_j(x)\right) \tag{5.35}$$

and the martingale part of this process will be the target martingale M.

Remark 81. *To extract a martingale $\widehat{M}$ from this approximation, one could try to use*

$$\widehat{\Delta}_j = \widehat{V}_j(X_j) - \widehat{C}_{j-1}(X_{j-1}), \quad j = 1, \dots, \mathcal{J} \tag{5.36}$$

and $\widehat{M_j} = \sum_{i=1}^{j} \widehat{\Delta}_i$. However, it is not clear that this choice fulfills the martingale property. Thus, inserting $\widehat{M}$ into Proposition 61 could give an estimator $Y(\widehat{M})$ that is no longer high biased, which will yield misleading results.

In other words, one should take care that

$$\widehat{C}_{j-1}(x) = \mathrm{E}\left[\widehat{V}_j(X_j)\middle|X_{j-1} = x\right] \tag{5.37}$$

is fulfilled, as it is in case of the true value function and the true continuation value.

Remark 82. *In order to ensure that condition* (5.37) *is fulfilled one could use an estimator $\widehat{C}_j$ that is a linear combination of basis functions that allow to calculate the conditional expectation explicitly (according to the underlying process). This is the main idea of Glasserman and Yu [43]. More precisely, they use a set of H basis functions $\phi_j^h : \mathbb{R}^D \to \mathbb{R}$, such that*

$$\mathrm{E}\left[\phi_{j+1}^h(X_{j+1})|X_j\right] = \phi_j^h(X_j), \quad h = 1, \dots, H \tag{5.38}$$

where $j = 0, \dots, \mathcal{J} - 1$ indicates the time step as before. Now one can find coefficients β_i^h by regression so that writing the estimators like

$$\widehat{C}_j(x) = \sum_{h=1}^{H} \beta_i^h \phi_j^h(x), \quad h = 1, \dots, H \tag{5.39}$$

and

$$\widehat{V}_{j+1}(x) = \sum_{h=1}^{H} \beta_j^h \phi_{j+1}^h(x), \quad h = 1, \dots, H \tag{5.40}$$

is possible and the martingale based on increments

$$\widehat{\Delta}_j = \sum_{h=1}^{H} \beta_j^h \left(\phi_{j+1}^h(X_{j+1}) - \phi_j^h(X_j)\right) \tag{5.41}$$

obviously fulfills (5.37) *due to* (5.38).

However, in higher dimensions and for underlyings that the transition density in not known of this approach of Glasserman and Yu may fail to be feasible. Subsimulations are the way out to ensure (5.37). In effect, one has to compute

$$\widehat{\Delta}_j^k = \widehat{V}_j(X_j) - \frac{1}{k}\sum_{i=i}^{k} \widehat{V}_j\left(X_j^{(i)}\right), \quad j = 0, \dots, \mathcal{J} \tag{5.42}$$

where $X_j^{(1)}, \dots, X_j^{(k)}$ are random variables with distribution of X_j drawn conditionally $\mathcal{F}_{j-1}$. Those can be called “one step subsimulations” starting at X_{j-1} at time step t_{j-1}. The martingale corresponding to those increments $\widehat{\Delta}_j^k$ based on k subsimulations is then $M_j^k = \sum_{i=1}^{j} \widehat{\Delta}_i^k$. It is a martingale adapted to the filtration

$$\mathcal{F}_j' = \mathcal{F}_j \vee \sigma\big\{X_p^{(i)}, p = 0, \dots, j, i = 1, \dots, k\big\}. \tag{5.43}$$

and it is easy to show that this procedure yields an upper bound $Y(M^k)$ of $Y(M)$ like in Remark 80, since obviously

$$\mathrm{E}_{\mathcal{F}_{\mathcal{J}}}[M_j^k] = M_j, \quad j = 0, \dots, \mathcal{J}, \tag{5.44}$$

since the target martingale was given by

$$M_j = \sum_{i=1}^{j} \widehat{V}_j(X_j) - \mathrm{E}[\widehat{V}_j(X_j)|\mathcal{F}_{j-1}]. \tag{5.45}$$

With some assumptions depending on the underlying process and the payoff, some more calculations show that (AR) and (AR') can indeed be fulfilled by this method.

Corollary 83. *If* $\mathrm{E}[\widehat{V}_j^2(X_j)] < \infty$ *for* $j = 0, \dots, \mathcal{J}$*, then assumption (AR) holds with* $\beta = 1$*.*

Corollary 84. *If* $\widehat{V}_j(X_j) < c$ *almost surely for* $j = 0, \dots, \mathcal{J}$*, then assumption (AR') holds with arbitrary large* α *and* $\beta = 1$*.*

The assumption $\widehat{V}_j^2(X_j) < c$ that can be used in both of these two corollaries is likely to be fulfilled in case of a bounded payoff. To achieve boundedness, one could simply truncate the payoff function. One could use $g_j'(x) = \min(g_j(x), M)$, where M is the amount of all the money in the world. It will not be difficult then to implement an estimator fulfilling $\widehat{V}_j(X_j) < c$.

5.2.2 Martingales from Stopping Rules

What is presented in the following is the approach that was originally introduced by Leif Andersen and Mark Broadie[2] in their article [2] . Essentially, they present a method that tries to approximate the martingale part of the true value process V_0 via what is called "testing step" in Chapter 2. Correspondingly, we will fix a (in general suboptimal) stopping time τ. More precisely, the method demands a consistent family of stopping times

$$\tau \equiv \tau_0, \dots, \tau_j, \dots, \tau_{\mathcal{J}} \equiv \mathcal{J} \tag{5.46}$$

and consistency is defined in definition 2. For example, these could be stopping times

$$\tau_s = \inf\Big\{j \in \{s, \dots, \mathcal{J}\} \;:\; C_j^{k'}(X_j) \le g_j(X_j)\Big\}, \tag{5.47}$$

based on some estimate $C_j^{k'}$ of the continuation value issued by some fast approximation method with a fixed parameter k'. Of course, stopping times from other methods can be plugged in here as well.

[2] There is some confusion about the denomination. Quite frequently, the expression "Andersen Broadie method" will be used to refer to the approach using martingales from continuation estimates as given in the previous Subsection.

Because of Theorem 62, we know that the optimal martingale M^* is given by

$$M_j^* = \sum_{i=0}^{j} \mathrm{E}\left[h_{\tau_i^*}(X_{\tau_i^*})\middle|\mathcal{F}_j\right] - \mathrm{E}\left[h_{\tau_i^*}(X_{\tau_i^*})\middle|\mathcal{F}_{i-1}\right], \quad j = 0, \ldots, \mathcal{J}. \tag{5.48}$$

Here, we denote by τ_j^* the optimal stopping time for an option issued newly at time step t_j at price X_j. Those stopping times $\tau_0^*, \ldots, \tau_{\mathcal{J}}^*$ fulfill the definition of consistency.

Analogously, when using a suboptimal family of stopping times $\tau_0, \ldots, \tau_{\mathcal{J}}$, it suggests itself to define

$$M_j = \sum_{i=0}^{j} \mathrm{E}\left[h_{\tau_i}(X_{\tau_i})\middle|\mathcal{F}_i\right] - \mathrm{E}\left[h_{\tau_i}(X_{\tau_i})\middle|\mathcal{F}_{i-1}\right], \tag{5.49}$$

which will be the target martingale of the sequence $(M^k)_{k\in\mathbb{N}}$. The consistency of stopping times allows us to write

$$E[g_{\tau_i}(X_{\tau_i})|\mathcal{F}_i] = \begin{cases} g_i(X_i) & \text{if } \tau_i = i \\ \mathrm{E}\left[g_{\tau_{i+1}}(X_{\tau_{i+1}})|\mathcal{F}_i\right] & \text{if } \tau_j \neq j \end{cases}, \tag{5.50}$$

so by defining

$$F_i := \mathrm{E}\left[g_{\tau_{i+1}}(X_{\tau_{i+1}})|\mathcal{F}_i\right], \quad i = 1, \ldots, \mathcal{J} \tag{5.51}$$

the following simple representation is possible because of a telescopic effect in (5.49).

$$M_j = F_j - F_0 + \sum_{i=0}^{j} \left(g_i(X_i) - F_i\right) 1_{\tau_i = i}. \tag{5.52}$$

The main idea of Andersen and Broadie is to generate a set of k subsamples in each time step t_j to estimate the conditional expectations [3]. More precisely, this means that in each time step, we calculate a quantity F_j^k based on k subsimulations to estimate $\mathrm{E}\left[g_{\tau_{i+1}}(X_{\tau_{i+1}})|\mathcal{F}_i\right]$ and obtain

$$M_j^k = F_j^k - F_0^k + \sum_{i=0}^{j} \left(g_j(X_j) - F_i^k\right) 1_{\tau_i = i}, \tag{5.53}$$

In addition, to use the idea of Theorem 71 for this approach, one should notice that the term (5.53) is independent of F_i^k if $i < j$ and $i \notin \mathfrak{D}$. In summary, we have the following procedure.

1. Draw a sample X from the underlying process.

2. For each time step t_j, $j = 0, \ldots, \mathcal{J}$ run the following three steps to estimate F_j.

[3] Andersen and Broadie recommend to use subsimulations. In Belomestny, Schoenmakers and Dickmann [10] the approach is quite general and includes cases that these expectations are estimated by Monte Carlo simulations that use other techniques than testing the stopping rule on subsamples.

(a) If $j > 0$ and $\tau_j \neq j$ and $j \notin \mathfrak{N}$, the calculations in b) and c) are not necessary, set $F_j^k = 0$.

(b) Generate k subsamples $\left(\widetilde{X}_{\cdot}^{j,(1)}, \tilde{\tau}_{j+1}^{(1)}\right), \ldots, \left(\widetilde{X}_{\cdot}^{j,(k)}, \tilde{\tau}_{j+1}^{(k)}\right)$ starting from X_j at time step t_j.

(c) Estimate F_j via

$$F_j^k = \frac{1}{k} \sum_{r=1}^{k} g_{\tilde{\tau}_{j+1}^{(r)}} \left(\widetilde{X}^{j,(r)}_{\tilde{\tau}_{j+1}^{(r)}} \right). \tag{5.54}$$

3. Calculate the martingale via

$$M_j^k = F_j^k - F_0^k + \sum_{i=0}^{j} \left(g_j(X_j) - F_i^k \right) 1_{\tau_i = 1} \tag{5.55}$$

4. Obtain a realization of $\mathcal{Z}(M^k)$ via

$$\max_{j \in \mathfrak{N}} \left(g_j(X_j) - M_j^k \right) \tag{5.56}$$

We can obtain an estimator like in (5.5) by repeating this procedure N times. The suitable filtration is now

$$\mathcal{F}_j' = \mathcal{F}_j \vee \sigma \left\{ \widetilde{X}_p^{i,(r)}, i = 0, \ldots, j, r = 1, \ldots, k, p = j, \ldots, \mathcal{J} \right\}, \tag{5.57}$$

since the stopping times $\tau_{j+1}^{(r)}$ are adapted to the filtration generated by the subsimulations. It will then also be fulfilled that

$$\mathrm{E}_{\mathcal{F}_{\mathcal{J}}}[M_j^k] = M_j, \quad j = 0, \ldots, \mathcal{J} \tag{5.58}$$

holds, which is in sense of Remark 80.

To make the complexity analysis of this method feasible, we have again two results in order to check the relevance of Theorem 75 and Theorem 76. Both of them include that the bias of the method can be expected to be $O(1/k)$, see Remark 77.

Corollary 85. *If* $\mathrm{E}[g_j(X_j)^2] < \infty$ *for* $j = 0, \ldots, \mathcal{J}$*, assumption (AR) holds with* $\beta = 1$*.*

Corollary 86. *If* $g_j(X_j) < c$ *almost surely for* $j = 0, \ldots, \mathcal{J}$*, assumption (AR') holds with arbitrary large* α *and* $\beta = 1$*.*

Once again, the bounded payoff function is not a crucial assumption. When compared to the method in the previous subsection, it is obvious that the complexity is much higher here, because the subsimulations have to run for more than one time step. Especially in case of many exercise dates, this will become essential. On the other hand, Theorem 18 in Section 2.1 states that finding a good estimate of the continuation value is much more difficult than finding a good stopping time[4], at least when using a fast approximation method. Thus,

[4] The advantage of this approach when compared to the previous subsection is similar to policy iteration. Compare Remark 35 with a stopping time τ^0 implied by some continuation estimates. Then, the usage of the expectation of the payoff under τ_j^1 in (5.42) is the suitable idea.

when the desired precision is very high, the Andersen Broadie approach will be the better choice.

In order to test the estimator (5.31), we retain Benchmark Example 48 and get a stopping rule based on continuation estimates from global regression. This leads to very good upper bounds near to the true value 8.08, which can be seen in Table 5.1. This table is based on 1000 repetitions of the algorithm where the trajectories (not the subsimulations) have been generated using the antithetic variates technique, which reduces the variance by a constant factor.

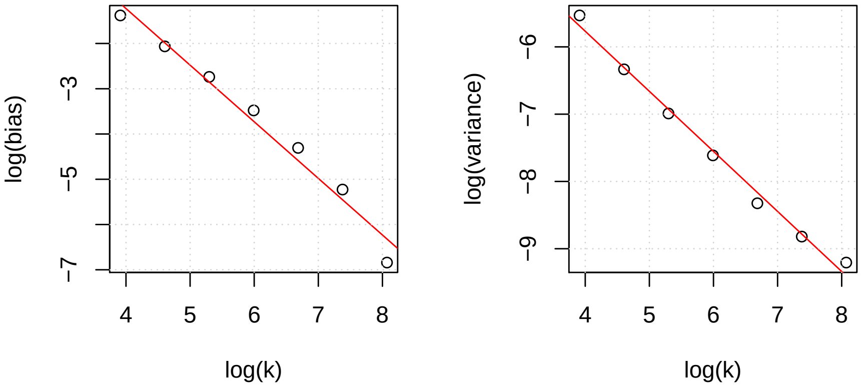

Figure 5.1: The results of Table 5.1 are shown in a log-log plot. The slope of the two red regression lines is about −1.1.

k	100	200	400	800	1600	3200
$\mathrm{E}[\mathcal{Z}(M^k)]$	8.20706	8.14471	8.11085	8.09348	8.08537	8.08107
$\mathrm{Var}[\mathcal{Z}(M^k)]$	1.78	0.92	0.49	0.24	0.15	0.10
$\mathrm{E}[\mathcal{Z}(M^k)] - V_0$	0.12706	0.06471	0.03085	0.01348	0.00537	0.00107

Table 5.1: Results of the Andersen Broadie algorithm.

We find experimentally that the bias and the variance decrease at an order $1/k$. The bias is indicated in the table as $\mathrm{E}[\mathcal{Z}(M^k)] - V_0$. Actually, we should use $\mathrm{E}[\mathcal{Z}(M^k)] - \mathrm{E}[\mathcal{Z}(M)]$, but the stopping time used here is so close to the optimal stopping time τ^*, so that M is very close to M^*. It is also clearly visible that for very small ε, the order of variance decreases, since M is only nearly optimal.

Remark 87. *Many authors, for example Andersen and Broadie [2], use the* *formulation*

$$\Delta_0 = \inf_{M \in \mathfrak{M}} \mathrm{E}\left[\max_j \left(g_j(X_j) - M_j\right)\right], \tag{5.59}$$

where $\mathfrak{M}$ is the set of all $\mathcal{F}$-adapted martingales not necessarily starting at $M_0 = 0$. An upper bound for the option price is then given by $\Delta_0 + M_0$. Obviously, this approach is equivalent to Proposition 64 with $M' = M - M_0$. The first step of the algorithm above is to estimate F_0. This is nothing else but testing the stopping time $\tau(= \tau_0)$, because $\tau_0 \equiv \tau_1$ in a non-degenerate example and exercising at t_0 is suboptimal.

Thus, we could set $\widehat{F}_0^k = 0$ and the result of the algorithm would be an estimator of $\Delta_0^{N,k}$, which should be added to the lower estimator that can be calculated separately. The advantage is that different numbers of trajectories can be used for the two procedures. The quantity Δ_0 will be zero for many trajectories.

Furthermore, Broadie and Cao [16] suggest other methods to improve the efficiency of the Andersen Broadie technique and of lower bound estimators. One of those is called "boundary distance grouping". For each path, one has to check if it approaches the vicinity of the exercise region. If this is the case, then subsamples are created to estimate the value of Δ_0. If not, the procedure is skipped.

Another one of their ideas is called "local policy enhancement". If the trajectory is likely to enter the exercise region, the decision whether to exercise is not based on the given stopping time τ. Instead, the stopping time τ is tested on a large number of subsimulations and the result is compared to the current payoff. This procedure can be seen as a one-step policy iteration, see Remark 35, in the vicinity of the exercise boundary.

They also introduce "sub-optimality checking", which is an improvement of policy fixing. With all those methods at hand, they state that the "computational time is reduced by a factor of several hundred".

5.2.3 Complexity Analysis of Nested Methods

In this section we want to examine the complexity of the Monte Carlo estimator as given in Section 5.2 with a martingale M^K under (AC). Therefore, we recall the definition [5]of $Y^{N,K}$

$$Y^{N,K} = \frac{1}{N} \sum_{n=1}^{N} \max_{j=0,\ldots,\mathcal{J}} \left(g_j(X_j^{(n)}) - M_j^{K,(n)}\right), \tag{5.60}$$

that is based on a set of trajectories

$$\left(X_\cdot^{(n)}, M_\cdot^{K,(n)}\right), \quad n = 1, \ldots, N,$$

which are i.i.d. samples of the vector $(X_\cdot, M_\cdot^K)$. We want to determine the order of complexity that is needed for a desired accuracy ε. The latter is again measured in terms of the root-mean-squared error with respect to the dual upper

[5] An upper-case K is used for $Y^{N,K}$ to distinguish it from the multilevel estimator.

bound of the target martingale (not the true option price). That is, we want to achieve

$$\sqrt{\mathrm{E}\left[\left(Y^{N,K}-Y(M)\right)^2\right]}\leq\varepsilon$$

with minimal cost, where the cost (or complexity) is measured in terms of the number of simulated subsamples

$$C=NK.$$

When the integer numbers N and K are treated as reals, we have the minimization problem

$$\mathscr{C}(\varepsilon):=\inf_{N,K>0}\left\{NK:\mathrm{E}\left[\left(Y^{N,K}-Y(M)\right)^2\right]\leq\varepsilon^2\right\},$$

The solution will be easy, since the bias-variance decomposition says that

$$\begin{aligned}\mathrm{E}\left[\left(Y^{N,K}-Y(M)\right)^2\right]&=\mathrm{E}\left[\left(Y^{N,K}-\mathrm{E}[\mathcal{Z}(M^K)]+\mathrm{E}[\mathcal{Z}(M^K)]-\mathrm{E}[\mathcal{Z}(M)]\right)^2\right]\\&=\mathrm{E}\left[(Y^{N,K}-\mathrm{E}[\mathcal{Z}(M^K)])^2\right]+(\mathrm{E}[\mathcal{Z}(M^K)]-\mathrm{E}[\mathcal{Z}(M)])^2\\&=\mathrm{Var}\left[\mathcal{Z}(M^K)\right]/N+|\,\mathrm{E}[\mathcal{Z}(M^K)-\mathcal{Z}(M)]|^2.\end{aligned}$$

We will call $v_K=\mathrm{Var}[\mathcal{Z}(M^K)]$ and distinguish two cases. Firstly, the case of a surely optimal martingale M and its approximation M^K. Secondly, the usual case that the target martingale M is not surely optimal and we assume the convergence $v_K\to v_\infty\neq 0$. The following corollary is about the second case.

Corollary 88. *Assuming that v_K is non-increasing, under (AR) alone we have that*

$$\mathscr{C}(\varepsilon)\in O\left(\frac{v_{\left\lceil\frac{(2C)^{1/\beta}}{\varepsilon^{2/\beta}}\right\rceil}}{\varepsilon^{2+2/\beta}}\right)\tag{5.61}$$

and in particular, if $v_K\to v_\infty$

$$\mathscr{C}(\varepsilon)\in O\left(\varepsilon^{-2-2/\beta}\right).$$

Hence $\mathscr{C}(\varepsilon)\in O(\varepsilon^{-4})$ for the Andersen-Broadie algorithm. To see that ε^{-4} can be realized and is not just an upper bound for the complexity, we look again at Example 78 in Section 5.2. We saw that in this case, Theorem 75 provides the best result possible and in the proof of Corollary 88, the choice for K^* and N^* is optimal. Furthermore we have that

$$v_K=\mathrm{Var}\left[\max\left(0,\frac{1}{K}\sum_{l=1}^{K}\xi^{(l)}-\xi\right)\right]\longrightarrow\mathrm{Var}[\xi^+]=:v_\infty,$$

which is equal to $\frac{1}{2}-\frac{1}{2\pi}$ and thus different from zero. Using Theorem 76 instead of Theorem 75 for the complexity analysis of course yield better results, so we have the following corollary.

Corollary 89. *Assuming that v_K is non-increasing, under the more restrictive assumptions (AR'), (AL) and (AQ), we have*

$$\mathscr{C}(\varepsilon) \in O\left(\frac{v_{\left\lceil \frac{(2C)^{1/2\gamma}}{\varepsilon^{1/\gamma}} \right\rceil}}{\varepsilon^{2+1/\gamma}}\right) \tag{5.62}$$

and in particular, when $\alpha = \infty$, $\beta = 1$ and $v_K \to v_\infty \neq 0$,

$$\mathscr{C}(\varepsilon) \in O\left(\frac{1}{\varepsilon^{2+\frac{2}{a+1}}}\right).$$

When referring to the situation of Remark 77, we now know that in case of the Andersen Broadie algorithm and assumption (AL) with $0 < a < 1$, the complexity will be of order $\mathscr{C}(\varepsilon) \in O(\varepsilon^{-3-\delta})$ for arbitrary small $\delta > 0$.

For the first case that the target martingale is surely optimal, we have the following results.

Corollary 90. *When using a surely optimal target martingale M, we have under assumption (AR) alone that*

$$\mathscr{C}(\varepsilon) = O\left(\frac{1}{\varepsilon^{2/\beta}}\right). \tag{5.63}$$

In particular, if (AR) is fulfilled with $\beta > 1$ then the complexity is less than ε^{-2}, which is often considered plainly “the complexity of Monte Carlo algorithms”.

Corollary 91. *When using a surely optimal target martingale M, under the more restrictive assumptions (AR'), (AL) and (AQ), we have*

$$\mathscr{C}(\varepsilon) \in O\left(\frac{1}{\varepsilon^{2+(1-\beta)/\gamma}}\right). \tag{5.64}$$

The complexity (5.64) is not better than (5.63) in case of $\beta = 1$, so if the target martingale is M^*, nothing is won when using the Andersen Broadie algorithm.

Remark 92. *In the proof of Corollary 91, we see that the optimal choice for N^* and K^* to achieve the optimal complexity implies $N^* \sim \varepsilon^{\beta/\gamma-2}$ which is constant in the case of (AR), i.e. $\gamma = \beta/2$. This is illustrated at the points where $\gamma \leq \beta/2$ in Figure 5.2. The figure shows the complexity depending on γ and β and is thus a summary of Corollary 90 and Corollary 91.*

So counterintuitively, if the target martingale is surely optimal and $\beta = 1$, it is efficient only to increase the number of subsimulations, but not the number of trajectories. In the general case of a suboptimal martingale, the situation will be similar. Provided that the martingale is not too bad, one should use a high number of subsimulations and a rather small number of trajectories.

In summary, we can state that for the Andersen Broadie method, assumptions (AR'), (AL) and (AQ) allow us to reduce the order of complexity by a factor of nearly ε^{-1} from $O(\varepsilon^{-4})$ down to $O(\varepsilon^{-3-\delta})$. This is also the true order and not just an upper bound, since the assumptions of Theorems 75 and 76 are efficient as shown by Example 78 and Figure 5.2.2. This result will be compared to the multilevel version of the Andersen Broadie estimator in Section 6.1.

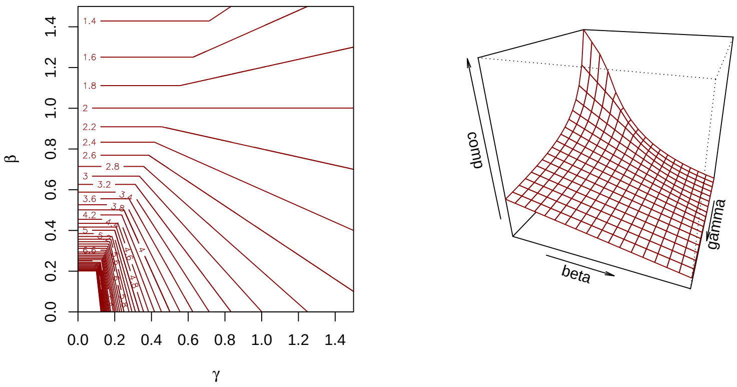

Figure 5.2: The order of complexity following Corollary 91 and Corollary 90 depending on γ and β. If $\beta = 1$, the complexity is constant in γ and equal to ε^{-2}.

5.3 Generalized BSDE Recursion

The optimal stopping problem as defined in Chapter 1 can be seen as a special case of a more general class of stochastic dynamic programming problems. Those problems arise for example when discretizing BSDEs [6] with convex generator or in case of fully nonlinear second order parabolic PDEs [7].

The generalized formulation reads as follows. We are interested in finding an estimator for Y_0 and we are given the recursion

$$Y_j = F_j \left(\mathrm{E}\left[\beta_{j+1} Y_{j+1} \right] \right), \qquad j = 0, \ldots, \mathcal{J} - 1, \tag{5.65}$$

with terminal condition

$$Y_{\mathcal{J}} = F_{\mathcal{J}}(0). \tag{5.66}$$

In particular, the solution of the problem may be multi dimensional, i.e. we have

$$\begin{aligned} F &: \{0, \ldots, \mathcal{J}\} \times \Omega \times \mathbb{R}^{D+1} \to \mathbb{R}, \\ \beta &: \{0, \ldots, \mathcal{J}\} \times \Omega \to \mathbb{R}^{D+1}, \end{aligned}$$

such that D denotes the dimension of the solution, no longer of the underlying. The assumptions on the generator F and β that we ask to obtain a class of problems that are treatable with our techniques will be specified in detail below.

First of all, this formulation obviously includes optimal stopping problems. Therefore, insert $F_j(z) = \max(g_j(X_j), z)$, $\beta \equiv 1$ and $F_{\mathcal{J}} = g_j(X_{\mathcal{J}})$ with some payoff function $g_j(X_j)$. Here, we have $D = 0$ irrespectively of the possibly multi-dimensional underlying X.

[6] Backwards Stochastic Differential Equations

[7] Partial Differential Equations

Since the main purpose of (5.65) is the treatment of BSDEs that typically lead to recursions of this type, let us mention a typical finance related example about funding cost. It is well known that the problem of finding a replicative strategy for the (max-) call option requires borrowing money. The amount of money to borrow increases or decreases form time step to time step. In contrast, when considering a (min-) put option or the negative of a (max-)call option, lending more or less money will be necessary. Thus, in both cases there will be one process modeling the bond with one interest rate. However, when constructing the replicating strategy for the sum of several different options, lending and borrowing can both be necessary. For example, consider the payoff

$$(\max_d X_T^d - \varkappa_1)^+ - 2(\max_d X_T^d - \varkappa_2)^+, \tag{5.67}$$

i.e. the sum of a positive and a negative call option. There will be two interest rates, R^b for borrowing and R^l for lending money, which are now decisive for the solution of the corresponding recursion and thus for the fair price. Suppose D risky assets modeled via Geometric Brownian Motion and the following dependence structure

$$dX_t^d = X_t^d \left(\mu_t^d dt + \sum_{k=1}^{D} \sigma_t^{d,k} dW_t^k \right), \tag{5.68}$$

where σ is assumed to be almost surely invertible with bounded inverse. From El Karoui [33] we learn that the solution of this problem can be represented via a BSDE of the form

$$d\mathscr{Y}_t = -f(t, \mathscr{Y}_t, \mathscr{Z}_t)dt + \mathscr{Z}_t^T dW_t \tag{5.69}$$

with f given by

$$f(t,y,z) = -R_t^l y - z^T \sigma_t^{-1} \left(\mu_t - R_t^l \bar{1} \right) + \left(R_t^b - R_t^l \right) \left(y - z^T \sigma_t^{-1} \bar{1} \right)^- \tag{5.70}$$

and terminal condition

$$\mathscr{Y}_T = (\max_d X_T^d - \varkappa_1)^+ - 2(\max_d X_T^d - \varkappa_2)^+. \tag{5.71}$$

When discretizing this BSDE, we get

$$Y_j = \mathrm{E}[Y_{j+1}|\mathcal{F}_j] + (t_{j+1} - t_j) f\left(t_j, \mathrm{E}\left[Y_{j+1}|\mathcal{F}_j\right], Z_i\right) \tag{5.72}$$

where Z_i is given by

$$Z_i = \mathrm{E}\left[\frac{W_{t_{i+1}} - W_{t_i}}{t_{i+1} - t_i} Y_{i+1} | \mathscr{F}_i \right]. \tag{5.73}$$

At each time step, the amount of money that is invested into the stocks is then given by $Z_t^T \sigma_t^{-1}$. There are other typical finance related examples that lead to a recursion (5.65), like option pricing under model uncertainty or credit value adjustment, for the latter see [27].

Analogously to optimal stopping theory, there are now primal and dual methods that provider lower and upper bounds. Let us note the following three assumptions to make the calculation of dual upper bounds possible.

(R) The process $\beta = (\beta_0, \dots, \beta_D)$ is bounded, adapted and $D+1$ dimensional and fulfills $\beta_0 \equiv 1$. The mapping F is Lipschitz continuous in $z \in \mathbb{R}^{D+1}$ uniformly in (j, ω) and satisfies $\mathrm{E}[|F_j(0)|^2] < \infty$ for every $j = 0, \dots, \mathcal{J}$.

(Comp) For every j and any two $\mathcal{F}_{j+1}$-measurable, integrable real-valued random variables $V, \bar{V}$ such that $V \geq \bar{V}$ a.s., it holds that

$$F_j\left(\mathrm{E}\left[\beta_{j+1}V \middle| \mathcal{F}_j\right]\right) \geq F_j\left(\mathrm{E}\left[\beta_{j+1}\bar{V} \middle| \mathcal{F}_j\right]\right). \tag{5.74}$$

(Conv) The map $z \to F_j(\omega, z)$ is convex for every j and almost every ω.

Assumption (R) ensures that the solution of (5.65) stays square integrable for all time steps. The other ones will be used below. A generalized version of the dual approach makes use of the following definition.

Definition 93. *Let us define* $\theta^{up} : \{0, \dots, \mathcal{J}\} \times \mathcal{M}_{D+1} \to \mathbb{R}$ *recursively via* $\theta^{up}_{\mathcal{J}}(M) = F_{\mathcal{J}}(0)$ *and*

$$\theta^{up}_j(M) = F_j(\beta_{j+1}\theta^{up}_{j+1}(M) - (M_{j+1} - M_j)), \quad j = 0, \dots, \mathcal{J} - 1.$$

Here, $\mathcal{M}_D$ denotes the set of all D-dimensional martingales. This definition is analogue to the expression $\mathcal{Z}(M)$ from Chapter 5, because we have for optimal stopping that

$$\theta^{up}_j(M) = \max(h_j(X_j), \theta^{up}_{j+1}(M) - (M_{j+1} - M_j)) \tag{5.75}$$

which leads to

$$\mathrm{E}\left[\theta^{up}_0(M)\right] = \mathrm{E}\left[\max_{j=0,\dots,\mathcal{J}}(g_j(X_j) - M_j)\right], \tag{5.76}$$

if we restricted ourselves to the use of martingales starting at $M_0 = 0$.

Theorem 94. *Under (R), (Comp) and (Conv), we have that*

$$Y_0 = \inf_{M \in \mathcal{M}_{D+1}} \mathrm{E}\left[\theta^{up}_0(M)\right] \tag{5.77}$$

and even

$$Y_0 = \theta^{up}_0(M^*) \quad a.s. \tag{5.78}$$

for the martingale part M^* *of the* $D+1$ *dimensional process* βY.

As Bender, Schweizer and Zhuo in [11] express it: “Roughly speaking, the idea is to remove all conditional expectations from equation and substact martingale increments, wherever conditional expectations were removed.” In particular, Theorem 94 says that each $D+1$-dimensional martingale $M \in \mathcal{M}_{D+1}$ provides an upper bound via

$$Y_0 \leq \mathrm{E}[\theta^{up}_0(M)], \tag{5.79}$$

which can again be exploited with a Monte Carlo estimator.

In order to realize a nested estimator again, we fix some approximation $\widetilde{Y}$ of Y and estimate the martingale part of $\beta\widetilde{Y}$ via

$$M_j^K = \sum_{i=1}^{j} \left(\beta_i \widetilde{Y}_i - \frac{1}{K} \sum_{\nu=1}^{K} \xi_j^{(\nu)} \right), \tag{5.80}$$

where $\xi_j^{(\nu)}$ are subsamples fulfilling $Law(\xi_j^{(\nu)}) = Law(\beta_j \widetilde{Y}_j | \mathcal{F}_{j-1})$ for all $j = 0, \ldots, \mathcal{J}$. The Monte Carlo estimator is now

$$Y_0^{N,K} = \frac{1}{N} \sum_{n=1}^{N} \theta_0^{up,(n)} \left(M^{K,(n)} \right) \tag{5.81}$$

based on i.i.d. replications

$$\left\{ \left(F_j^{(n)}, \beta_j^{(n)}, M^{K,(n)} \right), \ j = 0, \ldots, \mathcal{J}, \ n = 1, \ldots, N \right\}.$$

and the corresponding multilevel estimator (see Chapter 6) based on $N_0, \ldots, N_L$ paths and $K_0, \ldots, K_L$ subsamples is given by

$$Y_0^{\mathbf{N},\mathbf{K}} = \frac{1}{N_0} \sum_{n=1}^{N_0} \theta^{up,(n)} (M^{K_0,(n)}) \tag{5.82}$$

$$+ \sum_{l=1}^{L} \frac{1}{N_l} \sum_{n=1}^{N_l} \left[\theta_0^{up,(n)} (M^{K_l,(n)}) - \theta_0^{up,(n)} (M^{K_{l-1},(n)}) \right], \tag{5.83}$$

where

$$\left\{ (F_j^{(n)}, \beta_j^{(n)}, M_j^{K_{l-1},(n)}, M_j^{K_l,(n)}), \ n = 1, \ldots, N_l, \ j = 0, \ldots, \mathcal{J} \right\}$$

are i.i.d. replications, coupled as much as possible. Since assumptions like (AL) are much more difficult to derive in the general setting, we have at least that it holds again

$$\mathrm{E}\left[|M_j^K - M_j|^2 \right] = \mathrm{E}\left[\left| \sum_{j=1}^{j} \mathrm{E}_{\mathcal{F}_{i-1}}[\beta_i \widetilde{Y}_i] - \frac{1}{K} \sum_{l=1}^{K} \xi_i^{(l)} \right|^2 \right] \leq \frac{1}{K} \sum_{i=1}^{\mathcal{J}} \mathrm{E}\, |\beta_i \widetilde{Y}_i|^2], \tag{5.84}$$

which belongs to $O(K^{-1})$ if $\mathrm{E}[|\beta_i \widetilde{Y}_i|^2] < \infty$, $i = 1 \ldots, \mathcal{J}$,

We can thus perform the same complexity analysis as it will be done for the optimal stopping case. The standard Monte Carlo estimator (5.81) in general leads to a complexity of ε^{-4}, whereas the multilevel technique reduces the order to ε^{-2}. In practice, we can observe that such problems might have less complexity. We have a bias given asymptotically by $K^{-0.7}$ in the funding cost example, which leads to a complexity of about $\varepsilon^{-3.4}$ for the standard Monte Carlo estimator, see proof of Corollary 89. However, proving such results in the general BSDE case will be very difficult.

Of course, there is not only a dual but also a primal approach available for BSDE recursions like (5.65). Karoui, Peng, Quenez [33] present a primal representation using the convex conjugate of the generator F.

The most popular approach to construct lower bounds are again regression methods, see for example Lemor, Gobet and Warin [56]. Thereby, it is recommended to use martingale basis functions, so that the conditional expectations can be calculated explicitly. As explained before in Remark 82, Glasserman and Yu [42] use such functions for optimal stopping problems. Bender and Steiner [12] use the same idea for the BSDE case.

5.4 Martingales from Convex Optimization

Instead of approximating M^* better and better in order to get an upper bound $Y(M)$ that becomes smaller and smaller, we now want to discuss a procedure that works the other way around. Let us look for some martingale M that minimizes $Y(M)$ irrespectively of its relation to M^*.

Since the set $\mathfrak{M}_0$ of all $\mathcal{F}$-martingales with initial value zero is too large and contains M^*, the idea is senseless when formulated like that. Let us fix a set of martingales that are easy to simulate from. Therefore let (Ψ, ρ) denote a, possibly infinitely dimensional, metric space that will be used as a parameter space for the martingales and define

$$\mathcal{M} = \{M_{\cdot}(\psi) : \psi \in \Psi\}, \tag{5.85}$$

where the function $M(\cdot)$ yields the martingale depending on some parameter. Now we have an upper bound for the true option price V_0 via

$$\inf_{M \in \mathcal{M}} \mathrm{E}\left[\mathcal{Z}(M)\right], \tag{5.86}$$

see Theorem 61 and Monte Carlo simulation can be used to evaluate this expression by "training" and "testing" again. Therefore, look at the empirical formulation of the optimization problem

$$M_N := \arg \inf_{M \in \mathcal{M}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{Z}^{(i)}(M), \tag{5.87}$$

where $\mathcal{Z}^{(i)}(M) := \max_{j=0,\ldots,\mathcal{J}}(Z_j^{(i)} - M_j^{(i)})$ are based on the i.i.d. samples $(Z_{\cdot}^{(1)}, M_{\cdot}^{(1)}), \ldots, (Z_{\cdot}^{(N)}, M_{\cdot}^{(N)})$ of the vector process $(Z_{\cdot}, M_{\cdot})$. Then, simulate R i.i.d. copies of the minimizing martingale and we have

$$\frac{1}{R} \sum_{r=1}^{R} \mathcal{Z}^{(r)}(M_N), \tag{5.88}$$

which is a high biased Monte Carlo estimator for V_0. The testing step (5.88) is necessary to avoid overfitting that could occur in (5.87). Roughly speaking, the minimum in (5.87) could be to well adapted to the N realizations $(Z_{\cdot}^{(i)}, M_{\cdot}^{(i)})$.

However, numerically realizing this procedure is quite complicated. The first question is how to parametrize martingales, so we recall the martingale representation theorem taken from Steele[68]. It is about the continuous American case, but will be helpful. We will turn to the discrete Bermudan case again later.

Theorem 95. *Suppose that X_t is an $\mathcal{F}_t$-martingale, where $\mathcal{F}_t$ is the standard Brownian filtration. If there is a $T > 0$ such that $E[X_T^2] \leq \infty$, then there is a $\varphi(\omega, s) \in \mathcal{L}_a^2[0, T]$ such that* [8]

$$X_t = \int_0^t \varphi(\omega, s) dW_s \quad \textit{for all} \;\; 0 \leq t \leq T. \tag{5.89}$$

Moreover, the representation in this equation is unique up to a set of $dP \times dt$ measure zero.

Applying this theorem to the martingale part of the true price process M^* proofs the possibility to work with basis functions in the one dimensional case. It is not hard to see that in our Markovian setting, it will even hold

$$M_t^* = \int_0^t \varphi^*(X_s, s) dW_s, \tag{5.90}$$

with some φ^*, which is stronger than (5.89) because φ^* is no longer directly depending on ω. It is a self-suggesting idea to fix a set of functions $\varphi^1, \ldots, \varphi^K$ and try to find a linear combination of them such that $\beta_1 \varphi^2 + \ldots + \beta_K \varphi^K \approx \varphi^*$. To get an idea how the basis functions should look like in practice, we consider the next result taken from Wang and Caflisch [75].

Theorem 96. *Suppose the price of an American option V_t^{AM} based on a couple of assets $X_\cdot^1, \ldots, X_\cdot^D$ given by the stochastic differential equations*

$$dX_t^d = \mu^d(t, X_t) dt + \sigma^d(t, X_t) dW_t^d, \quad d = 1, \ldots, D. \tag{5.91}$$

The martingale part M^* of it is then given by

$$M_t^* = \int_0^t \sum_{d=1}^D \frac{\partial V^{AM}(u, X_u)}{\partial x^d} \sigma^d(u, X_u) dW_u^d, \tag{5.92}$$

which depends on the true value V_t^{AM}.

As before, the integrand in (5.92) is square integrable, i.e. we have a vector-valued function $\varphi^* : \mathbb{R} \times \mathbb{R}^D \to \mathbb{R}^D$, such that

$$M_t^* = \int_0^t \varphi^*(s, X_s) dW_s, \tag{5.93}$$

written in vector notation and $\int_0^t \mathrm{E}[|\varphi^*(s, X_s)|^2] ds < \infty$. In practice, we now have to find some heuristic ideas to define functions $\varphi^1, \ldots, \varphi^K : \mathbb{R} \times \mathbb{R}^D \to \mathbb{R}^D$, so that $\sum_{d=1}^D \frac{\partial V}{\partial x^d} \sigma^d$ is included in their span. When defining

$$M(\psi) = \int_0^\cdot \sum_{d=1}^D \sum_{k=1}^K \psi_k \varphi_d^k(s, X_s) dW_s^d \tag{5.94}$$

in accordance with (5.85), the set of martingales becomes

$$\mathcal{M} = \left\{ \sum_{k=1}^K \sum_{d=1}^D \psi_k \int_0^\cdot \varphi_d^k(s, X_s) dW_s^d, \psi \in \mathbb{R}^K \right\}, \tag{5.95}$$

[8] $\mathcal{L}_a^2[0, T]$ is the set of all adapted processes X_t with $\int_0^T \mathrm{E}[X_t^2] dt < \infty$.

so we have $\Phi = \mathbb{R}^K$ as a parameter space. It is possible to reformulate $\mathcal{M}$ as a linear span of martingales

$$\mathcal{M} = \left\{\beta_1 M^1 + \ldots + \beta_K M^K, \beta_1, \ldots, \beta_K \in \mathbb{R}\right\}, \tag{5.96}$$

by setting

$$M_t^k = \sum_{d=1}^{D} \int_0^t \varphi_d^k(s, X_s) dW_s^d. \tag{5.97}$$

The optimization problem $\inf_{M \in \mathcal{M}} \mathrm{E}\left[\mathcal{Z}(M)\right]$ is now finite dimensional and the empirical counterpart (training step)

$$M_N = \arg \inf_{M \in \mathcal{M}} \frac{1}{N} \sum_{n=1}^{N} \mathcal{Z}^{(n)}(M) \tag{5.98}$$

can be reformulated in terms of the coefficients. Therefore, fix N i.i.d. copies

$$(Z_\cdot^{(1)}, W_\cdot^{(1)}, X_\cdot^{(1)}), \ldots, (Z_\cdot^{(N)}, W_\cdot^{(N)}, X_\cdot^{(N)}) \tag{5.99}$$

of the vector process $(Z_\cdot, W_\cdot, X_\cdot)$ and for each $n = 1, \ldots, N$ base the corresponding trajectories of the martingales $M^{1,(n)}, \ldots, M^{K,(n)}$ on those copies via (5.97). Now we have $M_N = M(\beta^*)$, where

$$\beta^* = \arg \inf_{\beta \in \mathbb{R}^K} \frac{1}{N} \sum_{r=1}^{N} \left[\max_j \left(g_j \left(X_j^{(r)} \right) - \sum_{k=1}^{K} \beta_k \, M_j^{k,(r)} \right) \right]. \tag{5.100}$$

The optimization problem (5.100) is called “pathwise optimization problem” by Desai, Farras and Moallemi [31]. It is convex, since it is a “nonnegative linear combination of a set of pointwise suprema of affine functions of β”, as they point out. These mappings preserve convexity. In their article, they state that in the Bermudan case it is possible to reformulate the problem as

$$\min_{\beta \in \mathbb{R}^K, u \in \mathbb{R}^N} \quad \frac{1}{N} \sum_{n=1}^{N} u_n \tag{5.101}$$

$$\text{subject to} \quad u_i + \sum_{i=1}^{j} \sum_{k=1}^{K} \beta^k \left(M_i^{k,(r)} - M_{i-1}^{k,(r)} \right) \geq g_j \left(X_j^{(r)} \right), \tag{5.102}$$

where the constraint (5.102) must hold for all $n = 1, \ldots, N$ and $j = 0, \ldots, \mathcal{J}$. This is a linear program that can be solved with standard methods, for example “GLPK”. For a numerical experiment, we retain Benchmark Example 48 with $D = 5$ assets. Let us recall the properties of the exercise region from Section 1.2. With some inspiration we define the following three classes of basis functions. Denote

$$\mathrm{trig}^k(x) = \begin{cases} \cos(\frac{k}{2}x), & k \text{ even}, \\ \sin(\frac{k+1}{2}x), & k \text{ odd} \end{cases}$$

as an abbreviation for trigonometric functions. Let $\iota_t : [0, T] \to \{1, \ldots, D\}$ be a random ordering at time t, where $\iota_t(1)$ is the index of the highest asset at time t, $\iota_t(2)$ is the index of the second highest and so on, such that

$$X_t^{\iota_t(D)} \leq X_t^{\iota_t(D-1)} \leq \ldots \leq X_t^{\iota_t(1)}.$$

Set $y^d(x) = \log(x^d)/\sqrt{T-t}$, $d = 1, \dots, D$, where the dependence on t is omitted for the sake of notation. For $k = 0, \dots, 24$, choose

$$\varphi_d^k(x,t) = x^d \begin{cases} \operatorname{trig}^k(y^{\iota_t(1)}(x)), & d = \iota_t(1) \\ 0, & \text{otherwise} \end{cases}$$

to approximate (5.92). Here, x^d corresponds to the volatility function σ up to a constant, since the assets are modeled via geometric Brownian motion and the bracket approximates $\frac{\partial V}{\partial x^d}$ via trigonometric functions that depend on the maximum of all assets. In case of a max-call option, the derivative of the true value indeed strongly depends on this maximum, especially in the vicinity of the exercise region (and completely within the exercise region).

For $k = 25, \dots, 49$, define the functions

$$\varphi_d^k(x,t) = x^d \begin{cases} \operatorname{trig}^{k-25}(y(x)^{\iota_t(1)} - y(x)^{\iota_t(2)}), & d = \iota_t(1) \\ -\operatorname{trig}^{k-25}(y(x)^{\iota_t(1)} - y(x)^{\iota_t(2)}), & d = \iota_t(2) \\ 0, & \text{otherwise,} \end{cases}$$

that are motivated by the properties of the "push", which is the difference between the highest and second highest asset. The higher the push, the more likely exercising is optimal, see Section 1.2. Thus, the true value and so its derivative might depend on the push.

For $k = 50, \dots, 74$, we set

$$\varphi_d^k(x,t) = x^d \operatorname{trig}^{k-50}(y(x)^d)$$

to model dependence from each single asset and finally have 75 martingales defined by

$$M_t^k = \sum_{d=1}^{D} \int_0^t \varphi_d^k(X_s^d, s) dW_s^d.$$

Let us return to the Bermudan setting. It is convenient to use a discretization according to the exercise dates and obtain

$$\widehat{M}_j^k = \sum_{d=1}^{D} \sum_{i=0}^{j-1} \varphi_d^k(X_i^d, t_i)(W_{i+1}^d - W_i^d), \tag{5.103}$$

which are still martingales in the discrete setting. Unfortunately, for a small number of exercise dates the approximation (5.103) is not a good choice. The martingale representation theorem holds for the continuous case, so the finite sum must be close to the integral to provide the existence of a suitable integrand φ^*. Thus, it is convenient to add more time steps to the sum approximating the integral without changing the number of exercise dates. In this example, we used 300 such time steps and obtain martingales $\widetilde{M}$. We finally have a discretized family

$$\mathcal{M} = \left\{ \beta_1 \widetilde{M}_\cdot^1 + \dots + \beta_{75} \widetilde{M}_\cdot^{75},\ \beta_1, \dots, \beta_{75} \in \mathbb{R} \right\}.$$

Many numerical optimization methods prefer smooth functions. However, our problem is not that smooth, since the maximum function

$$\arg \max_{j=0,\dots,\mathcal{J}} (Z_j - M_j(\psi)) \tag{5.104}$$

may "jump" for small changes of the parameter ψ, so it will be difficult for the gradient methods to find the right direction to look for the minimum. For a vector $(x_0, \dots, x_{\mathcal{J}})$, we define a smoothed version of the maximum function via

$$\overset{p}{\max} x = \frac{1}{p} \log \left(\sum_{j=0}^{\mathcal{J}} \exp(px_j) \right) \tag{5.105}$$

with some parameter $p > 0$ so that

$$\overset{p}{\max} x \to \max x, \text{ as } p \to \infty. \tag{5.106}$$

The reason is that

$$\begin{aligned}
&\overset{p}{\max} x - \max x \\
&= \frac{1}{p} \log \left(\sum_{j=0}^{\mathcal{J}} \exp(px_j) \right) - 1/p \log(\exp(p \max x)) \\
&= 1/p \log \left(\sum_{j=0}^{\mathcal{J}} \exp(px) / \exp(p \max x) \right) \\
&\leq 1/p \log(\mathcal{J} + 1) \to 0, \quad \text{as} \quad p \to \infty.
\end{aligned}$$

The results based on 5000 paths for estimating the coefficients $\beta_1, \dots, \beta_{75}$ and 5000 testing paths are presented in Table 5.2. For comparison, we also give the confidence intervals from Andersen Broadie [2] based on 10000 inner simulations and the estimates from Rogers [64] who makes use of the Black-Scholes formula and Remark 50. In parenthesis the deviations from the corresponding upper confidence intervals of Andersen and Broadie are given. The computation time is of order of 5 minutes on 2.1Gz processor.

X_0^d	A&B Interval	Rogers Dual	Sieve Dual	Standard deviation
90	[16.602, 16.655]	16.98 (1.95%)	17.02 (2.19%)	0.03142
100	[26.109, 26.292]	26.75 (1.74%)	26.60 (1.17%)	0.03997
110	[36.704, 36.832]	37.61 (2.11%)	37.27 (1.19%)	0.04734

Table 5.2: Convex optimization in case of a 5-dimensional max-call Bermudan option.

It is very unlikely that the true value can be reached when the martingales are constructed in such a way. However, in case of some simple options it might be the case that there are martingales in $\mathcal{M}$ that have the surely optimal property. Let $\Psi^* \subset \Psi$ denote the subset of all parameters ψ such that $M(\psi)$ is a martingale with this surely optimal property.

Belomestny [7] suggests to introduce a penalization term motivated by the statement of Theorem 66, which leads to the optimization problem

$$\inf_{M \in \mathcal{M}} \mathrm{E}\left[\mathcal{Z}(M)\right] + \lambda \sqrt{\mathrm{Var}\left[\mathcal{Z}(M)\right]}, \tag{5.107}$$

where $\lambda > 0$ determines the degree of penalization by the variance. For a good set of martingales, the penalization will have hardly any disadvantages. Especially for small λ, it will only prefer a solution with small variance from all solutions that are nearly optimal in (5.86). If Ψ^* is nonempty, there is no disavdantage, since every martingale associated with Ψ^* is also a solution of (5.107) for all $\lambda > 0$. However, their are some advantages.

We recall that the number of martingales fulfilling the surely optimal property might be large, and additionally there may be infinitely many martingales that minimize $E[\mathcal{Z}(M)]$ without having the surely optimal property, see Theorem 65 and the following remarks.

Thus, the approach (5.107) is advantageous compared to (5.86) irrespectively whether $\Psi^* \neq \emptyset$ holds or not because it provides a martingale with smaller variance for the testing step anyway, so that less samples will be needed. The variance can be seen as a regularization term with regularization parameter λ, see for example Hofmann [48]. Now consider the empirical version of (5.107)

$$M_N := \arg \inf_{M \in \mathcal{M}} \left(\frac{1}{N} \sum_{n=1}^{N} \mathcal{Z}^{(n)}(M) + \lambda \sqrt{V_N(M)} \right), \quad \lambda > 0, \tag{5.108}$$

where

$$V_N(M) := \frac{1}{N(N-1)} \sum_{1 \le n < m \le N} (\mathcal{Z}^{(n)}(M) - \mathcal{Z}^{(m)}(M))^2.$$

is a well known unbiased estimator for the variance. Again, in case of a linear span of martingales as in (5.100), the problem is numerically feasible by optimizing over the coefficients. This problem can no longer be solved via linear programming. Methods for convex optimization are necessary here, see Chapter 7.

If Ψ is infinite-dimensional, minimizing (5.108) over $\psi \in \Psi$ may not be well-defined. Even if a minimizer exists, it is generally difficult to compute or has a very slow rate of convergence. Therefore, the introduction of "sieves" is motivated as explained in the next subsection.

5.4.1 Sieves Method

In case of an infinitely dimensional parameter space Ψ, difficulties may arise because the problem of optimization over an infinite-dimensional noncompact space may no longer be well-posed. Thus, it is convenient to look for an approximating sequence of non-decreasing parameter subspaces $\Psi_1 \subseteq \Psi_2 \subseteq \ldots \subseteq \Psi$, called "sieves" in the following. Then we have suitable sequence of families of martingales $\mathcal{M}_1, \mathcal{M}_2, \ldots$, where $\mathcal{M}_\nu = \{M(\psi), \psi \in \Psi_\nu\}$.

We consider compact sieves that provide the existence of some projection mapping $\pi_\nu : \Psi \to \Psi_\nu$, which maps each parameter from Ψ to some point in Ψ_ν such that $\rho(\psi, \pi_\nu \psi) \to 0$ as $\nu \to \infty$. In the following, this role is played by the functions that are used as integrands in the martingale representation in Theorem 96. More precisely, we now use $\Psi = \mathcal{L}^2_P([0,T] \times \mathbb{R}^D)$ instead of $\Psi = \mathbb{R}^D$, but we retain linear sieves of the form

$$\Psi_\Lambda := \{\alpha_1 \varphi^1 + \ldots + \alpha_\Lambda \varphi^\Lambda : \alpha_1, \ldots, \alpha_\Lambda \in \mathbb{R}\}, \tag{5.109}$$

as before. As the choice $\lambda = 0$ is senseless for all grades of approximation, we look at the martingales that are solutions to the slightly modified problem

$$M_N := \arg \inf_{M \in \mathcal{M}_{\Lambda_N}} \left(\frac{1}{N} \sum_{n=1}^{N} \mathcal{Z}^{(n)}(M) + (1 + \lambda_N) \sqrt{V_N(M)} \right), \tag{5.110}$$

where $\Lambda_N \to \infty$ and $\lambda_N \to 0$ as $N \to \infty$. It was shown by Belomestny [7] that under a proper choice of λ_N and Λ_N

$$\max \left\{ E[\mathcal{Z}(M_N)] - V_0, \sqrt{\mathrm{Var}[\mathcal{Z}(M_N)]} \right\} = O_P \left(\delta_N + \Lambda_N^{D+1} \log(\Lambda_N)/\sqrt{N} \right),$$

where $\delta_N = \inf_{\psi \in \Psi_{\Lambda_N}, \psi^* \in \Psi^*} \rho(\psi, \psi^*)$. In other words, the bias and the variance are converging to zero, where the convergence rate mainly depends on the density of the sieves measured by δ_n. Without the penalization it is not possible to achieve this bound. The sequence $(M_N)_{N \in \mathbb{N}}$ could get stack at a martingale which is minimizing (5.86) without having the optimal surely property. The notation $O_P(f)$ is called the stochastic Landau symbol and is defined following van der Vaart [73].

Definition 97. *The stochastic Landau symbol O_P is defined as follows.*

$$X_n \in O_P(R_n) \text{ means } X_n = Y_n R_n \text{ and } Y_n \in O_P(1),$$

where $O_P(1)$ denotes a sequence that is bounded in probability.

5.5 Proofs

Proof of Theorem 62

From the Doob decomposition, i.e. Theorem 63, we know that there is a martingale M^* such that

$$V_j = V_0 - A_j + M_j^*$$

with $A_0 = M_0^* = 0$ and A is an increasing process since the true value process V is a supermartingale. Inserting M^* into (5.1) leads to

$$\begin{aligned} \mathcal{Z}(M^*) &= \max_j \left(Z_j - M_j^* \right) \qquad (5.111) \\ &= \max_j \left(Z_j - V_j + V_0 - A_j \right) \\ &= V_0 + \underbrace{\max_j \left(Z_j - V_j - A_j \right)}_{\leq 0} \leq V_0. \end{aligned}$$

The underbraced term is less or equal to zero since the true value is always greater or equal to the current payoff by definition and A_j is an increasing process starting at initial value 0. At the same time, equation (5.1) tells us that the expectation of $\mathcal{Z}(M^*)$ is greater or equal to V_0, so equality

$$V_0 = \max_j (Z_j - M_j^*)$$

must hold almost surely.

Proof of Theorem 69

The left hand sides of the two inequalities are implied by Theorems 61 and 68. To prove the right hand sides of the inequalities, check that

$$\begin{aligned} W_j^M &= \mathrm{E}\left[\max_{j\le i\le\mathcal{J}}(Z_i - M_i)|\mathcal{F}_j\right] + M_j & (5.112)\\ &\le \mathrm{E}\left[\max_{j\le i\le\mathcal{J}}(W_i - M_i)|\mathcal{F}_j\right] + M_j & (5.113)\\ &= \mathrm{E}\left[\max_{j\le i\le\mathcal{J}} A_i|\mathcal{F}_j\right] + M_j = A_j + M_j = W_j, & (5.114)\end{aligned}$$

which is basically the same argument as in the proof of Theorem 62. The first equality in (5.114) follows since A is nonincreasing. The multiplicative case follows analogous:

$$\begin{aligned} W_j^B &= \mathrm{E}^B\left[\max_{j\le i\le\mathcal{J}}\frac{g_i(X_i)}{B_i}\Big|\mathcal{F}_j\right]B_j\\ &\le \mathrm{E}^B\left[\max_{j\le i\le\mathcal{J}}\frac{W_i}{B_i}\Big|\mathcal{F}_j\right]B_j\\ &= \mathrm{E}^B\left[\max_{j\le i\le\mathcal{J}}L_i\Big|\mathcal{F}_j\right]B_j = L_jB_j = W_j,\end{aligned}$$

because L is nonincreasing.

Proof of Theorem 71

There are two possibilities for the proof, we will use one for the two statements either.

We define an "equivalent payoff" with the help of the true continuation value function C_j via

$$g_j'(x) = \begin{cases} -\infty & , C_j(x) > g_j(x)\\ g_j(x) & , \text{otherwise}\end{cases}.$$

In particular, $g'_{\mathcal{J}} \equiv g_{\mathcal{J}}$, because $C_{\mathcal{J}} \equiv -\infty$ by definition. Now, since $\tau^* = \mathcal{J}$ in case of $g_i(X_i) = 0$ for all $i = 0, \ldots, \mathcal{J}-1$, we have

$$\sup_{\tau\in\mathcal{T}}\mathrm{E}\left[g'_\tau(X_\tau)|X_0 = x_0\right] = \sup_{\tau\in\mathcal{T}}\mathrm{E}\left[g_\tau(X_\tau)|X_0 = x_0\right].$$

This can be interpreted as two options with the same true value. So we have

$$\begin{aligned}\mathrm{E}[\max_{\mathfrak{N}} g_j(X_j) - M_j] \ge \mathrm{E}[\max_{\mathfrak{O}} g_j(X_j) - M_j] = \mathrm{E}[\max_j g_j'(X_j) - M_j]\\ = \sup_{\tau\in\mathcal{T}}\mathrm{E}\left[g'_\tau(X_\tau)|X_0 = x_0\right] = \sup_{\tau\in\mathcal{T}}\mathrm{E}\left[g_\tau(X_\tau)|X_0 = x_0\right] = V_0\end{aligned}$$

which proves (5.24).

The following idea is taken from Caflisch and Wang [75]. In case of the Doob martingale M^*, let t be a time step such that $(t, X_t) \in \mathcal{C}$ is a continuation point and let s^* be the next optimal exercise time, i.e. $s^* = \min\{s > t : (s, X_s) \in \mathcal{E}\}$

and exercising at the last time step is optimal per definition. We know from Proposition 70 that $M^*_{s^*} = M^*_s - V_s - V_{s^*}$, since the path is contained in the continuation region. Thus

$$Z_t - M_t^* < V_t - M_t^* = V_{s^*} - M^*_{s^*} = Z_{s^*} - M^*_{s^*}, \tag{5.115}$$

so the maximum of $Z_t - M_t^*$ cannot be attained at a time steps t where exercising is suboptimal.

Proof of Corollary 74

Because of the Doob decomposition $V_j = V_0 + M_j^* - A_j$, we note at first that

$$\begin{aligned}\Lambda &= \min_{j\notin\mathcal{Q}}\left(V_0 - Z_j + M_j^*\right)\\ &= \min_{j\notin\mathcal{Q}}\left(V_j - Z_j + A_j^*\right)\\ &\geq \min_{j\notin\mathcal{Q}}\left(V_j - Z_j\right) \geq \min_{\{j:V_j>Z_j\}}\left(V_j - Z_j\right),\end{aligned}$$

where the last inequality follows because we know from Theorem 71 that the maximum of $Z_t - M_t^*$ cannot be attained at a point where exercising is suboptimal. In other words $j \notin \mathcal{Q}$ leads to $V_j > Z_j$. We now have up to a constant (for the sake of notation, fix j to be the "worst time step") that

$$\begin{aligned}P(\Lambda^{-a} \leq x) &\geq P((V_j - Z_j)^{-a} \leq x | V_j > Z_j)\\ &= P((V_j - Z_j) \geq x^{-1/a} | V_j > Z_j),\end{aligned}$$

which is the conditional probability of the event that the asset stays in the vicinity of the exercise boundary.

Since we may assume the boundary assumption (AB)

$$\mathrm{P}(|C_j - Z_j| \leq \delta) \leq A\delta^\alpha, \quad \delta \to 0, \tag{5.116}$$

even for all time steps, we can bound $P(V_j - Z_j \geq \delta | V_j > Z_j)$ with the help of the law of total probability

$$\begin{aligned}P(|C_j - Z_j| \leq \delta) =& P(C_j - Z_j \leq \delta | C_j > Z_j) P(C_j > Z_j)\\ &+ P(Z_j - C_j \leq \delta \wedge C_j \leq Z_j).\end{aligned}$$

Notice that $C_j = V_j$ if $C_j > Z_j$, so we have

$$\begin{aligned}P(V_j - Z_j \leq \delta | V_j > Z_j) &= \frac{P(|C_j - Z_j|) \leq \delta) - P(Z_j - C_j \leq \delta \wedge C_j \leq Z_j)}{P(V_j > Z_j)}\\ &\leq \frac{A\delta^\alpha - P(Z_j - C_j \leq \delta \wedge C_j \leq Z_j)}{P(V_j > Z_j)},\end{aligned}$$

which is smaller than $A\delta^\alpha$ if $P(V_j > Z_j)$ is greater than zero. This can be assumed for a nondegenerate example. So using $\delta = x^{-1/a}$ yields

$$\begin{aligned}
\mathrm{E}[\Lambda^{-a}] &= \int_0^\infty (1 - P(\Lambda^{-a} \leq x))dx \\
&\leq \int_0^\infty (1 - P(V_j - Z_j \geq x^{-1/a} | V_j > Z_j))dx \\
&= \int_0^\infty P(V_j - Z_j < x^{-1/a} | V_j > Z_j)dx \\
&\leq \int_0^\infty A x^{-\alpha/a} dx,
\end{aligned}$$

which is finite if $a < \alpha$ and so (AL) is fulfilled.

Proof of Theorem 75

On the one hand, it holds for each $k \in \mathbb{N}$ and $j_k^{\max} := \min \mathcal{Q}^k$ that

$$\begin{aligned}
\mathcal{Z}(M^k) - \mathcal{Z}(M) &= \max_{j=0,\dots,\mathcal{J}} (Z_j - M_j^k) - \max_{j=0,\dots,\mathcal{J}} (Z_j - M_j) \\
&\leq M_{j_k^{\max}} - M_{j_k^{\max}}^k, \quad \text{a.s.,} \qquad (5.117)
\end{aligned}$$

and on the other hand, we get for each $k \in \mathbb{N}$ and $\mathcal{Q} =: \{j^{\max}\}$,

$$\mathcal{Z}(M^k) - \mathcal{Z}(M) \geq M_{j^{\max}} - M_{j^{\max}}^k. \qquad (5.118)$$

By (5.117) and (5.118) we thus have

$$\mathrm{E}\left[\left(\mathcal{Z}(M^k) - \mathcal{Z}(M)\right)^2\right] \leq \mathrm{E}\left[\max_{j=1,\dots,\mathcal{J}} \left(M_j - M_j^k\right)^2\right] \leq B k^{-\beta}.$$

Further, by the Cauchy-Schwarz inequality we so have immediately,

$$|\,\mathrm{E}[\mathcal{Z}(M^k) - \mathcal{Z}(M)]| \leq \left\{\mathrm{E}\left[\max_{j=1,\dots,\mathcal{J}} \left(M_j - M_j^k\right)^2\right]\right\}^{1/2} \leq \sqrt{B} \cdot k^{-\beta/2}.$$

Proof of Theorem 76

Let us now turn to the case where in addition assumptions (AR'), (AL) and (AQ) are fulfilled. From (5.117) we obtain for $k \in \mathbb{N}$,

$$\mathrm{E}_{\mathcal{F}_\mathcal{J}}[\mathcal{Z}(M^k) - \mathcal{Z}(M)] \leq \mathrm{E}_{\mathcal{F}_\mathcal{J}}\left[M_{j_k^{\max}} - M_{j_k^{\max}}^k\right]$$

$$= \underbrace{\mathrm{E}_{\mathcal{F}_\mathcal{J}}\left[\left(M_{j_k^{\max}} - M_{j_k^{\max}}^k + M_{j^{\max}}^k - M_{j^{\max}}\right) 1_{j_k^{\max} \neq j^{\max}}\right]}_{(I)} + \mathrm{E}_{\mathcal{F}_\mathcal{J}} \underbrace{\left[M_{j^{\max}} - M_{j^{\max}}^k\right]}_{(II)}$$

Note that $Z_{j_k^{\max}} - M_{j_k^{\max}}^k \geq Z_{j^{\max}} - M_{j^{\max}}^k$ and hence

$$\{j_k^{\max} \neq j^{\max}\} \subset \left\{\max_{j=1,\dots,\mathcal{J}} (M_j - M_j^k + M_{j^{\max}}^k - M_{j^{\max}}) \geq \Lambda\right\}.$$

We thus have

$$\begin{aligned}
\mathrm{P}_{\mathcal{F}_{\mathcal{J}}}(j_k^{\max} \neq j^{\max}) &\leq \mathrm{P}_{\mathcal{F}_{\mathcal{J}}}\left(\max_{j=1,\dots,\mathcal{J}}(M_j - M_j^k + M_{j^{\max}}^k - M_{j^{\max}}) \geq \Lambda\right) \\
&\leq \mathrm{P}_{\mathcal{F}_{\mathcal{J}}}\left(\max_{j=1,\dots,\mathcal{J}}(M_j - M_j^k) \geq \Lambda/2\right) + \mathrm{P}_{\mathcal{F}_{\mathcal{J}}}\left(M_{j^{\max}}^k - M_{j^{\max}} \geq \Lambda/2\right) \\
&\leq \mathrm{P}_{\mathcal{F}_{\mathcal{J}}}\left(\max_{j=1,\dots,\mathcal{J}}(M_j - M_j^k) \geq \Lambda/2\right) + \mathrm{P}_{\mathcal{F}_{\mathcal{J}}}\left(\max_{j=1,\dots,\mathcal{J}}(M_j^k - M_j) \geq \Lambda/2\right).
\end{aligned}$$

By (AR') and the conditional Markov inequality it follows that

$$\mathrm{P}_{\mathcal{F}_{\mathcal{J}}}\left(\max_{j=1,\dots,\mathcal{J}}(M_j - M_j^k) \geq \Lambda/2\right) \leq \frac{4B}{\Lambda^2}k^{-\beta}, \quad \mathrm{P}_{\mathcal{F}_{\mathcal{J}}}\left(\max_{j=1,\dots,\mathcal{J}}(M_j^k - M_j) \geq \Lambda/2\right) \leq \frac{4B}{\Lambda^2}k^{-\beta}.$$

Hence

$$\mathrm{P}_{\mathcal{F}_{\mathcal{J}}}(j_k^{\max} \neq j^{\max}) \leq \frac{8B}{\Lambda^2}k^{-\beta}$$

for all k. Furthermore, due to (AR') and a conditional version of the generalized Hölder inequality

$$\mathrm{E}\,|XY| \leq \|X\|_p \, \|Y\|_q$$

for $\frac{1}{p} + \frac{1}{q} \leq 1$ and $p, q \geq 1$, we obtain by taking $q = 2$ and $p = \max\{2, 2/a\}$,

$$\begin{aligned}
(I) &\leq [\mathrm{P}_{\mathcal{F}_{\mathcal{J}}}(j_k^{\max} \neq j^{\max})]^{1/p} \sqrt{4\,\mathrm{E}_{\mathcal{F}_{\mathcal{J}}}\left[\max_{j=1,\dots,\mathcal{J}} \left(M_j - M_j^k\right)^2\right]} \\
&\leq \frac{2\,(8B)^{1/p}}{\Lambda^{2/p}} k^{-\beta/p} B k^{-\beta/2} =: \frac{B_1}{\Lambda^{2/p}} k^{-\beta(1/p+1/2)}.
\end{aligned}$$

Combining (5.117) with (5.118) and using assumption (AR') again for the term (II), we arrive at the inequality

$$-Ak^{-\alpha} \leq \mathrm{E}[\mathcal{Z}(M^k) - \mathcal{Z}(M)] \leq B_1 k^{-\beta(1/p+1/2)}\, \mathrm{E}\left[\frac{1}{\Lambda^{2/p}}\right] + Ak^{-\alpha} \leq Ck^{-\gamma}$$

with $\gamma = \min\{\alpha, \beta \min\{1, (a+1)/2\}\}$ and some $C > 0$.

Proof of Remark 79

In Example 78, there are three values possible for M_1, so

1. $\xi = 3/2b \;\; \Rightarrow \;\; M_1 = 1/2b,\ \mathcal{Z}(M) = 0,\ \mathcal{Q} = \{0\},\ \Lambda = 1/2b$
2. $\xi = b \;\; \Rightarrow \;\; M_1 = 0,\ \mathcal{Z}(M) = 0,\ \mathcal{Q} = \{0, 1\},\ \Lambda = \infty$ per definition
3. $\xi = 1/2b \;\; \Rightarrow \;\; M_1 = 1/2b,\ \mathcal{Z}(M) = 1/2b,\ \mathcal{Q} = \{1\},\ \Lambda = 0,$

so we have

$$\mathrm{E}[\Lambda^{-a}] = \quad \mathrm{P}(\xi = 3b/2) \times (1/2b)^{-a} + \mathrm{P}(\xi = b) \times \infty^{-a} + \mathrm{P}(\xi = b/2) \times 0^{-a}.$$

Hence, $\mathrm{E}[\Lambda^{-a}] < \infty$ is not fulfilled for $a > 0$.

Furthermore, with positive probability we have that $\xi = \mathrm{E}[\xi]$ in the second case and we have the vector $Z - M = (0, E[\xi] - \xi) = (0, 0)$, which violates (AQ).

Proof of Corollary 83

We check that

$$\begin{aligned}
&\mathrm{E}_{\mathcal{F}_\mathcal{J}}[\max_j (M_j^k - M_j)^2] \\
&\le \mathrm{E}_{\mathcal{F}_\mathcal{J}}\left[\sum_{j=0}^{\mathcal{J}} \left\{M_j^k - M_j\right\}^2\right] \\
&= \mathrm{E}_{\mathcal{F}_\mathcal{J}}\left[\sum_{j=0}^{\mathcal{J}} \left\{\sum_{i=1}^{j} \left(\mathrm{E}_{\mathcal{F}_{i-1}}\left[\widehat{V}_i(X_i)\right] - \frac{1}{k}\sum_{l=1}^{k} \widehat{V}_i\left(X_i^{(l)}\right)\right)\right\}^2\right] \\
&= \mathrm{E}_{\mathcal{F}_\mathcal{J}}\left[\sum_{j=0}^{\mathcal{J}} \left\{\sum_{i=1}^{j} \mathrm{E}_{\mathcal{F}_{i-1}}\left(\mathrm{E}_{\mathcal{F}_{i-1}}\left[\widehat{V}_i(X_i)\right] - \frac{1}{k}\sum_{l=1}^{k} \widehat{V}_i\left(X_i^{(l)}\right)\right)^2\right\}\right]
\end{aligned}$$

holds and because of the law of the subsamples we thus have

$$\begin{aligned}
&\mathrm{E}_{\mathcal{F}_\mathcal{J}}[\max_j (M_j^k - M_j)^2] \\
&\le \sum_{j=0}^{\mathcal{J}} \sum_{i=1}^{j} \mathrm{E}_{\mathcal{F}_\mathcal{J}} \frac{1}{k}\left[\mathrm{Var}_{\mathcal{F}_{i-1}}[\widehat{V}_i(X_i)]\right] \\
&= \frac{\mathcal{J}}{k} \sum_{i=1}^{j} \mathrm{E}_{\mathcal{F}_\mathcal{J}}\left[\mathrm{Var}_{\mathcal{F}_{i-1}}[\widehat{V}_i(X_i)]\right] \\
&\le \frac{\mathcal{J}}{k} \sum_{i=1}^{\mathcal{J}} \mathrm{E}_{\mathcal{F}_\mathcal{J}}\left[\left(\mathrm{E}_{\mathcal{F}_{i-1}}\left[\widehat{V}_i(X_i)\right]\right)^2\right] \\
&\le \frac{\mathcal{J}}{k} \sum_{i=1}^{\mathcal{J}} \mathrm{E}_{\mathcal{F}_\mathcal{J}}\left[\widehat{V}_i^2(X_i)\right].
\end{aligned}$$

For the unconditional expectation, it thus follows

$$\begin{aligned}
\mathrm{E}\,\mathrm{E}_{\mathcal{F}_\mathcal{J}}\left[\max_j (M_j^k - M_j)^2\right] &\le \frac{\mathcal{J}}{k}\sum_{i=1}^{\mathcal{J}} \mathrm{E}\,\mathrm{E}_{\mathcal{F}_\mathcal{J}}\left[\widehat{V}_i^2(X_i)\right] \\
\frac{\mathcal{J}}{k}\sum_{i=1}^{\mathcal{J}} \mathrm{E}\left[\widehat{V}_i^2(X_i)\right] &\le \frac{c\mathcal{J}}{k} =: k^{-1}B,
\end{aligned}$$

since we may assume that $\mathrm{E}[\widehat{V}_j^2(X_j)] < \infty$, $j = 0, \ldots, \mathcal{J}$.

Proof of Corollary 84

We know from the proof of Corollary 83 that

$$\mathrm{E}_{\mathcal{F}_\mathcal{J}}\left[\max_j (M_j^k - M_j)^2\right] \le \frac{\mathcal{J}}{k}\sum_{i=1}^{\mathcal{J}} \mathrm{E}_{\mathcal{F}_\mathcal{J}}\left[\widehat{V}_i^2(X_i)\right],$$

which is smaller than $\mathcal{J}\frac{c^2}{k}$ if $\widehat{V}_j(X_j) < c$ almost surely and $\beta = 1$ is shown. Furthermore, (5.44) can be expressed by writing $\alpha = \infty$.

Proof of Corollary 85 and 86

It holds

$$\begin{aligned}
&\mathrm{E}_{\mathcal{F}_{\mathcal{J}}}\left[\max_j (M_j^k - M_j)^2\right] \\
&\le \mathrm{E}_{\mathcal{F}_{\mathcal{J}}}\left[\sum_{j=0}^{\mathcal{J}} (M_j^k - M_j)^2\right] \\
&= \sum_{j=0}^{\mathcal{J}} \mathrm{E}_{\mathcal{F}_{\mathcal{J}}}\left[\left\{F_j^k - F_j + F_0^k - F_0 + \sum_{i=0}^{j} \left(F_i - F_i^k\right) 1_{\tau_i = i}\right\}^2\right] \\
&= \sum_{j=0}^{\mathcal{J}} \mathrm{E}_{\mathcal{F}_{\mathcal{J}}}\left[\left\{F_j^k - F_j\right\}^2 1_{\tau_j \neq j} + \left\{F_0^k - F_0\right\}^2 + \sum_{i=0}^{j-1} \left\{F_i - F_i^k\right\}^2 1_{\tau_i = i}\right] \\
&= \sum_{j=0}^{\mathcal{J}} \mathrm{E}_{\mathcal{F}_{\mathcal{J}}}\left[\sum_{i=0}^{\mathcal{J}} \left\{F_i - F_i^k\right\}^2\right] \\
&\le \frac{\mathcal{J}}{k} \sum_{i=0}^{\mathcal{J}} \mathrm{E}_{\mathcal{F}_{\mathcal{J}}}\left[\mathrm{Var}_{\mathcal{F}_i}[F_i]\right] \le \frac{\mathcal{J}}{k} \sum_{i=0}^{\mathcal{J}} \mathrm{E}_{\mathcal{F}_i}\left[F_i^2\right].
\end{aligned}$$

We have for Corollary 85 that

$$\mathrm{E}\left[\frac{\mathcal{J}}{k} \sum_{i=0}^{\mathcal{J}} \mathrm{E}_{\mathcal{F}_i}\left[F_i^2\right]\right] \le \frac{\mathcal{J}}{k} \sum_{i=0}^{\mathcal{J}} \mathrm{E}[(g_{\tau_{i+1}}(X_{i+1}))^2] =: k^{-1} B, \tag{5.119}$$

since we may assume $\mathrm{E}[g_i(X_i)^2] < \infty$ for $i = 0, \dots, \mathcal{J}$ and

$$F_j = \mathrm{E}_{\mathcal{F}_j}[g_{\tau_{j+1}}(X_{\tau_{j+1}})]. \tag{5.120}$$

For Corollary 86 it suffices to note that

$$F_j = \mathrm{E}_{\mathcal{F}_j}[g_{\tau_{j+1}}(X_{\tau_{j+1}})] \le \max_{i>j}(g_i(X_i)) \le c. \tag{5.121}$$

for some in general suboptimal rule τ, so $\beta = 1$ and $\alpha = \infty$ again.

Proof of Corollary 88 and 90

See proofs of Corollary 89 and Corollary 91 with $\gamma = \beta/2$.

Proof of Corollary 89

By Theorem 75 it is clear that

$$\mathrm{E}\left[\left(Y^{N,K} - Y(M)\right)^2\right] \lesssim N^{-1} v_K + C K^{-2\gamma} \tag{5.122}$$

as $K \to \infty$ and to ensure that $\mathrm{E}\left[\left(Y^{N,K} - Y(M)\right)^2\right] \lesssim \varepsilon^2$, it is obviously optimal to choose K and N as small as possible such that $N^{-1} v_K \le \varepsilon^2/2$ and $K^{-2\gamma} \le \varepsilon^2/2$ are fulfilled. So we have

$$K^*(\varepsilon) = \frac{(2C)^{1/2\gamma}}{\varepsilon^{1/\gamma}}, \qquad N^*(\varepsilon) = \frac{2 v_K}{\varepsilon^2}, \tag{5.123}$$

yielding a complexity of

$$N^*(\varepsilon) \times K^*(\varepsilon) \in O\left(\frac{v_{\left\lceil \frac{(2C)^{1/2\gamma}}{\varepsilon^{1/\gamma}} \right\rceil}}{\varepsilon^{2+1/\gamma}}\right), \tag{5.124}$$

which is optimal since v_K is non-increasing.

Proof of Corollary 91

In case of a surely optimal target martingale M, it holds

$$v_K = \operatorname{Var}\left[\mathcal{Z}(M^K)\right] \leq \mathrm{E}\left[(\mathcal{Z}(M^K) - \mathcal{Z}(M))^2\right] \leq BK^{-\beta}.$$

From Corollary 89 we know that

$$\mathscr{C}(\varepsilon) \in O\left(\frac{v_{\left\lceil \frac{(2C)^{1/2\gamma}}{\varepsilon^{1/\gamma}} \right\rceil}}{\varepsilon^{2+1/\gamma}}\right), \tag{5.125}$$

so with $v_K \leq K^{-\beta}$, we have

$$\mathscr{C}(\varepsilon) \in O\left(\frac{(\frac{1}{\varepsilon^{1/\gamma}})^{-\beta}}{\varepsilon^{2+1/\gamma}}\right) = O\left(\frac{1}{\varepsilon^{2+1/\gamma-\beta/\gamma}}\right) \tag{5.126}$$

since v_K is non-increasing.

Proof of Theorem 94

Let us define $Y_j^{up} = \mathrm{E}[\theta_j^{up}|\mathcal{F}_j]$, where the dependence on M is omitted. Assumption (Conv) allows us to apply Jensen's inequality such that

$$Y_j^{up} \geq F_j\left(\mathrm{E}[\beta_{j+1}\theta_{j+1}^{up}|\mathcal{F}_j]\right) = F_j\left(\mathrm{E}[\beta_{j+1}Y_{j+1}^{up}|\mathcal{F}_j]\right). \tag{5.127}$$

Since it is clear that $Y_{\mathcal{J}} = F_{\mathcal{J}}(0) = Y_{\mathcal{J}}^{up}$, we can show inductively that $Y_j^{up} \geq Y_j$ for all j using (Comp) and in particular $\mathrm{E}\left[\theta_0^{up}\right] \geq Y_0$.

When inserting M^*, equality holds in (5.78). This can be checked inductively with $M_j^* - M_{j-1}^* = \beta_j Y_j - \mathrm{E}\left[\beta_j Y_j \middle| \mathcal{F}_{j-1}\right]$.

Proof of Theorem 96

At first we calculate the differential of V_t according to Itô's formula:

$$dV_t = \frac{\partial V_t}{\partial t}dt + \sum_{d=1}^{D}\frac{\partial V_t}{\partial x^d}dX_t^d + \frac{1}{2}\sum_{d=1}^{D}\sum_{e=1}^{D}\frac{\partial^2 V_t}{\partial x^d \partial x^e}dX_t^d dX_t^e$$

Then, by using the rules of box calculus, see e.g. Steele [68], we obtain

$$dV_t = \left(\frac{\partial V}{\partial t} + \sum_{d=1}^{D}\mu^d\frac{\partial V}{\partial x^d} + \frac{1}{2}\sum_{d=1}^{D}\sum_{e=1}^{D}\frac{\partial^2 V}{\partial x^d \partial x^d}\sigma^d\sigma^e\rho_{de}\right)dt + \sum_{d=1}^{D}\frac{\partial V}{\partial x^d}\sigma^d dW_t^d$$

$$= \widetilde{\mathcal{L}}_{BS} V_t dt + \sum_{d=1}^{D} \frac{\partial V_t}{\partial x^d} \sigma^d dW_t^d,$$

where $\widetilde{\mathcal{L}}_{BS}$ is the discounted Black-Scholes operator defined by

$$\widetilde{\mathcal{L}}_{BS} = \frac{\partial}{\partial t} + \sum_{d=1}^{D} \mu^d \frac{\partial}{\partial x^d} + \frac{1}{2} \sum_{d=1}^{D} \sum_{e=1}^{D} \sigma^d \sigma^e \rho_{de} \frac{\partial^2}{\partial x^d \partial x^e}.$$

Integration provides

$$V_t - V_0 = \int_0^t \widetilde{\mathcal{L}}_{BS} V_t du + \int_0^t \sum_{d=1}^{D} \frac{\partial V_t}{\partial x^d} \sigma^d dW_u^d. \tag{5.128}$$

By comparison with the Doob decomposition and its uniqueness, see Theorem 63, it is clear that

$$M_t^* = \int_0^t \sum_{d=1}^{D} \frac{\partial V_u}{\partial x_u^d} \sigma^d(u, X_u) dW_u^d.$$

Chapter 6

Multilevel for Nested Dual Methods

It is not only possible to use the multilevel technique to reduce the complexity of lower biased estimators as in Chapter 3, but also to reduce the complexity of nested dual methods. Therefore, we consider the standard Monte Carlo estimator $Y^{N,K}$ introduced in Section 5.2 and compare it to its multilevel counterpart $Y^{\mathbf{n},\mathbf{k}}$. The complexity $\mathscr{C}_{ML}(\varepsilon)$ of the latter will then be compared to $\mathscr{C}(\varepsilon)$ from Section 5.2.3. It turns out that if only (AR) is fulfilled with $\beta = 1$, it holds that $\mathscr{C}_{ML}(\varepsilon) \in O(\varepsilon^{-2} \ln^2(\varepsilon))$, regardless of the assumptions (AR'), (AL), and (AQ).

The complexity analysis will take place in the next section. Afterwards, in Section 6.2, we want to analyse the complexity of the multilevel estimator given a fixed number of levels. The results are then tested in case of a "worst-case example", see Section 6.3.

Section 6.4 is exclusively about the complexity of the Andersen Broadie method. That complexity analysis will be more precise and the asymptotic behavior of the computational gain will be analysed more exactly under some mild heuristics. Finally, in Section 6.5 the two estimators have to compete and their efficiency will be judged by some variance criterion. In a numerical example modeling LIBOR rates, it is shown as expected that the multilevel estimator becomes better and better as the available computational budget increases.

Let us recall the standard Monte Carlo estimator

$$Y^{N,K} = \frac{1}{N} \sum_{n=1}^{N} \max_{j=0,\dots,\mathcal{J}} \left(Z_j^{(n)} - M_j^{K,(n)} \right) \tag{6.1}$$

from Section 5.2. Now, fix some natural number $L > 0$. Let $\mathbf{k} = (k_0, \dots, k_L)$ be a sequence of natural numbers satisfying $1 \leq k_0 < k_1 < \ldots < k_L$ and write

$$\begin{aligned} Y(M^{k_L}) &= Y(M^{k_0}) + \sum_{l=1}^{L} [Y(M^{k_l}) - Y(M^{k_{l-1}})] \qquad (6.2) \\ &= \mathrm{E}[\mathcal{Z}(M^{k_0})] + \sum_{l=1}^{L} \mathrm{E}[\mathcal{Z}(M^{k_l}) - \mathcal{Z}(M^{k_{l-1}})]. \end{aligned}$$

For a given sequence $\mathbf{n} = (n_0, \ldots, n_L)$ with $n_0 > \ldots > n_L \geq 1$, we first simulate the initial set of trajectories

$$\left\{ \left(Z_j^{(i)}, M_j^{k_0,(i)} \right), \quad i = 1, ..., n_0, \quad j = 0, \ldots, \mathcal{J} \right\}$$

of the vector process $(Z_\cdot, M_\cdot^{k_0})$ and then for each level $l = 1, \ldots, L$ independently a set of trajectories

$$\left\{ \left(Z_j^{(i)}, M_j^{k_{l-1},(i)}, M_j^{k_l,(i)} \right), \quad i = 1, ..., n_l, \quad j = 0, \ldots, \mathcal{J} \right\}$$

of the vector process $(Z_\cdot, M_\cdot^{k_{l-1}}, M_\cdot^{k_l})$. As in Chapter 3, it is not strictly prescribed how to sample $M^{k_{l-1}}$ and M^{k_l} at the same time. For each application of the multilevel technique, there might be possibilities to sample in a way such that $M^{k_{l-1}}$ and M^{k_l} are particularly correlated to further increase the efficiency. Only the marginal distributions of the two martingales must not be changed. It is clear that both of the martingale realizations will be based on the same trajectory of the payoff, i.e. $Z^{(i)}$. In case of nested methods, it is convenient to reuse the subsamples from M^{k_l} when generating $M^{k_{l-1}}$, which will be done in the following numerical examples. This turns out to be a very worthwile enhancement.

Based on these simulations, we define the multilevel estimator

$$Y^{\mathbf{n},\mathbf{k}} := \frac{1}{n_0} \sum_{i=1}^{n_0} \mathcal{Z}^{(i)}(M^{k_0}) + \sum_{l=1}^{L} \frac{1}{n_l} \sum_{i=1}^{n_l} \left[\mathcal{Z}^{(i)}(M^{k_l}) - \mathcal{Z}^{(i)}(M^{k_{l-1}}) \right], \tag{6.3}$$

where $\mathcal{Z}^{(i)}(M^k) := \max_{j=0,\ldots,\mathcal{J}} \left(Z_j^{(i)} - M_j^{k,(i)} \right)$.

6.1 Complexity Analysis

We obtain the bias of the multilevel algorithm by taking expectations in (6.2) which leads to a telescopic sum, such that

$$\left| \mathrm{E}\left[Y^{\mathbf{n},\mathbf{k}} \right] - Y(M) \right| = \left| \mathrm{E}\left[\mathcal{Z}(M^{k_L}) \right] - \mathrm{E}\left[\mathcal{Z}(M) \right] \right| \leq C k_L^{-\gamma} \tag{6.4}$$

if Theorem 76 is fulfilled or with $\gamma = \beta/2$ in case of Theorem 75. Since the samples used in the different levels are independent and thus uncorrelated, we have for the variance that

$$\mathrm{Var}\left[Y^{\mathbf{n},\mathbf{k}} \right] = n_0^{-1} \mathrm{Var}[\mathcal{Z}(M^{k_0})] + \sum_{l=1}^{L} \frac{1}{n_l} \mathrm{Var}\left[\mathcal{Z}(M^{k_l}) - \mathcal{Z}(M^{k_{l-1}}) \right]. \tag{6.5}$$

We note that for $l > 0$ it holds

$$\begin{aligned} \mathrm{Var}\left[\mathcal{Z}(M^{k_l}) - \mathcal{Z}(M^{k_{l-1}}) \right] &\leq \mathrm{E}\left[\left(\mathcal{Z}(M^{k_l}) - \mathcal{Z}(M^{k_{l-1}}) \right)^2 \right] \\ &\leq 2\,\mathrm{E}\left[\left(\mathcal{Z}(M^{k_l}) - \mathcal{Z}(M) \right)^2 \right] \\ &\quad + 2\,\mathrm{E}\left[\left(\mathcal{Z}(M^{k_{l-1}}) - \mathcal{Z}(M) \right)^2 \right] \\ &\leq 2(Bk_l^{-\beta} + Bk_{l-1}^{-\beta}) \leq 4Bk_{l-1}^{-\beta} \leq \widetilde{B}k_{l-1}^{-\beta}, \end{aligned}$$

by Theorem 75. For notational convenience, we assume that $\widetilde{B}$ is such that also $\mathrm{Var}[\mathcal{Z}(M^{k_0})] \leq \widetilde{B}k_0^{-\beta}$. We now arrive at the following complexity theorem.

Theorem 98. *Suppose that $k_l = k_0\kappa^l$ for some integer $k_0, \kappa > 1$, and $l = 0, \ldots, L$. Assume that the inequalities* (5.29) *hold with $\gamma \geq 1/2$. Fix some $0 < \varepsilon < 1$ and set*

$$L = \left\lceil \frac{-\ln \frac{k_0^\gamma \varepsilon}{C\sqrt{2}}}{\gamma \ln \kappa} \right\rceil . \tag{6.6}$$

Let

$$n_l = \begin{cases} \left\lceil 2\varepsilon^{-2}\widetilde{B}k_0^{-\beta}\kappa^{L(1-\beta)/2}(1-\kappa^{-(1-\beta)/2})^{-1}\kappa^{-l(1+\beta)/2} \right\rceil, & \beta < 1, \\ \left\lceil 2\varepsilon^{-2}\widetilde{B}(L+1)k_0^{-1}\kappa^{-l} \right\rceil, & \beta = 1, \\ \left\lceil 2\varepsilon^{-2}\widetilde{B}k_0^{-\beta}(1-\kappa^{-(\beta-1)/2})^{-1}\kappa^{-l(1+\beta)/2} \right\rceil, & \beta > 1. \end{cases}$$

and the complexity of the estimator $Y^{\mathbf{n},\mathbf{k}}$ is given by

$$\mathscr{C}_{ML}(\varepsilon) := \sum_{l=0}^{L} k_l n_l \in \begin{cases} O(\varepsilon^{-2-(1-\beta)/\gamma}), & \beta < 1, \\ O(\varepsilon^{-2}\ln^2 \varepsilon), & \beta = 1, \\ O(\varepsilon^{-2}), & \beta > 1. \end{cases} \tag{6.7}$$

This result should be compared to the complexity of the standard dual nested method

$$\mathscr{C}(\varepsilon) \in O\left(\frac{1}{\varepsilon^{2+1/\gamma}}\right) \tag{6.8}$$

in case of $v_k \to v_\infty \neq 0$ from Theorem 89. It is eye-catching that (6.7) is independent of γ, while (6.8) is not.

When comparing Theorem 75 and Theorem 76, it is clear that particularly in cases that (AR'), (AQ) or (AL) are not fulfilled the order of complexity can be reduced tremendously by the multilevel technique. If $\beta = 1$ and (AR) only holds with a bias rate of $\gamma = 1/2$, we have $\mathscr{C}_{ML}(\varepsilon) \in O(\varepsilon^{-2}\ln^2 \varepsilon)$ compared to $\mathscr{C}(\varepsilon) \in \varepsilon^{-4}$ of the standard Monte Carlo estimator.

In the better case that $\gamma = 1$ also holds, the gain will still be of order ε^{-1}. This Andersen Broadie case will be examined more precisely in Section 6.4.

6.2 Alternative Adjustment of Levels

In the approach of Giles it is assumed that $k_l = k_0\kappa^l$ for all levels $l = 1, \ldots, L$ and some positive natural number κ. This choice makes the complexity analysis a little easier. However, a priori it is not clear if this choice for $\mathbf{k}$ is optimal. In particular, the order of complexity that can be achieved with the multilevel technique might be higher with another choice. Another disadvantage is that it is difficult to measure the order of the multilevel estimator numerically. As the desired precision ε becomes smaller, more and more levels have to be introduced. This typically leads to a saw tooth function as shown in Section 3.1.1.

In this section, we want to fix the number of Levels L and calculate the optimal rate of complexity that can be achieved with the optimal choice of k_l and n_l depending on ε. We will denote the resulting complexity by $\chi(L)$. It turns out that asymptotically for $L \to \infty$, there will be no improvement compared to Giles' result. However, our result makes it possible to achieve nearly optimal complexity rates with a finite number of levels, which makes implementation and numerical analysis easier.

Let us recall the bias-variance decomposition of a multilevel estimator

$$\underbrace{k_L^{-2\gamma}}_{\text{squared bias}} + \underbrace{\frac{1}{n_0} + \sum_{l=1}^{L} \frac{k_{l-1}^{-\beta}}{n_l}}_{\text{variance}} \lesssim \varepsilon^2 \tag{6.9}$$

to ensure a root-mean-squared error less than ε. Suppose the complexity of the multilevel estimator to fulfill

$$\sum_{l=0}^{L} k_l n_l \lesssim \varepsilon^{-\chi(L)}, \tag{6.10}$$

asymptotically for $L \to \infty$, i.e. we already assume the existence of an order of complexity given by a number $\chi(L) > 0$. The task is now to find maximal n_l and k_l in the sense that (6.10) is fulfilled with minimal $\chi(L)$ subject to (6.9). First of all, we know from (6.9) that $k_L = \varepsilon^{-1/\gamma}$ must hold to ensure that the bias is decreasing quickly enough as $\varepsilon \to 0$. In order to fulfill (6.10) we can now decide that the largest choice of n_L allowed is given by

$$n_L = \varepsilon^{-\chi}/k_L = \varepsilon^{-\chi+1/\gamma}.$$

This result can in turn be inserted into (6.9) and we know about the optimal choice for k_{L-1} to ensure that each of the summands in (6.9) is asymptotically smaller than ε^2. Thus, it is possible to use (6.9) and (6.10) alternately which leads to the following result:

$$\begin{aligned}
&k_L = \varepsilon^{-1/\gamma}\\
&\quad \Rightarrow n_L = \varepsilon^{1/\gamma-\chi}\\
&k_{L-1} = \left(\varepsilon^{2+1/\gamma-\chi}\right)^{-1/\beta} = \varepsilon^{-2/\beta-\frac{1}{\beta\gamma}+\chi/\beta}\\
&\quad \Rightarrow n_{L-1} = \varepsilon^{-\chi}/\varepsilon^{-2/\beta-\frac{1}{\beta\gamma}+\chi/\beta} = \varepsilon^{-\chi+2/\beta+\frac{1}{\beta\gamma}-\chi/\beta}\\
&k_{L-2} = \left(\varepsilon^{2-\chi+2/\beta+\frac{1}{\gamma\beta}-\chi/\beta}\right)^{-1/\beta} = \varepsilon^{-2/\beta+\chi/\beta-2/\beta^2-\frac{1}{\gamma\beta^2}+\chi/\beta^2}\\
&\quad \Rightarrow n_{L-2} = \varepsilon^{-\chi+\frac{1}{\gamma\beta^2}+2/\beta+2/\beta^2-\chi/\beta-\chi/\beta^2}\\
&k_{L-3} = \varepsilon^{-2/\beta+\chi/\beta-\frac{1}{\gamma\beta^3}-2/\beta^2-2/\beta^3+\chi/\beta^2+\chi/\beta^3}\\
&\quad \Rightarrow n_{L-3} = \varepsilon^{-\chi+2/\beta-\chi/\beta+\frac{1}{\gamma\beta^3}+2/\beta^2+2/\beta^3-\chi\beta^2-\chi/\beta^3}
\end{aligned}$$

It is quite obvious that this iterative procedure yields the formulas

$$\log(k_{L-i}) = -\frac{1}{\gamma\beta^i} + (\chi-2)\sum_{k=1}^{i} \frac{1}{\beta^k} \tag{6.11}$$

$$\Rightarrow \log(n_{L-i}) = -\chi + \frac{1}{\gamma\beta^i} + (2-\chi)\sum_{k=1}^{i} \frac{1}{\beta^k} \tag{6.12}$$

In order to further analyze these quite general expressions for n_l and k_l, we have to distinguish two different cases.

1. In case of $\beta = 1$ we have

   $$\log(k_{L-i}) = -\tfrac{1}{\gamma} + i(\chi - 2) \quad \log(n_{L-i}) = -\chi + \frac{1}{\gamma} + i(2 - \chi).$$

   Now, from $\frac{1}{n_0} = \varepsilon^2$ we infer that

   $$\begin{aligned} -2 &= -\chi + 1/\gamma + i(2 - \chi) \\ \Leftrightarrow \chi &= \frac{2 + 1/\gamma + 2L}{L + 1} \end{aligned}$$

   must be the order of complexity. We have that $\chi \to 2$ as $L \to 0$, which is in agreement with Theorem 98.

2. In case of $\beta \neq 1$ we have a geometric series and so

   $$\log(k_{L-i}) = -\tfrac{1}{\gamma\beta^i} + (\chi - 2)\tfrac{\beta^i - 1}{\beta^{i+1} - \beta^i} \quad \log(n_{L-i}) = -\chi + \frac{1}{\gamma\beta^i} + (2 - \chi)\frac{\beta^i - 1}{\beta^{i+1} - \beta^i}$$

   and from $\frac{1}{n_0} = \varepsilon^2$ we infer that now

   $$\begin{aligned} -2 &= -\chi + \frac{1}{\gamma\beta^L} + (2 - \chi)\frac{\beta^L - 1}{\beta^{L+1} - \beta^L} \\ \Leftrightarrow \chi &= \frac{2\gamma\beta^{L+1} + \beta - 2\gamma - 1}{\gamma(\beta^{L+1} - 1)}. \end{aligned}$$

   It thus holds that for $\beta < 1$

   $$\chi \to 2 + (1 - \beta)/\gamma, \quad \text{as} \quad L \to 0 \tag{6.13}$$

   and for $\beta > 1$

   $$\chi \to 2, \quad \text{as} \quad L \to 0, \tag{6.14}$$

   which is also in agreement with Theorem 98.

When inserting the optimal χ, we can also reformulate n_l and k_l independently and arrive at the following theorem.

Theorem 99. *When using a fixed number of levels L, the complexity of the multilevel estimator $\mathscr{C}_L(\varepsilon)$ as in* (6.9) *and* (6.10) *and the optimal choice of n_l and k_l are given in the following:*

- *In case of $\beta = 1$:*

  $$\mathscr{C}_L(\varepsilon) = \varepsilon^{-\frac{2+1/\gamma+2L}{L+1}}$$

  $$k_l(\varepsilon) = \varepsilon^{-\frac{l+1}{\gamma(L+1)}}, \quad n_l(\varepsilon) = \varepsilon^{\frac{l}{\gamma(L+1)} - 2}$$

- *In case of $\beta \neq 1$:*

  $$\mathscr{C}_L(\varepsilon) = \varepsilon^{-\frac{2\gamma\beta^{L+1}+\beta-2\gamma-1}{\gamma(\beta^{L+1}-1)}}$$

  $$k_l(\varepsilon) = \varepsilon^{-\frac{\beta^{1+l}-1}{\gamma(\beta^{L+1}-1)}}, \quad n_l(\varepsilon) = \varepsilon^{\frac{(2-2\beta^{L+1})\gamma+\beta^{l+1}-\beta}{\gamma(\beta^{L+1}-1)}}$$

As the results from Theorem 99 converge to the result of Giles [1], it is now clear that it is no hard restriction to assume $k_l = k_0\kappa^l$.

When inserting γ and β for dual nested methods, we have at first the good natured case that (AL), (AR) and (AR') are fulfilled. Following Corollary 89, it may happen that $\gamma = 1$ and $\beta = 1$ and so

$$\mathscr{C}_L(\varepsilon) = \varepsilon^{-\frac{2L+3}{L+1}}$$

$$k_l(\varepsilon) = \varepsilon^{-\frac{l+1}{L+1}}, \quad n_l(\varepsilon) = \varepsilon^{\frac{l}{L+1}-2},$$

which implies $\mathscr{C}_L(\varepsilon) \to 2$, as $l \to \infty$. Those numbers are shown in Table 6.1.

L	k_0	k_1	k_2	k_3	n_0	n_1	n_2	n_3	$\chi(L)$
0	-1				-2				3
1	-1/2	-1			-2	-3/2			2.5
2	-1/3	-2/3	-1		-2	-5/3	-4/3		$\frac{7}{3} = 2.\bar{6}$
3	-1/4	-2/4	-3/4	-1	-2	-7/4	-6/4	-5/4	$\frac{9}{4} = 2.5$
4	...	...	...	...	...	...	...	...	$\frac{11}{5}$

Table 6.1: The optimal adjustment of the multilevel estimator given in powers of ε.

Secondly, in the worst case of a dual nested method that those assumptions are not fulfilled, we only have $\gamma = 1/2$ and $\beta = 1$, so

$$\mathscr{C}_L(\varepsilon) = \varepsilon^{-\frac{4+2L}{L+1}}$$

$$k_l(\varepsilon) = \varepsilon^{-\frac{2l+2}{L+1}}, \quad n_l(\varepsilon) = \varepsilon^{\frac{2l}{L+1}-2},$$

which is used in Section 6.3. The theorem is also used for the table about lower bounds in Section 3.2.3 with $\gamma = 1$ and $\beta = 1/2$.

6.3 Single-Period Example

To numerically illustrate the results of the previous section, let us consider a "worst case scenario", i.e. let us assume that only Theorem 75 is fulfilled and 76 is not. We take again the single-period Example 78 with $b = 10$ and recall from Remark 79 that (AL) and (AQ) are violated, so $\beta = 1$, $\gamma = 1/2$ and we cannot expect an order of ε^{-3} for the standard MC, but ε^{-4}.

Since measuring the order of complexity when using Theorem 45 will be difficult, we use a fixed number of levels L as in Section 6.2 and we choose some suitable constants for n_l and k_l by experience.

$$\mathscr{C}_L(\varepsilon) = \varepsilon^{-\frac{4+2L}{L+1}}$$

[1] Note that our approach (6.9) includes the result of Giles [39], as $k_{l-1} = k_l/\kappa$ in his setting.

$$\begin{aligned} k_L &= 100 \times \varepsilon^{-2}, \qquad k_l(\varepsilon) = 10 \times \varepsilon^{-\frac{2l+2}{L+1}}, \;\; l = 0, \dots, L-1 \qquad (6.15)\\ n_0 &= 1000 \times \varepsilon^{-2}, \qquad n_l(\varepsilon) = \varepsilon^{\frac{2l}{L+1}-2}, \;\; l = 1, \dots, L \end{aligned}$$

L	k_0	k_1	k_2	k_3	n_0	n_1	n_2	n_3	$-\log(\mathscr{C}_L(\varepsilon))$
0	-2				-2				4
1	-1	-2			-2	-1			3
2	-2/3	-4/3	-2		-2	-4/3	-2/3		$\frac{8}{3} = 2.\bar{6}$
3	-1/2	-1	-3/2	-2	-2	-3/2	-1	-1/2	$\frac{10}{4} = 2.5$
4	...	...	...	...	...	...	...	...	2.4

Table 6.2: Level adjustment for $\beta = 1$ and $\gamma = 1/2$ given in powers of ε.

Table 6.2 shows which orders to expect. The results are presented in Figure 6.1. First of all, the two log-log plots on the top measure the order of the bias depending on the number of subsimulations k and the order of variance of the levels. The latter is measured for some levels $k_l = 10 \times 2^l$, $l = 0, \dots, 5$. The red regression lines indicate an order of $\varepsilon^{-0.53}$ and $\varepsilon^{-1.007}$ respectively. Furthermore, we measure via very exact calculations that $Y[M] = 1.255$ and $v_\infty = 4.65$. Thus, we can infer that given the desired complexity *comp*, the standard Monte Carlo is optimal with $K = \sqrt{comp/v_\infty} \times 0.74$ and $N = comp/K$ following Corollary 88.

On the bottom, there are two plots comparing the standard Monte Carlo algorithm indicated by the dashed line to the multilevel Monte Carlo with different numbers of levels L. The latter is indicated by the coloured lines which are blue except for $L = 2$. For this case of $L = 2$ as well as for the standard Monte Carlo, we test the algorithm three hundred times in order to check the root-mean-squared error. The results are indicated as points in the lower right plot. It is visible that they prove the order of complexity as calculated before. For a complexity of greater than 20 million, the multilevel with $L = 2$ is better than the standard MC. This corresponds to approximately 1 minute on a modern computer.

The slopes of the blue lines are very accurately conform to the values in Table 6.2. In the log-log plot, we can see that their orders are converging to ε^{-2}.

6.4 Complexity Analysis of AB Method

The complexity analysis in Section 6.1 is not only asymptotic in the sense that it analyses the complexity for $L \to \infty$, but it is also quite general as it considers arbitrary $\gamma > \beta/2$ and $\beta > 0$. As explained in the previous section, Theorem 89 tells us that under (AR'), (AL) and (AQ) the Andersen Broadie method may lead to the good natured case of $\beta = 1$ and $\gamma = 1$. Under these circumstances, a better complexity analysis is possible.

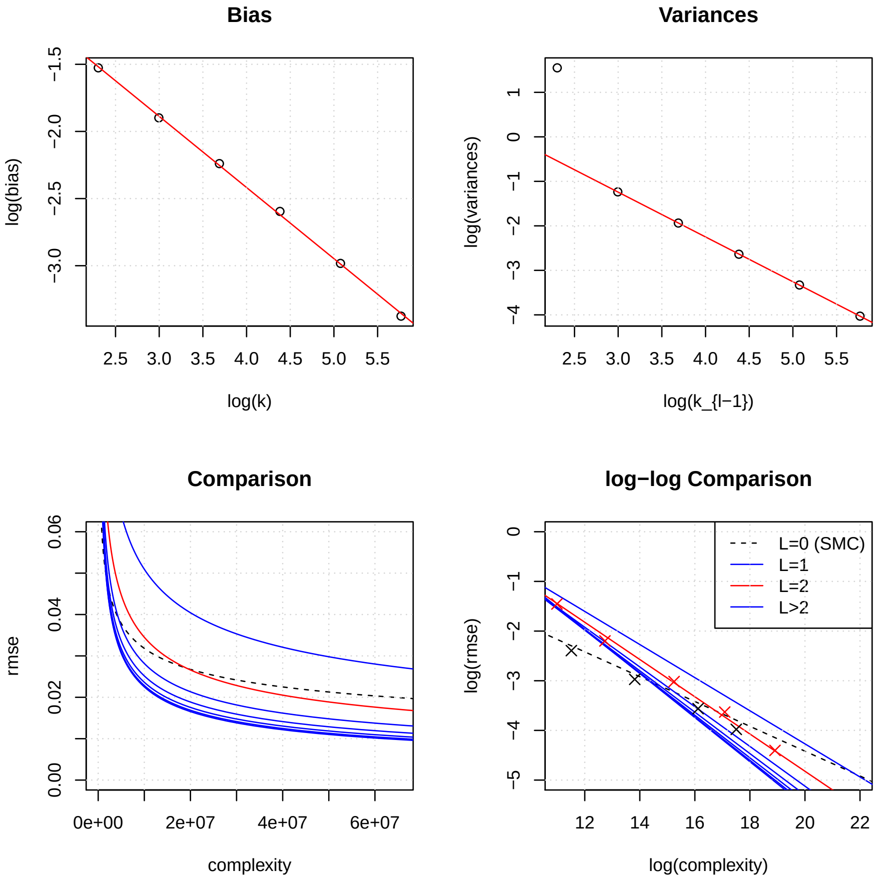

Figure 6.1: Results with parameters according to (6.15).

We assume $\mathscr{C}_{AB}(\varepsilon) = \varepsilon^{-3}$ (omitting δ) because of Remark 77. Furthermore, Theorem 98 provides the result $\mathscr{C}_{ML}(\varepsilon) = \varepsilon^{-2}\ln^2\varepsilon$, so one hopes to have a reduction of complexity by a factor of

$$\mathscr{R}(\varepsilon) := \frac{\mathscr{C}_{AB}(\varepsilon)}{\mathscr{C}_{ML}(\varepsilon)} = \varepsilon^{-1}. \tag{6.16}$$

Additionally, we are not only interested in the order, but also the quantity of the complexity improvement. Therefore, we want to optimize the asymptotic behavior of the the quantities k_l, n_l and the resulting complexity under some mild heuristics, which in general are indeed suboptimal.

Since the role of n_0 in (6.3) it totally different from $n_1, \ldots, n_L$, it will be fruitful to optimize n_0 separately. As a heuristic model, the relation of $n_1, \ldots, n_l$ among each other is assumed to be inverse to k_l, so the approach reads

$$k_l = k_0\kappa^l \quad \text{for} \quad 0 \le l \le L, \quad \text{and} \quad n_l = n_1\kappa^{1-l} \quad \text{for} \quad 1 \le l \le L. \tag{6.17}$$

It is now clear that in our setting, the bias $\mathrm{E}\left[Y^{\mathbf{n},\mathbf{k}}\right] - Y(M)$ is asymptotically converging to $\frac{\mu_\infty}{k_0\kappa^L}$, which is the definition of some constant $\mu_\infty > 0$.

Our model leads to the simple advantage that the complexity is the same in all levels, since $n_l k_l = k_0 n_1 \kappa$ for all $l \geq 1$.

$$\begin{aligned} \mathscr{C}_{ML} &= n_0 k_0 + \sum_{l=1}^{L} n_l k_l \\ &= n_0 k_0 + n_1 k_0 \kappa L. \end{aligned} \tag{6.18}$$

Our model also allows us to find an easy expression for the variance:

$$\begin{aligned} \operatorname{Var}[Y^{\mathbf{n},\mathbf{k}}] &= n_0^{-1} \operatorname{Var}[\mathcal{Z}(M^{k_0})] + \sum_{l=1}^{L} n_l^{-1} \operatorname{Var}\left[\mathcal{Z}(M^{k_l}) - \mathcal{Z}(M^{k_{l-1}})\right] \\ &= n_0^{-1}\sigma_\infty^2 + \sum_{l=1}^{L} n_l^{-1} k_l^{-1} \mathfrak{V}_\infty \\ &= n_0^{-1}\sigma_\infty^2 + n_1^{-1} k_0^{-1} \mathfrak{V}_\infty \kappa^{-1} L, \end{aligned}$$

where σ_∞^2 and $\mathfrak{V}_\infty$ are constants greater than zero. From the variance-bias decomposition we know that in order to achieve a mean-squared error smaller than ε^2, first of all the bias has to be smaller than ε. Thus, from $\mu_\infty k_0^{-1} \kappa^{-L} < \varepsilon$ it follows that

$$L > \frac{\ln \varepsilon^{-1} + \ln(\mu_\infty / k_0)}{\ln \kappa}. \tag{6.19}$$

The second part that is necessary to achieve such a mean-squared error less than ε^2 is to achieve a variance less than $\varepsilon^2 - \mu_\infty^2 k_0^{-2} \kappa^{-2L}$. To achieve this with minimal effort, we optimize the number of trajectories as in proof of Theorem 45.

$$\begin{aligned} n_0^* &= \frac{\sigma_\infty^2 + \sigma_\infty L \sqrt{k_0^{-1} \mathfrak{V}_\infty}}{\varepsilon^2 - \mu_\infty^2 k_0^{-2} \kappa^{-2L}}, \\ n_1^* &= \frac{\sigma_\infty \kappa^{-1} \sqrt{k_0^{-1} \mathfrak{V}_\infty} + \kappa^{-1} L k_0^{-1} \mathfrak{V}_\infty}{\varepsilon^2 - \mu_\infty^2 k_0^{-2} \kappa^{-2L}} \\ &= n_0^* \kappa^{-1} \sigma_\infty^{-1} \sqrt{k_0^{-1} \mathfrak{V}_\infty}, \end{aligned}$$

This choice leads to a complexity of

$$\begin{aligned} \mathscr{C}_{ML}^*(k_0, L, \varepsilon) &= k_0\, n_0^*(k_0, L, \varepsilon) + k_0 \kappa L\, n_1^*(k_0, L, \varepsilon) \\ &= \frac{\left(L\sqrt{\mathfrak{V}_\infty} + \sigma_\infty \sqrt{k_0}\right)^2}{\varepsilon^2 - \mu_\infty^2 k_0^{-2} \kappa^{-2L}}, \end{aligned} \tag{6.20}$$

which is optimal considering the number of trajectories, but the number of levels is not necessarily optimal yet. To optimize L, we differentiate (6.20) as a function of L and obtain:

$$\kappa^{2L} \varepsilon^2 = \underbrace{\sigma_\infty \mu_\infty^2 \mathfrak{V}_\infty^{-1/2} k_0^{-3/2} \ln \kappa + \mu_\infty^2 k_0^{-2}}_{p} + L \underbrace{\mu_\infty^2 k_0^{-2} \ln \kappa}_{q} \tag{6.21}$$

As expected, $L \to \infty$ as $\varepsilon \downarrow 0$. Taking logarithms on both sides of (6.21) leads to

$$\begin{aligned} 2L\ln(\kappa) + 2\ln\varepsilon &= \ln(p+qL) \\ \Rightarrow L &= \frac{\ln\varepsilon^{-1}}{\ln\kappa} + \frac{1}{2\ln\kappa}\ln(p+qL) \qquad (6.22) \\ &= \frac{\ln\varepsilon^{-1}}{\ln\kappa} + \frac{\ln(qL)}{2\ln\kappa} + O(L^{-1}), \quad \varepsilon \downarrow 0, \end{aligned}$$

since $\ln(a+b) = \ln(b) + \ln(a/b+1)$ and the logarithmus naturalis has slope 1 at $x = 1$. Further we have

$$L = \frac{\frac{\ln\varepsilon^{-1}}{\ln\kappa} + O(L^{-1})}{1 - \frac{\ln(qL)}{2L\ln\kappa}},$$

where $O(L^{-1})$ and $\frac{\ln(qL)}{2L\ln\kappa}$ tend to zero. Hence we have $L = O(\ln\varepsilon^{-1})$, as $\varepsilon \downarrow 0$. This implies by (6.22)

$$L = \frac{\ln\varepsilon^{-1}}{\ln\kappa} + O\left(\ln\ln\varepsilon^{-1}\right) \qquad (6.23)$$

and by iterating (6.22) once again we obtain

$$\begin{aligned} L &= \frac{\ln\varepsilon^{-1}}{\ln\kappa} + \frac{\ln(qL)}{2\ln\kappa} + O(L^{-1}) \\ &\underset{(6.23)}{=} \frac{\ln\varepsilon^{-1}}{\ln\kappa} + \frac{\ln(q)}{2\ln\kappa} + \frac{\ln\left(\frac{\ln(\varepsilon^{-1})}{\ln\kappa} + O(\ln\ln\varepsilon^{-1})\right)}{2\ln\kappa} + O(L^{-1}). \end{aligned}$$

By inserting $q = \mu_\infty^2 k_0^{-2}\ln\kappa$ this becomes

$$L = \frac{\ln\varepsilon^{-1}}{\ln\kappa} + \frac{\ln(\mu_\infty/k_0\sqrt{\ln\kappa})}{\ln\kappa} + \frac{\ln\ln\varepsilon^{-1} - \ln\ln\kappa}{2\ln\kappa} + O\left(\frac{\ln\ln\varepsilon^{-1}}{\ln\varepsilon^{-1}}\right) + O(L^{-1}),$$

where we used again the logarithm rule as above and the terms containing $\ln\ln\kappa$ cancel out each other. It holds $O\left(L^{-1}\right) \in O\left(\frac{\ln\ln\varepsilon^{-1}}{\ln\varepsilon^{-1}}\right)$ and we finally define

$$L^*(k_0,\varepsilon) := \frac{\ln\varepsilon^{-1}}{\ln\kappa} + \frac{\ln\ln\varepsilon^{-1}}{2\ln\kappa} + \frac{\ln\left(\mu_\infty/k_0\right)}{\ln\kappa} + O\left(\frac{\ln\ln\varepsilon^{-1}}{\ln\varepsilon^{-1}}\right), \qquad \varepsilon \downarrow 0. \quad (6.24)$$

To check whether relation (6.19) is fulfilled, note that $\frac{\ln\ln\varepsilon^{-1}}{\ln\varepsilon^{-1}} \to 0$ as $\varepsilon \to 0$. Thus, $L^* \to \infty$ and (6.19) will not lead to problems for ε small enough. Inserting the optimal number of levels from (6.24) yields

$$\begin{aligned} \mathcal{C}^*_{ML}(k_0,L,\varepsilon) =& \mathfrak{V}_\infty\varepsilon^{-2}\left(\frac{\ln\varepsilon^{-1}}{\ln\kappa} + \frac{\ln\ln\varepsilon^{-1}}{2\ln\kappa} + \frac{\ln\left(\mu_\infty/k_0\right)}{\ln\kappa} + \frac{\sigma_\infty\sqrt{k_0}}{\sqrt{\mathfrak{V}_\infty}}\right)^2 \\ &\times\left(1 + O\left(\frac{\ln\ln\varepsilon^{-1}}{\ln\varepsilon^{-1}}\right)\right). \qquad (6.25) \end{aligned}$$

The final step to find the minimal complexity under the heuristics (6.17) is the choice of k_0. The expression $\frac{\ln(\mu_\infty/k_0)}{\ln\kappa} + \frac{\sigma_\infty\sqrt{k_0}}{\sqrt{\mathfrak{V}_\infty}}$, which is the part of (6.25) containing k_0 becomes minimal at

$$k_0^* = \frac{4\mathfrak{V}_\infty}{\sigma_\infty^2\ln^2\kappa}.$$

Of course, this is not the global minimum, since equation (6.25) includes terms of higher order that are not respected in this consideration. Anyway, since we are interested in the coefficients of the asymptotic complexity, we collect the term of highest order in (6.25) and obtain

$$\mathscr{C}^*_{ML}(k_0, L, \varepsilon) \asymp \mathfrak{V}_\infty \varepsilon^{-2} \left(\frac{\ln \varepsilon^{-1}}{\ln \kappa} \right)^2 .$$

From the proof of Corollary 89, we know that

$$\mathscr{C}^*_{AB}(\varepsilon) = \frac{2\sqrt{2}}{\varepsilon^3} \mu_\infty \sigma^2_\infty,$$

For the ratio $\mathscr{R}(\varepsilon) := \frac{\mathscr{C}^*_{AB}(\varepsilon)}{\mathscr{C}^*_{ML}(\varepsilon)}$ it thus holds that

$$\begin{aligned}\mathscr{R}(\varepsilon) &\asymp \left(\frac{2\sqrt{2}}{\varepsilon^3} \mu_\infty \sigma^2_\infty \right) / \left(\mathfrak{V}_\infty \varepsilon^{-2} \left(\frac{\ln \varepsilon^{-1}}{\ln \kappa} \right)^2 \right) \\ &= \frac{2\sqrt{2} \ln^2 \kappa}{\varepsilon \ln^2 \varepsilon^{-1}} \mu_\infty \sigma^2_\infty / \mathfrak{V}_\infty,\end{aligned}$$

which is sound with the expectation (6.16).

6.5 Interest Rate Example

In the previous section, the standard Andersen Broadie estimator $Y^{N,K}$ was compared to its multilevel counterpart $Y^{\mathbf{n},\mathbf{k}}$ by considering the minimal complexity necessary to realize the accuracy ε. One could ask the inverse question and look for the maximal accuracy ε^* under a given complexity C. We would expect the ratio of accuracies to behave according to

$$\frac{\varepsilon^*_{ML}(C)}{\varepsilon^*_{AB}(C)} \in O(C^{-1/6}), \tag{6.26}$$

where the logarithm and δ were again omitted. However, this comparison is not appropriate for a numerical example, since the complexities are calculated conditional that the number of trajectories and subsimulations are chosen optimally and estimation of the order is very sensitive, especially for a complex numerical example like the following.

Therefore, the two estimators will be compared following the same criterion as in Section 3.2.3: The complexity is fixed and the number of inner paths must ensure the same bias for both methods in order to compare their variances under equal circumstances. Thereby, we continue the previous section and assume

$$k_l = k_0 \kappa^l \quad \text{for} \quad 0 \le l \le L, \quad \text{and} \quad n_l = n_1 \kappa^{1-l} \quad \text{for} \quad 1 \le l \le L. \tag{6.27}$$

Additionally, we now demand

$$NK = n_0 k_0 + n_1 k_0 \kappa L \tag{6.28}$$

to fix the same complexity for both methods and

$$k_L = k_0 \kappa^L = K, \tag{6.29}$$

which determines the bias. So we have

$$k_0 = K\kappa^{-L}, \qquad n_1 = NL^{-1}\kappa^{L-1} - n_0\kappa^{-1}L^{-1}.$$

The variances of $Y^{\mathbf{k},\mathbf{n}}$ and $Y^{K,N}$ are now

$$\operatorname{Var}\left[Y^{K,N}\right] = \frac{v(0,K)}{N}, \quad \operatorname{Var}\left[Y^{\mathbf{k},\mathbf{n}}\right] = \frac{v(0,k_0)}{n_0} + \sum_{l=1}^{L} n_l^{-1} v(k_{l-1},k_l), \tag{6.30}$$

where

$$v(k_{l-1},k_l) := \operatorname{Var}\left[\mathcal{Z}^{(r)}(M^{k_l}) - \mathcal{Z}^{(r)}(M^{k_{l-1}})\right]$$

denotes the variance wthin each level and $\mathfrak{V}_\infty$ and σ_∞ as before. So we have the total variance

$$\operatorname{Var}\left[Y^{\mathbf{k},\mathbf{n}}\right] = \frac{\sigma_\infty^2}{n_0} + \sum_{l=1}^{L} n_1^{-1}\kappa^{l-1}\frac{\mathfrak{V}_\infty}{k_0\kappa^l} = \frac{\sigma_\infty^2}{n_0} + \frac{k_0^{-1}\mathfrak{V}_\infty L^2}{N\kappa^L - n_0},$$

that becomes minimal for

$$\begin{aligned} n_0^* &= \frac{N\kappa^L}{1 + L\sqrt{\frac{\mathfrak{V}_\infty \kappa^L}{\sigma_\infty^2 K}}} = \frac{N\kappa^L}{1 + L\sqrt{v(k_{L-1},K)\kappa^L/\sigma_\infty^2}} \\ n_1^* &= n_0^*\sqrt{\frac{v(k_{L-1},K)\kappa^{L-2}}{\sigma_\infty^2}} \end{aligned} \tag{6.31}$$

which, if inserted yields

$$\operatorname{Var}\left[Y^{\mathbf{k},\mathbf{n}^*}\right] = \frac{1}{N\kappa^L}\left(\sigma_\infty + L\sqrt{v(k_{L-1},K)\kappa^L}\right)^2 =: \frac{\Theta_{K,L}^2}{N}. \tag{6.32}$$

So we have for the ratios of the two variances

$$\mathfrak{R}(K,L) := \frac{\operatorname{Var}\left[Y^{\mathbf{k},\mathbf{n}^*}\right]}{\operatorname{Var}\left[Y^{K,N}\right]} = \left(\kappa^{-L/2} + \frac{Lv(k_{L-1},K)}{\sigma_\infty}\right)^2, \tag{6.33}$$

which is only depending on K and L now.

Suppose the Andersen Broadie algorithm is run with optimal K and N, such that the root-mean-squared error ε is minimized given a complexity C. We then have an estimator $Y^{K^*(C),N^*(C)}$. Since any choice of K will be the optimal choice for some number of trajectories $N(K)$, analysing the convergence

$$\mathfrak{R}(K,L) \to 0 \tag{6.34}$$

is indeed sufficient to show that

$$\frac{\varepsilon_{ML}^*}{\varepsilon_{AB}^*} \leq \frac{\varepsilon_{ML}}{\varepsilon_{AB}^*} \to 0.$$

This is possible, as (6.33) is independent of N and C.

The following example is about a Bermudan swaption. It is considered in Sec. 7 of Kolodko and Schoenmakers [54] in the context of the well known LIBOR Market Model. Since the complexity analysis was done under the assumption that $v_k \to v_\infty \neq 0$, it is natural to choose an example where the optimal martingale is unknown and very difficult to approximate. Benchmark Example 48 is therefore not appropriate here. Because of the very complex structure of the interest rates, it is not easy to find a good stopping rule. We will use the simple rule (6.38) that is implied by the maximum of all European values. This corresponds to a policy iteration, see Remark 35 with window parameter $\kappa = \infty$ and $\tau_j^0 \equiv j$ for all $j = 0, \ldots, \mathcal{J}$. The first iterated stopping rule τ^1 is then equal to (6.38). We will thus have a target martingale that is clearly suboptimal, which is necessary for the multilevel technique to be advantageous, compare Corollary 90.

The upper bounds obtained in the following are definitely higher than the true value and the variance is clearly non-zero. Thus, it should be possible to reach a complexity, such that the multilevel algorithm becomes faster than the standard MC algorithm. This will indeed be visible.

Benchmark Example 100. *The dynamics of the LIBOR Market Model with respect to a tenor structure $0 < T_1 < \ldots < T_n$ in the spot LIBOR measure P^*. are given by the following system of SDE's*

$$dL_i = \sum_{j=\kappa(t)}^{i} \frac{\delta_j L_i L_j \gamma_i \cdot \gamma_j}{1 + \delta_j L_j}\, dt + L_i\, \gamma_i \cdot dW^* \qquad 0 \leq t \leq T_i, \qquad 1 \leq i < n, \tag{6.35}$$

with $\delta_i := T_{i+1} - T_i$, $t \to \gamma_i(t) = (\gamma_{i,1}(t), \ldots, \gamma_{i,d}(t))$ being deterministic factor loadings, and $\kappa(t) := \min\{m : T_m \geq t\}$ being the next LIBOR fixing date after t. In (6.35), $(W^*(t) \mid 0 \leq t \leq T_{n-1})$ is a d-dimensional standard Brownian motion under the measure P^* induced by the numeraire

$$B_*(t) := \frac{B_{\kappa(t)}(t)}{B_1(0)} \prod_{i=1}^{\kappa(t)-1} (1 + \delta_i L_i(T_i))$$

with $B_i(t)$, $t \leq T_i$, being zero coupon bonds with face value \$1 at their respective maturities T_i, $1 \leq i \leq n$.

A Bermudan swaption issued at $t = 0$ gives the holder the right to exercise once a cash-flow

$$S(T_i) := \left(\sum_{j=i}^{n-1} B_{j+1}(T_i) \delta_j \left(L_j(T_i) - \theta \right) \right)^+,$$

that is the positive part of the value of a swap contract with settlement dates $T_{i+1}, ..., T_n$ and strike θ, at an exercise date out of the set $\{\mathcal{T}_1, \ldots, \mathcal{T}_{\mathcal{J}}\} \subset \{T_1, \ldots, T_n\}$ specified in the option contract. The discounted cashflow process reads

$$Z_j := S_{\mathcal{T}_j} / B_*(\mathcal{T}_j), \qquad j = 1, ..., \mathcal{J}.$$

The problem data is also the same as in Kolodko and Schoenmakers [54]. There are forty-one tenor dates, so $n = 41$ which implies that the dimension of the stochastic process is also 41. The maturity time is $T_n = 10.25$, so $\delta_i = 0.25$, for all $i = 1, \ldots, n$. The exercise dates are equally distributed, i.e. $\mathcal{T}_i = T_{4i}$, $i = 1, \ldots, 10$. The initial value is 10% all over the tenor structure and the strike price is also $\theta = 10\%$, so this is an at-the-money example. The underlying process modeled by the differential equation (6.35) is simulated via a log-Euler scheme using a discretization with fineness $\Delta t = \delta/5$. The LIBOR volatility structure is determined by

$$\gamma_i(t) = cg(T_i - t)e_i, \quad g(s) = g_\infty + (1 - g_\infty + as)e^{-bs} \tag{6.36}$$

with $c = 0.2,\ a = 1.5,\ b = 3.5,\ g_\infty = 0.5$. Here, the vectors e_i are unit vectors, such that

$$\rho_{ij} := e_i^\top e_j = \exp(-\varphi|i-j|), \tag{6.37}$$

for all $i, j = 1, \ldots, n-1$ and $\varphi = 0.0413$. In other words, a Cholesky-decomposition is necessary to obtain the coefficients of the stochastic differential equation.

In the simulation, we use martingales from stopping rules, see Section 5.2.2, with the stopping time

$$\tau_i = \inf\left\{ j : i \leq j \leq \mathcal{J},\ \max_{p:\, j \leq p \leq \mathcal{J}} \mathrm{E}^j Z_p \leq Z_j \right\}, \quad i = 0, \ldots, \mathcal{J}. \tag{6.38}$$

Here, $\mathrm{E}^j Z_p$ is the discounted price of the corresponding European option. They are computed via the formula in [65] that has "accuracy better than than 0.3% relative for this example" according to Schoenmakers. The exact numbers are given in the next two tables. Therein, we use the abbreviations

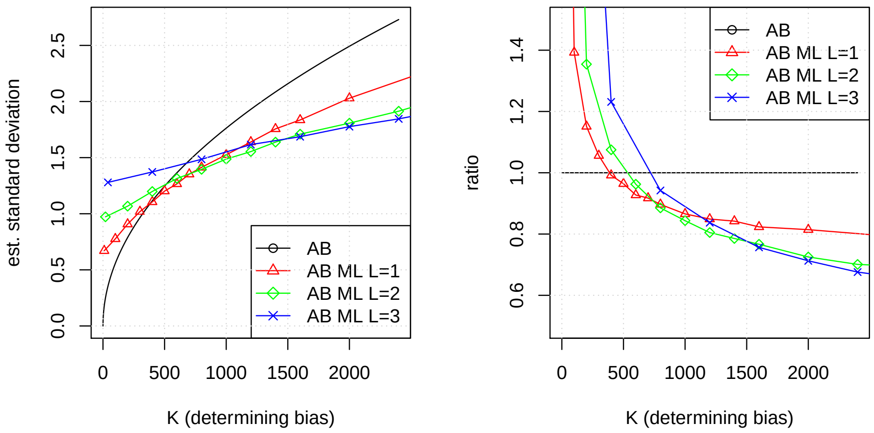

Figure 6.2: The plot on the right hand side shows the variance ratio function $\mathcal{R}(K, L)$ for $L = 1, 2, 3$ from the numerical experiment. It is based on the results for ε_{ML} and ε_{AB} given in the left hand plot. It is clearly visible that the introduction of higher levels becomes fruitful as the demanded bias is getting smaller.

$$\xi_{0,k_0}(r) = \mathcal{Z}^{(r)}(M^{k_0}), \quad \xi_{k_{l-1},k_l}(r) = \mathcal{Z}^{(r)}(M^{k_l}) - \mathcal{Z}^{(r)}(M^{k_{l-1}}). \tag{6.39}$$

In the first step, we only estimate the variances of the levels. We use $n_0 = 10000$, trajectories, $k_0 = 50$, subsamples in level 0, $\kappa = 2$, $n_l = n_0\kappa^{-l}, k_l = k_0\kappa^l, l = 0, ..., L$, and compute for $L = 3$ levels the following quantities

$$\widehat{v}(k_{l-1}, k_l) = \frac{1}{n_l - 1}\sum_{r=1}^{n_l}(\xi_{k_{l-1},k_l}(r) - \bar{\xi}_{k_{l-1},k_l})^2, \quad \widehat{\Theta}_{K,L} = \frac{\widehat{\sigma}_\infty}{\kappa^{L/2}} + L\widehat{v}(k_{L-1}, K).$$

Here, $\widehat{\sigma}_\infty = \sqrt{\widehat{v}(0, 300)}$ denotes the estimator of σ_∞ and $\widehat{\Theta}_{K,L}$ estimates $\Theta_{K,L}$ from (6.32).

To reduce variance, the same subsimulations that were used for $\mathcal{Z}^{(r)}(M^{k_{l-1}})$ are reused in $\mathcal{Z}^{(r)}(M^{k_l})$ in each level. The following table gives information about the estimates of $v(k_{l-1}, k_l)$ and $\Theta_{K,L}$ and also gives advice how to invest the computational time into the different levels. The latter is done by listing n_0^*/N and n_1^*/N based on (6.31).

Afterwards, we fix a complexity of $NK = 10^6$ and predict how the multilevel algorithm would perform compared to the standard MC algorithm in such a case. Those results are given in the second table and are illustrated in Figure 6.2.

6.6 Proofs

Proof of Theorem 98

The number of levels is chosen in (6.6) such that it holds

$$\left|\mathrm{E}\left[Y^{\mathbf{n},\mathbf{k}}\right] - Y(M)\right| \leq C(k_0\kappa^L)^{-\gamma} = \varepsilon/\sqrt{2}. \tag{6.40}$$

So the squared bias is exactly half of the accuracy ε^2. For the three cases of β, we keep that in mind and show that the allocation of the trajectories $\mathbf{n}_l$ as given in the theorem lead to a total mean-squared error of ε^2.

1. Case $\beta < 1$: By (6.5), we obtain for the total variance of the multilevel estimator

$$\begin{aligned}
\mathrm{Var}\left[Y^{\mathbf{n},\mathbf{k}}\right] &\leq \widetilde{B}\sum_{l=0}^{L}\widetilde{B}^{-1}2^{-1}\varepsilon^2 k_0^\beta\kappa^{-L(1-\beta)/2}(1-\kappa^{-(1-\beta)/2})\kappa^{l(1+\beta)/2}k_0^{-\beta}\kappa^{-\beta l}\\
&= 2^{-1}\varepsilon^2\kappa^{-L(1-\beta)/2}(1-\kappa^{-(1-\beta)/2})\frac{\kappa^{(L+1)(1-\beta)/2}-1}{\kappa^{(1-\beta)/2}-1}\frac{\kappa^{-(1-\beta)/2}}{\kappa^{-(1-\beta)/2}}\\
&= 2^{-1}\varepsilon^2\kappa^{-L(1-\beta)/2}\left(\kappa^{(L+1)(1-\beta)/2}-1\right)\kappa^{-(1-\beta)/2}\\
&= 2^{-1}\varepsilon^2\left(1-\kappa^{-(L+1)(1-\beta)/2}\right) \leq \varepsilon^2/2,
\end{aligned}$$

which is sufficient because of the variance-bias decomposition. The order of $\mathscr{C}_{ML}(\varepsilon)$ follows from the estimate

$$\begin{aligned}
\sum_{l=0}^{L} k_l n_l &\leq \sum_{l=0}^{L} k_0 \kappa^l \left(2\varepsilon^{-2} \widetilde{B} k_0^{-\beta} \kappa^{L(1-\beta)/2} (1-\kappa^{-(1-\beta)/2})^{-1} \kappa^{-l(1+\beta)/2} + 1 \right) \\
&= \sum_{l=0}^{L} \kappa^l \left(2\varepsilon^{-2} \widetilde{B} k_0^{1-\beta} \kappa^{L(1-\beta)/2} (1-\kappa^{-(1-\beta)/2})^{-1} \kappa^{-l(1+\beta)/2} + k_0 \right) \\
&= k_0^{1-\beta} \left(2\varepsilon^{-2} \widetilde{B} \kappa^{L(1-\beta)/2} \left(\kappa^{L(1-\beta)/2} - \kappa^{-(1-\beta)/2} \right) + k_0 \frac{\kappa^{L+1}-1}{\kappa-1} \right) \\
&\leq 2\varepsilon^{-2} \widetilde{B} k_0^{1-\beta} \kappa^{L(1-\beta)} + k_0 \kappa^{L+1} \\
&\leq 2\varepsilon^{-2} \widetilde{B} k_0^{1-\beta} \kappa^{\left(\frac{-\ln \frac{k_0^\gamma \varepsilon}{C\sqrt{2}}}{\gamma \ln \kappa} + 1 \right)(1-\beta)} + k_0 \kappa^{\frac{-\ln \frac{k_0^\gamma \varepsilon}{C\sqrt{2}}}{\gamma \ln \kappa} + 2} \\
&= 2\widetilde{B} \frac{\left(C\sqrt{2} \right)^{(1-\beta)/\gamma}}{\varepsilon^{2+(1-\beta)/\gamma}} \kappa^{(1-\beta)} + \frac{\left(C\sqrt{2} \right)^{1/\gamma}}{\varepsilon^{1/\gamma}} \kappa^2 \\
&= O(\varepsilon^{-2-(1-\beta)/\gamma}), \quad \varepsilon \to 0,
\end{aligned}$$

where one should note that $\gamma \geq 1/2$.

2. Case $\beta = 1$: We calculate that it holds

$$\begin{aligned}
\operatorname{Var}\left[Y^{\mathbf{n},\mathbf{k}} \right] &\leq \widetilde{B} \sum_{l=0}^{L} 2^{-1} \varepsilon^2 \widetilde{B}^{-1} (L+1)^{-1} k_0 \kappa^l k_0^{-1} \kappa^{-l} \\
&= 2^{-1} \varepsilon^2 \sum_{l=0}^{L} (L+1)^{-1} = \varepsilon^2/2.
\end{aligned}$$

For $\mathscr{C}_{ML}(\varepsilon)$ we now have

$$\begin{aligned}
\mathscr{C}_{ML}(\varepsilon) &\leq \sum_{l=0}^{L} k_0 \kappa^l \left(2\varepsilon^{-2} \widetilde{B} (L+1) k_0^{-1} \kappa^{-l} + 1 \right) \\
&= \sum_{l=0}^{L} \left(2\varepsilon^{-2} \widetilde{B} (L+1) + k_0 \kappa^l \right) \\
&= 2\varepsilon^{-2} \widetilde{B} (L+1)^2 + k_0 \frac{\kappa^{L+1}-1}{\kappa-1} \\
&\leq 2\varepsilon^{-2} \widetilde{B} (L+1)^2 + k_0 \kappa^{L+1} \\
&\leq 2\varepsilon^{-2} \widetilde{B} \left(\frac{-\ln \frac{k_0^\gamma \varepsilon}{C\sqrt{2}}}{\gamma \ln \kappa} + 2 \right)^2 + \frac{\left(C\sqrt{2} \right)^{1/\gamma}}{\varepsilon^{1/\gamma}} \kappa^2 \\
&= O(\varepsilon^{-2} \ln^2 \varepsilon), \quad \varepsilon \to 0
\end{aligned}$$

since $\gamma \geq 1/2$.

3. Case $\beta > 1$: The variance fulfills again

$$\begin{aligned}
\mathrm{Var}\left[Y^{\mathbf{n},\mathbf{k}}\right] &\leq \widetilde{B}\sum_{l=0}^{L} 2^{-1}\varepsilon^2\widetilde{B}^{-1}k_0^{\beta}(1-\kappa^{-(\beta-1)/2})\kappa^{l(1-\beta)/2}k_0^{-\beta} \\
&= 2^{-1}\varepsilon^2(1-\kappa^{-(\beta-1)/2})\frac{1-\kappa^{(L+1)(1-\beta)/2}}{1-\kappa^{(1-\beta)/2}} \\
&= 2^{-1}\varepsilon^2\left(1-\kappa^{(L+1)(1-\beta)/2}\right) \leq \varepsilon^2/2
\end{aligned}$$

and for the complexity we have in this case

$$\begin{aligned}
\mathscr{C}_{ML}(\varepsilon) &\leq \sum_{l=0}^{L} k_0\kappa^l\left(2\varepsilon^{-2}\widetilde{B}k_0^{-\beta}(1-\kappa^{-(\beta-1)/2})^{-1}\kappa^{-l(1+\beta)/2}+1\right) \\
&\leq 2\varepsilon^{-2}\widetilde{B}k_0^{1-\beta}\sum_{l=0}^{L}(1-\kappa^{-(\beta-1)/2})^{-1}\kappa^{l(1-\beta)/2}+k_0\kappa^{L+1} \\
&= 2\varepsilon^{-2}\widetilde{B}k_0^{1-\beta}\left(1-\kappa^{(L+1)(1-\beta)/2}\right)+k_0\kappa^{L+1} \\
&\leq 2\varepsilon^{-2}\widetilde{B}k_0^{1-\beta}+\frac{\left(C\sqrt{2}\right)^{1/\gamma}}{\varepsilon^{1/\gamma}}\kappa^2 \\
&= O(\varepsilon^{-2}), \quad \varepsilon\to 0, \quad \text{since} \quad \gamma \geq 1/2.
\end{aligned}$$

l	(k_{l-1}, k_l)	$\sqrt{\widehat{v}(k_{l-1}, k_l)}$	n_l^*/N
0	(0,50)	0.006928	0.006803
1	(50,100)	0.002918	0.002199
2	(100,200)	0.002048	0.001100
3	(200,400)	0.001386	0.000550
	$K = 400$	$\widehat{\Theta}_K = 0.006300$	
0	(0,100)	0.006055	0.003993
1	(100,200)	0.002044	0.001001
2	(200,400)	0.001521	0.000501
3	(400,800)	0.001075	0.000250
	$K = 800$	$\widehat{\Theta}_K = 0.005366$	
0	(0,150)	0.006083	0.003127
1	(150,300)	0.001581	0.000590
2	(300,600)	0.001118	0.000295
3	(600,1200)	0.000809	0.000148
	$K = 1200$	$\widehat{\Theta}_K = 0.004569$	
0	(0,200)	0.005986	0.002416
1	(200,400)	0.001443	0.000431
2	(400,800)	0.001040	0.000215
3	(800,1600)	0.000764	0.000108
	$K = 1600$	$\widehat{\Theta}_K = 0.004435$	
0	(0,250)	0.005918	0.001995
1	(250,500)	0.001360	0.000334
2	(500,1000)	0.000971	0.000167
3	(1000,2000)	0.000718	0.000084
	$K = 2000$	$\widehat{\Theta}_K = 0.004297$	
0	(0,300)	0.005954	0.001809
1	(300,600)	0.001198	0.000254
2	(600,1200)	0.000822	0.000127
3	(1200,2400)	0.000602	0.000064
	$K = 2400$	$\widehat{\Theta}_K = 0.003949$	

Table 6.3: Estimates for $v(k_{l-1}, k_l)$, n_l^* and Θ_K, $l = 1, \ldots, 3$

l	n_l^*	k_l	$\frac{1}{n_l}\sum_{r=1}^{n_l}\xi_{k_{l-1},k_l}(r)$	$\sqrt{\widehat{v}(k_{l-1},k_l)}$
0	6600	50	0.0341611	0.00686113
1	2230	100	1.12787e-05	0.00288741
2	1110	200	2.45243e-05	0.00193951
3	550	400	2.74035e-05	0.00140730
			$Y^{\mathbf{n}^*,\mathbf{k}} = 0.0340843$	$\mathrm{sd}(Y^{\mathbf{n}^*,\mathbf{k}}) = \mathbf{0.0001336}$
AB	$N = 2500$	$K = 400$	$Y^{N,K} = 0.0341167$	$\mathrm{sd}(Y^{N,K}) = \mathbf{0.0001111}$

l	n_l^*	k_l	$\frac{1}{n_l}\sum_{r=1}^{n_l}\xi_{k_{l-1},k_l}(r)$	$\sqrt{\widehat{v}(k_{l-1},k_l)}$
0	3880	100	0.0341174	0.00630161
1	1010	200	6.64818e-06	0.00208278
2	500	400	4.17533e-05	0.00144304
3	250	800	6.16153e-05	0.00101637
			$Y^{\mathbf{n}^*,\mathbf{k}} = 0.0340224$	$\mathrm{sd}(Y^{\mathbf{n}^*,\mathbf{k}}) = \mathbf{0.0001510}$
AB	$N = 1250$	$K = 800$	$Y^{N,K} = 0.0341235$	$\mathrm{sd}(Y^{N,K}) = \mathbf{0.0001676}$

l	n_l^*	k_l	$\frac{1}{n_l}\sum_{r=1}^{n_l}\xi_{k_{l-1},k_l}(r)$	$\sqrt{\widehat{v}(k_{l-1},k_l)}$
0	3050	150	0.034024	0.00613309
1	600	300	5.96397e-06	0.00167245
2	300	600	-1.46135e-05	0.00114155
3	150	1200	1.42701e-05	0.00078627
			$Y^{\mathbf{n}^*,\mathbf{k}} = 0.0340522$	$\mathrm{sd}(Y^{\mathbf{n}^*,\mathbf{k}}) = \mathbf{0.0001595}$
AB	$N = 850$	$K = 1200$	$Y^{N,K} = 0.0340616$	$\mathrm{sd}(Y^{N,K}) = \mathbf{0.0001903}$

l	n_l^*	k_l	$\frac{1}{n_l}\sum_{r=1}^{n_l}\xi_{k_{l-1},k_l}(r)$	$\sqrt{\widehat{v}(k_{l-1},k_l)}$
0	2360	200	0.0340537	0.00593753
1	430	400	4.03549e-05	0.00157739
2	210	800	-7.94923e-06	0.00095657
3	100	1600	-7.50145e-05	0.00090936
			$Y^{\mathbf{n}^*,\mathbf{k}} = 0.0340111$	$\mathrm{sd}(Y^{\mathbf{n}^*,\mathbf{k}}) = \mathbf{0.0001826}$
AB	$N = 625$	$K = 1600$	$Y^{N,K} = 0.0340312$	$\mathrm{sd}(Y^{N,K}) = \mathbf{0.0002326}$

Table 6.4: The performance of ML estimates with the optimal choice of n_l^*, $l = 0, \ldots, 3$, compared to the performance of the standard AB estimate with $NK = 10^6$ and $K = k_L$.

Chapter 7

Implementation

The numerical examples that appear in this work have been implemented in C++. The source code can be found on on the webpage

www.uni-due.de/mathematik/dickmann/

or directly on the github account

https://github.com/cfdickmann

Many of the algorithms that can be found there make use of the library ALGLIB, which is an open source, cross-platform library for numerical algorithms. It was designed by Sergey Bochkanov and is available at

http://www.alglib.net/

The random numbers are generated by a Mersenne Twister, which is a popular type of pseudorandom number generator. In the experiments, the implementation of Makoto Matsumoto and Takuji Nishimura was used, which can be found on

$$http://www.math.sci.hiroshima-u.ac.jp/\sim m-mat/MT/emt.html$$

Another interesting implementation related issue is the solution of convex optimization problems which is discussed in the following section.

7.1 Smooth Minimization of Non-Smooth Functions

In Section 5.4 a algorithm for convex optimization problems of the type (5.108) was necessary. A variety of methods is available for this task, in particular gradient methods. The so-called BFGS algorithm belongs to the class of quasi-Newton methods because it approximates the Hessian matrix to find the best direction of descent. It was developed by four researchers at the same time, namely Broyden [21], Fletcher [36], Goldfarb [44] and Shanno [67] in 1970. An implementation of it can be found in ALGLIB as well as in the library of the scripting language R.

Another interesting method can be found in Nesterov [61]. In his article, he describes a method to solve the minimization problem

$$\min_x \{f(x) : x \in Q\} \tag{7.1}$$

for a function $f \in C_L^{1,1}(Q)$, which is a convex real valued function defined on a closed convex set $Q \subset E$. The gradient is Lipschitz continuous such that

$$\|\nabla f(x) - \nabla f(y)\|^* \leq L\|x - y\|, \quad \forall x, y, \in Q.$$

To describe the minimization algorithm, let us define the quantity

$$T_Q(x) = \arg\min_{y \in Q} \left\{ \langle \nabla f(x), y - x \rangle + \frac{1}{2} L \|y - x\|^2 : y \in Q \right\},$$

which is an unambiguous definition in case of a convex norm $\|\cdot\|$. Secondly, let $d(x)$ be a prox-function of Q that is strongly convex with convexity parameter $\sigma > 0$. Nesterov proposes the following algorithm.

Algorithm 2 Smooth minimization of a non-smooth function

```
for $k \geq 0$ do
    1. Compute $f(x_k)$ and $\nabla f(x_k)$
    2. Find $y_k = T_Q(x_k)$.
    3. Find $z_k = \arg\min_x \left\{ \frac{L}{\sigma} d(x) + \sum_{i=0}^{k} \frac{i+1}{2} [f(x_i) + \langle \nabla f(x_i), x - x_i \rangle] : x \in Q \right\}$
    4. Set $x_{k+1} = \frac{2}{k+3} z_k + \frac{k+1}{k+3} y_k$
end for
```

Theorem 101. *For any $k \geq 0$, we have that*

$$f(y_k) - f(x^*) \leq \frac{4Ld(x^*)}{\sigma(k+1)(k+2)} \tag{7.2}$$

where x^* is the optimal solution to the problem (7.1).

It turns out that the BFGS, as well as Nesterov's methods are appropriate for the solution of the problem (5.108). An implementation of both of them can be found on github in the project "AmericanOptionsBFGS". However, rather high values for L will be recommended and its choice is quite difficult, which is in favor of the BFGS.

Chapter 8

Perspectives

At the end of this work, some perspectives for future research are mentioned. The first one originates from Section 4.3.3. If it was possible to further examine the term

$$\mathrm{Var}\left[\mathrm{E}\left[Y_{\tau^A} - Y_{\tau^B} \middle| \mathcal{F}_{\tau^\wedge}\right] \middle| \tau^A \neq \tau^B\right] \tag{8.1}$$

one might achieve an order of complexity even less than $\varepsilon^{-2.5}$, which was the multilevel complexity in the usual mesh case. Two other ideas are listed below.

8.1 Unbiased Estimators

As explained in the introduction of Chapter 3, Rhee and Glynn [63] suggest an algorithm that can be seen as an extension for the multilevel technique. Roughly speaking, they randomize the number of levels and thus construct an unbiased estimator.

Let us apply this idea for the high biased estimator $\mathrm{E}[\theta_0^{up}(M)]$ for a BSDE recursion problem as given in Section 5.3. We could include the idea of Rhee and Glynn in our multilevel technique by fixing $L \in \mathbb{N}^+$ and an increasing sequence of positive integers

$$k_0, k_1, \dots, k_{L-1}, k_L, k_{L+1}, \dots. \tag{8.2}$$

Now, let η be a random variable taking all the numbers $L, L+1, \dots$ with positive probability. Denote by

$$p_i = P(\eta \geq i), \quad i = L, L+1, \dots.$$

and introduce the new estimator

$$\begin{aligned} \widehat{Y}_0^{N,K} &= \frac{1}{n_0} \sum_{n=1}^{n_0} \theta_0^{up,(n)}(M^{k_0,(n)}) \qquad (8.3) \\ &+ \sum_{l=1}^{L-1} \left\{ \frac{1}{n_l} \sum_{n=1}^{n_l} \theta_0^{up,(n)}(M^{k_l,(n)}) - \theta_0^{up,(n)}(M^{k_{l-1},(n)}) \right\} \\ &+ \frac{1}{n_L} \sum_{n=1}^{n_L} \left\{ \sum_{i=L}^{\eta^{(n)}} \frac{1}{p_i} \left(\theta_0^{up,(n)}(M^{k_i,(n)}) - \theta_0^{up,(n)}(M^{k_{i-1},(n)}) \right) \right\} \end{aligned}$$

where $\eta^{(1)}, \ldots, \eta^{(n_L)}$ are i.i.d. copies of η. It is clear that the expectation of our estimator fulfills

$$E[\widehat{Y}_0^{N,K}] = E[\theta_0^{up}(M^{k_{L-1}})] + E_L,$$

where E_L denotes the expectation of the highest level, i.e.

$$\begin{aligned} E_L &:= E\left[\sum_{i=L}^{\eta^{(n)}} \frac{1}{p_i}\left(\theta_0^{up,(n)}(M^{k_i,(n)}) - \theta_0^{up,(n)}(M^{k_{i-1},(n)})\right)\right] \\ &= E[\theta_0^{up}(M)] - E[\theta_0^{up}(M^{k_{L-1}})] \end{aligned}$$

because of the telescopic sum, so $E[\widetilde{Y}_0^{N,K}] = E[\theta_0^{up}(M)]$ under some assumptions on the sequence $(M^k)_{k\in\mathbb{N}}$.

Secondly, the variance is clear for the lower levels but must be analysed for the highest level. Rhee and Glynn [63] can show that for approximation schemes fulfilling

$$\mathrm{E}\left[\left(\theta_0^{up}(M^k) - \theta_0^{up}(M)\right)^2\right] \leq Ck^{2r} \tag{8.4}$$

with some $r > 1/2$ the variance of the highest level of our approach will be be finite if a suitable choice for k_L, k_{L+1} and a suitable distribution of η is used. However, as we take from (5.84) we only have (8.4) fulfilled with $r = 1/2$.

Perhaps it is possible to find a little change in (8.3) or to improve the result of Rhee and Glynn. Perhaps it is possible to randomize the number of levels in a different way to somehow avoid this problem. Another loophole could be the use Quasi Monte Carlo techniques that could improve (5.84).

8.2 Upper Bounds via Semi-Infinite Programming

In this section, a quite new idea to construct upper bounds is introduced. Like all other methods, it is working well in case of $D = 1$. Unfortunately, it turns out that it is very unreliable in higher dimensions. It is still a challenge to get it working at least for $D = 3, 4, 5$.

The basic idea exploits the characterization of the true solution V_j as the smallest supermartingale dominating the payoff process $g_j(X_j)$, see Remark 60. With some effort, one can show that

$$V_0 = \mathrm{E}\left[\sum_{j=0}^{\mathcal{J}} (g_j(X_j) - C_j(X_j))^+\right] \tag{8.5}$$

and given a lower bound $l_j(x)$ for the continuation value $C_j(x)$ for all $x \in \mathbb{R}^D$, we have an upper bound via

$$V_0 \leq \mathrm{E}\left[\sum_{j=0}^{\mathcal{J}} (g_j(X_j) - l(X_j))^+\right]. \tag{8.6}$$

When constructing such a bound backwards from $t_{\mathcal{J}}$ to t_0, we have the following recursive problem for each time step: Try to find a function $l_j : \mathbb{R}^D \to \mathbb{R}$ such that for each $x \in \mathbb{R}^D$, $l_j(x)$ is a lower biased estimator for $o_j(x)$, where

$$o_j(x) := \mathrm{E}\left[\sum_{i=j+1}^{\mathcal{J}} (g_j(X_j) - l_j(X_j))^+ \,\middle|\, X_j = x \right]. \tag{8.7}$$

We have the following convenient effect: If there is a small error at time step j such that $l_j(x)$ is too high in some region, then (8.7) will cause that $l_{j-1}(y)$ will be a little higher than expected for all y that have a high transition density to x at j. This will lead to some numerical ease, i.e. the ith summand in (8.7) will contribute a little less and the jth summand a little more.

The following algorithm works with a set of basis functions $\phi^j : \mathbb{R}^D \to \mathbb{R}$, $j = 1, \ldots, J$ and supporting points $x_p \in \mathbb{R}^D$, $p = 1, \ldots, p$ that must be fixed before. For each exercise date t_i beginning from $t_{\mathcal{J}-1}$ to t_0, carry out the following three steps:

1. For each x_p, generate M independent subsimulations $X_\cdot^{(1)}, \ldots, X_\cdot^{(M)}$ starting from x_p at time i and estimate

$$E_p = \frac{1}{M} \sum_{m=1}^{M} \left\{ \sum_{i<k\leq\mathcal{J}} \left(g_i\left(X_i^{(m)}\right) - \sum_{j=1}^{J} \beta_i^j \phi_i^j \left(X_i^{(m)}\right) \right)^+ \right\}, \tag{8.8}$$

   where $\beta_N \equiv 0$ per definition.

2. Solve the linear optimization problem

$$\alpha := \operatorname*{arginf}_{\beta\in\mathcal{A}} \left\{ \sum_{p=1}^{P} \left(E_p - \sum_{j=1}^{J} \beta^j \phi_i^j(x_p) \right) \right\}$$

   where

$$\mathcal{A} = \left\{ \beta \in \mathbb{R}^J : \sum_{j=1}^{J} \beta^j \phi_i^j(x_p) \leq E_p, \quad p = 1, \ldots, P \right\}.$$

3. Repeat steps 1 and 2 five times with different subsimulations to obtain $\alpha_1, \ldots, \alpha_5$ and define β_i as their arithmetic mean.

Afterwards, we generate again H testing paths $Y_\cdot^{(1)}, \ldots, Y_\cdot^{(H)}$ and obtain

$$\frac{1}{H} \sum_{h=1}^{H} \left\{ \sum_{0\leq i\leq\mathcal{J}} \left(g_i\left(Y_i^{(h)}\right) - \sum_{j=1}^{J} \beta_i \phi_i^j \left(Y_i^{(h)}\right) \right)^+ \right\} \tag{8.9}$$

as a high biased estimator of the option price. To solve the linear optimization problems, we use the simplex algorithm of GLPK (GNU Linear Programming Kit).

In case of one asset, choosing basis functions and the number of supporting points must be done very carefully, as too many basis functions will cause the linear optimization problem to become ill-conditioned, so little errors can cause

X_0	low	σ(low)	high	σ(high)
80	21.5110	0.012	21.6188	0.009
90	14.8359	0.012	14.9351	0.012
100	9.8836	0.013	9.9460	0.009
110	6.3762	0.012	6.4340	0.005
120	4.0245	0.011	4.0489	0.005

Table 8.1: We used $J = 16$ basis functions, $P = 25$ supporting points, $M = 5000$ subsimulations and $H = 10$ million testing paths. The algorithm needs about 6 seconds and about 3 minutes are needed for testing on an Intel Dual Core (2.3 GHz).

large problems. In case of a big number of basis functions, $90 - 98\%$ the coefficients calculated by the GLPK turn out to be zero and summing in (8.8) and (8.9) will be faster when saving only those coefficients different from zero and their positions.

To treat a one-dimensional Bermudan put option with Strike price $\varkappa$ on a set of $\mathcal{J}$ exercise dates, we choose uniformly distributed supporting points according to

$$x_p = \varkappa\left(0.1 + \frac{p}{P}\right),\ p = 1, \ldots, P,$$

which in case of a call option would have been $x_p = \varkappa(0.9 + \frac{p}{P})$. When choosing basis functions $\{\phi^j\}_{j=1,\ldots,J}$ a first idea are polynomials and the payoff function

$$1, X_t, X_t^2, X_t^3, X_t^4, g_t(X_t).$$

Furthermore, we additionally use the functions

$$\left(4\frac{j-6}{J-6}K - x\right)^+,\ j = 6, \ldots, \mathcal{J}$$

for the put option, where 4 has been considered optimal by experience. In this example, there are $\mathcal{J} = 9$ uniformly distributed exercise dates and maturity time $T = 0.5$. There is one asset of GMB with interest rate $r = 0.06$, dividend yield of $\delta = 0.4$, and Strike $\varkappa = 100$. The results are presented in Table 8.1 and Table 8.2, where the lower bounds were calculated by global regression, see Section 2.1.2.

X_0	low	σ(low)	high	σ(high)
80	21.5110	0.012	21.5576	0.007
90	14.8359	0.012	14.8814	0.012
100	9.8836	0.013	9.9154	0.007
110	6.3762	0.012	6.4095	0.011
120	4.0245	0.011	4.0436	0.009

Table 8.2: Investing more time, i.e. one minute for the algorithm and 4 minutes for testing, using $J = 56$ basis functions, $P = 80$ supporting points and $M = 10000$ subsimulations yields better results.

Bibliography

[1] A. Agarwal and S. Juneja. Nearest neighbor based estimation technique for pricing Bermudan options.

[2] L. Andersen and M. Broadie. A primal-dual simulation algorithm for pricing multidimensional American options. *Management Sciences*, 50(9):1222–1234, 2004.

[3] S. Asmussen and K. Binswanger. Simulation of ruin probabilities for subexponential claims. *Astin Bulletin*, 27(2):297–318, 1997.

[4] A. N. Avramidis and H. Matzinger. Convergence of the stochastic mesh estimator for pricing bermudan options. *Journal of Computational Finance*, 7:73–91, 2004.

[5] K. W. Bauer Jr, S. Venkatraman, and J. R. Wilson. Estimation procedures based on control variates with known covariance matrix. In *Proceedings of the 19th conference on Winter simulation*, pages 334–341. ACM, 1987.

[6] D. Belomestny. Pricing Bermudan options using nonparametric regression: optimal rates of convergence for lower estimates. *Finance and Stochastics*, 15(4):655–683, 2011.

[7] D. Belomestny. Solving optimal stopping problems via empirical dual optimization. *The Annals of Applied Probability*, 23(5):1988–2019, 10 2013.

[8] D. Belomestny, C. Bender, and J. Schoenmakers. True upper bounds for Bermudan products via non-nested Monte Carlo. *Math. Financ.*, 19(1):53–71, 2009.

[9] D. Belomestny, T. Nagapetyan, and V. Shiryaev. Multilevel path simulation for weak approximation schemes: myth or reality. *arXiv preprint arXiv:1406.2581*, 2014.

[10] D. Belomestny, J. Schoenmakers, and F. Dickmann. Multilevel dual approach for pricing american style derivatives. *Finance and Stochastics*, 17(4):717–742, 2013.

[11] C. Bender, N. Schweizer, and J. Zhuo. A primal-dual algorithm for bsdes. *arXiv preprint arXiv:1310.3694*, 2013.

[12] C. Bender and J. Steiner. Least-squares monte carlo for backward sdes. In *Numerical methods in finance*, pages 257–289. Springer, 2012.

[13] A. Bensoussan and J.-L. Lions. *Applications of variational inequalities in stochastic control.* Elsevier, 2011.

[14] B. Bouchard and X. Warin. Monte-Carlo valuation of American options: facts and new algorithms to improve existing methods. *Numerical Methods in Finance*, pages 215–255, 2012.

[15] P. Boyle, M. Broadie, and P. Glasserman. Monte Carlo methods for security pricing. *Journal of Economic Dynamics and Control*, 21(8):1267–1321, 1997.

[16] M. Broadie and M. Cao. Improved Lower and Upper Bound Algorithms for Pricing American Options by Simulation. *Quantitative Finance*, 8(8):845–861, 2008.

[17] M. Broadie and J. Detemple. The valuation of American options on multiple assets. *Mathematical Finance*, 7(3):241–286, 1997.

[18] M. Broadie, Y. Du, and C. C. Moallemi. Efficient risk estimation via nested sequential simulation. *Management Science*, 57(6):1172–1194, 2011.

[19] M. Broadie and P. Glasserman. A stochastic mesh method for pricing high-dimensional American options. *Journal of Computational Finance*, 7(4):35–72, 2004.

[20] M. Broadie, P. Glasserman, and Z. Ha. Pricing american options by simulation using a stochastic mesh with optimized weights. In *Probabilistic Constrained Optimization*, pages 26–44. Springer, 2000.

[21] C. G. Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, 1970.

[22] R. Carmona, P. Del Moral, P. Hu, and N. Oudjane. An Introduction to Particle Methods with Financial Applications. In R. A. Carmona, P. Del Moral, P. Hu, and N. Oudjane, editors, *Numerical Methods in Finance*, volume 12 of *Springer Proceedings in Mathematics*, pages 3–49. Springer Berlin Heidelberg, 2012.

[23] J. Carriére. Valuation of early-exercise price of options using simulations and nonparametric regression. *Insur. Math.Econ.*, 19, 1996.

[24] N. Chen and P. Glasserman. Additive and multiplicative duals for american option pricing. *Finance and Stochastics*, 11(2):153–179, 2007.

[25] E. Clément, D. Lamberton, and P. Protter. An analysis of a least squares regression method for american option pricing. *Finance and Stochastics*, 6(4):449–471, 2002.

[26] J. C. Cox, S. A. Ross, and M. Rubinstein. Option pricing: A simplified approach. *Journal of financial Economics*, 7(3):229–263, 1979.

[27] S. Crépey. Bilateral counterparty risk under funding constraintspart ii: Cva. *Mathematical Finance*, 2012.

[28] S. Dahlke, W. Dahmen, M. Griebel, W. Hackbusch, K. Ritter, R. Schneider, C. Schwab, and H. Yserentant. *Extraction of quantifiable information from complex systems.*

[29] M. Davis and I. Karatzas. A deterministic approach to optimal stopping. *Probability, Statistics and Optimisation (ed. FP Kelly). NewYork Chichester: John Wiley & Sons Ltd*, pages 455–466, 1994.

[30] S. Dereich and S. Li. Multilevel monte carlo for lévy-driven sdes: central limit theorems for adaptive euler schemes. *Preprint*, 161.

[31] V. V. Desai, V. F. Farias, and C. C. Moallemi. Pathwise optimization for optimal stopping problems. *Management Science*, 58(12):2292–2308, 2012.

[32] D. Egloff. Monte Carlo algorithms for optimal stopping and statistical learning. *Ann. Appl. Probab.*, 15:1396–1432, 2005.

[33] N. El Karoui, S. Peng, and M. C. Quenez. Backward stochastic differential equations in finance. *Mathematical finance*, 7(1):1–71, 1997.

[34] M. Emsermann and B. Simon. Improving Simulation Efficiency with Quasi Control Variates. *Stochastic Models*, 18, 2002.

[35] Y. Feng and G. Gallego. Optimal starting times for end-of-season sales and optimal stopping times for promotional fares. *Management Science*, 41(8):1371–1391, 1995.

[36] R. Fletcher. A new approach to variable metric algorithms. *The computer journal*, 13(3):317–322, 1970.

[37] H. Föllmer and A. Schied. *Stochastic Finance: An Introduction in Discrete Time.* De Gruyter studies in mathematics, 2002.

[38] S. Gerhold et al. The longstaff–schwartz algorithm for lévy models: results on fast and slow convergence. *The Annals of Applied Probability*, 21(2):589–608, 2011.

[39] M. B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.

[40] P. Glasserman. *Monte Carlo Methods in Financial Engineering.* Springer, 2004.

[41] P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic. Multilevel splitting for estimating rare event probabilities. *Operations Research*, 47(4):585–600, 1999.

[42] P. Glasserman and B. Yu. Number of paths versus number of basis functions in american option pricing. *The Annals of Applied Probability*, 14(4):2090–2119, 2004.

[43] P. Glasserman and B. Yu. Simulation for american options: Regression now or regression later? In *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 213–226. Springer, 2004.

[44] D. Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24(109):23–26, 1970.

[45] M. B. Gordy and S. Juneja. Nested simulation in portfolio risk measurement. *Management Science*, 56(10):1833–1848, 2010.

[46] M. Haugh and L. Kogan. Pricing American options: a duality approach. *Operations Research*, 52(2):258–270, 2004.

[47] S. Heinrich. Multilevel Monte Carlo methods. In *Large-scale scientific computing. 3rd international conference, LSSC 2001, Sozopol, Bulgaria, June 6–10, 2001.*, pages 58–67. Springer, 2001.

[48] B. Hofmann. *Mathematik inverser Probleme.* Teubner Stuttgart, 1999.

[49] J. Jacod and A. Shiryaev. *Limit Theorems for Stochastic Processes.* Springer, 2003.

[50] P. Jaillet, D. Lamberton, and B. Lapeyre. Variational inequalities and the pricing of american options. *Acta Applicandae Mathematica*, 21(3):263–289, 1990.

[51] F. Jamshidian. Numeraire-invariant option pricing & american, bermudan, and trigger stream rollover. 2004.

[52] M. Kohler, A. Krzyżak, and N. Todorovic. Pricing of high-dimensional american options by neural networks. *Mathematical Finance*, 20(3):383–410, 2010.

[53] A. Kolodko and J. Schoenmakers. Upper bounds for Bermudan style derivatives. Monte Carlo Methods and Appl. 10(3-4):331–343, 2004.

[54] A. Kolodko and J. Schoenmakers. Iterative construction of the optimal bermudan stopping time. *Finance and Stochastics*, 10(1):27–49, 2006.

[55] S. S. Lavenberg, T. L. Moeller, and P. D. Welch. Statistical results on control variables with application to queueing network simulation. *Operations Research*, 30(1):182–202, 1982.

[56] J.-P. Lemor, E. Gobet, X. Warin, et al. Rate of convergence of an empirical regression method for solving generalized backward stochastic differential equations. *Bernoulli*, 12(5):889–916, 2006.

[57] F. Longstaff and E. Schwartz. Valuing American options by simulation: a simple least-squares approach. *Rev. Financ. Stud.*, 14:113–147, 2001.

[58] P. G. M. Broadie. A stochastic mesh method for pricing high-dimensional American options. *Journal of Computational Fincance*, 7(4):35–72, 2004.

[59] D. McLeish. A general method for debiasing a monte carlo estimator. *arXiv preprint arXiv:1005.2228*, 2010.

[60] B. L. Nelson. Control variate remedies. *Operations Research*, 38(6):974–992, 1990.

[61] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[62] G. Peskir and A. Shiryaev. *Optimal stopping and free-boundary problems.* Springer, 2006.

[63] C.-h. Rhee and P. W. Glynn. A new approach to unbiased estimation for sde's. In *Proceedings of the Winter Simulation Conference*, page 17. Winter Simulation Conference, 2012.

[64] L. C. G. Rogers. Monte Carlo Valuation of American Options. *Mathematical Finance*, 12(3):271–286, July 2002.

[65] J. Schoenmakers. Robust Libor Modelling and Pricing of Derivative Products. *BocaRaton London NewYork Singapore: Chapman & Hall – CRC Press 2005*, 2005.

[66] J. Schoenmakers, J. Zhang, and J. Huang. Optimal dual martingales, their analysis, and application to new algorithms for bermudan products. *SIAM Journal on Financial Mathematics*, 4(1):86–116, 2013.

[67] D. F. Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24(111):647–656, 1970.

[68] J. M. Steele. *Stochastic calculus and financial applications.* Springer, New York, 2001.

[69] L. Stentoft. Assessing the least squares monte-carlo approach to american option valuation. *Review of Derivatives research*, 7(2):129–168, 2004.

[70] K. T. Talluri and G. J. van Ryzin. *The theory and practice of revenue management.* Springer, New York, 2005.

[71] J. Tsitsiklis and B. V. Roy. Optimal stopping of Markov processes: Hilbert Space Theory, approximation algorithms, and an application to pricing high-dimensional derivatives. *IEEE Transactions on Automatic Control*, 44:1840–1851, 1999.

[72] J. Tsitsiklis and B. V. Roy. Regression methods for pricing Amirican-style options. *IEEE Transactions on Neural Networks*, 12:694–703, 2001.

[73] A. W. Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

[74] M. Villén-Altamirano and J. Villén-Altamirano. Analysis of RESTART simulation: Theoretical basis and sensitivity study. *European Transactions on Telecommunications*, 13(4):373–385, 2002.

[75] Y. Wang and R. Caflisch. Fast computation of upper bounds for american-style options without nested simulation. 2010.

[76] D. Z. Zanger. Convergence of a least-squares monte carlo algorithm for bounded approximating sets. *Applied Mathematical Finance*, 16(2):123–150, 2009.

[77] D. Z. Zanger. Quantitative error estimates for a least-squares monte carlo algorithm for american option pricing. *Finance and Stochastics*, 17(3):503–534, 2013.

List of Research Papers

1. Chapter 6 is partly based on:
   D. Belomestny, J. Schoenmakers, and F. Dickmann. *Multilevel dual approach for pricing American style derivatives.* Finance and Stochastics, 17(4):717-742, 2013

2. Chapter 3 is mainly based on:
   D. Belomestny, F. Dickmann, and T. Nagapetyan. *Pricing American options via multi-level approximation methods.* arXiv preprint arXiv:1303.1334, 2013.

3. Chapter 4 is based on:
   F. Dickmann and N. Schweizer. *Faster comparison of stopping times by nested conditional monte carlo.* arXiv preprint arXiv:1402.0243, 2014.

4. Chapters 5 and 2 are partly based on:
   D. Belomestny, C. Bender, F. Dickmann and N. Schweizer. *Solving Stochastic Dynamic Programs by Convex Optimization and Simulation*, first chapter of *Extraction of Quantifiable Information from Complex Systems*, Springer 2014 [28].

Acknowledgement

First of all, I want to thank my advisor Prof. Denis Belomestny for his support and patience during the last three years. He made it possible for me to participate in the Priority Program 1324 “Mathematical methods for extracting quantifiable information from complex systems” of Deutsche Forschungsgemeinschaft.

This offered to me the great opportunity to become a part of the scientific research community and get in contact with other young researchers. With more than 170 preprints and the book [28] published at Springer, it was really a successful program. I would also like to thank all the authors of our articles listed one page before. In alphabetical order, they are Prof. Christian Bender, PD John Schoenmakers, Nikolaus Schweizer and Tigran Nagapetyan.

Another very positive thing I encountered was the friendly and motivating working atmosphere in our group of Applied Statistics at the University of Duisburg-Essen. Thus, I want to thank my working colleagues Anna N., Anna S., Mikhail, Nicole, Nikolaus, Vladimir and Volker. It was really a great time.

The most important thank goes to my parents who deserve my gratitude for all the years of caring and understanding. There would not have been so many opportunities for me without them.

Zusammenfassung in deutscher Sprache

Der Multilevel Ansatz wurde vor allem durch die Arbeiten von Giles [39] und Heinrich [47] in der Stochastik populär. Dabei handelt es sich um eine Idee, die dazu genutzt werden kann, die Komplexität von Monte-Carlo Simulationen zu reduzieren. Genauer gesagt geht es darum, dass ein stochastischer Algorithmus gegeben ist, welcher ein nicht-deterministisches, vom Zufall abhängiges Ergebnis A_k ausgibt [1]. Dabei ist k ein Parameter, welcher sowohl Einfluss auf die Rechenzeit als auch auf die Genauigkeit des Algorithmus hat. Nun kann wie gewöhnlich $\mathrm{E}[A_k]$ durch einen Monte Carlo Schätzer

$$\frac{1}{n}\sum_{i=1}^{n} A_k^{(i)}$$

geschätzt werden, wobei $A_k^{(1)}, \ldots, A_k^{(n)}$ unabhängige, identische verteilte Ergebnisse des Algorithmus sind. Für einen Vektor von Parametern $k_0, \ldots k_L$ stellt man fest, dass

$$\mathrm{E}[A_{k_L}] \quad = \quad \mathrm{E}[A_{k_0}] + \sum_{l=1}^{L} \mathrm{E}\left[A_{k_l} - A_{k_{l-1}}\right]$$

und versucht dann, jede der Erwartungen auf der rechten Seite durch eine eigene, unabhängige Monte Carlo Simulation zu schätzen. So erhält man einen Schätzer für $\mathrm{E}[A_{k_L}]$, welcher im Folgenden als "Multilevel Schätzer" bezeichnet wird. Im Falle $L = 1$ kann man dies so ausdrücken: Der Algorithmus wird mit geringerer Genauigkeit als quasi-Kontrollvariate für sich selbst benutzt. Der Sinn dieses Vorgehens besteht in der Reduktion der Komplexität des Problems. Dieses wird möglich sein, wenn für kleine Parameter k die Genauigkeit des Algorithmus etwas, die Komplexität aber sehr viel geringer ist als für große k.

Der Multilevel Ansatz wird in dieser Arbeit auf folgende Problemstellung angewendet: Es sei ein stochastischer Prozess $X.$ gegeben, der zu $\mathcal{J} + 1$ verschiedenen Zeitpunkten $t_0, \ldots, \mathcal{J}$ gestoppt werden kann. Findet diese Stoppentscheidung zum Zeitpunkt mit Index j statt, so erhält der Besitzer der Option eine Auszahlung in Höhe von $g_j(X_j)$, wobei die Abzinsung gemäß des Zinssatzes r bereits in der Funktion g_j enthalten ist. Diese Problemstellung

[1] Die Zufälligkeit wird durch die Benutzung von Pseudo-Zufallszahlen realisiert. Diese können im Rechner zum Beispiel durch lineare-Kongruenz Generatoren oder einen Mersenne-Twister generiert werden.

entspricht der Bestimmung des fairen Preises einer so genannten “Bermuda” Option. Dieser ergibt sich aus dem No-Arbitrage Prinzip und kann als

$$V_0(x) = \sup_{\tau \in \mathcal{T}} \mathrm{E}\left[g_\tau(X_\tau) | X_j = x\right] \tag{8.10}$$

ausgedrückt werden. Hier bezeichnet $\mathcal{T}$ die Menge aller adaptierten Stoppzeiten mit Werten in $0, \ldots, \mathcal{J}$. Um die optimale Stoppzeit zu approximieren und (8.10) auswerten zu können, betrachten wir die Klasse der so genannten “fast approximation methods”. Diese arbeiten mit Hilfe von Regressionsmethoden rekursiv vom letzten bis zum ersten Zeitschritt und induzieren eine Stoppregel. Dieses Vorgehen ist charakteristisch für die grundlegende Problematik bei der Lösung von BSDEs [2] und zugehörige Rekursionen: Die numerische Lösung schreitet in der Zeit rückwärts, die Prozesse jedoch vorwärts voran.

Anschließend kann eine untere Schranke für den fairen Preis der Options per Monte Carlo Simulation approximiert werden, indem diese Stoppregel “getestet” wird. Man erhält nun als Schätzer

$$V_0^N = \frac{1}{N} \sum_{i=1}^{N} g_{\tau^{(i)}} \left(X^{(i)}_{\tau^{(i)}} \right),$$

wobei $\left(X^{(i)}_\cdot, \tau^{(i)} \right)$, $i = 1, \ldots, N$ unabhängige Realisierungen von X und den dazugehörigen Stoppzeiten sind. Bewertet man die Qualität einer solchen Methode mit Hilfe des mittleren quadratischen Fehlers ε, so ergibt sich für das Grundproblem eine Komplexität von ε^{-3} im gewöhnlichen Fall, siehe Kapitel 2. Hier ist unter dem gewöhnlichen Fall ein gutgestelltes Problem und die Verwendung eines stochastischen Netzes zu verstehen. Mit Hilfe des Multilevel Ansatzes lässt sich in diesem Fall eine Komplexität von $\varepsilon^{-2.5}$ erreichen, siehe Kapitel 3. Die Verbesserung ist im gewöhnlichen Fall also von Ordnung $\varepsilon^{-0.5}$, kann jedoch in anderen Fällen bis zu ε^{-1} betragen.

Zwei Stoppzeiten, welche beide nahe dem Optimum sind, werden in fast allen Fällen gleich ausfallen. D.h. für viele Realisierungen des Prozesses $X_\cdot$ wird der Beitrag zum Monte Carlo Schätzer 0 sein. Wäre es möglich, nur solche Realisierungen zu simulieren, die zu verschiedenen Stoppzeiten führen, könnte sich die Komplexität weiter reduzieren lassen. In Kapitel 4 wird daher eine Methode “NCMC” [3] vorgestellt, die durch eine Art von Gabelungen künstlich mehr Pfade solcher Art erzeugt. Es stellt sich heraus, dass dadurch eine klare Verringerung der Komplexität, jedoch keine geringere Ordnung erreicht werden kann.

Um obere Schranken für den Wert einer Bermuda Option zu berechnen, kann die duale Formulierung des Problems genutzt werden:

$$V_0 = \inf_{M \in \mathfrak{M}_0} \mathrm{E}\left[\max_{j=0,\ldots,\mathcal{J}} (g_j(X_j) - M_j) \right] \tag{8.11}$$

Hierbei ist $\mathfrak{M}_0$ die Menge aller adaptierten Martingale mit Startwert 0. Es geht also nun darum, ein Martingal zu wählen, welches die rechte Seite von (8.11) möglichst klein werden lässt. Es stellt sich heraus, dass der Martingalteil M^* des Prozesses V die optimale Wahl ist. Dabei ist V der so genannte “true

[2] Backwards Stochastic Differential Equations

[3] Nested Conditional Monte Carlo

value process", der zu jedem Zeitpunkt dem fairen Preis der Option entspricht. Obwohl letzterer gerade das gewünschte Ergebnis ist und damit als unbekannt vorausgesetzt werden muss, gibt diese Erkenntnis verschiedene Anhaltspunkte, wie nach einem geeigneten Martingal gesucht werden sollte.

Andersen und Broadie [2] nutzen in ihrer sehr bekannten Arbeit diesen Ansatz, indem sie das optimale Martingal mit Hilfe von geschachtelten Simlationen, so genannten "Subsimulations" approximieren. Hierbei sind zwei sehr ähnliche Vorgehensweisen möglich. Diese Subsimulations führen jedoch zu hohem Rechenaufwand, welcher erneut durch den Multilevel Ansatz reduziert werden kann. Je nach Problemstellung kann das Grundproblem eine Komplexität von Ordnung ε^{-3} oder sogar ε^{-4} besitzen, siehe Kapitel 5. Der Multilevel Ansatz senkt die Komplexität in jedem Fall (bis auf einen logarithmischen Faktor) auf ε^{-2}, wie in Kapitel 6 dargelegt wird.

Da in der Praxis nicht nur die Ordnung der Komplexität, sondern auch deren absolute Größe entscheidend sind, ist der Multilevel Ansatz bei der Berechnung unterer Schranken immer empfehlenswert, bei der oberer Schranken nur bei großer Komplexität, das heißt bei sehr genauen Berechnungen.

Selbständigkeitserklärung

Hiermit erkläre ich, gem. §7 Abs. (2) c)+ e) der Promotionsordnung der Fakultäten für Biologie, Chemie und Mathematik zur Erlangung des Dr. rer. nat., dass ich vorliegende Dissertation selbständig verfasst und mich keiner anderen als der angegebenen Hilfsmittel bedient habe.